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

Exploiting temporal and spatial constraints in traffic sign detection from a moving vehicle

Siniša Šegvić · Karla Brkić · Zoran Kalafatić · Axel Pinz

Received: date / Accepted: date

Abstract This paper addresses detection, tracking and recognition of traffic signs in video. Previous research has shown that very good detection recalls can be obtained by state-of-the-art detection algorithms. Unfortunately, satisfactory precision and localization accuracy are more difficultly achieved. We follow the intuitive notion that it should be easier to accurately detect an object from an image sequence than from a single image. We propose a novel two-stage technique which achieves improved detection results by applying temporal and spatial constraints to the occurrences of traffic signs in video. The first stage produces well-aligned temporally consistent detection tracks, by managing many competing track hypotheses at once. The second stage improves the precision by filtering the detection tracks by a learned discriminative model. The two stages have been evaluated in extensive experiments performed on videos acquired from a moving vehicle. The obtained experimental results clearly confirm the advantages of the proposed technique.

Keywords video analysis, object detection, object tracking, discriminative models, supervised learning, traffic signs

S. Šegvić, K. Brkić, and Z. Kalafatić
Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia
Tel: +385-1-6129 935 Fax: +385-1-6129 653
E-mail: name.surname@fer.hr

A. Pinz
Institute of Electrical Measurement and Measurement Signal Processing, Graz University of Technology, Austria
Tel: +43-316-873 5021 Fax: +43-316-873 7266
E-mail: axel.pinz@tugraz.at

1 Introduction

Automated detection and recognition of traffic signs is an exciting application field of computer vision, in which the research community and the industry have achieved significant recent progress [2,31,41]. The research in the field has been motivated by attractive applications such as driver assistance [18,19], intelligent vehicles capable of autonomous operation [14,35], and automated traffic inventories [31,48]. Traffic inventories have been employed as a tool in road safety inspection assessments [10,33] due to their capability to provide a comprehensive insight into the required state of a road. By periodically comparing the current conditions of the road against the reference state in the inventory, one can detect anomalies such as broken, covered, worn-out or stolen traffic signs, and erased or incorrectly painted road surface markings. Unfortunately, both i) the initial creation of the inventory and ii) the assessments themselves are very costly in terms of expert time, and that has recently spurred the interest in achieving at least partial automation of these two procedures [1,30, 6,31,48].

The uniform appearance of traffic signs lends well to the object detection paradigm based on binary classification within the sliding window [47,12,38,16]. The well-known approach based on cascading boosted Haar classifiers [47] is especially interesting due to fair detection accuracy and a very efficient use of computing resources [16]. In our experiments on Croatian warning and prohibition signs imaged from a moving service vehicle [48], properly trained boosted Haar cascades achieve about 95% recall for signs ranging between 26 and 80 pixels in scale. This significantly outperforms our implementations of more specific approaches based on pixelwise colour segmentation [31] and shape-based

detection such as radial symmetry [29], and Hough transform [21]. The efficiency of boosted Haar cascades enables near real-time performance of complex systems containing several additional components besides raw object detection, especially when modern multicore power is harnessed.

Nevertheless, we had to conclude that the raw performance of boosted Haar cascades is still insufficient for real applications, which due to safety implications require extremely high precisions, recalls and recognition accuracies. The two main shortcomings of traffic sign detection based on boosted Haar cascades are:

1. poor precision [8,9]: when a near 100% detection is desired, one typically obtains more than one false positive per processed image;
2. poor localization accuracy [48]: the detection responses often considerably deviate from the true location (this often disturbs the subsequent classification).

The problem of poor precision stems from the fact that it is quite difficult to capture the variation of the negative class in classification-based object detection. One way to deal with this problem would be to attempt to learn the background variation online, however, as a downside, that would require managing the inevitable label noise [38]. The problem is aggravated by the typical sparsity of the considered objects: traffic signs from the most common class A (triangular warning signs) appear in less than 5% of the frames in our videos.

Localization inaccuracies are intrinsic to general object detection algorithms which produce responses at many locations near the true positive detection. These responses are subsequently grouped by a heuristic geometric criterion without any relation towards the pixel-based evidence in the source image. Consequently, the final responses are often fairly inaccurate, especially for triangular traffic signs where fragments appear similar to the complete objects of the class (cf. Fig. 1).

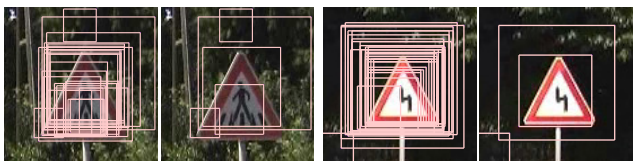


Fig. 1 The grouping algorithm may produce multiple responses in the vicinity of a sign. These responses often deviate from the true location (left). A dark background can be detected as the rim of the sign, which results in an additional oversized group of responses (right).

In this paper, we approach the problem of traffic sign detection by exploiting the fact that traffic signs are typically visible in many frames of the input sequence. We accomplish the detection goals of precision, recall and localization accuracy by exploiting spatio-temporal relations within the processed video. The main idea of the proposed technique is to extract temporally consistent detection sequences, which possess spatio-temporal properties typical for traffic signs imaged from a moving vehicle.

Temporal consistency is enforced by requiring that all detections have a consistent warped appearance throughout the sequence. In practice, we achieve this by combining the evidence from raw object detections [47] and differential tracker [44]. Since the raw detections randomly deviate from the true object location, we hypothesize a new detection sequence from many seed detections over the time. We finally choose the best hypothesis by employing the following two criteria: i) correct alignment towards the seed detection, and ii) count of confirmations from the raw detections. This approach rules out spurious raw detections which occur only in a small fraction of image frames, as well as most inaccurately localized raw detections which tend to be badly tracked due to changing background pixels. The details of the procedure are described in Sect. 5.

Even though insisting on temporal consistency considerably improves the precision, still many false positives remain. We notice that the image locations of properly placed traffic signs are strongly constrained, and that some additional constraints can be exploited by looking at (x,y,scale,t) trajectories obtained by tracking traffic signs over time (cf. Fig. 2). Consequently, we try to capture these spatio-temporal constraints by a discriminative model obtained by supervised machine learning. This procedure is detailed in Sect. 6.

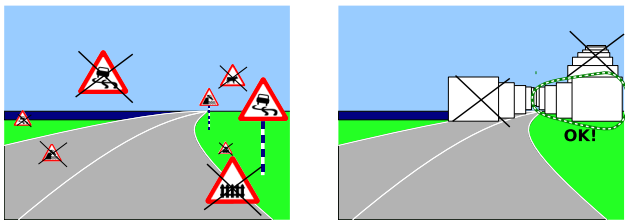


Fig. 2 Image locations of traffic signs are constrained with respect to scale and image coordinates (x,y) (left). Additional constraints exist on (x,y,scale,t) trajectories (right). The crossed-out signs and trajectories should not appear in images of roads which comply to relevant regulations.

The diagram of the proposed system for detection, tracking and recognition of traffic signs is shown in Fig. 3. The input image is processed by a suitably trained boosted Haar cascade to produce raw object detections. Raw detections are used as seeds for a cluster of track

hypotheses. When all hypotheses from a cluster die-off, we forward the most consistent hypothesis to the next processing stage. The module for enforcing the spatio-temporal constraints converts the tracks into fixed-length feature vectors corresponding to (x,scale) and (y,scale) trajectories. The tracks are subsequently classified into signs and non-signs by employing the trained discriminative model. Representative appearances of the remaining tracks can in the end be used for recognition.

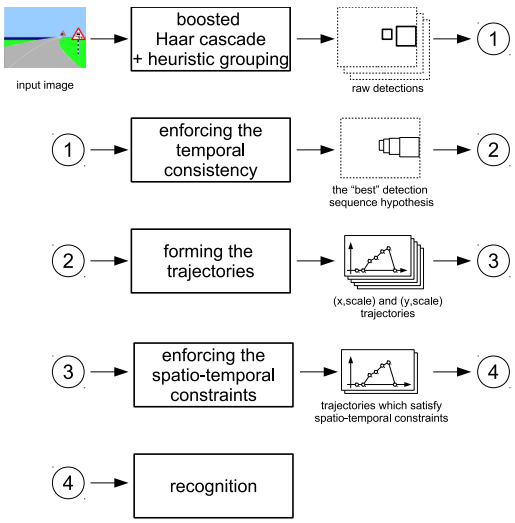


Fig. 3 The diagram of the proposed system for detection, tracking and recognition of traffic signs.

2 Related work

Although common application areas of traffic sign detection (e.g. driver assistance and road inspection) typically involve sequences of many image frames, we first review a large body of previous research which focuses on processing single images. These techniques are relevant since they can be integrated as lower level building blocks of a higher level temporal framework.

Appearance of a traffic sign is strictly constrained - its colour, shape, size and interior are prescribed by state legislation and intended to stand out from the environment. A logical way to design a detector is then to consider colour and shape cues.

Colour has been used as a dominant cue for traffic sign detection in the early studies [37,14]. Some state-of-the-art approaches [41,40,45] use colour as a component of their traffic sign detectors. Image segmentation is usually accomplished by comparing pixel values to a set of heuristically determined thresholds. However, we notice that the traffic signs colours significantly depend on lighting conditions, especially when video is acquired from a moving vehicle. Therefore, a single set of thresholds would unlikely be able to adequately segment all images. Some authors tried to alleviate this problem by choosing a colour space supposed to be less sensitive to illumination changes, usually HSI [37,34], or by observing the ratios between the colour components [14]. Gao et al. [20] used the CIECAM97 colour model, but still needed separate sets of thresholds for images captured under different weather conditions.

