A single target voting scheme for traffic sign detection

Article · June 2011

DOI: 10.1109/IVS.2011.5940429

Sebastian Houben
University of Bonn

A single target voting scheme for traffic sign detection

Sebastian Houben*

Abstract— **Traffic sign detection and recognition is an important part of advanced driver assistance systems. Many prototype solutions for this task have been developed, and first commercial systems have just become available. Their image processing chain can be devided into three steps, preprocessing, detection, and recognition. Albeit several reliable sign recognition algorithms exist by now sign detection under real-world conditions is still unstable. Therefore, we address the first two steps of the processing chain presenting an analysis of widely used detectors, namely Hough-like methods. We evaluate several preprocessing steps and tweaks to increase their performance. Hence, the detectors are applied to a large, publicly available set of images from real-life traffic scenes. As main result we establish a new probabilistic measure for traffic sign colour detection and, based on the findings in our analysis, propose a novel Hough-like algorithm for detecting circular and triangular shapes. These improvements significantly increased detection performance in our experiments.**

I. Introduction

Traffic signs are important to warn the driver of possibly unexpected road conditions, to control traffic flow and to prevent dangerous situations for him and other road users. However, they draw attention from the driver and, if used too extensively, are likely to be overlooked. Therefore, an automated traffic sign recognition system is desirable to assist the driver. It improves road safety and seems indispensable for the future development of autonomous driving systems.

A vast amount of research covering the development and evaluation of complete traffic sign recognition systems has been published. Although first commercial systems have reached the market all of them are restricted to particular sign classes, e.g. speed limit signs, and only work under limited weather and lighting conditions.

Normally these systems perform three steps: image preprocessing, sign detection and sign recognition. Great progress has been made in the field of sign recognition [1], [2], [3], the detection stage, however, is less reliable. Common assumptions are that the first appearance of a traffic sign in an image sequence is fixed to a certain size, that it follows a straight movement towards the camera over time, that it does not undergo affine transformation, and that its pixels lie in a well-defined colour interval. These preconditions are reasonable but not fully generalizable. Any violation usually results in missed detections.

In this study, we take several well-known sign detection algorithms and apply them to a set of single images from a public available dataset with challenging cluttered urban street scenes (c.f. Section IV). Different preprocessing steps are proposed to gradually contribute more and more a priori knowledge to the detection stage (c.f. Section III).

Furthermore, we propose a novel fuzzy colour evaluation step that enhances detection results with a learned set of traffic signs. Finally, we discuss disadvantages of the current voting schemes (c.f. Section II) and propose a new approach that proves more reliable than the established ones.

II. Traffic sign detectors

The great amount of literature on traffic sign detection focusses on two main branches: Viola-Jones detectors [4], [5] (often based on the originally proposed Haar wavelet features [6]) and model-based Hough-like approaches. Viola-Jones methods are fast and have high detection rates. However their reliability depends crucially on the training set and untrained signs will therefore not be detected. Which of the two approaches is more promising is still undecided. Nevertheless in this paper we will restrict ourselves to Hough-like methods [7], [8].

For our study these can be generalized best by defining a set $V_{\mathbf{p}}$ of features that contribute to the vote of point $\mathbf{p}$ in the vote space. The vote space is a parameter space that describes the shape one wants to detect, in our case $\{(\mathbf{x}, r) \mid \mathbf{x} \text{ midpoint of shape and } r \text{ radius}\}$. We define the radius of a regular polygon as the shortest distance from the midpoint to an outline point. Every parameter n-tuple is mapped to a real number that measures its likelihood to represent a detected shape. By extracting the local maxima of this mapping one obtains the most likely detections. The advantages of Hough-like procedures are that they are fast and can work on sparse data. By adjusting the vote mapping they can easily be adapted to a priori knowledge.

In the upcoming section we will cover several proposals of voting schemes that have been developed and used for real-time traffic sign detection. The features are obtained from a gradient image and consist of one to three pixel positions $\mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}$ and their orientations $\theta_1, \theta_2, \theta_3$.

