Video-based Traffic Sign Detection, Tracking and Recognition

Andrzej Ruta

School of Information Systems, Computing and Mathematics
Brunel University

Thesis submitted to the Brunel University
for the degree of Doctor of Philosophy
· 2009 ·

To my loving parents

Abstract

Detecting and recognising objects in natural scenes is a challenging machine vision problem. The complexity of this problem grows even more when an additional real-time processing requirement is to be met. This is common in all kinds of traffic-related applications and in the driver support systems in particular, as a rapid decision may have a critical impact on the safety of a human. Computationally efficient image and video processing requires smart algorithms. Frequently, these algorithms must strike a balance between complexity and accuracy and the limitations arising from this compromise must be compensated elsewhere, e.g. through an efficient modelling of temporal dependencies between the consecutive frame observations.

In this work, the area of visual object detection, tracking and recognition in the traffic environment is explored. The primary focus is put on the problem of video-based traffic sign recognition (TSR), which is one of the major tasks in the contemporary visual driver assistance systems. Certain algorithms hereby presented are used for solving related problems, such as pedestrian detection or classification of car models. Some of them are formulated flexibly enough to be considered in a much broader application context. Despite these multiple applications, the substance of the presented research is centred around the TSR problem and can be logically divided into three parts, reflecting the natural order of tasks performed in the typical processing pipeline: detection, tracking and recognition.

At the detection stage several techniques are analysed and evaluated using different image and video datasets. These techniques include both a cascade of boosted classifiers, which is a general-purpose object detector that requires large volumes of natural data for training, and the application-specific, training-free regular shape detectors based on the circular Hough Transform (HT) and its extensions. Discussion of these methods is conducted from the perspective of their suitability for efficient scanning of large regions of an input image. In such a usage scenario an additional fast interest region extraction algorithm is required to enable frame-rate operation of the TSR system. Such an algorithm based on a quad tree processing and an integral image technique is proposed. Moreover, the known detection algorithms suffer from the problem of oversensitivity, i.e. generate multiple redundant positive hypotheses. A *Confidence-weighted Mean Shift* (CWMS) clustering algorithm is proposed to address this problem by finding modes of the detector's response distribution, where the confidence of a single hypothesis is related to the soft response of the detector.

Tracking of the target is an important task in visual surveillance. It greatly reduces computation by avoiding exhaustive search of the object candidates in every frame of the input video and improves the overall accuracy of the detection and recognition. In this study the geometry of the target over time is modelled. Several methods are compared in solving the traffic sign tracking problem: a combined Kalman Filter-Particle Filter tracker (KF-PF) with a limited pose/viewpoint invariance built in, a contour tracker based on an evolving Pixel Relevance Map (PRM), suitable for tracking the road signs in visual clutter, and the regression tracker (RT) which models the full structure of the apparent affine target deformation. All methods are shown to be useful for tracking of the road signs at a significant distance from the camera, where modelling of the position and scale changes suffices. The latter technique is shown to model the affine distortions of signs the most efficiently, which enables reconstruction of the full-face view of the target seen even under significant viewpoint changes and at little computational cost. All presented algorithms are evaluated experimentally on synthetic and real traffic video.

Several methods are considered for robust road sign classification. A general direction followed is towards developing a robust class similarity measure together with a suitable discriminative feature representation of the pictograms. One proposed similarity measure is designed for comparing the objects essentially containing only a few discrete colours, like traffic signs do. It utilises so-called *Colour Distance Transform* (CDT) and is inferred from the idealised template sign images via *one-vs-all* dissimilarity maximisation. More generic similarity estimation methods presented include *SimBoost*, a novel variant of AdaBoost algorithm, and the *Similarity-Learning Kernel Regression Trees* (S-KRT), both of which are learned from the pairs of images representing either the same or different classes. Good discriminative power of these algorithms and their other interesting properties, such as flexibility with respect to the choice of the feature representation of the images, are demonstrated using different sign and non-sign image datasets. The latter experiments are conducted to show usefulness of the pairwise similarity-based classifiers in generic, multi-class object recognition.

Acknowledgments

I am deeply grateful to my supervisor, Dr. Yongmin Li, for his constant guidance and motivation given to me over the last three years. If it had not been for his valuable suggestions and critical comments, I would have never found myself at the end of this long journey. I am also indebted to my second supervisor, Prof. Xiaohui Liu, for arranging an extraordinary environment in which I could conveniently conduct my research and for the organisation of the Intelligent Data Analysis group meetings, both formal and informal, which were great fora for exchanging research ideas. I wish to thank both for agreeing to fund my attendance on several interesting conferences.