Shape-based approaches for traffic sign detection include Hough transform, fast radial transform and corner detection. Hough transform was used to locate straight lines or circles from which the outline of a sign can be inferred [21,41]. Motivated by the problem of traffic sign detection, Loy and Barnes [29] proposed a generalization of Hough transform to equiangular polygons, entitled fast radial symmetry transform. Having a detector for equiangular polygons means being able to detect many kinds of signs (circular, triangular, square, octagonal). Shape can also be inferred from corners. By performing corner detection, one can reason about the possible shapes in an image using the configurations of detected corners [14].

There are also approaches based on machine learning. For instance, Fang et al. [18] used neural networks to model the shape and colour of traffic signs.

After the milestone in face detection set by the work of Viola and Jones [47], there is an increased interest in applying cascades of boosted Haar classifiers to traffic sign detection. Baro and Vitria [4] detected signs using the extended feature set proposed by Lienhart and Maydt [27]. They further filtered false positives using fast radial symmetry transform [29] and principal component analysis. Bahlmann et al. [2] incorporated colour information into a cascade of boosted Haar classifiers by generating Haar-like features from several colour planes (R, G, B, normalized r, g, b, and grey-scale). In that way, the most discriminative colour features were extracted by machine learning. Similarly, Ruta et al. [39] used a cascade of boosted classifiers with colour-parameterized Haar-like features, computed from images with enhanced red, blue and green colours. The detector cascade is used to quickly establish regions of interest in an image and thereafter they detect equiangular polygons using an improvement of fast radial symmetry transform. Recent research on boosted Haar classification focuses on the design of new features: dissociated dipoles [5], polygonal features [36], and quantum features [24].

Boosted Haar cascade detectors usually produce multiple responses around the true target instances. The number of neighbouring responses can be used for discriminating true from false positives, but on the other

hand they deteriorate the localization accuracy. The responses are usually grouped by some heuristic algorithm and each group is substituted by a single compound detection region [27], which rarely fits the detected object well. Ruta et al. [41] presented a technique for improving the localization accuracy of a multiple-response detector based on mean shift clustering, which was modified to incorporate confidence measures of the detector's responses. A similar idea was presented by Grabner et al. [22], who used the *wobble* transform to increase the number of detector responses and then applied the mean shift clustering.

In our particular application of traffic sign detection, a single sign is visible through multiple frames, so the chances that it will be detected and recognized properly are much higher when taking spatio-temporal relations into account. Piccioli et al. [37] used a Kalman filter for temporal integration of the information extracted from individual image frames. Several other studies utilized Kalman filter tracking in order to reduce the search time for traffic sign detection [18,39,40]. Liu et al. [28] employed a feature tracker in order to establish correspondences between sign detections in consecutive frames. An alternative tracking method has been presented by Ruta et al. [41]. Their system learns a motion model from random affine transformations applied to the full-face view of a detected sign. The main role of the tracking subsystem is to reveal the affine transformation parameters and compensate the geometrical distortions of the sign appearance in order to make the detector pose-invariant.

In general, detection of objects in still images can benefit from learned spatial constraints between 3D scene geometry and the camera pose, as noted by Hoiem et al. [23]. This concept has been extended towards modeling *space-time* relationships for particular applications, for instance by exploring a mutual feedback scheme between spatial detection and temporal tracking of faces in crowds [11]. Ess et al. [17] address multi-person tracking in stereo video acquired from a mobile platform. The central contribution of their work is a graphical model for tracking-by-detection which combines image-based evidence obtained by 2D detection with spatio-temporal constraints provided by visual odometry and ground plane estimation. Similar to Ess et al., Timofte et al. [45] also require the temporal tracking of camera pose (they combine GPS and Structure from Motion), but they focus on stereo acquisition and on 3D spatial constraints to improve detection and recognition rates for traffic signs.

This paper explores multi-frame cues for suppressing false positive responses and improving the localization accuracy in traffic sign detection from a moving vehicle. Our approach is based on temporally consistent sequences of traffic sign appearances which we call detection tracks. We achieve improved localization accuracy by hypothesizing many detection tracks for each physical sign, and choosing the hypothesis supported by the strongest raw detection evidence. The surviving detection tracks are subsequently filtered by a discriminative trajectory model which succeeds to reduce false positive detections by enforcing 2D spatio-temporal constraints. Unlike previous approaches [45, 17] we avoid relying on SfM and 3D reconstruction, since, in single camera systems, these are prone to occasional instabilities. To the best of our knowledge, this approach has not been tried out yet in the context of traffic scenes.

3 Assumptions and datasets

The experimental part of this paper has been performed on production videos supplied by our industrial partner. The videos have been acquired in the scope of a commercial road maintenance assessment service which has been contracted by several Croatian counties since 2005. The service is chiefly concerned with signs which occur on the right side of the road, approximately perpendicular to the viewing direction. The videos are acquired by a higher-level consumer-grade camera mounted on the top of the service vehicle. The optical axis of the camera is roughly aligned with the longitudinal axis of the vehicle. Several cameras have been employed, but all of them feature the horizontal field of view of about 48°. Although positional readings are also recorded we do not employ them in this work for simplicity and generality. The provided materials contain more than 50 hours of compressed SDTV video (720×576 pixels) acquired on Croatian local roads, mainly in the countryside and in small cities. During the acquisition, the car has to drive at an adequate speed for the corresponding part of the road (40–60 km/h) in order not to disturb the regular traffic. The maximal resolution of the imaged traffic signs rarely exceeds 80 pixels, which in combination with the motion blur makes the traffic sign detection a challenging task. The road surface markings and the road texture vary considerably, so that an approach relying on road detection would not be straightforward.

In order to collect groundtruth samples for training and/or evaluating algorithms for object detection and recognition, we developed the application Marker[1]. The

[1] Marker can be freely downloaded and employed for non-commercial activities. The stable version (Marker v1.0) can be downloaded from: `http://www.zemris.fer.hr/~ssegvic/mastif/marker/marker.zip`.

purpose of this application is manual annotation of traffic sign locations in video frames (cf. Fig. 4). Annotating a sign consists of placing a tight bounding box around the sign and selecting the respective class. The application supports storing the annotated frames as individual images, along with the accompanying groundtruth in the form of a text file. Our annotation protocol establishes a guideline to annotate each physical sign at four different scales, from about 25×25 to the scale attained when the sign reaches the image border. Over the time, we collected a corpus of about 7000 annotations, which correspond to about 1750 distinct traffic signs.

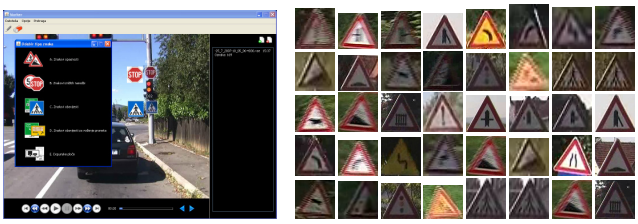
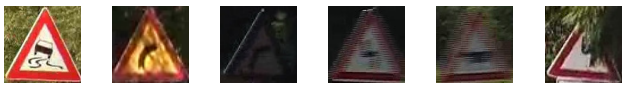


Fig. 4 A sign sample is annotated by (i) enclosing it in a bounding box, and (ii) selecting the sign type (left). Using this procedure we have manually annotated about 3000 warning signs (triangular shape, class A) (right).

In this work we focus at the superclass of triangular warning signs due to their prevalence in our videos. These signs correspond to the category A from the Vienna Convention [25], which is in most of Europe distinguished by a triangular shape with a thick red rim. We group about 3000 annotations of triangular signs from our corpus into two datasets: T2009 and T2010. The dataset T2009 contains about 2000 annotations collected in 2009 on videos acquired with an interlaced camera[2]. Some annotations from T2009 are shown in Fig. 4(right) and Fig. 5. Depending on the lighting conditions, the imaged colours of the red rim often get considerably desaturated, sometimes even to the point of being indistinguishable from the grey colour. The dataset T2010 contains about 1000 annotations of triangular signs collected in 2010 on videos acquired by a newer progressive scan camera. The signs from T2010 have better quality, however, the colour still appears unreliable. The resolution of the signs from both sets typically ranges between 20×20 and 80×80 pixels.

[2] Our datasets can be freely downloaded for the purposes of academic research:
`http://www.zemris.fer.hr/~ssegvic/mastif/datasets.shtml`.
We note that two similar datasets have been recently published. They can be accessed at `http://benchmark.ini.rub.de/` and at `http://www.cvl.isy.liu.se/research/traffic-signs-dataset` [26].



(a) (b) (c) (d) (e) (f)

Fig. 5 Sample images from the T2009 dataset: a normal sign (a), shadow (b), color inconsistency (c), interlacing (d), motion blur and interlacing (e), occlusion (f).

4 The low level algorithms

In this section we briefly present the two algorithms which have been employed as building blocks of the proposed technique. We first present the object detection algorithm based on a cascade of boosted Haar classifiers. Subsequently, we introduce the differential tracker with warp correction and checking.

4.1 Object detection by boosted Haar cascades

A cascade of boosted Haar classifiers [47] is a binary classifier suitable for object detection due to its favorable ratio of performance vs. computational complexity [16]. The cascade is applied within a sliding window over all feasible image locations and scales, and the detections are reported at all locations with a positive response. Each stage of the cascade is constructed by boosting the Haar classifiers obtained by exhaustive search on the common set of positive samples and a per-stage set of negative samples. Haar classifiers are chosen because they can be extremely efficiently evaluated with the help of integral image.