A. Regular Polygon Detector (RPD)

In 2004 Loy and Barnes [9], [10] published a detector for regular polygons with n corners that has since been applied very successfully to traffic sign detection. For every pixel one determines the lines $l(r)$, $l'(r)$ orthogonal to the pixel's orientation. Its midpoints lie on the line through $\mathbf{z}$ with orientation θ and have distance r from $\mathbf{z}$ (c.f. Figure 1). The length of the line is twice the length of a n-sided polygon's edge, i.e. $2r \tan \frac{\pi}{n}$. We therefore define

$$V_{(\mathbf{x},r)} = \{(\mathbf{z}, \theta) \mid \mathbf{x} \text{ lies on } l(r) \text{ or } l'(r)\}$$

*Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany.
`sebastian.houben@ini.rub.de`

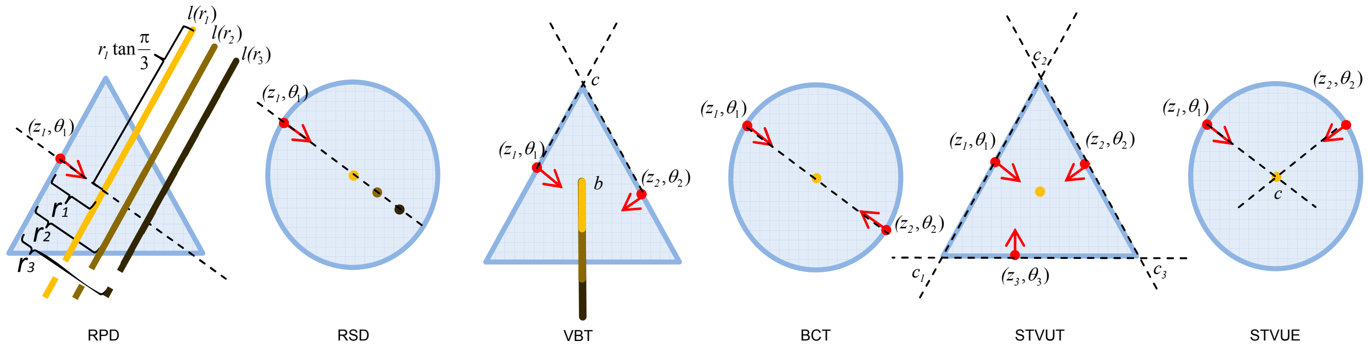


Fig. 1. Schemes for the voting methods. RPD and RSD will also vote for positions and radii converse to the considered pixel's orientation since it is unknown if the orientation is directed into the interior or exterior of the shape. Please note that for clarity reasons these vote positions were omitted.

(c.f. Figure 1). Each pixel therefor votes for every possible center and radius of every n-sided polygon it could be edge of.

B. Radial Symmetry Detector (RSD)

The Radial Symmetry Detector [11] may be regarded as the Regular Polygon Detector for $n \to \infty$ and is considered the classical voting scheme for circle detection. Every pixel votes for the midpoint in the direction of the gradient:

$$V_{(\mathbf{x},r)} = \{(\mathbf{z},\theta) \mid \mathbf{x} \text{ has distance } r \text{ from } \mathbf{z} \\ \text{in direction } \theta \text{ or } -\theta\}$$

C. Vertex Bisector Transform (VBT)

Belaroussi and Tarel have recently proposed two voting schemes based on two pixel positions and orientations:

To detect triangles and other regular polygons the Vertex Bisector Transform [12] considers every pair of pixel positions $\mathbf{z_1}, \mathbf{z_2}$ and non-parallel orientations $\theta_1 \neq \theta_2$ to calculate the intersection point $\mathbf{c}$ of the two lines through $\mathbf{z_1}$ and $\mathbf{z_2}$ orthogonal to θ_1 and θ_2 respectively. It then votes for every point on the bisector b of the two lines' intersection angle (c.f. Figure 1):

$$V_{(\mathbf{x},r)} = \left\{(\mathbf{z_1}, \theta_1, \mathbf{z_2}, \theta_2) \mid \mathbf{x} \text{ is on } b \text{ and } r = ||\mathbf{x} - \mathbf{c}|| cos\frac{\pi}{n}\right\}$$

This method was proposed for triangle detection but can easily be extended to detection of regular polygons.

D. Bilateral Chinese Transform (BCT)

The Bilateral Chinese Transform [13] considers every pair of pixels with parallel or converse orientations directed to each other and votes for their center. Let α be the orientation of the line $\overline{\mathbf{z_1}\mathbf{z_2}}$:

$$V_{(\mathbf{x},r)} = \left\{(\mathbf{z_1}, \theta_1, \mathbf{z_2}, \theta_2) \mid \mathbf{x} = \frac{1}{2}(\mathbf{z_1} + \mathbf{z_2}), \right. \\ |\theta_1 - \theta_2| = 0 \ mod \ \pi, |\alpha - \theta_1| = 0 \ mod \ \pi, \\ \left. |\alpha - \theta_2| = 0 \ mod \ \pi, r = \frac{1}{2}||\mathbf{z_1} - \mathbf{z_2}||\right\}$$

The BCT is able to detect e.g. circles, squares, rectangles and octagons.