I am grateful to the members of IDA group: Dr. Allan Tucker and Prof. Zidong Wang for several inspirational discussions and comments on my work and the fellow PhD students: Kyriakos Chrysostomou, Emma Steele, Jian Li, Jose Polo and Ana Salazar for exchanging ideas and for creating a warm atmosphere in our research lab. I wish to thank other IDA group memers: Dr Stephen Swift, Dr Stanislao Lauria, Dr Sherry Chen for making this group so diverse and thus so interesting. My thanks also go to the technical staff: Gareth Hirst, Jeremy Baxter, Jason Hobbs and Andrew Kendall on whom I could count having hardware/software problems, and to the research program coordinators: Ms Julie Whittaker and Ms Ela Heaney who were irreplaceable in dealing with various administrative issues arising in the course of my work.

I would like to thank Dr. Fatih Porikli from Mitsubishi Electric Research Laboratories for the prolific research cooperation and for the numerous valuable ideas I exploited in this thesis and in the papers he co-authored. Much appreciation to my brother Dymitr, who, as a scientist, commented many times on my work and helped me develop the correct attitude to the research. I am also grateful to my friend, Michal Czardybon, who was a great advisor regarding the computer programming issues.

Finally, I would like to express my deepest gratitude to all those who wholeheartedly supported me in pursuing the PhD study, first of all my beloved mum and dad, but also the best friends: Gosia, Tadek, Michal, Magda and Kasia. I wish I could make up for the time I could not spend with them during the last three long years.

Declarations

Certain parts of the work presented in this thesis have been accepted for publication or have been already published in the following articles:

1. Ruta, A., Porikli, F., Li, Y., Watanabe, S., Kage, H. and Sumi, K.: A New Approach for In-Vehicle Camera Traffic Sign Detection and Recognition, accepted for presentation in *the 11th IAPR Conference on Machine Vision Applications*, Yokohama, Japan, May 2009

2. Ruta, A., Li, Y. and Liu, X.: Detection, Tracking and Recognition of Traffic Signs from Video Input, In *Proc. of the 11th International IEEE Conference on Intelligent Transportation Systems*, 55-60, Beijing, P.R. China, October 2008

3. Ruta, A., Li, Y. and Liu, X.: Low-Resolution Human Detection and Gait Recognition in Natural Scenes, In *Proc. of the 14th International Conference on Automation and Computing*, 213-218, Uxbridge, United Kingdom, September 2008

4. Ruta, A., Li, Y. and Liu, X.: Towards Real-Time Traffic Sign Recognition by Class-Specific Discriminative Features. In *Proc. of the 18th British Machine Vision Conference*, Coventry, United Kingdom, 1:399-408, September 2007 (**Best Poster Prize**)

5. Ruta, A., Li, Y. and Liu, X.: Traffic Sign Recognition Using Discriminative Local Features. In *Proc. of the 7th International Symposium on Intelligent Data Analysis*, Ljubljana, Slovenia, 355-266, September 2007

The following articles discussing parts of the work hereby presented are under review at the time of submitting this dissertation:

1. Ruta, A., Li, Y. and Liu, X.: Kernel Regression Tree Ensembles for Estimating Similarity Between Object Classes, submitted in March 2009 to *the 12th IEEE International Conference on Computer Vision*

2. Ruta, A., Porikli, F., Watanabe, S. and Li, Y.: In-Vehicle Camera Traffic Sign Detection and Recognition, submitted in March 2009 to *Machine Vision and Applications*

3. Ruta, A., Li, Y. and Liu, X.: Robust Class Similarity Measure for Traffic Sign Recognition, submitted in January 2009 to *IEEE Transactions on Intelligent Transportation Systems*

4. Ruta, A., Li, Y. and Liu, X.: Real-Time Traffic Sign Recognition from Video by Class-Specific Discriminative Features, submitted in December 2007 to *Pattern Recognition*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cars became a primary means of transport in the 20th century and their number has been dynamically growing since the time they were invented. At the present the motorways and the urban roads in many developed and developing countries are full of vehicles, which has exposed the drivers to various risks. This, together with the technological progress of the post-war decades, boosted the development of car industry and the research aimed at increasing driving safety and automation of the vehicle navigation process.

First recognised computer vision-based driver assistance systems were developed in 1980's when adequate computing hardware and cameras could already be fitted in a standard passenger car. As road signs are an important part of the traffic infrastructure which plays a key role in regulating flow of the vehicles, it was soon found necessary to include in the Driver Support Systems (DSS), as they were often called, the visual traffic sign recognition (TSR) functionality. Nowadays, TSR is considered only a single aspect of the computer-aided driver assistance, along with obstacle detection, pedestrian detection, parking assistance or lane departure alerting, as well as a range of non-visual components like GPS-based vehicle positioning or intelligent route planning.

There are three potential usage scenarios of a traffic sign recognition system, ordered by the level of interference of TSR in the human activity:

- **the present-day scenario**: TSR is used to detect and recognise the traffic signs passed by and to present the appropriate information to the driver. No further actions are taken. This information can have a form of an arbitrary sound, an understandable natural-language audio message, or a visual presentation of the detected and interpreted sign's prototype on the dashboard, as well as their combinations.