Viola and Jones [47] have noticed that the typical number of true negative responses (background) in any given image is immensely greater than the number of true positives (objects). Thus, it is usually possible to rule out more than 50% of the locations in the source image with a very simple boosted classifier. The key idea of the approach is to place such a simple boosted classifier at the first stage of the cascade, and then to assign gradually more and more complicated tests to the higher stages. The simplest boosted classifier is applied at all image locations, while the more involved classifiers are applied only where needed. Consequently, good performance is obtained at a small computational price.

The training procedure requires the common set of positive samples and a set of background images. At the first stage of the cascade the per-stage negative set is produced by sampling the specified quantity of random samples from the backgrounds. The subsequent stages are trained in a way to be smarter and more complex than the previous stages, by supplying ever harder per-stage negative sets. This is achieved by constructing the negative set for a given stage from back-

ground patches which pass all of the previous stages! Thus, the collection of the required number of negative samples gradually becomes more and more computationally demanding. In fact, the collection of negatives very soon begins to dominate the total training time, and the training with more aggressive parameters often needs to be aborted because the required number of negative samples can not be found in feasible time.

In our experiments, we used the implementation of the training procedure from OpenCV [27,7]. The program `haartraining` has the following main parameters: i) minimum hit rate (`mhr`), ii) maximum false alarm (`mfa`), iii) number of negative samples (`nneg`), and iv) number of stages (`nstages`). At each stage of the cascade, the training proceeds by adding a new Haar classifier until the constructed boosted Haar classifier reaches the specified parameters `mhr` and `mfa`. The training stops when the equivalent performance of the specified number of stages is reached. Best results have been obtained by selecting `mhr`=0.99, `mfa`=0.40, `nneg`=10000, `nstages`=14, and by choosing the basic set of features and Gentle AdaBoost [27]. These settings yield a total false alarm rate of $0.4^{14} = 2.7 \cdot 10^{-6}$. Depending on the parameters, the training typically took about one or two days on a recent computer with four cores (implicit multithreading is achieved by employing OpenMP). The selected Haar features from the first stage of a trained detector are shown in Fig. 6.

Fig. 6 The four features from the first stage of the cascade superimposed on typical images of traffic signs. The leftmost feature is sensitive to the bottom edge of a sign, while the other three detect the structure near the top of the sign.

4.2 Tracking with warp correction and checking

Point feature tracking is a technique for establishing correspondence between rectangular patches in neighbouring video frames. By chaining pairwise correspondences between neighbouring frames, one can construct point features trajectories. In applications which need to avoid error accumulation, the pairwise correspondences are corrected by aligning the current feature towards the reference appearance stored in the first frame of tracking [44].

Perhaps the most popular alignment approach[3] is based on Gauss-Newton gradient descent optimization as proposed by Lucas and Kanade [44,3,43]. The algorithm recovers the parameters $\hat{\mathbf{p}}$ of the warp which minimizes the pixel-wise error norm between the reference image $\mathbf{I}_\mathrm{R}$ and the warped current image $\mathbf{I}_\mathrm{W}$:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{\mathbf{x}} \|\mathbf{I}_\mathrm{W}(\mathbf{x}, \mathbf{p}) - \mathbf{I}_\mathrm{R}(\mathbf{x})\| \,. \tag{1}$$

Different warps have been used in various tracking contexts. A widely used tracking technique known as the KLT tracker [44] first employs a simple translation warp for recovering the displacement towards the previous neighbouring frame. Subsequently, the full affine warp is employed for correcting the alignment towards the reference image and for diagnosing bad convergence of the alignment procedure. In order to be able to tolerate large inter-frame camera motion, the translation tracking is usually performed across a suitable resolution pyramid.

The employed implementation of the KLT tracker derives from the public library maintained by Stan Birchfield at Clemson university[4]. The most important additions concern isotropic warp with contrast compensation, better sampling of large warped images [43], dynamic update of reference images, and improved tracking of small objects across the resolution pyramid. We have also introduced a number of smaller improvements in order to be able to smoothly integrate the library with the rest of our program.

In the implementation, the tracker resolution was set to 23×23 pixels, and the warp model was set to isotropic scaling with contrast compensation (5 DOF). The reference is updated with the mean of the last few appearances whenever the relative scale of the current feature with respect to the reference reaches 1.25. The introduced errors due to reference switching were not noticeable in informal experiments. The tracking is abandoned whenever the RMS residual between the current appearance and the reference becomes greater than 15. Such relatively high threshold was chosen due to fine texture in some signs (e.g. pedestrian crossing) which can not be accurately tracked with constant enlargement of the current feature. The tracking would be considerably easier if we tracked backwards in the video, because then the reference would hold more information than the current image. However we do not do that in order to preserve the generality of the approach.

[3] There is a recent alignment approach which promises better performance [46].

[4] URL `http://www.ces.clemson.edu/~stb/klt/`

5 Enforcement of temporal consistency

In this section we describe our procedure for extracting well-localized temporally consistent detection tracks originating from a distinct physical object. The proposed approach is based on averaging out the uncertainty of individual detection responses (cf. Fig. 1) by accumulating evidence from many consecutive frames. In order to be able to relate the evidence recovered in different frames, inter-frame correspondence must be established. Differential tracking appears as a suitable candidate for this task since most ideogram-based signs give rise to good features to track [44]. However, in videos acquired at conventional driving speeds, the inter-frame point-feature displacements are often greater than the dimensions of the traffic signs. This unfortunately precludes straightforward applications of existing tracking algorithms (especially if we wish to approach 100% recall) and calls for more involved solutions.

One approach to deal with large inter-frame displacements would be to predict current feature positions by employing a geometric model. This model would comprise parameters such as camera placement, car motion, road geometry and the sign displacement. However, the evolution of such model would be very complicated [13] while the errors in its estimation might eventually hurt the gains.

Instead, we propose a simpler, pure 2D approach, in which the feature positions are predicted at raw detection responses[5] whenever the tracker fails to converge. Note that this is especially effective in the *second* frame of the detection track, when previous feature dynamics is unavailable. In most other frames of the detection track the tracker succeeds to establish the correspondence by relying on extrapolation-based prediction, which alleviates the impact of false negative detections. The proposed approach achieves a synergy between the tracker and the detector, since it simultaneously achieves: i) temporal consistency due to tracking, and ii) inter-frame displacement tolerance due to detection. If the tracked reference patch does not cross the boundary of a planar object, the failures occur extremely rarely: only on simultaneous divergence of a tracker and a complete miss-detection. In order to make this approach work with non-rectangular sign classes, we internally track a suitably displaced rectangular patch in the interior of the sign, as annotated by the interior red rectangle in Fig. 15(b). We do not see this as a particular shortcoming, since this displacement can be easily learned from the training set.

[5] This is exactly oposite from [11], where the detector uses predictions provided by the tracker

Each detection track is bootstrapped by seeding the tracker within a suitable raw detection. We have no way of assessing the localization accuracy of particular raw detections, and so a question arises: Which raw detection to track? In order to avoid the trap of committing to a certain raw detection too early, we hypothesize many detection tracks and manage them concurrently. Thus, a new track hypothesis is initialized whenever there is a raw detection which is more than a few pixels away from any of the existing tracks. For each track, we keep count of "confirmations" as the number of subsequent detections in the immediate vicinity of the current feature location. In order to save computing resources, we terminate the tracks which are seldom confirmed or poorly aligned with respect to the reference.

In order to be able to identify distinct physical objects we associate nearby detection tracks into clusters. At each given moment we can tell which track hypothesis from the cluster is likely to be more accurate than the others by looking at the number of confirmations and the age of the feature measured in frames. The definitive decision is postponed until all tracks from the cluster die off, after having processed all available evidence. Thus, the proposed approach chooses the best among many autonomously evolving hypotheses for each physical object, somewhat in the spirit of a particle filter.

We note that the detection tracks which are initiated at inaccurately localized raw detections are unlikely to receive a high score due to the following two reasons. First, such tracks are likely discarded due to bad alignment, since their reference image contains background which most often changes due to motion parallax. Second, such tracks receive less support from the raw detections, at least if the detector is not heavily biased. Thus, besides enforcing the temporal consistency, the proposed approach additionally improves the localization accuracy of the detections.

In the implementation, the decisions are based on distances between the tracked features and the raw detections. An attempt of resuming a lost feature at a raw detection position is performed if the distance between the previous feature position and a raw detection is less than `thResume`. A new track is seeded at each raw detection which is farther than `thNewFeature` from all existing tracks. A new cluster is started whenever a new detection track is farther than `thNewCluster` from all other detection tracks. A confirmation is recorded at each detection track whose current position is closer than `thConfirm` to some raw detection. In most cases we employed the distance metric which measures a normal-

ized overlap between the two windows d_i and d_j:

$$\text{distance1}(d_i, d_j) = 1 - \frac{\text{area}(d_i \cap d_j)}{\max(\text{area}(d_i), \text{area}(d_j))} \ . \quad (2)$$

A shortcoming of this function is that it returns 1 for all disjoint windows, which is problematic for locating a detection for resuming the tracking of a lost feature. Consequently, when the windows are disjoint, we instead employ a scale-normalized Euclidean distance [38] with a penalization factor on scale difference. In all experiments, the four thresholds were set to: `thResume`=3, `thNewFeature`=0.2, `thNewGroup`=0.6, and `thConfirm`=0.3. Note that `thConfirm` is greater than `thNewFeature` in order to avoid starvation of competing hypotheses. In the end, the cluster is accepted for further processing only if its best hypothesis has more than 5 confirmations, and the total increase in feature scale is larger than 1.25.

To conclude, the proposed approach generates well-localized temporally consistent detection tracks such as the ones shown in the bottom row in each subfigure of Fig. 13 and Fig. 14. In most cases we obtain only one detection track for each physical object. By requiring temporal consistency, the approach suppresses many false positive responses. Due to evaluation across many images, the resulting detection tracks are better localized than the original raw detections. Experiments which confirm the above considerations are presented in 7.2.