With the schemes introduced so far one often perceives the problem that the voting maxima are ambiguous and the results suffer from noisy or cluttered images (c.f. Figure 2). This is caused by the schemes' property that every pixel votes for a large number of points in the parameter space which therefore contains too many potentially worthless votes. To circumvent this problem we propose to use exactly as many pixel positions and orientations as are needed to cast a single vote to the parameter space.

E. Single Target Vote for Upright Triangles (STVUT)

Three pixel positions with pairwise non-parallel orientations will suffice to define a triangle. Having these one can easily calculate the corners $\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}$ as the pairwise intersections of the lines through $\mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}$ orthogonal to $\theta_1, \theta_2, \theta_3$ respectively. We therefor set

$$V_{(\mathbf{x},r)} = \left\{(\mathbf{z_1}, \theta_1, \mathbf{z_2}, \theta_2, \mathbf{z_3}, \theta_3) \mid \mathbf{x} = \frac{1}{3}(\mathbf{c_1} + \mathbf{c_2} + \mathbf{c_3}), \right. \\ \left. r = ||\mathbf{x} - \mathbf{c_1}|| cos\frac{\pi}{3}\right\}$$

Since taking every triple of gradient pixels into account is infeasible for real-time applications we used the following heuristic to choose appropriate vote features: For every pixel $\mathbf{z_1}$ oriented in the vicinity of $-\frac{\pi}{6}$ (with respect to the x-axis) a pixel $\mathbf{z_2}$ with mirrored orientation is searched in the same image row. To obtain the third pixel position $\mathbf{z_3}$ one chooses a column between $\mathbf{z_1}$ and $\mathbf{z_2}$ and looks for orientations in a vicinity of $\frac{\pi}{2}$. This column is selected by calculating the y-coordinate of $\mathbf{z_1}$ modulo the distance of $\mathbf{z_1}$ and $\mathbf{z_2}$. This would guarantee to consider every column with same probability if the distance between $\mathbf{z_1}$ and $\mathbf{z_2}$ was constant.

F. Single Target Vote for Upright Ellipses (STVUE)

To define midpoint and radius of a circle one needs two pixel positions including orientations. Similar to the BCT we define

$$V_{(\mathbf{x},r)} = \{(\mathbf{z_1}, \theta_1, \mathbf{z_2}, \theta_2) \mid \mathbf{x} = \mathbf{c} \text{ and } r = ||\mathbf{z_1} - \mathbf{c}||\}$$

where $\mathbf{c}$ is the intersection of the lines through $\mathbf{z_1}$ and $\mathbf{z_2}$ with orientation θ_1 and θ_2 respectively.

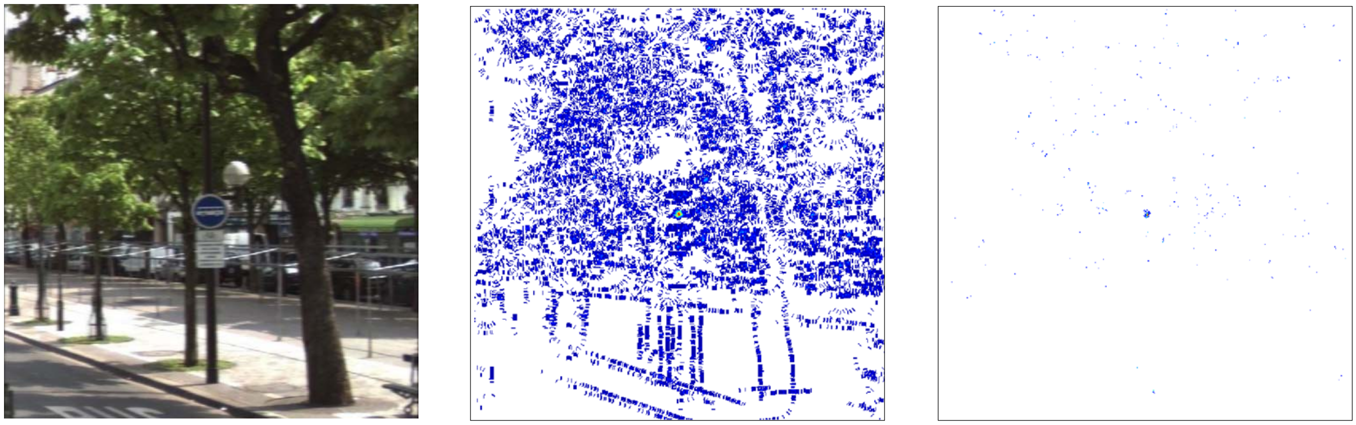

Original image
Radial Symmetry Detector (RSD)
Single Target Vote for Upright Triangles (STVUE)