- **the near-future scenario**: TSR system not only detects and recognises road signs, but is also allowed to trigger certain, limited mechanical actions preventing the vehicle from achieving various dangerous states, and hence ensuring safety of a human. Such a limited intervention could involve for example an electronic speed reduction or a fuel inflow cut-off in presence of speed limit signs, or to give the right of way to the other, privileged traffic participants.

- **the future scenario:** TSR is used to detect and recognise traffic signs and is fully integrated with the other DSS components, which jointly take full responsibility for a vehicle navigation. In this scenario the vehicle would be entirely autonomous and the role of a human would be reduced to merely providing the journey endpoints and possibly some additional requirements to the system, just in a way the GPS is nowadays used.

Undoubtedly, the first stage of the TSR systems development has already been reached. However, the current sign detection and recognition algorithms are still far from perfect. First of all, the existing solutions do not seem to have achieved maturity in providing their basic functionality on a practically acceptable level. For example, the current TSR systems cannot handle all types of traffic signs. They can merely recognise narrow subcategories exhibiting similar shape and colour characteristics, where the intra-category appearance variability is relatively low, e.g. speed limit signs. Recognition of more complex signs, like those giving directions, which have unstandarised shapes and colours and contain textual information to be interpreted, is still a great challenge. Secondly, there are numerous undesirable factors and driving circumstances in which the reliability of the existing TSR approaches significantly drops. These include adverse illumination, rainfall/snowfall, or vibrations which might be caused by the vehicle moving on an uneven road surface. Only when the abovementioned limitations are properly addressed, a way towards fully autonomous vehicle navigation will be open.

In this thesis a three-years research on traffic sign recognition is presented. In this research some of the limitations of the previous approaches have been addressed. The detailed discussion of our approach to detecting, tracking and recognising road signs from a moving vehicle will be given in chapters 3, 4 and 5. Below, we only formulate the TSR problem and provide an outline of the presented methods, emphasising the original contributions of this thesis.

1.1 Problem Formulation and Goals

The aim of the traffic sign recognition system operating on board of a vehicle is to detect and track the sign instances over time and to correctly interpret their pictograms, so that the driver can react properly to the encountered traffic situation. The input to a TSR system is a live video stream captured by an in-vehicle camera/cameras and its output are the desired-form signals providing a human-understandable interpretation of the detected and recognised signs. Such a system can be conceptually visualised using a block diagram with three main components, as shown in Figure 1.1. The arrows between each pair of components are drawn by default in both directions. However, depending on the actual system architecture, certain interactions may be unidirectional, or may not exist at all. This diagram will be presented in more detail in the following sections.

Consider a single front-looking camera mounted on board of a vehicle in front of the windscreen. Whenever a relevant traffic sign is detected within the field of view of the camera, as shown in Figure 1.2a, the TSR system should analyse the sign's

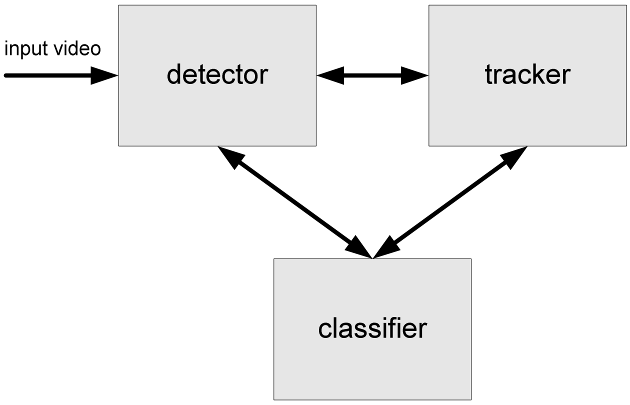
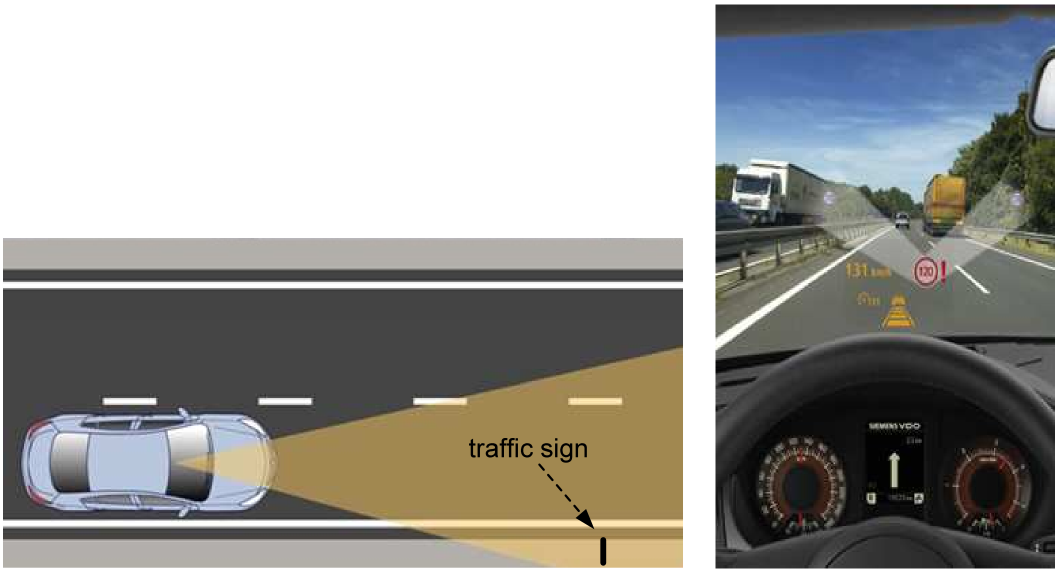