6 Enforcement of spatio-temporal constraints

In this section, we explore the possibilities of learning spatio-temporal constraints which govern the behavior of traffic signs in our videos. A traffic sign typically appears somewhere to the right in the video frame, at a learnable scale and location, and increases in size and finally exits the scene in a learnable manner. Learning these constraints requires temporally consistent detection tracks and can enhance the quality of the detection process.

6.1 Representing temporally consistent detection tracks

Enforcing temporal consistency significantly reduces the number of false positive detections. Nevertheless, a number of false positives are still tracked through multiple frames, as any false positive with a stable background is a candidate for successful tracking. In order to further reduce the number of false positives, we take into account not only the temporal consistency of a sign candidate, but also the spatio-temporal constraints valid for traffic signs. As illustrated in Fig. 2, traffic signs appear at specific combinations of scales and image locations. The expected behavior of a traffic sign through time is to appear at a predictable position, increase in size and finally exit the scene in a predefined manner. If one could check whether these spatio-temporal constraints are valid on the obtained detection tracks, one could infer whether an object is a sign or a false positive. However, the actual constraints are subject to parameters such as: i) geometry of the road ahead, ii) dynamic position of the car with respect to the road (lateral displacement, orientation), iii) the actual placement of the sign with respect to the road (lateral displacement, height of the pole, orientation), iv) the physical size of the sign. All of the parameters above may vary, and it is hard to tell in advance within which intervals. Hence a machine learning approach appears appropriate.

In order to learn typical spatio-temporal behavior of a traffic sign, we use a training set consisting of labeled detection tracks. Each track belonging to a sign is hand-labeled as a positive, and each track belonging to some other object as a negative. To formalize, a detection track with identifier id is a set of detections:

$$\tau_{\text{id}} = \{d_t\}_{t=1}^{\text{nframes}}, d_t = (x_t, y_t, w_t, h_t) \ , \quad (3)$$

where d_t denotes a detection in frame t. A detection is a tuple (x_t, y_t, w_t, h_t), with x_t and y_t being the coordinates of the upper left corner of the detection rectangle, and w_t and h_t being the width and the height of the rectangle. In this paper, the detector is configured so that the width and the height of the detection rectangles are always equal. For simplicity, in the rest of the paper we refer to them as scale. It is important to notice that different tracks might consist of a different number of detections, i.e. be defined for a different number of frames. For instance, if a car is driving fast, we expect to see a sign in fewer frames and therefore obtain a shorter track than if it were driving slowly. In the rest of the paper, we refer to this number of detections as the length of the track.

Fig. 7 shows the tracks of three different signs. The z axis shows the passage of time and corresponds to the ordinal number of a frame. The figure illustrates that detection tracks contain a lot of important spatio-temporal information which we could learn. Just by looking at Fig. 7, it is easy for a human to conclude where a traffic sign typically appears in a video, for how long, at which scales and how its size changes through time.

It is our ambition to build a classifier which would be able to distinguish valid tracks from invalid ones, hence implicitly inferring all these important cues. In building a classifier, the key problem is always finding a way to represent the input data as feature vectors.

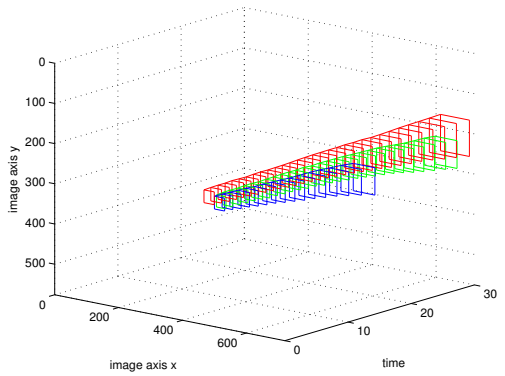


Fig. 7 Tracks of three different signs. The z axis ("time") corresponds to the ordinal number of the detection in the detection track.

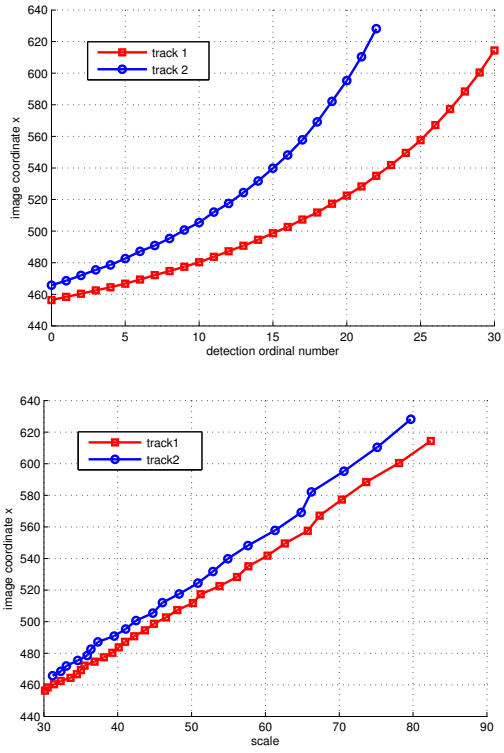


Fig. 8 Plots of x coordinates of two tracks dependent on time (above) and scale (below). Plotting against scale results in a much more consistent representation.

Data which are semantically similar should be close in the feature vector space. In our case, the data are labeled detection tracks. If all tracks had the same length, i.e. consisted of the same number of detections, it might be feasible to directly use them as feature vectors. However, this is not the case. Hence, we need to find a way to convert them into vectors of fixed length. These feature vectors have to be meaningfully comparable. To illustrate, consider Fig. 7 again. A problem arises when plotting the detection rectangles against time: if one track starts with a relatively small detection rectangle, and another with a large one, both rectangles will be plotted at $t = 0$. If comparing these rectangles, one might conclude that they belong to two different objects, one smaller and one larger, while in fact they belong to the same sign which was detected at different distances from the camera! To avoid this problem, our feature vectors should be normalized both for scale and for speed. For scale, the corresponding elements of two feature vectors should represent same scales. For speed, the length of the original tracks, which depends on the speed of the car, should not influence our feature vectors.

To solve this normalization issue, we propose to represent x and y coordinates of a track as dependent on scale [9], rather than on time. This is illustrated in Fig. 8 where two trajectories are drawn. In the top figure, the x and y coordinates of the detections in a track are plotted against time, i.e. the first point of the first track corresponds to the detection in the first frame, the second point to the detection in the second frame and so on. As the car is driving faster for track 2, the tracks start to diverge approximately after $t = 10$. Fig. 8 (bottom) shows how this problem can be addressed by plotting the values of x and y against scale. When using scale in this manner, the flow of time is implicitly shown, assuming that the scales of the detection rectangles increase monotonically. If the car were driving faster, our curves would be defined with fewer points, but corresponding scales would still have corresponding coordinates. This representation is more suitable for extracting fixed-length feature vectors.

A straightforward way of extracting the vectors is to sample all trajectories at a predefined set of scales. For instance, when viewing Fig. 8 (bottom) one might consider sampling the values of x and y for scales 50, 60 and 70 and building a feature vector out of those values. But it is unsafe to assume that all tracks belonging to traffic signs will contain detections of these scales. A sign might be occluded and then reappear - perhaps because a tree is blocking the view, or a parked car is in the way.

As a solution to this problem, we first scan the entire training set of tracks and find the minimal and maximal scale of the union of all positive tracks appearing in the set. We then construct a set of sampling scales $\{s_i\}_{i=1}^{n}$ by dividing the range between the minimum and the maximum scale into $n-1$ equally spaced intervals. In order to convert a track to a feature vector, x and y coordinates of a track are sampled for scale values $\{s_i\}$. As the values $\{s_i\}$ are computed to be equidistant, they

will be arbitrary numbers which satisfy the condition that they are within the acceptable scale range. However, there is no guarantee that there are detection rectangles in the training set with the scale exactly equal to one of the values $\{s_i\}$. For instance, the set of sampling scales might be $\{50, 60, 70\}$ and we might have a track with detection rectangles of scales 48, 52, 58 and 64. This can be viewed as a problem of missing values [32]. We propose to solve it using the following three methods:

1. linear inter/extrapolation
2. cubic inter/extrapolation
3. linear interpolation with zero imputation

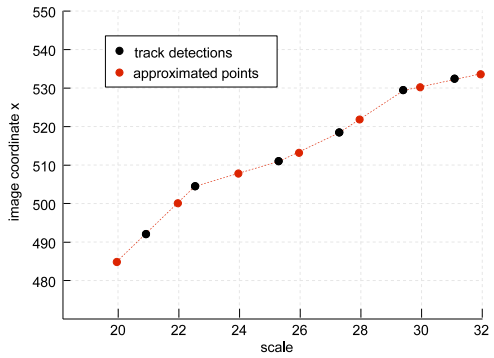


Fig. 9 Linear inter/extrapolation over a set of points. Out-of-interval estimates are obtained by drawing a line through two nearest data points.

An illustration of the first method is shown in Fig. 9. The horizontal axis shows the scale range. Sampling scales are $\{20, 22, ..., 32\}$. The vertical axis shows the image coordinate x for different scales. Small black dots denote measurements taken from the detection tracks, while red dots denote interpolated points. For sampling at values which fall within the range of known scales $(22 - 30)$, linear interpolation is used. For values which lie outside this range (20 and 32), we use linear extrapolation. Linear extrapolation is a method for constructing new data points which lie outside a discrete set of known data points. When constructing a point x_* lying outside the known interval, a line is drawn through the two nearest known points, (x_{k-1}, y_{k-1}) and (x_k, y_k), and the missing value is obtained as

$$y(x_*) = y_{k-1} + \frac{x_* - x_{k-1}}{x_k - x_{k-1}}(y_k - y_{k-1}) \quad (4)$$

Our second method is analogous to the first one, except we use cubic inter/extrapolation. Cubic extrapolation is similar to linear extrapolation, but a third order polynomial is used instead of a line.