Fig. 2. Comparing the vote images by RSD (middle) and STVUE (right) of a common urban traffic scene (left) one realizes that pixel-wise schemes voting for a multitude of parameters suffer from noise and clutter. The vote images shown are taken for a fixed radius of 15 pixels which is also the radius of the sign to detect. For this image RSD casts 878,601 votes while STVUE only casts 4,449.

Again we propose a helpful heuristic by considering only those points that lie on the same x-coordinate and should hence have a mirrored orientation.

Please note that STVUT and STVUE are not restricted to detect regular shapes (regular upright triangles and circles) but also stretched forms of them (isosceles upright triangles and upright ellipses). However the votes these schemes cast lie in the common parameter space considerung position and radius. Thus, the detected radius can be considered as a measure of size of the detection rather than the radius in the geometric sense.

III. Preprocessing steps

In the following section we propose seven preprocessing steps that transform the acquired colour image into a gradient image that assigns magnitude and orientation to every pixel. We will later limit the number of pixel positions considered in the voting schemes to the ones with the greatest magnitude. The gradient magnitude also affects the amount a tuple contributes to a parameter vote. We set the vote of a pixel or n-tuple of pixels to be

$$ln(1+||grad(\mathbf{z_1})||)\cdot ln(1+||grad(\mathbf{z_2})||)\cdot ln(1+||grad(\mathbf{z_3})||)$$

where $grad(\mathbf{z_k})$ denotes the gradient at the position $\mathbf{z_k}, k = 1, 2, 3$ (the single and double voting schemes consider only the first and first two factors respectively).

A. Gradient Magnitude Threshold (GMT)

A first step is to consider every edge regardless of any colour or orientation information. We therefor calculate the gradient of the gray image via a Sobel filter, apply Non-Maxima-Suppression and threshold the result to limit the number of pixels the voting scheme operates on.

B. Colour Gradient (CG)

To strengthen edges of objects with distinctively different colours we take the gradient of the gray image and weight the magnitude of each gradient vector with the difference of the neighbouring pixels in colour space. We found the YUV colour space to be adequate. The colour information is stored in the U- and V-components, the intensity is given by the Y-component. We take the distance of two points in the UV-plane as a measure for colour disparity.

