Proc. of the 21st DAGM Symposium für Mustererkennung,
pp. 86-93, Springer Verlag, 1999.

Traffic Sign Recognition Revisited

D.M. Gavrila

Image Understanding Systems, DaimlerChrysler Research, 89081 Ulm, Germany

Abstract. The first part of this paper provides an overview of previous work on traffic sign recognition. Various components are discussed, such as detection, classification and temporal integration. The second part of this paper covers a recently developed shape-based system, based on distance transforms. This system has been quite successful in detecting and recognizing traffic signs in real-time; we report single-image recognition rates of above 90 % in preliminary experiments both offline as on-board our demo vehicle.

1 Introduction

Vision-based systems on-board vehicles hold quite some promise in assisting future drivers at their driving task; they might even allow autonomous navigation under certain simplified conditions (e.g. [7] [6]). The underlying aim would be to increase traffic safety and driving-comfort. An important capability of such systems is to recognize traffic signs; this may, for example, allow an on-board system to warn the driver for inappropriate actions (e.g. speeding, taking a wrong turn in an one-way street).

The outline of this paper is as follows. Section 2 provides an overview of past work on traffic sign recognition. Section 3 discusses a recently developed shape-based system; it consists of a detection component based on distance transforms (DTs) and a classification component based on radial basis functions (RBFs). We conclude in Section 4.

2 Overview of past work

One can distinguish three main components in work on traffic sign recognition: detection, classification and temporal integration. Detection involves using color or shape features to generate candidate regions where traffic sign might reside in a particular image. The resulting (normalized) pixel intensities are subsequently analyzed in a classification phase to determine whether they represent valid traffic signs. The final step, temporal integration consists of increasing the reliability of detection and/or classification by considering image sequences; this requires a tracking capability. We now discuss these components in turn.

2.1 Detection

Color-based detection methods [3] [10] [13] aim to segment the typical colors of traffic signs (i.e. red, blue and yellow) in order to provide a region of interest for further processing. In some systems, the boundaries in color space are hardcoded by the user [3], others use learning approaches [10] [11] For example, Janssen *et al.* [10] use a pixel-based segmentation scheme where a look-up table is generated offline by a polynom classifier after a training phase. The training set involves a large amount of manually labeled pixels obtained under a variety of lighting conditions. After the classification of individual pixels in various colors classes, [10] [13] follow up with a region growing and filtering step, in which candidate regions for traffic signs are selected based on their color, global shape features, and topological relations (e.g. neighboring, enclosing).

Shape-based detection methods do not require color information. Blancard [4] relies on edge segmentation and linking to obtain a contour representation, from which various features are extracted (e.g. perimeter, curvature). Others used template matching techniques, either using entire traffic sign shapes as templates [1] [8] [5] or only subparts [3] [15] (verifying consistency afterwards). For correlation, images are preprocessed with high-pass band filters.

Comparing color- and shape-cues one observes that color provides a very immediate focussing mechanism for detecting traffic signs. But whether color information alone can provide an accurate boundary of the traffic sign region is doubtful. In practice, measured color values vary significantly from their underlying "true" values, given different lighting conditions, so that a useful partitioning of color space in regions corresponding to particular traffic sign colors and non-traffic sign colors is problematic, irrespective of the color space (e.g. RGB, HSV) used [10] [5]. Indeed, color regions such as those for "blue" and "red" can overlap; additional problems arise with a color-only approach when considering traffic signs which are inherently "white" (i.e. those clearing a particular speed-limit). Furthermore, region growing methods that do not contain model information tend to have unwanted "spill-over" effects.

Shape-based methods based on gradient features tend to be more robust with respect to lighting conditions. One needs to distinguish though between methods which place strong demands on edge segmentation (i.e. detection and linking) and methods which are more model-driven (i.e. template matching), the latter typically being more robust, at higher computational cost.

2.2 Classification

The classification step is usually preceeded by a normalization procedure on the pixels of the candidate regions (i.e. the potential pictographs); typically, regions are scaled to a fixed size and measures are taken to factor out lighting conditions. The resulting intensity features are subsequently fed into one of the established classifier tools: nearest neighbour (NNB) [2], radial basis functions (RBF) [10], polynom classifiers (PC) [12] and neural networks (NN) [1] [3] [12]. or input to self-designed schemes [5]. Work by Kressel *et al.* [12] uses a combined classifier approach, employing a feed-forward multi-layer NN with spatial receptive fields for dimensionality reduction; this has the advantage that relevant features can be learned by the NN and the dimensionality reduction results in speed gains. Furthermore, they compare the performance of a local approximator (RBF) with that of a global approximator (PC) on large traffic sign databases.