Linear and cubic extrapolation tend to be sensitive to measurement noise. Furthermore, the assumption of the linearity of consecutive measurements does not hold if the track is obtained while the car is driving down a curve. As linear extrapolation is performed by constructing a line through two nearest data points, measurement noise at these two points can significantly alter the slope of the line, leading to errors in point estimation. An example of linear and cubic extrapolation on a real detection track is shown in Fig. 10. Cubic extrapolation might work better on this particular example if more data points were used for constructing the third-order polynomial. However, it is difficult to decide exactly how many points one should use. If one used all the available points and found a third-order polynomial which models them nearly perfectly, one would risk chaotic out-of-interval behavior due to overfitting the known data. If too few points were used, one would risk poor estimation, such as in Fig. 10 (bottom).

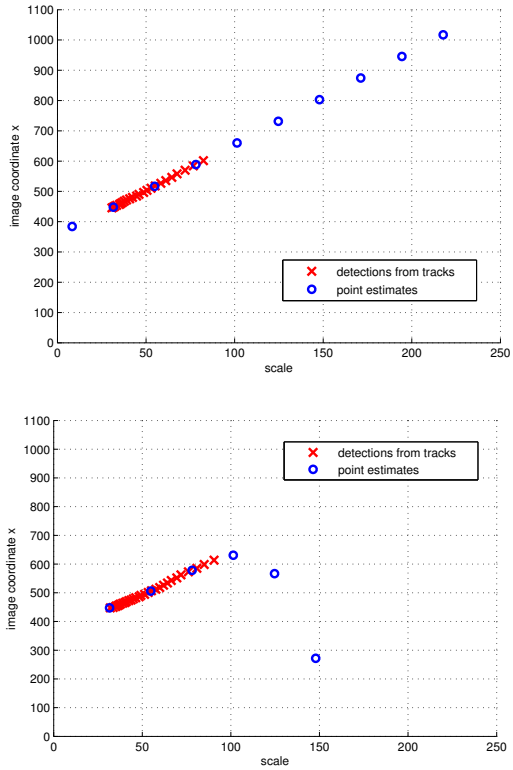


Fig. 10 Linear inter/extrapolation (above) and cubic inter/extrapolation (below) of a real track taken from the training set. Red crosses indicate real measurements obtained from detection tracks, and blue circles represent estimated values. Notice the noise sensitivity in the case of cubic extrapolation.

To avoid extrapolation problems, we propose our third method - a combination of linear interpolation and zero imputation. We use linear interpolation for

sampling values lying inside the known interval, and zero imputation for out-of-interval values. Zero imputation is a common statistical technique for dealing with missing values [32]. In general, imputation is the substitution of a missing data point by some predefined constant. Zero imputation simply assumes that the value of the missing point is zero. An illustration for this method is shown in Fig. 11.

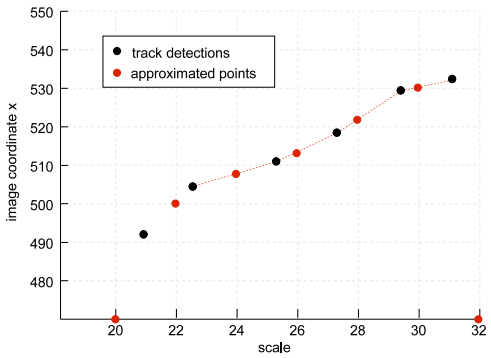


Fig. 11 Linear interpolation with zero imputation over a set of points. Out-of-interval values are set to zero, while the remaining values are interpolated.

6.2 Learning valid detection tracks

The three methods outlined above are directly used to construct feature vectors from tracks. First we find the minimal and the maximal scale appearing in the training set. We then construct a set of sampling scales $\{s_i\}_{i=1}^{n}$, as described in the previous section. Next, each track is represented in x-scale and y-scale coordinate systems. In case some tracks have multiple detections at the same scale, the detections are averaged to form a single detection[6]. For each track τ_{id}, values of x and y are sampled at scale values $\{s_i\}_{i=1}^{n}$. Thus we obtain two sets of interpolated coordinates, $\{x_i\}_{i=1}^{n}$ and $\{y_i\}_{i=1}^{n}$. The feature vector φ_{id} describing a track τ_{id} is then constructed as:

$$\varphi_{\mathrm{id}} = (x_1, x_2, ..., x_n, y_1, y_2, ..., y_n)^T \tag{5}$$

To train a classifier, a training set consisting of k labeled tracks is converted into a set of k feature vectors $\{\varphi_i\}_{i=1}^{k}$ with corresponding labels $\{c_i\}_{i=1}^{k}$, $c_i \in \{+1, -1\}$. The number n of sampling scales $\{s_i\}_{i=1}^{n}$ will determine the dimensionality of the feature vectors. As for each scale we obtain two coordinates, x and y, the length of the feature vector will be $2n$.

To test the classifier on a new track, the track needs to be converted to a feature vector. The conversion needs to be carried out using the same sampling values $\{s_i\}_{i=1}^{n}$ as in the training phase.

[6] This occasionally occurs in sharp turns for some false positive detection tracks.

7 Experimental results

In this section, we thoroughly evaluate the components of our detection system. We build upon a simple boosted Haar cascade detector, studying the influence of enforcing temporal consistency and spatio-temporal constraints on the total number of false positives it produces. We also show how introducing temporal consistency influences track quality in terms of reduced object localization error, and we explore the inner workings of our temporal consistency subsystem on six hard cases from our training set. Finally, we demonstrate that good traffic sign tracks obtained by our system can be useful in motion-based background segmentation.

7.1 Results of raw detection

In this subsection we briefly review our results on raw detection of triangular traffic signs [8,48]. Best detection results have been obtained by boosted Haar cascades [47] with classic native resolution of 24×24 pixels. The evaluation was performed on a subset of T2010 containing 918 annotations larger than 25×25 pixels, (the detection is considerably less accurate for small signs). The achieved detection performances are summarized in Table 1, depending on the count of training samples $\mathrm{N_{pos}}$. $\mathrm{N_{bg}}$ and $\mathrm{N_{test}}$ denote the number of employed background images and the number of evaluation samples. The table shows that boosted Haar cascades achieve quite encouraging recalls when enough training samples are available. The best results are located in the bottom row of the table where we train the detector on the entire interlaced dataset T2009.

Table 1 Impact of the training set to the detection performance of the boosted Haar cascade

$\mathrm{N_{pos}}$	$\mathrm{N_{bg}}$	$\mathrm{N_{test}}$	recall	precision
352	110	72	68%	46%
898	230	918	80%	64%
2154	711	918	96%	54%

Besides the precision, the localization accuracy also leaves to desire. The distribution of the localization error obtained in detection responses for the set T2010

is shown in Fig. 12. Such deviation has been found responsible for a 12 percent points decrease in recognition accuracy [48]. The deviation is expressed as normalized non-overlapping area defined in (2). The total number of samples is 994. The mean of the distribution is at 0.18, which means that an average raw detection has an 82% overlap with the groundtruth.

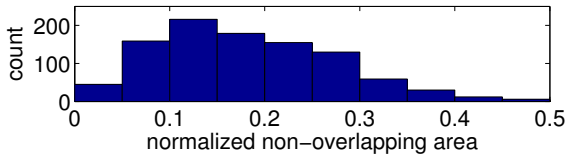


Fig. 12 Distribution of the localization error in T2010 responses.

7.2 Extracting temporally consistent detection tracks

This subsection presents experiments targeting the proposed approach for extracting temporally consistent detection tracks. We first present experiments providing a qualitative insight into the inner workings of the approach in 7.2.1. Next, in 7.2.2 and 7.2.3, we provide quantitative experiments which evaluate the detection performance and the localization accuracy. Finally, we discuss some additional benefits of the approach in 7.2.4.

7.2.1 Qualitative experiments

The proposed approach for extracting temporally consistent detection tracks chooses the most prominent track hypothesis after having collected all available evidence. Typically, this occurs when the object is about to leave the field of view. The choice involves determining a relative majority of votes, which makes it possible to identify candidates which may be supported in less than 50% of the individual frames. This is illustrated in Fig. 13, for two different traffic signs. The top row in each subfigure presents a hand-picked detection chain consisting of the most accurately localized raw detection responses in the corresponding frame[7]. The bottom row in each subfigure shows the chosen detection track hypotheses. We observe that the temporal consistency of the raw detection chains is rather poor: they oscillate around the true object position, while the size at times decreases along the sequence. On the other hand, the detection tracks are quite consistent, while their sizes increase monotonically.

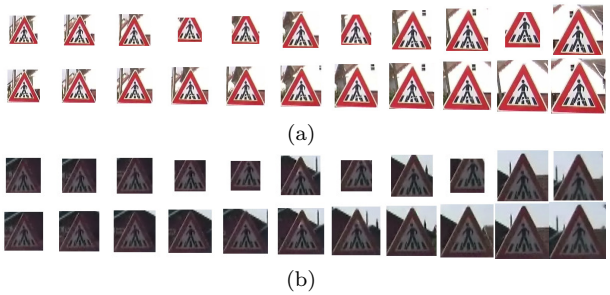


Fig. 13 Each of the two subfigures (a-b) shows the best hand-picked raw detection chain (top row) and the corresponding elements of the chosen detection track (bottom row). The approach tollerates ill-localized raw detections when the relative majority supports the correct hypothesis.