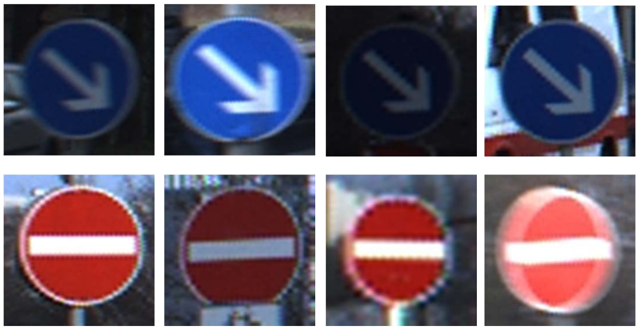


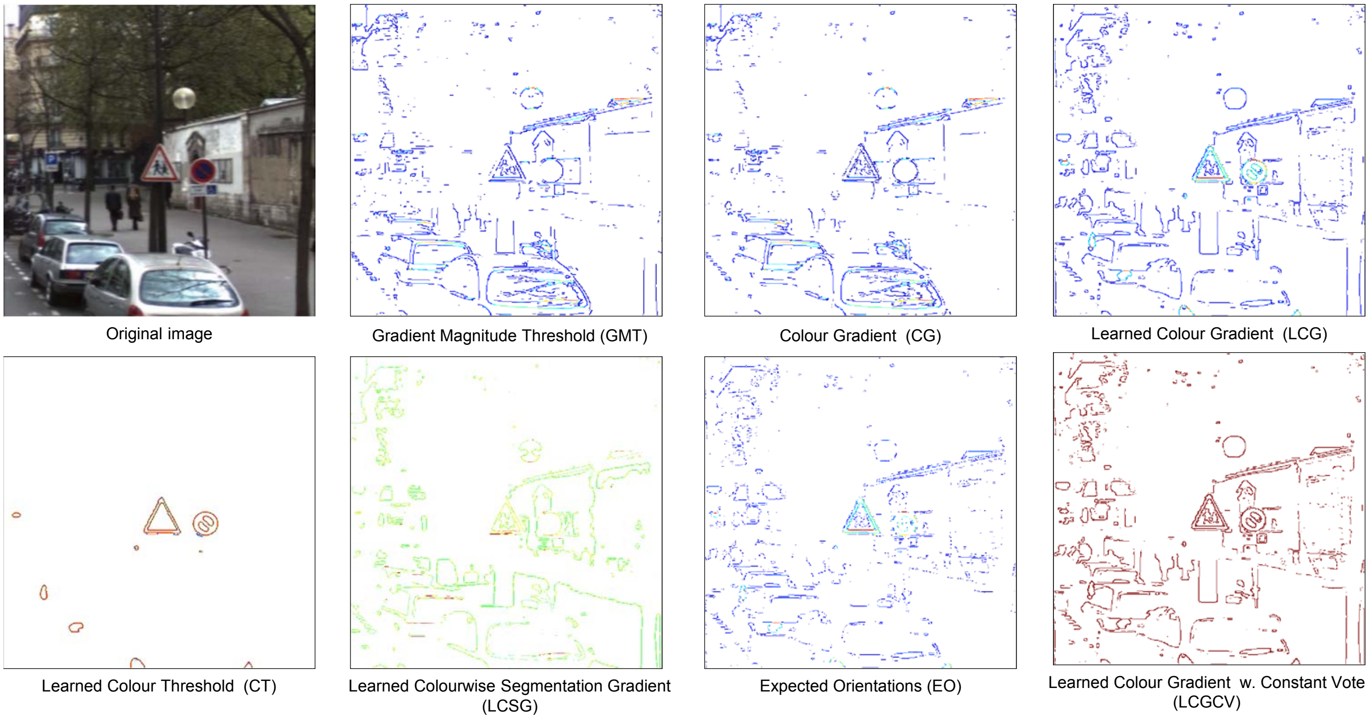
Fig. 4. Some training examples for the traffic sign colours blue (total 54) and red (total 50)

C. Learned Colour Gradient (LCG)

In the next step we want to go more specificly for traffic sign colours. We therefor took a dataset of red and blue traffic signs, semi-automatically segmented the colour of interest and took the average colour of every sign (c.f. Figure 4 for some examples). To obtain a likelihood for every pixel to be of a traffic sign colour we assumed its distribution to be a mixture of Gaussians with the averaged colours as means and standard deviation 0.1 . This way we received a likelihood image we determined the gradient image of (c.f. Section III-A). The edges of traffic signs will hence likely be gradients with strong magnitude increasing the vote in the several schemes.

D. Learned Colour Threshold (CT)

Currently many sign detection approaches start with a threshold of the given image. To compare the probabilistic LCG to this method we threshold the likelihood image and take the gradient.


Original image

Gradient Magnitude Threshold (GMT)

Colour Gradient (CG)

Learned Colour Gradient (LCG)

Learned Colour Threshold (CT)

Learned Colourwise Segmentation Gradient (LCSG)

Expected Orientations (EO)

Learned Colour Gradient w. Constant Vote (LCGCV)

Fig. 3. The gradient images by the proposed preprocessing steps. The relative magnitude is given in false colours.

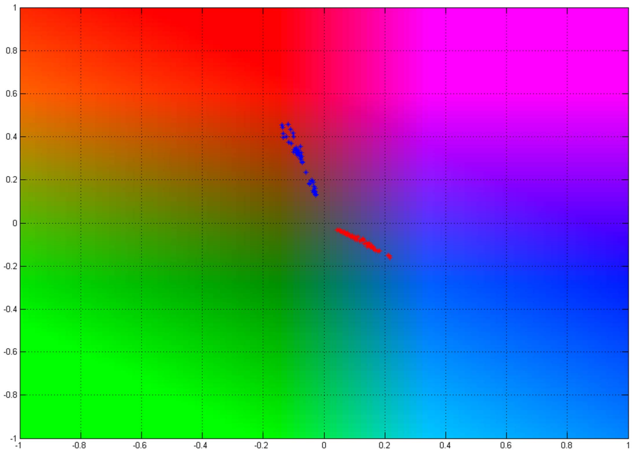


Fig. 5. Distribution of average traffic sign colours in YUV colour space for the red (blue asterisks) and blue signs (red asterisks). The Y-component was fixed to the average sign intensity $Y = 0.33$ that ranged from 0.07 to 0.54

E. Learned Colourwise Segmentation Gradient (LCSG)

Figure 5 shows that the range of possible traffic sign colours is wide. This might cause edge detection problems when the sign is placed in front of a background consisting of a traffic sign colour as well. We therefor assign the index of the sign colour a pixel resembles most to its position. Since artificial differences between two traffic sign colour shades are now included into the image applying the gradient should also detect edges between them. To endow similar shades with similar indexes we projected the learned data on its principal component and numbered the points consecutively according to their projected position.