Figure 1.1: Block diagram of traffic sign recognition system operating on board of a vehicle.

pictogram over time, classify it, possibly before the sign is passed by, and present the outcome to the driver, for instance in a way shown in Figure 1.2b.



(a) (b)

Figure 1.2: Usage scenario of a traffic sign recognition system: a) schematic depiction of a vehicle approaching a traffic sign, b) an example way of presenting the information about a detected and recognised sign to the driver. The right illustration by courtesy of *Siemens AG*.

In the detection stage we want to capture possibly all instances of traffic signs which fall into the field of view of the camera, and which apparent size in the image is in between a predefined minimum and maximum. This apparent scale range corresponds to the range of distances of a sign from the camera and is defined such that only the signs mounted in the close vicinity of the road are included in the analysis. As scene segmentation and object detection always precede the higher-level analysis in a processing pipeline, we would like our sign detector to produce very few false alarms, even at the cost of missing certain true positives in a single image. This choice is based on an assumption that the potential cost of false positives for the driver is generally higher than the negative impact of not detecting the true signs, at least when the TSR system is only intended to play a supportive role. Besides, when video stream is processed, missing the target in one frame can always be made up for in another, where it is possibly easier to discriminate it from the background.

The goal of the road sign tracker is to maintain the track of an initially detected sign over time, until it disappears from the field of view of the camera. We want our tracker to go beyond the commonly adopted scheme in which it is only used to reduce the local search region for the sign detector. First, we would like to model the evolution of a tracked sign, or at least its boundary, on a feature or pixel level. Secondly, our goal is to develop a framework for modelling the full structure of the affine apparent transformations of the target in the image plane so that its frontal view can always be retrieved, regardless of the actual camera's viewpoint. Secondly, the desired tracker should be able to operate efficiently in real time, in natural, possibly cluttered urban street scenes. To certain extent it should also be robust to illumination changes and ought to have an ability to retrieve the geometry of the target even when for several frames it remains occluded.

As the diversity of traffic signs and their pictograms all over the world is huge [166], in this work we focus solely on the symbolic signs defined in a single chosen country. The appearances of such signs are standarised in a highway code. However, it should not preclude easy re-training of the system for operation in any other country. This country-related limitation, imposed mainly for practical reasons and to facilitate field tests, should not be reflected in the design of the classifier. Our goal is therefore to make it efficiently discriminate between even a large number of potentially very similar signs, without the upper limit explicitly defined. Besides, we focus on the problem of extracting a possibly the most discriminative lower-dimensionality representation of traffic signs without recourse to the large volumes of difficult to collect real-life images. We want this representation to efficiently combine different visual cues, such as shape and colour.

1.2 Outline of the Approach

The proposed framework for traffic sign detection, tracking and recognition is illustrated in Figure 1.3. The system is composed of three components: detector, tracker and classifier. Unless otherwise stated, the input to our system are image sequences captured by a single car-mounted wide-angle camera. However, the system remains fully operational if static images are passed on input. In that case the tracking component is disabled and the processing only involves detection and recognition.

When performing a dynamic road sign recognition, the input video stream is assumed to be already split into individual frames. An in-vehicle TSR system should capture traffic signs at a considerable distance from the camera in order to allow time for an appropriate analysis, temporal information fusion, and ultimately a reliable recognition. As such distant signs appear nearly stationary, i.e. their apparent