We illustrate the competition between the hypotheses by two experiments shown in Fig. 14. As in Fig. 13, columns of the two subfigures correspond to distinct video frames, while the top row shows the best hand-picked detection chain. The bottom rows of each subfigure show two hypotheses from the same cluster, which received the highest support from the raw detections (typically, tens of hypotheses are managed for each traffic sign). In both cases a better localized hypothesis happened to be initialized *after* a close competitor. This is unfavourable for the better hypothesis since an earlier initialization implies more opportunities to obtain support. However, in both experiments the more accurate hypothesis eventually gathered more support, which means that at a certain point a hypothesis switch had occured.

These experiments show that the proposed approach succeeds to favour more accurately located hypotheses. The approach works since, statistically, the detector responses are likely to be more dense near the true object location. Specific background structure may temporarily disturb the detector, however, on a larger temporal scale, these disturbances often cancel out [22], especially when the raw detector is well trained. Thus, the approach succeeds by providing more opportunities to average-out the systematic error of the raw detector.

7.2.2 Case study: the six traffic signs

We have performed automated experiments on the evaluation video of about 130000 frames, by employing the annotated groundtruth corresponding to the previously introduced dataset T2010. The results show that temporal consistency reduces the count of false positives

[7] One could attempt to identify such detection chains by carefully associating raw detection responses across the neighbouring frames. However, that may be difficult to achieve, especially when raw detections are missing in some frames, and in the presence of multiple responses. The proposed approach makes such blind association unnecessary, while at the same time achieving a better localization accuracy.

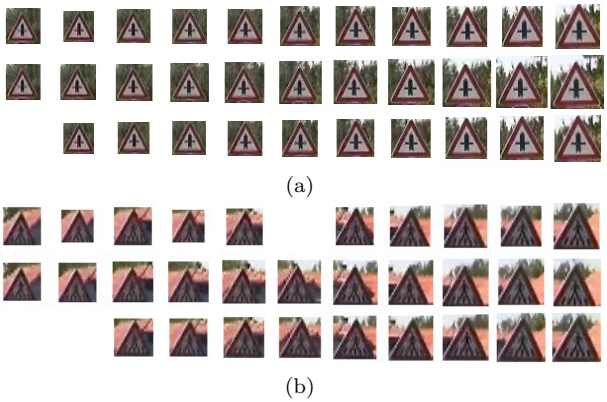


Fig. 14 The two subfigures (a-b) illustrate competition between the track hypotheses. The top row in each subfigure contains the best hand-picked chain of raw detections, while the bottom two rows show two distinct hypothesized tracks belonging to the dominant cluster. In both experiments the better localized bottom hypothesis receives more confirmations, despite being initialized later than the competing middle row hypothesis.

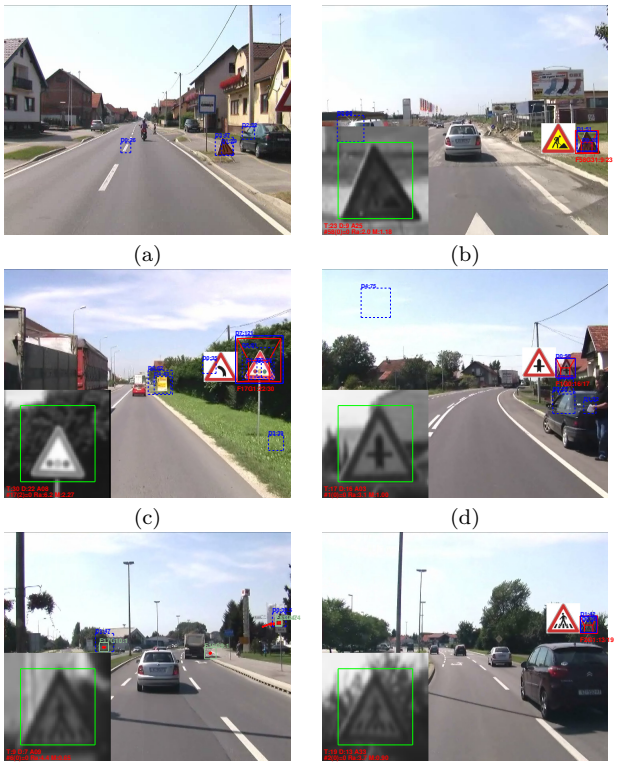


Fig. 15 Case studies of the six difficult instances identified by automated testing. Please see the text for details.

when compared to ad-hoc raw detection chaining, while preserving near-100% recall. The six most difficult cases out of about 250 traffic signs in the test video are shown in Fig. 15. In all of the six images, red lines indicate dominant track hypotheses for each cluster, if available. The current position is denoted by the thick outermost rectangle, while the thin innermost rectangle shows the position of the patch which is actually tracked. Blue rectangles designate raw detections. By default, they are rendered with dashed lines, while solid lines emphasize support of the dominant hypothesis. The warped appearance of a local neighbourhood around each track hypothesis is shown in the bottom-left angle of the image. The green frame shows the position of the tracked sign in the neighbourhood. The first line of text below warped appearance shows the count of frames in which the object was tracked (T), and the number of raw detection confirmations (D). The second line indicates the tracking status, the current alignment residual towards the reference (Ra), and the scale of the feature (M) where M=1 means 45×45 pixels. Where applicable, the result of 1-NN classification in the LDA subspace [15, 48] is shown besides the track hypothesis. These conventions have also been used in the accompanying video.

The sign in Fig. 15(a) produced raw detections in only three frames, probably due to the fact that it was inclined for about 30%. The pyramidal translational tracker did not succeed to recover the first displacement due to considerable image motion and the tracking had not even started. The sign in Fig. 15(b) was successfully tracked and recognized despite being inclined. Our training set did not contain inclined signs, so we think these results are acceptable. The sign in Fig. 15(c) was detected, however the localization is inaccurate. This is caused by the effect also shown in Fig. 1(right), whereby smooth dark background appears to the detector as the rim of the sign. Correct position of the sign is also detected, but since the oversized detection was more frequent we were out of luck. The recognition is of course incorrect. We believe that this is the only large localization error obtained on this video. The bottom sign in Fig. 15(d) could not be detected due to an occlusion by a stopped car in the emergency lane. The top sign was correctly detected, tracked and recognized. The sign in Fig. 15(e) was extraordinarily far from the camera, since the vehicle left the rightmost lane. Due to a temporary disturbance, the tracking started a little later than it may have in the ideal case, so that only 7 confirmations were recorded. The requirement that there should be at least 1.25 total increase in scale was not met, and thus the detection was not reported. Please note that the image in Fig. 15(e) has been rendered in a special debug mode of the program whereby track hypotheses are shown in green. The debug mode also causes the warped current appearance to be shown despite the fact that the sign is not officially detected as implied by the absence of the thick

red border. The sign in Fig. 15(f) has been correctly detected and recognized despite the distance.

7.2.3 Evaluation of the localization accuracy

We continue the qualitative discussion on localization accuracy from 7.2.1, by automated quantitative evaluation on the dataset T2010. In this experiment each annotated sign is simultaneously compared both with the closest raw detection, and with the closest element of a detection track[8]. In order to make a fair comparison, we disregard the annotations for which either of the two rectangles is missing or deviates too much. This resulted in 300 discarded annotations which mostly correspond to small image plane rectangles. The obtained results are shown in Fig. 16. The figure shows empirically obtained distributions of the localization error in the raw detection responses (the top graph) and in the detection tracks (the bottom graph). As in 7.1, the deviation is defined by eq. (2). The total number of analyzed samples is 694. The mean of the raw detection deviation is at 0.17 while the track deviation mean is at 0.12. This implies that the detection tracks are considerably better localized than the raw detection responses.

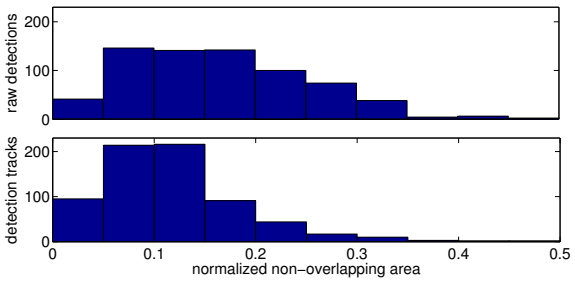


Fig. 16 Quantitative comparison between the localization accuracies of the raw detection responses (top) and the detection tracks obtained by the proposed approach (bottom).

7.2.4 Discussion of other benefits

Besides being useful for pruning false positive detections, the proposed detection approach could also improve the recognition accuracy. As mentioned in 7.1, our previous research indicates that better localization implies considerably better recognition. Additionally, temporally consistent detection sequences are more easily checked for spatio-temporal constraints than the raw detection chains, due to reliable scale. These experiments are discussed in section 7.3. Finally, temporally consistent detection sequences offer interesting potential for reliably solving the foreground/background segmentation which can serve as an additional recognition cue. Preliminary results along that line of research are presented in section 7.4.

[8] In other words, the comparison assumes that we would always be able to select the better detection when multiple responses are present.

7.3 Classifying detection tracks

Enforcing temporal consistency in detection tracks reduces the number of false positive detections, but some false positives still remain. In this section we investigate how the false positive count can be further reduced by adding more spatio-temporal constraints which hold for traffic signs, but not for false positives. The idea is to exploit the fact that a traffic sign typically appears at predictable positions in the image and at predictable scales. This is achieved by training a classifier which would discriminate between detection tracks of true traffic signs and detection tracks of false positives, based on the positions and the scales of the elements in the detection track.