F. Expected Orientations (EO)

We perform a LCG preprocessing step and ignore those gradient pixels that are significantly off to the orientations one would expect for an upright triangle. Please note that due to the limited number of gradient pixels the exclusion will bring correctly oriented points with less magnitude into consideration. This preprocessing step does not affect the circle detection.

G. Learned Colour Gradient with Constant Vote (LCGCV)

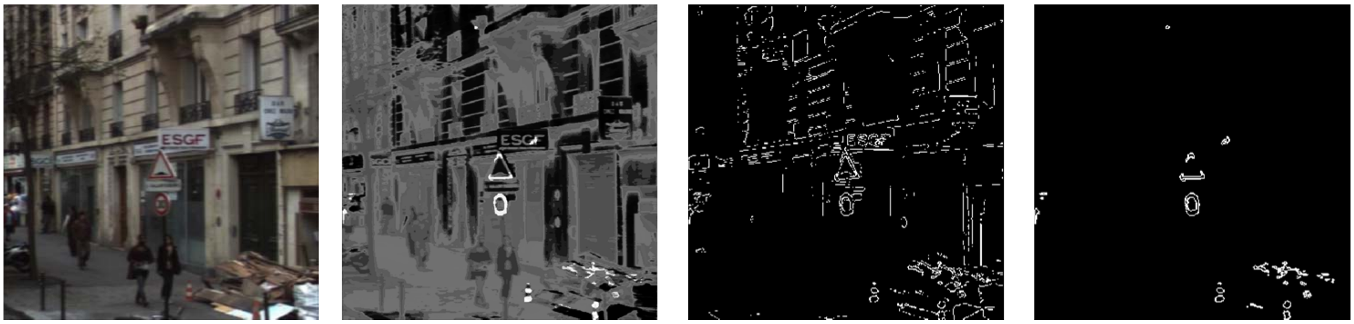
To test the meaning of the weighted vote we also tried the LCG preprocessing step for voting schemes with constant vote. This way the gradient magnitude will only influence the choice of considered gradient pixels but not the amount a pixel or n-tuple of pixels contribute to the vote.

IV. Experimental Results

To evaluate the performance of the proposed voting schemes and preprocessing options we applied them to the public available Stereopolis database [14] that provides a set of 847 images with 251 road signs from a crowded city traffic scene in Paris, France. We restrained the test to circular (RSD, BCT, STVUE, 176 signs) and triangular (RPD, VBT, STVUT, 27 signs) signs .

As a show of generality we trained the LCG, LCSG, EO and LCGCV preprocessing on a seperate set of 50 red and 54 blue traffic signs we acquired in Bochum, Germany. Figure 4 contains a choice of them.

To provide a realistic scenario with real time limitations we choose the 10,000 pixels with largest gradient magnitude after preprocessing to be passed to the voting schemes. Furthermore, we took into account the 10 strongest votes



Original image Sign colour llikelihood Learned Colour Gradient (LCG) Colour Threshold (CT)

Fig. 6. Example of a traffic scene (left) where the sign colour likelihood image (middle left) did not perfectly detect the whole sign border. Taking the gradient of the likelihood image (middle right) can account for this lack while a common threshold (right) will result in a incomplete segmentation and false edges.