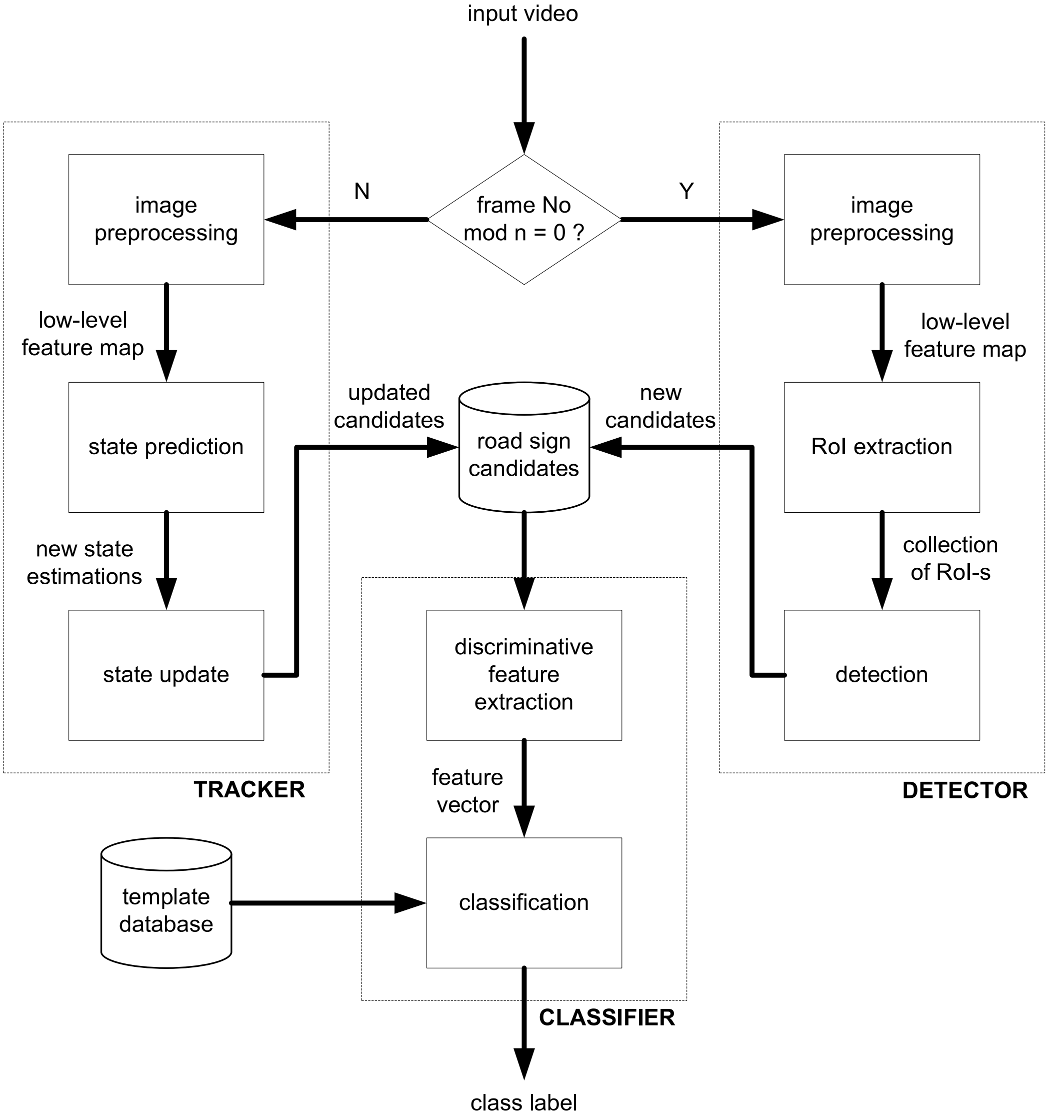


Figure 1.3: Architecture of the traffic sign detection, tracking and recognition system.

motion in the image plane is very slow, performing an exhaustive search for new traffic sign candidates in every frame of the input video is unnecessary. Therefore, at the beginning of the processing pipeline the frame number is checked. Full scene exploration is done only every n frames. In all other frames only the states of the already established sign candidates are updated.

Detection of new likely traffic signs is preceded by an appropriate preprocessing of the input image. It is intended to filter the image in order to extract the desired low-level features, e.g. compute image gradients, determine edges, or amplify patches of certain colour. The details of the image preprocessing techniques used in our approach will be discussed in Chapter 3. With a low-level feature map available, a fast, top-down, quad-tree scene analysis is performed. The goal in this step is to rapidly establish the initial regions of interest (RoI-s) where the likelihood of signs' presence is high. In principle, these regions are determined in a very conservative way, i.e. such that the risk of not capturing all true sign instances in the scene is kept to minimum, even at the cost of some RoI-s covering unnecessarily large fragments of the scene.

Once RoI-s are found, the localised traffic sign detection is performed at multiple scales in each RoI. The detector utilises a binary classifier that for each possible image location within a given RoI yields a soft response associated with a likelihood of the traffic sign's presence in this particular location. Two state-of-the-art object detection techniques are investigated in this work: 1) Hough Transform and its regular-polygon extension and 2) a boosted classifier cascade. In order to improve the accuracy of the detector and to address the problem of multiple positive hypotheses produced around each true sign in the scene, an appropriate kernel-based clustering in the detector's response space is carried out. It is implemented using a *Confidence-Weighted Mean Shift* algorithm. Road sign detectors used and the details of the aforementioned postprocessing technique are discussed in Chapter 3.