In order to evaluate different classifiers, we have collected a set of 268 positive and 601 negative hand-labeled detection tracks derived from a video in which dataset T2010 was annotated. The tracks are converted into feature vectors (cf. subsection 6.2) using 10 sampling scales $\{s_i\}_{i=1}^{10}$. The feature vectors are then input into the following classifiers[9]:

- AdaBoost with decision stumps as base classifiers
- Random forest with varying numbers of trees and random features
- Bayesian network using a simple estimator which estimates probabilities directly from data and hill climbing algorithm K2
- Multilayer perceptron with varying numbers of hidden layers and corresponding neurons

Our main goal in track classification is retaining almost all true positive tracks, while discarding a maximal number of false positives. In spirit of that requirement, table 2 compares the false positive rates obtained by employing i) linear inter/extrapolation[10] and ii) linear interpolation with zero imputation, when the true positive rate is set to 0.98. The obtained rates correspond to percentages of false positives if we allow 2% of positive samples to be classified incorrectly. The table shows that imputation achieves better false positive rates for all considered classifiers.

[9] The data mining tool Weka [49] was used for the experiments.

[10] The results obtained for cubic inter/extrapolation are worse than the results for linear inter/extrapolation, so we are omitting them.

Table 2 Classification results of feature vectors constructed by linear inter/extrapolation and zero imputation with linear interpolation. False positive rates are shown, assuming a true positive rate of 0.98.

Classifier	False positive rate	
	extrapolation	imputation
AdaBoost	0.53	0.24
Bayesian network	0.58	0.18
Multilayer perceptron	0.97	0.29
Random forest	0.37	0.22

All rates in the table have been obtained on a randomly chosen evaluation set with 53 positive and 120 negative detection tracks. Control parameters for each classification method have been obtained by grid optimization with respect to ten-fold cross-validation performance on the training set. The training set consists of 696 tracks, with the negative to positive track ratio of 2.24.

We are allowing a misclassification of 2% because our training set has several true positives whose spatio-temporal configuration is not typical and is therefore very hard to learn. For example, there is only one instance of a traffic sign placed on the left side of the road, one sign is placed on an extremely high pole, while some signs are placed behind the bicycle track which makes them appear unusually small.

Most classifiers have trouble classifying feature vectors derived by linear extrapolation, and the false positive rates are much better with zero imputation. We believe this is due to the sensitivity of the x/scale and y/scale relationships to the local variations of road curvature, which can lead to unreliable point estimates (cf. subsection 6.2). By simply setting the unknown points in a feature vector to zero, the classifiers get the chance to learn that zero indicates a missing value [32]. All classifiers perform better with zero imputation, which is visible by comparing false positive rates in table 2. The same applies to areas under the ROC curves.

Other experiments have shown that setting the number of sampling scales to 5 leads to degraded recognition performance, although not all methods are equally affected. On the other hand, increasing the number of sampling scales does not significantly alter the results.

The best classifier manages to discard 82% of false positive detections while retaining 98% of true positives. The resulting area under the ROC curve is 0.96. We believe that even better reduction rates can be achieved by expanding the training set.

Fig. 17 shows ROC curves of selected classifiers: Bayesian network with linear extrapolation and imputation, Multilayer Perceptron with imputation, and Random Forest with extrapolation. ROC curves of classifiers working with zero imputed feature vectors cluster above the classifiers employing linear extrapolation.

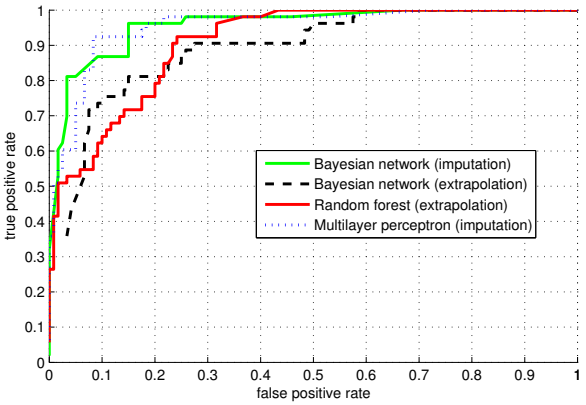


Fig. 17 ROC curves of selected classifiers: classifiers which employ zero imputation perform better then their counterparts relying on linear extrapolation.

7.4 Motion-based background segmentation

Consider the two similar traffic signs in motion depicted in Fig. 18. Due to their similar greyscale appearance, it would be hard to discriminate them by only considering individual image frames. This task would be much easier if one could consider a motion sequence, since the two signs have different occlusion boundaries.

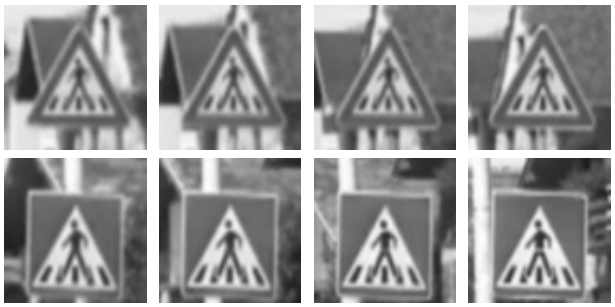


Fig. 18 Two traffic signs which are quite similar in grayscale appearance are easily distinguished in a motion sequence due to different occlusion boundaries. The shape difference is noticeable only when the two objects are observed in motion.

Multiple views onto a rigid moving object can provide rich cues about the 3D shape of the object. In order to exploit these cues, individual views somehow need to be put into correspondence. However, correspondence between the detection tracks considered in this paper is temporally consistent by design (this is

relatively easily achieved since the signs are rigid and flat). Thus we have all preconditions for detecting the occlusion boundaries of the tracked signs and determining shape masks such as those shown in Fig. 19 (right).

Fig. 19 Two similar signs with different shapes (left). The corresponding shape masks recovered from the motion sequences shown in Fig. 18 (right).

The shape masks can be recovered by looking at the variance of each particular pixel in the warped neighbourhood around the tracked sign [42]. Since we actually track only the interior of the sign we can be confident that the variance does not disturb the convergence of the tracker. Surprisingly encouraging results have been obtained in many cases, especially in urban areas where the backgrounds of the traffic signs usually contain significant structure. The results tend to be less interesting outside the cities where the backgrounds often contain featureless areas such as woods or the sky. The obtained shape masks for the first 9 traffic signs in the test video are presented in Fig. 20.

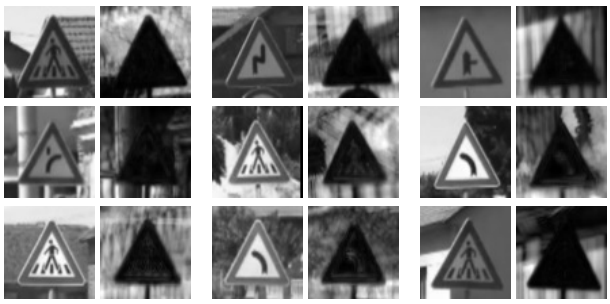


Fig. 20 The recovered shape masks for the first 9 traffic signs of the test video. For each traffic sign we show the warped neighbourhood in the last frame of the detection sequence (left part of each image pair) and the equalized image of estimated pixel variance (right image of each image pair).

8 Conclusions

We have presented a technique for exploiting temporal and spatial constraints in traffic sign detection and recognition across a sequence of image frames. The first stage of the technique organizes raw responses of a boosted Haar cascade detector into representative detection tracks with consistent appearance. The second stage subsequently classifies the extracted detection tracks into signs and not-signs by a discriminative model obtained by supervised learning. Our experiments presented in Section 7 show that the technique significantly improves the localization accuracy of the detections, and simultaneously achieves a substantial decrease of false positive detections.

The main idea of the proposed approach for extracting temporally consistent detection tracks is to require a proper alignment of all member detections. We experienced considerable difficulties in making this idea work in practice, due to large inter-frame camera motions implied by typical speeds of the acquisition vehicle. Additionally, the alignment of non-rectangular signs turned out to be rather sensitive to background changes, since the employed tracking algorithm assumes rectangular image patches. The proposed solution overcomes these problems by managing a cluster of redundant track hypotheses for each physical traffic sign, and by performing the alignment only on suitable rectangular patches in the interior of the sign. The most representative hypothesis is chosen after having collected all available evidence, at the moment when the tracking of all hypotheses from the cluster is over.

Spatio-temporal constraints are enforced by feeding the extracted detection tracks to a discriminative binary classifier. In order to achieve invariance with respect to the speed of the acquisition vehicle, the tracks are represented as fixed-length feature vectors of x and y image coordinates at discrete detection scales. As individual tracks rarely span the whole feasible range of scales, we faced the problem of choosing the values for missing coordinates. Experiments have shown that imputing hardwired values provides consistent and satisfactory results over a range of machine learning algorithms.

Despite the high baseline results presented in 7.1, the proposed approaches significantly improve the detection performance in comprehensive experiments on a very large real-life dataset. We therefore conclude that the presented spatial and temporal constraints contribute an essential improvement to our compound system for automatic traffic sign recognition in video [48]. We believe that the proposed technique offers significant potential towards achieving well-localized and false-positive-free traffic sign detection.

The main direction for the future work in improving temporal consistency concerns increasing the lengths of the representative trajectories by fusing the information contained in all salient hypotheses. Additionally, we would also like to employ the pixel variance images to improve the recognition accuracy. The future work on spatial consistency shall address approaches to ex-

ploit the recovered camera motion by SfM techniques, and to learn contextual constraints arising from other typical constituents of the road scenes.