TABLE I
DETECTION RATE

	RPD	RSD	VBT	BCT	*STVUT*	*STVUE*
GMT	11%	65%	15%	44%	*26%*	*62%*
CG	19%	91%	15%	47%	*52%*	*68%*
LCG	52%	90%	74%	86%	*78%*	*91%*
CT	63%	56%	63%	45%	*70%*	*72%*
LCSG	11%	47%	7%	17%	*52%*	*43%*
EO	48%	-	81%	-	*81%*	-
LCGCV	41%	90%	70%	83%	*74%*	*84%*

TABLE II
HIT RANKING

	RPD	RSD	VBT	BCT	*STVUT*	*STVUE*
GMT	1.3 ± 0.6	2.2 ± 2.1	3.0 ± 2.7	2.2 ± 2.4	*1.6 ± 1.5*	*2.8 ± 2.4*
CG	2.8 ± 3.5	1.6 ± 1.7	2.8 ± 2.4	2.2 ± 2.1	*2.1 ± 2.0*	*2.6 ± 2.0*
LCG	2.1 ± 1.5	1.5 ± 1.4	2.5 ± 2.5	1.4 ± 1.4	*1.7 ± 2.1*	*1.8 ± 1.7*
CT	1.9 ± 2.2	2.7 ± 2.5	2.6 ± 2.5	2.7 ± 2.6	*1.9 ± 2.5*	*2.0 ± 1.9*
LCSG	2.3 ± 2.3	2.2 ± 2.2	2.0 ± 0.0	2.9 ± 2.5	*2.1 ± 1.5*	*3.1 ± 2.4*
EO	1.3 ± 0.9	-	1.5 ± 1.1	-	*2.1 ± 2.5*	-
LCGCV	3.2 ± 2.2	1.5 ± 1.4	2.2 ± 2.3	1.6 ± 1.6	*1.9 ± 1.7*	*2.0 ± 1.8*

which were local maxima in the parameter space. Both limitations seem reasonable if one wanted to guarantee a limited runtime. A sign is assumed to be detected if the estimated position's error is less then 20% of the true sign size and the estimated radius' relative error is below 45%. This is the exact criterion used in [14]. For this benchmark all schemes were implemented in Matlab and were executed with the exact same set of parameters (i.e. gradient image, discretization of parameter space, allowed discrepancy of measured orientations). For sake of comparability no further optimizations were applied.

Table I shows the detection rate of all presented voting schemes with each proposed preprocessing stage (Figure 7 covers a choice of them in more detail). Please note that the algorithms operated on single images. In real-life applications one would achieve significant improvement by combining the detection results of several time steps. The table clearly shows the importance of including specific colour information into the preprocessing steps. The LCG provides best results with all voting schemes and outperforms the very commonly used colour threshold excepting the RPD that strongly benefits from a small number of pixels after a presegmentation. The LCSG could not compare to LCG because of too many artificial edges. For the importance of the weight a vote contains one can look at the LCGCV preprocessing. Comparing this to LCG obviously all schemes benefit strongly from the weighted votes. Thus, we see that both the choice of pixels and the voting weight the LCG provides are valuable properties that clearly increase detection performance. Although we neglected runtime in our experiments, an implementation on a modern desktop system (Intel Core2 Quad 2.4Ghz, 4 GB RAM) could process a single image (1355 × 781 pixels) in less than 400 ms providing real-time capability for our algorithms.

Regarding the voting schemes one can state that for detection of circles the three presented methods (RSD, BCT, STVUE) perform comparably well. Among the triangle detectors VBT and STVUT have similar maximum detection rates although STVUT seems to be more reliable on sparse data, e.g. after common gradient threshold (c.f. GMT or CG). In our benchmark RPD had the significantly lowest detection rate but on the other hand is the least specialized method.

Table II contains the average rank a vote maximum was counted as detection if sorted by vote weight. This hit ranking goes along with the detection rates of the respective schemes. One can see that for the salient detection methods the correct hit can usually be found among the first five hypothesis.

V. Conclusion and Future Work

We were able to show that probabilistic preprocessing is well suited for providing pixel-wise colour information to following stages. Especially Hough-like methods are easy to adjust to this kind of initial estimations. However, they sometimes suffer from overvoting cluttered or noisy structures. Our proposal to extend current voting schemes to a