Existing traffic sign candidates need to be tracked over time. As a road sign is a rigid and planar object that is not subject to any physical deformations, it is often assumed that instead of temporal modelling of the sign's appearance, it is sufficient to model only its geometry. Based on the known 3D pose parameters any desired view of the target can be retrieved. In Chapter 4 of this thesis three different approaches to road sign tracking are presented where this direction is followed. One of our trackers is a predictor-corrector estimator which combines a Kalman Filter (KF) for centroid/scale prediction and search region reduction with a Particle Filter (PF) used for anisotropic scaling correction. An improvement to this baseline approach is a contour tracking framework based on the so-called *Pixel Relevance Model*, where the relevance of a pixel is defined as a likelihood of it belonging to the sign's contour. The relevance of the pixels contained in a local search region maintained by the KF are updated in each frame of the input video using a spatio-temporal voting technique. This tracker is particularly useful for tracking the road signs in visual clutter where the risk of confusing the true signs' contours with the other, surrounding edges is high.

The last proposed tracker parametrically models the full affine structure of the apparent target motion in the image plane. It utilises a matrix-valued tracking function that encompasses the relationship between the unique feature representation of the target and the affine transformations it is subject to while being approached by a camera. In system runtime this function is instantly learned and periodically updated via regression from the last stable view of a sign in known 3D pose. It is done by generating random affine deformations of the sign and the subsequent minimisation of the sum of squared geodesic distances between the estimated and the known motion matrices. The affine tracker enables reconstruction of the frontal view of a sign regardless of the actual camera's viewpoint, which is desirable from the recognition point of view.

In the recognition stage the classification component analyses each road sign candidate from the pool of tracked candidates. At this point a candidate is assumed to be a (possibly warped) centred image of a sign cropped from the current frame image. This image is first re-scaled to a predefined size. Then, depending on the actual recognition strategy adopted, it is subject to further preprocessing which facilitates extraction of the most discriminative image features that have been determined in the classifier training process. We have developed three different classifiers that are suitable for solving problems involving multiple similar object classes. However, all three classifiers are based on a common idea of representing each sign with a collection of distances/similarities to the class prototypes.

The first classifier is based on a discriminative representation of traffic signs, learned independently for each class based solely on the idealised template sign images. This representation is assembled from a compact set of local image regions where a given template differs possibly the most from all other templates with respect to a novel distance metric based on the so-called *Colour Distance Transform*. The two other classifiers require a number of realistic sign images for training. A soft sign similarity measure is inferred from the pairs of training images, where each pair is labelled: “same” or “different”, depending on whether or not the paired images represent the same or different pictograms. It is then used within the nearest neighbour classification framework. The learned similarity function combines differences between the responses of the most discriminative local descriptors evaluated at the corresponding locations of both images in an input pair. In the first case, a new variant of AdaBoost algorithm [60], called *SimBoost* is used to construct this function. In the second case it is built within a kernel regression tree framework. Details of these techniques will be given in Chapter 5.

1.3 Contributions

The contributions of this thesis in three main areas: detection, tracking and recognition include:

1. **Detection**

   - A quad tree focus of attention operator has been introduced for rapid detection of the regions of interest. This technique requires an integral image to be computed from a chosen low-level feature map of the input image. The realisation of the proposed interest operator presented in this work involves extraction of the regions with high concentration of the colour-specific gradients, which is useful for traffic sign detection. Other implementations, for example capturing motion in an input image, are outlined.

- A *Confidence-Weighted Mean Shift* algorithm has been developed for accurate estimation of the sign's location in the scene. This algorithm treats the detector's response space as a probability distribution with modes to be found. These modes, corresponding to the pursued target locations and scales, are iteratively found using a mean shift clustering technique which has been modified in order to incorporate the confidence of the hypotheses provided by the soft detector's responses. The proposed method is accurate and is shown to efficiently eliminate the multiple redundant candidate hypotheses produced during a dense scanning of the image by a sign detector.

2. **Tracking**

   - Combined Kalman Filter-Particle Filter (KF-PF) road sign tracker has been presented, where the KF part is responsible for centroid/scale prediction and the search region reduction, and the PF part is used for anisotropic scaling correction. This approach allows accurate target tracking in most typical traffic situations and has a limited pose invariance built-in.
   - A contour tracking framework has been proposed for robust detection of road signs in cluttered scenes over time. The tracker combines a Kalman Filter (KF) with an evolving *Pixel Relevance Model* (PRM). The Kalman filter is used for prediction of the sign position and scale and to reduce the search region for the detector. PRM uses the information provided by KF to construct a prior, motion-compensated map of relevances of the pixels contained in this region. The posterior pixel relevance map is built by incorporating the current-frame gradient magnitude information. It is modulated by a Distance Transform image computed around the hypothesised sign contour located at the position predicted by KF.
   - We have applied a regression tracking framework developed by Tuzel et al. [156] to track the signs in the input video. Our tracker is able to model the apparent affine road signs' deformations in various traffic situations. Therefore, reconstruction of a full-face view of a sign, regardless of the actual camera's viewpoint, is made possible. This enables the road sign classifier to fully exploit the image sequence for the entire period when the sign is visible in the field of view of the camera.