Acknowledgements This research has been jointly funded by Croatian National Foundation for Science, Higher Education and Technological Development, and Institute of Traffic and Communications, under programme Partnership in Basic research, grant #04/20. The project web site is at `http://mastif.zemris.fer.hr`

The research has been additionally supported by the Austrian-Croatian bilateral programme funded by the Austrian Agency for International Mobility and Cooperation in Education, Science and Research, and the Croatian Ministry of Science, Education and Sports.

The authors would like to thank Ivan Fratrić and Jan Šnajder for helpful suggestions and disussions.

References

1. P. Arnoul, M. Viala, J.P. Guerin, and M. Mergy. Traffic signs localisation for highways inventory from a video camera on board a moving collection van. In *Proc. of IV*, pages 141–146, Tokyo, Japan, September 1996.
2. Claus Bahlmann, Ying Zhu, Visvanathan Ramesh, Martin Pellkofer, and Thorsten Koehler. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In *Proc. of IV*, pages 255–260, Las Vegas, Nevada, June 2005.
3. Simon Baker and Iain Matthews. Lucas-Kanade 20 years on: A unifying framework. *Int. J. Comput. Vis.*, 56(3):221–255, March 2004.
4. X. Baro and J Vitria. Fast traffic sign detection on greyscale images. *Recent Advances in Artificial Intelligence Research and Development*, pages 131–138, October 2005.
5. Xavier Baró, Sergio Escalera, Jordi Vitrià, Oriol Pujol, and Petia Radeva. Traffic sign recognition using evolutionary adaboost detection and forest-ecoc classification. *IEEE Trans. ITS*, 10(1):113–126, 2009.
6. W. Benesova, Y. Lypetskyy, J.-Ph. Andreu, L. Paletta, A. Jeitler, and E. Hdl. A mobile system for vision based road sign inventory. In *Proc. 5th International Symposium on Mobile Mapping Technology*, Padova, Italy, May 2007.
7. Gary Rost Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly Media, Inc., 2008.
8. Karla Brkić, Axel Pinz, and Siniša Šegvić. Traffic sign detection as a component of an automated traffic infrastructure inventory system. In *Proc. of AAPR/ÖAGM*, Stainz, Austria, May 2009.
9. Karla Brkić, Siniša Šegvić, Zoran Kalafatić, Ivan Sikirić, and Axel Pinz. Generative modeling of spatio-temporal traffic sign trajectories. In *Proc. of UCVP*, pages 25–31, held in conjuction with CVPR2010, San Francisco, California, June 2010.
10. João L. Cardoso, Christian Stefan, Rune Elvik, and Michael Srensen. Road safety inspection - best practice guidelines and implementation steps. Technical report, Deliverable D5 of the EU FP6 project RIPCORD - ISEREST, 2007.
11. J. R. Casas, A. P. Sitjes, and P. P. Folch. Mutual feedback scheme for face detection and tracking aimed at density estimation in demonstrations. *Vision, Image and Signal Processing, IEE Proceedings* -, 152(3):334–346, 2005.
12. Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. of CVPR*, pages 886–893, 2005.
13. Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. PAMI*, 26(6):1052–1067, 2007.
14. A. de la Escalera, L.E. Moreno, M.A. Salichs, and J.M. Armingol. Road traffic sign detection and classification. *IEEE Trans. IE*, 44(6):848–859, December 1997.
15. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, New York, 2. edition, 2001.
16. Markus Enzweiler and Dariu M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Trans. PAMI*, 31(12):2179–2195, 2009.
17. A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Robust multi-person tracking from a mobile platform. *IEEE Trans. PAMI*, 31(10):1831–1846, 2009.
18. Chiung-Yao Fang, Sei-Wang Chen, and Chiou-Shann Fuh. Road-sign detection and tracking. *IEEE Trans. VT*, 52(5):1329–1341, September 2003.
19. Luke Fletcher, Nicholas Apostoloff, Lars Petersson, and Alexander Zelinsky. Vision in and out of vehicles. *IEEE Intell. Systems*, 18(3):12–17, 2003.
20. X.W. Gao, L.N. Podladchikova, D.G. Shaposhnikov, K. Hong, and N. Shevtsova. Recognition of traffic signs based on their colour and shape features extracted using human vision models. *Journal of Visual Communication and Image Representation*, 17(4):675–685, 2006.
21. M.A. Garcia-Garrido, M.A. Sotelo, and E. Martin-Gorostiza. Fast traffic sign detection and recognition under changing lighting conditions. In *Proc. ITSC*, pages 811–816, Toronto, Canada, September 2006.
22. Helmut Grabner, Csaba Beleznai, and Horst Bischof. Improving adaboost detection rate by wobble and mean shift. In *Proc. of CVWW*, pages 23–32, Zell an der Pram, Austria, 2005.
23. Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. *Int. J. Comput. Vis.*, 80(1):3–15, 2008.
24. Francisco Parada-Loira Iago Landesa-Vázquez and José L. Alba-Castro. Fast real-time multiclass traffic sign detection based on novel shape and texture descriptors. In *Proc. ITSC*, pages 1–8, Madeira, Portugal, 2010.
25. Inland transport comitee. *Convention on road signs and signals*. Economic comission for Europe, 1968.
26. Fredrik Larsson and Michael Felsberg. Using fourier descriptors and spatial models for traffic sign recognition. In *Proc. of SCIA*, volume 6688, pages 238–249, 2011.
27. Rainer Lienhart, Alexander Kuranov, and Vadim Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *Proc. of DAGM*, pages 297–304, Magdeburg, Germany, 2003.
28. Wei Liu, Xue Chen, Bobo Duan, Hui Dong, Pengyu Fu, Huai Yuan, and Hong Zhao. A system for road sign detection, recognition and tracking based on multi-cues hybrid. In *Proc. of IV*, pages 562 –567, June 2009.
29. G. Loy and N.M. Barnes. Fast shape-based road sign detection for a driver assistance system. In *Proc. of IROS*, pages 70– 75, Sendai, Japan, September 2004.
30. Sérgio R. Madeira, Luìsa C. Bastos, António M. Sousa, Joăo F. Sobral, and Luìs P. Santos. Automatic traffic signs inventory using a mobile mapping system for GIS applications. In *International Conference and Exhibition on Geographic Information*, Lisboa, Portugal, May 2005.
31. S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Siegmann, H. Gomez-Moreno, and F.J. Acevedo-Rodriguez. Traffic sign recognition system for inventory purposes. In *Proc. of IV*, pages 590–595, Eindhoven, Netherlands, June 2008.
32. Benjamin M. Marlin. *Missing Data Problems in Machine Learning*. doctoral dissertation, University of Toronto, 2008.

33. Timothy Mccarthy, Nui Maynooth, Conor Mcelhinney, Conor Cahalane, and Pankaj Kumar. Initial results from european road safety inspection (eursi) mobile mapping project. In *Proc. of ISPRS CRIMT*, Newcastle, UK, June 2010.
34. Yok-Yen Nguwi and Abbas Z. Kouzani. Detection and classification of road signs in natural environments. *Neural Comput. Appl.*, 17(3):265–289, 2008.
35. Jing Peng and Bir Bhanu. Learning to perceive objects for autonomous navigation. *Auton. Robots*, 6(2):187–201, 1999.
36. Minh-Tri Pham, Yang Gao, Viet-Dung D. Hoang, and Tat-Jen Cham. Fast polygonal integration and its application in extending haar-like features to improve object detection. In *Proc. of CVPR*, San Francisco, California, June 2010.
37. G. Piccioli, E. De Micheli, P. Parodi, and M. Campani. Robust method for road sign detection and recognition. *Image and Vision Computing*, 14(3):209 – 223, 1996.
38. Peter M. Roth. *On-line Conservative learning.* PhD thesis, Graz university of Technology, Institute for Computer Vision and Graphics, 2008.
39. A. Ruta, Yongmin Li, and Xiaohui Liu. Detection, tracking and recognition of traffic signs from video input. In *Proc. ITSC*, pages 55 –60, October 2008.
40. Andrzej Ruta, Yongmin Li, and Xiaohui Liu. Real-time traffic sign recognition from video by class-specific discriminative features. *Pattern Recog.*, 43(1):416–430, 2010.
41. Andrzej Ruta, Fatih Porikli, Shintaro Watanabe, and Yongmin Li. In-vehicle camera traffic sign detection and recognition. *Mach. Vision. Appl.*, 2009. accepted for publication.
42. Siniša Šegvić, Anthony Remazeilles, and François Chaumette. Enhancing the point feature tracker by adaptive modelling of the feature support. In *Proc. of ECCV*, Springer LNCS, pages 112–124, Graz, Austria, May 2006.
43. Siniša Šegvić, Anthony Remazeilles, Albert Diosi, and François Chaumette. A scalable mapping and localization framework for robust appearance-based navigation. *Comput. Vis. Image Underst.*, 113(2):172–187, February 2009.
44. Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc. of CVPR*, pages 593–600, Seattle, Washington, June 1994.
45. Radu Timofte, Karel Zimmermann, and Luc Van Gool. Multi-view traffic sign detection, recognition, and 3d localisation. In *Proc. of WACV*, pages 69–76, Snowbird, Utah, 2009.
46. Oncel Tuzel, Fatih Porikli, and Peter Meer. Learning on lie groups for invariant detection and tracking. In *Proc. of CVPR*, pages 1–8, Anchorage, Alaska, 2008.
47. Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vis.*, 57(2):137–154, 2004.
48. Siniša Šegvić, Karla Brkić, Zoran Kalafatić, Vladimir Stanisavljević, Marko Ševrović, Damir Budimir, and Ivan Dadić. A computer vision assisted geoinformation inventory for traffic infrastructure. In *Proc. ITSC*, pages 66–73, Madeira, Portugal, September 2010.
49. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 2005.