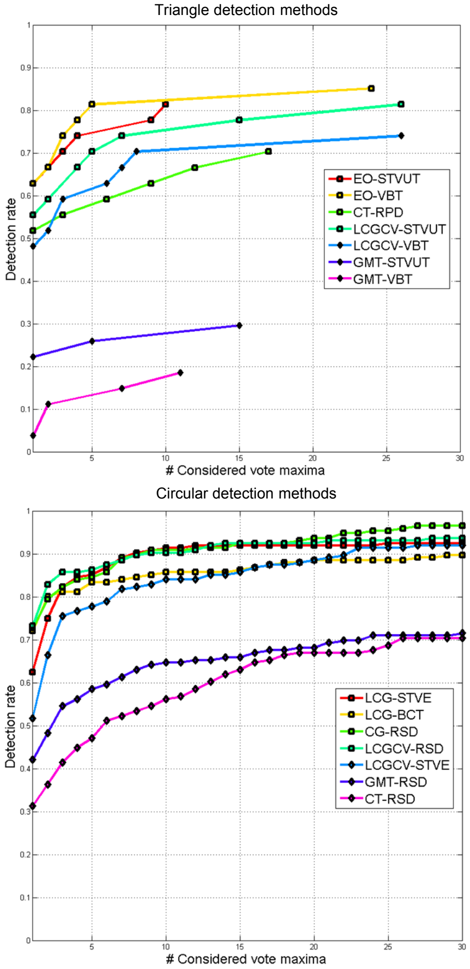


Fig. 7. The behaviour of detection rate against number of considered vote maxima for some proposed procedures (top = triangle detection, bottom = circle detection).

single target vote in order to minimize this turned out to be a promising approach.

Nevertheless, our algorithms were based on heuristic deliberations and not on statistical findings. In a next step we want to examine a larger dataset of traffic signs to extract valuable features and estimate the necessary parameters (e.g. the usual orientation of a triangular sign's edges) more accurately. We hope this will enable us to refine the initial likelihood estimation and the voting weight in the parameter space. Furthermore, we want to take advantage of the single target voting schemes' property of having significantly less votes in parameter space. Therefore, the common technique of accumulating the votes for every block in the discretized vote space does not fit. By simply storing every vote point in an array we can reduce memory costs and gain significant accuracy.

We will work on extending the idea of single target voting to rectangles and regular octagons to start the development of a real-time multi-purpose traffic sign detector.

Acknowledgments

I want to thank Johannes Stallkamp, Marc Schlipsing and Jan Salmen for acquiring the traffic sign colour learning set[1] and the iTowns team for kindly providing the Stereopolis database [14].

References

[1] A. Broggi, P. Cerri, P. Medici, P. P. Porta, and G. Ghisio, "Real Time Road Signs Recognition," in *Proceedings of the IEEE Symposium on Intelligent Vehicles*, 2007, pp. 981–986.

[2] C. Keller, C. Sprunk, C. Bahlmann, J. Giebel, and G. Baratoff, "Real-time recognition of u.s. speed signs," in *Proceedings of the IEEE Symposium on Intelligent Vehicles*, 2008, pp. 518 –523.

[3] C. Bahlmann, Y. Zhu, and V. Ramesh, "A system for traffic sign detection, tracking, and recognition using color, shape and motion information," in *Proceedings of the IEEE Symposium on Intelligent Vehicles*, 2005, pp. 255–260.

[4] K. Brkic', A. Pinz, and S. Łegvic', "Traffic sign detection as a component of an automated traffic infrastructure inventory system," in *Proceedings of the annual Workshop of the Austrian Association for Pattern Recognition*, 2009, pp. 1–12.

[5] X. Baró, S. Escalera, J. Vitrià, O. Pujol, and P. Radeva, "Traffic sign recognition using evolutionary adaboost detection and forest-ecoc classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 113–126, 2009.

[6] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, pp. 137–154, 2004.

[7] P. Hough, "Machine analysis of bubble chamber pictures," in *International Conference of High Energy Accelerators and Instrumentation*, 1959.

[8] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[9] G. Loy, "Fast shape-based road sign detection for a driver assistance system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 70–75.

[10] N. Barnes, G. Loy, D. Shaw, and A. Robles-Kelly, "Regular polygon detection," in *International Conference on Computer Vision*, 2005, pp. 778–785.

[11] N. Barnes, A. Zelinsky, and L. Fletcher, "Real-time speed sign detection using the radial symmetry detector," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 322–332, 2008.

[12] R. Belaroussi and J.-P. Tarel, "Angle vertex and bisector geometric model for triangular road sign detection," in *Proceedings of IEEE Workshop on Applications of Computer Vision*, 2009, pp. 577–583.

[13] ——, "A real-time road sign detection using bilateral chinese transform," in *Proceedings of IEEE International Symposium on Visual Computing*, 2009, pp. 1161–1170.

[14] R. Belaroussi, P. Foucher, J.-P. Tarel, B. Soheilian, P. Charbonnier, and N. Paparoditis, "Road sign detection in images: A case study," in *Proceedings of IEEE International Conference on Pattern Recognition*, 2010, pp. 484–488.

[1] The images are part of the German Traffic Sign Benchmark dataset (`http://benchmark.ini.rub.de/`)