3. **Recognition**

   - A discriminative, class-specific representation of traffic signs has been developed. It is learned individually for each sign type by selecting a compact set of local image regions where the clean template of this sign differs the most from the templates of all remaining signs. For measuring dissimilarity between two corresponding regions we utilise a novel distance metric based on a *Colour Distance Transform.* In order to make the proposed representation useful for recognising traffic signs in natural imagery, the colour of each pixel in the observed candidate image is discretised using an off-line learned Gaussian Mixture colour classifier. The classification of an entire observation is done by matching it to the database-stored templates with respect to the learned class-dependent discriminative region sets.

- We have introduced two algorithms for building a robust multi-class classifier that has been tested in TSR as well as for a more general visual object recognition. The proposed algorithms aim at constructing an object similarity function based on the information extracted from the labelled pairs of images representing either the same or different object classes. This similarity function combines the local distances between the paired images, understood as differences of the values of local image descriptors evaluated at the corresponding locations of both input images. The first similarity learning method is based on a modified AdaBoost algorithm which we call *SimBoost.* The second algorithm utilises kernel regression trees, further combined via bagging/boosting and in random forests. We have called these trees *Similarity-Learning Kernel Regression Trees.*

1.4 Roadmap

The rest of this thesis is organised as follows:

1. **Chapter 2** reviews the previous approaches to general visual object detection and recognition. State-of-the-art traffic sign recognition systems are separately discussed. Their limitations are pointed out and the space for improvement is identified.

2. **Chapter 3** presents our approach to road sign detection. First, a method for fast interest region extraction based on a quad-tree focus of attention operator is introduced. Afterwards, two types of detectors for localised capturing of traffic signs in the previously found RoIs are discussed, one based on the Hough Transform and its regular polygon extension, and the other utilising a boosted classifier cascade. A *Confidence-Weighted Mean Shift* algorithm for refining the detectors' responses is finally presented and evaluated experimentally.

3. **Chapter 4** is focused on the tracking aspect of TSR. Three road sign trackers are developed. One combines the traditionally used Kalman Filter with a Particle Filter (KF-PF), which jointly provide accurate geometrical road sign tracking with limited viewpoint invariance. The second technique, is aimed at tracking the sign's contour over time in potentially cluttered scenes and combines the baseline KF-PF framework with a temporally evolving map of

pixel relevance. The last tracker is grounded in the Lie group theory and utilises a function learned via regression from random deformations of the target in known 3D pose. This function enables robust estimation of the affine target geometry in the current frame based on the previous-frame state estimates and the current observation vector. Experimental evaluation of all trackers is provided.

4. **Chapter 5** contains discussion of the similarity-based traffic sign recognition. First, the class-specific discriminative representations of signs are developed based on the principle of *one-vs-all* template sign dissimilarity maximisation. Then, a nearest neighbour classifier operating on these representations is built. Second, two road sign similarity learning algorithms are introduced, one based on *SimBoost*, a modification of the well-known AdaBoost, the other utilising the kernel regression tree framework. Both algorithms learn the sign similarity function from the equivalence information, i.e. pairs of same/different images. Extensive experimental evaluation of the classifiers based on the learned similarity is carried out using both sign and non-sign image datasets and real traffic video.

5. **Chapter 6** concludes the work presented in this thesis and provides a general discussion of its contributions and possible further research directions.

Chapter 2

Overview

Humans are faced with the problem of recognition in their everyday lives. Whether it concerns as intangible matter as disease diagnosis, or more perceivable one, like face identification, there seems to be no universal recipe for an always-working solution. In oher words, every recognition problem is different. Regardless of the actual entity to be recognised, a natural approach involves identifying some number of features that make the object of interest distinguishable. It is usually possible to determine such distinctive features straightaway, based on the domain-level knowledge. However, sometimes the feature space is too large or the discriminative patterns in the data are not easy to spot. This requires aid of the advanced data analysis techniques from an area of data mining, statistical pattern recognition, machine learning, artificial intelligence and the related fields.

Humans are still much more successful in solving many visual recognition problems than the computers are. It is mainly due to the more advanced structure and operation of the biological visual sensors and the neural system responsible for propagating and interpreting this sensory information. As a result, humans can better cope with the uncertainty and ambiguity of the data and are able to capture the relationships between the visual object features extremely fast and reliably. This makes us superior to the computer machines in dealing with what is typically a challenging problem for a machine vision system: viewpoint variation, varying illumination, occlusions, geometrical deformations or background clutter. While the above observation is true for the problems where the relevant object features can easily be perceived, human's recognition capability seems to degrade rapidly with the increase in the number and complexity of the information cues to be analysed. This is the case in many applications, like DNA identification or fingerprint recognition.