Not all work on traffic sign recognition requires a separate detection step as described in the previous section. Some work uses pictograph-based classifiers directly in conjunction with a search algorithm. For example, Betke and Makris [2] use simulated annealing and Aoyagi and Asukara [1] use genetic algorithms. Because little or no (error-prone) segmentation is required for these types of approaches, they have the potential to be very robust. In practice however, they tend to be quite slow and impractical for real-time applications. The problem is that the classifier outcome is typically not a well-behaved evaluation measure to guide the search; its value can vary significantly at small parameter changes (e.g. position, scale), limiting the potential for efficient coarse-to-fine approaches. Brute-force approaches based on optical correlators have been used, though, with some success.

2.3 Temporal Integration

Simple tracking techniques involve few assumptions about the world and rely solely on what is observed in the images (i.e. tracking in 2D) to estimate object motion and establish correspondence. More sophisticated techniques model camera geometry and vehicle speed to achieve better motion estimates. For example, [5] considers a vehicle driving straight with constant velocity and uses a Kalman-filter framework to track the centers of detected traffic signs. Once correspondence is established over time, integration of recognition results is done by simple averaging techniques where larger weights are given to recognition results of traffic signs closer to the camera.

3 A DT-based traffic sign recognition system

Currently, our system consists of a detection (Section 3.1 and classification (Section 3.2) component.

3.1 Detection

The detection step uses a template-based correlation method to identify potential traffic signs in images; this involves so-called distance transforms (DTs). Matching with DT is illustrated schematically in Figure 1. It involves two binary images, a segmented template T and a segmented image I, which we'll call "feature template" and "feature image". The "on" pixels denote the presence of a feature and the "off" pixels the absence of a feature in these binary images. What the actual features are, does not matter for the matching method. Typically, one uses edge- and corner-points. The feature template is given off-line for a particular application (here it is a particular traffic sign shape), and the feature image is derived from the image of interest by feature extraction.

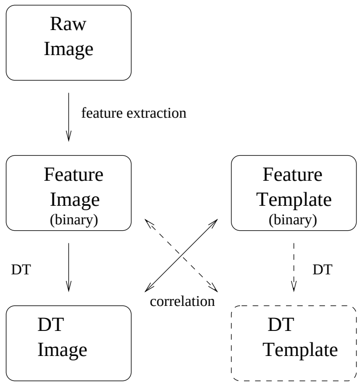


Fig. 1. Matching using a DT

Matching T and I involves computing the distance transform of the feature image I. The template T is transformed (e.g. translated, rotated and scaled) and positioned over the resulting DT image of I; the matching measure $D(T, I)$ is determined by the pixel values of the DT image which lie under the "on" pixels of the transformed template. These pixel values form a distribution of distances of the template features to the nearest features in the image. The lower these distances are, the better the match between image and template at this location. There are a number of matching measures that can be defined on the distance distribution. One possibility is to use the average distance to the nearest feature. This is the *chamfer* distance.

$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t) \tag{1}$$

where $|T|$ denotes the number of features in T and $d_I(t)$ denotes the distance between feature t in T and the closest feature in I. Thus, the chamfer distance consists of a correlation between T and the distance image of I, followed by a division. Other more robust (and costly) measures reduce the effect of missing features (i.e. due to occlusion or segmentation errors) by using the average truncated distance or the f-th quantile value (the *Hausdorff* distance) [9] [14].

In applications, a template is considered matched at locations where the distance measure $D(T, I)$ is below a user-supplied threshold θ

$$D(T, I) < \theta \tag{2}$$

Figure 2 illustrates the matching scheme of Figure 1 for the typical case of edge features. Figure 2a-b shows an example image and template. Figure 2c-d shows the edge detection and DT transformation of the edge image. The distances in the DT image are intensity-coded; lighter colors denote larger distance values.

The advantage of matching a template (Figure 2b) with the DT image (Figure 2d) rather than with the edge image (Figure 2c) is that the resulting similarity measure will be smoother as a function of the template transformation parameters. This enables the use of various efficient search algorithms to lock onto the correct solution. It also allows more variability between a template and an object of interest in the image.

For an overview of previous work on DT-based matching, see [8]. The proposed system extends basic DT-based matching in two ways. First, edge features are differentiated by their orientation. Separate DTs are computed for each orientation interval of a scene edge image. The edge templates are separated into subparts with similar edge orientation. Matching proceeds as before, but now the match measure between image and template is the sum of the match measures between template and DT image corresponding to the same edge orientation interval. Incorporating orientation information greatly reduces false-positives and speeds up the hierarchical detection process discussed next.

The second extension involves matching multiple (N) templates using a template hierarchy, in addition to employing a coarse-to-fine search in parameter space. The idea is that at a coarse level of search, when the image grid size of the search is large, it would be inefficient to match each of the N objects separately, if they are relatively similar to each other. Instead, one would group similar templates together and represent them by a prototype template; matching would be done with this prototype, rather than with the individual templates, resulting in a (potentially significant) speed-up. This grouping of templates is done at various levels, resulting in a hierarchy, where at the leaf levels there are the N templates one needs to match with, and on intermediate levels there are the prototypes. See Figure 3 for the traffic sign hierarchy used for the experiments. Matching can then be considered as a tree traversal process. When a particular node is reached during the traversal process, the corresponding template needs to be matched against the DT image at some particular image locations. For the locations where the match is below the node-specific distance threshold, the matching propagates to the children of the node, or ends successfully at the leaf level. For details, see [8].

3.2 Classification

After the detection step, regions corresponding to the interior of the contour templates are extracted from the images at the detected locations; these are the candidate traffic signs. A normalization step follows in which the pixel intensities are normalized w.r.t. mean and variance. Furthermore, cropped regions are scaled to a fixed square NxN size, filling non-data pixels with null values.

A radial basis function (RBF) network is used for classification [12]. During the training stage, reference vectors are set in feature space by an agglomerative clustering procedure. Linear ramps, rather than Gaussians, are used as radial functions, for efficiency purposes. Two radius parameters need to be specified for each such ramp; these parameters are set based on the distance to the nearest reference vector of the same class and to that of the nearest reference vector of one of the other classes, in a manner described in [12]. The test stage consists of summing probabilities that an unknown feature vector corresponds to a particular class, based on the contributions made by the various reference vectors.

3.3 Experiments

The aim was to recognize circular and triangular (up/down) traffic signs, as seen on highways and secondary roads. For starters, we built classifiers for a subset of traffic signs: 5 classes dealt with speed limitations (circular) and 5 dealt with various warning signs (triangular).

For detection, we used shape-templates with radii 7-18 pixels (the images were of size 360 by 288 pixels). This lead to a total of 36 templates, for which a template tree was specified "manually" as in Figure 3. Coarse-to-fine matching used a grid size of $\sigma = 8, 4, 1$ for the three levels of the template tree. For classification, we scaled image regions down to 16 x 16 pixels. The number of reference vectors used varied between 50 and 250, depending on the class.

The experiments involved both off- and online tests. Off-line, we used a database of 1000 images, taken during day-time (sunny, rainy) and night-time. See Figure 4. We obtained single-image detection rates of over 95%, when allowing solutions to deviate by 2 pixels and by radius 1 from the values obtained by a human. On the average, there were two or less false positives per image at the detection stage. Preliminary single-image recognition rates are above 90%, with less than 5% false positives. Although promising, due to small size of the examined test set, these figures must be considered with some caution. Difficult environmental conditions (e.g. rain drops, partial occlusion by window wiper, direct sunlight into camera) could furthermore reduce detection rates by as much as 15%.

On-board experiments were performed with our E-class T-model vehicle (see Figure 5). The system ran at about 6-8 Hz on a 266 Mhz dual-Pentium II with MMX.

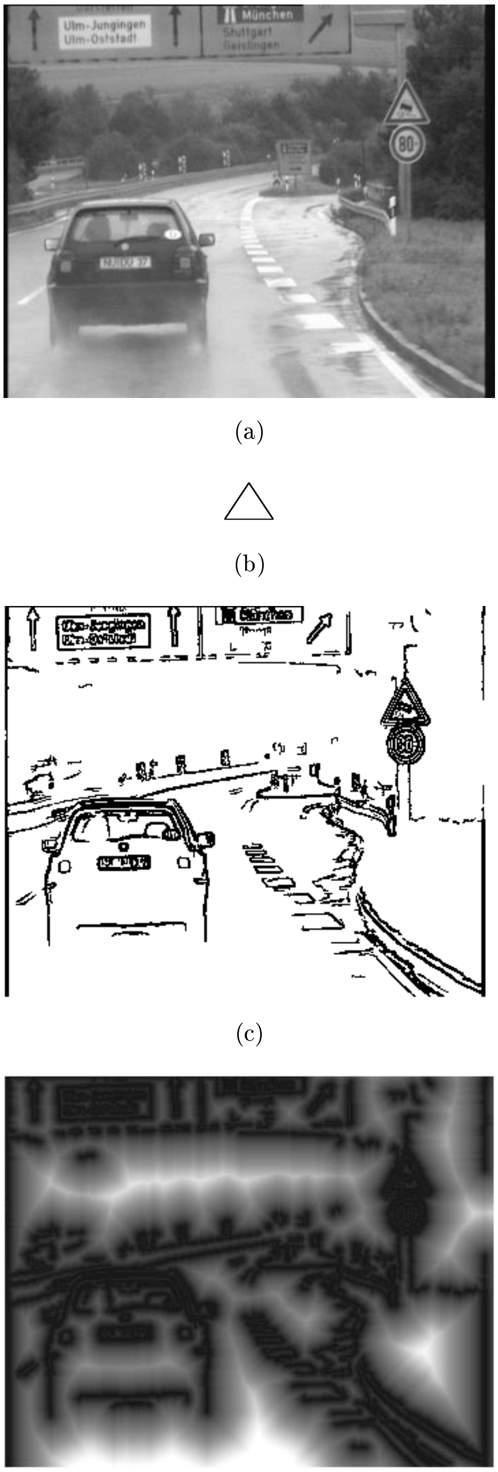
4 Conclusions

An overview was provided of existing work on traffic sign recognition; discussed were detection, classification and temporal integration components. A recently developed traffic sign recognition was subsequently introduced where an efficient hierarchical method is used for shape-based detection; it uses a simultaneous coarse-to-fine approach in parameter and shape space effectively enabling real-time implementation on a general-purpose processor. Subsequent classification was performed by a RBF network. Work in progress involves validating the promising initial results in ROC experiments, using a significantly larger database (10000) and furthrt improving them by temporal integration.

References

1. Y. Aoyagi and T. Asakura. A study on traffic sign recognition in scene image using genetic algorithms and neural networks. In *Proc. IEEE Conf. on Industrial Electronics, Control and Instrumentation*, pages 1838–1843, Taipei, Taiwan, 1996.
2. M. Betke and N. Makris. Fast object recognition in noisy images using simulated annealing. In *International Conference on Computer Vision*, pages 523–530, 1995.
3. A. de la Escalera, L. Moreno, M. Salichs, and J. Armingol. Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, 44(6), 1997.
4. M. de Saint Blancard. *Road Sign Recognition: A Study of Vision-based Decision Making for Road Environment Recognition*, chapter 7. Springer Verlag, 1991.
5. G. Piccioli *et. al.* Robust method for road sign detection and recognition. *Image and Vision Computing*, 14:209–223, 1996.
6. U. Handman *et al.* An image processing system for driver assistance. In *Proc. of Intelligent Vehicles Conference*, pages 481–486, Stuttgart, Germany, 1998.
7. U. Franke, D. Gavrila, S. Görzig, F. Lindner, F. Pätzhold, and C. Wöhler. Autonomous driving goes downtown. *IEEE Intelligent Systems*, 13(6):40–48, 1998.
8. D. Gavrila. Multi-feature template matching using distance transforms. In *International Conference on Pattern Recognition*, pages 439–444, Brisbane, 1998.
9. D. Huttenlocher, G. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.
10. R. Janssen, W. Ritter, F.Stein, and S. Ott. Hybrid approach for traffic sign recognition. In *Proc. of Intelligent Vehicles Conference*, pages 390–395, 1993.
11. N. Kehtarnavaz and A. Ahmad. Traffic sign recognition in noisy outdoor scenes. In *IEEE International Conference on Intelligent Vehicles*, pages 460–465, 1995.
12. U. Kressel, F. Lindner, C. Wöhler, and A. Linz. Hypothesis verification based on classification at unequal error rates. In *Submitted to ICANN*, 1999.
13. L. Priese, R. Lakmann, and V. Rehmann. Automatische verkehrszeichenerkennung mittels echtzeit-farbbildanalyse (in german). *Automatisierungstechnik*, 45(12), 1997.
14. W. Rucklidge. Locating objects using the hausdorff distance. In *International Conference on Computer Vision*, pages 457–464, 1995.
15. P. Seitz, G. K. Lang, B. Gilliard, and J.C. Pandazis. The robust recognition of traffic signs from a moving car. In *Proc. 13th DAGM Symposium on pattern recognition*, pages 287–294, 1991.

This article was processed using the LaTeX macro package with LLNCS style



(a) (b) (c) (d)

Fig. 2. (a) original image (b) template (c) edge image (d) DT image

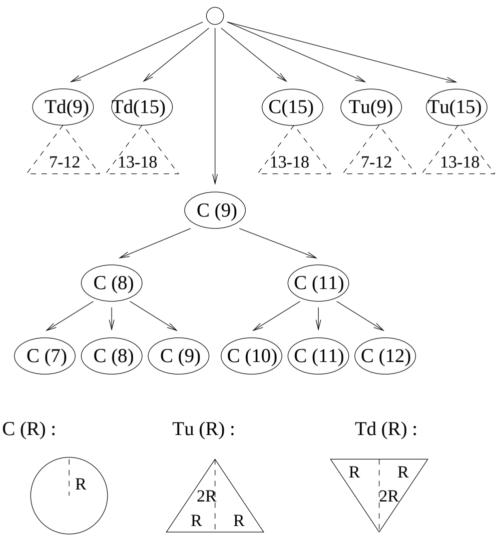


Fig. 3. A hierarchy for traffic sign shapes (hard-coded)

Fig. 4. Traffic sign recognition

Fig. 5. on-board camera and display