Pattern Recognition is a central term to the substance of our work. In this research a small area of pattern recognition, visual object detection, and recognition, is explored. This restricted area of interest is further reduced as we focus on a specific family of automotive machine vision applications – those related to the detection, tracking and recognition of traffic signs from a moving vehicle. Certain methods presented in this work also address several related issues like human detection and car model classification, or are devised in a sufficiently generic way to enable applying them to a much broader class of machine vision problems. Due to the latter fact we find it necessary to set this research against the existing approaches to a broadly defined object detection, tracking and recognition. This state-of-the-art literature is thoroughly discussed in Sections 2.1 and 2.2. Particular emphasis is put on the traffic-related applications which are analysed in section 2.3. In Section 2.4 limitations of these previous approaches to TSR are pointed out.

2.1 Visual Object Detection and Tracking

Visual object recognition is perhaps what comes to our minds most often when thinking about recognition in general. We recognise objects when trying to spot a colleague in a crowd, when reading a newspaper, or while driving a car. A similar sequence of actions is performed hundreds and thousands of times every day, when a human is trying to catch sight of a previously seen object in the environment, not necessarily on purpose, sometimes even subconsciously. When image and video processing became attainable on a computer machine, automation of these tasks was found a natural direction to follow. However, automatic object detection and recognition is still a difficult undertaking and only a limited progress has been made for over 30 years of research in this area. It seems that the way humans do it is still far more efficient and effortless compared to most of the existing computer algorithms.

A robust object detector must be able to cope with the diversity of the imagery constituting the scene. Normally, unless further restricted by the problem constraints, actual image may depict the object of interest in a non-uniform, possibly cluttered background, in varying pose, scale and illumination, all of it affecting its appearance. In certain applications not only the target object has to be detected, but also its specific identity should be determined. The latter task, classification, is frequently even more of a challenge than the detection, especially when a large number of unique but similar instances of the same object category exist. An example of such a challenging problem is face recognition, e.g. [71, 75, 77, 94, 105, 155] or fingerprint recognition [96]. Frequently, there is no clear distinction between the detection and the recognition, but even if they are performed sequentially, the latter may strongly depend on the former. In a video context, where a motion comes into play and the scene changes over time, also the third component becomes necessary, the target tracker. Depending on the actual problem characteristics, the tracking can be implemented at different levels to model solely the motion of the target, its temporal appearance variability, or both.

One popular approach to object detection is by first preprocessing the raw image in order to locate the salient regions likely containing the target objects. Further analysis is performed only in the found regions of interest (RoI). Discovery of such regions is typically performed by simple image filtering aimed at extracting from a raw image certain low-level features, like independent colour channel values, edges, or gradient directions/magnitudes. However, if this is insufficient, RoI finding can also be driven by a more advanced image segmentation technique, e.g. region growing [53, 10, 130, 68], clustering [70, 149], watersheds [23, 24], background subtraction [62, 63, 64, 89, 116] and many other. Actual choice of the segmentation method depends on the nature of the problem and the input feature space. For example, in the case of traffic sign detection, a natural approach is to concentrate on the highly-contrasting regions of characteristic colours. In most surveillance applications focus is primarily put on those regions of the scene where temporal appearance changes are observed. In many cases, however, probability of the target object existence is relatively regularly distributed over the entire image, which renders an early segmentation void. Furthermore, in many complex systems such a sequential approach to object detection may be a cause of problems. A failure in RoI detection at an early stage of the processing pipeline is irreversible, which makes the following components fail too. In such cases using dedicated, problem-specific focus operators is usually a workaround.

Segmentation and detection are seldom performed in a raw RGB colour space. For a majority of applications a more discriminative feature representation has to be constructed. Therefore, an identification of the features that most adequately describe an input image or image sequence is a critical step in the development of a successful object detector. Feature extraction from an image may involve colour space transformation [97, 129]. Another possibility is to search for various low-level descriptors, such as gradient orientations, edges, corners or primitive shapes. A range of well-established image processing techniques are available for all of these tasks. For specific colour extraction usually some sort of discretisation is performed. It is based on a set of predefined thresholds or simple pixel-wise transformations applied to the image in the selected colour appearance model. More advanced techniques rely on the training data from which the colours are learned and modelled probabilistically, usually with Gaussians or Gaussian Mixtures [15]. To extract gradient and edge information, convolution of the image with an appropriate kernel mask typically produces satisfactory results. Similar morphological operations can be used to detect certain structures, e.g. simple shapes or angles, but in general the convolution-based techniques are known to be computationally too expensive for many practical applications. For more accurate edge extraction any of the well-known operators can be used: Roberts, Sobel, Prewitt, Laplacian of Gaussian, Canny. They are reviewed in many computer vision handbooks, e.g. [53].

Alternative techniques have been proposed for edge detection, for example those based on watersheds [48] or genetic algorithms [69]. There is also a rich literature on detecting corners or, more generally, interest points, the term being first introduced by Schmid and Mohr [51]. A simple yet expensive way of doing this is by using the abovementioned morphology. Moravec operator [5], one of the earliest corner detector algorithms, analyses similarity between the image patch centered at a given pixel and the nearby, largely overlapping patches. The main deficiency of this operator is related to the fact that it easily gets confused by the edges which are not in the direction of the neighbours. More robust corner detectors take advantage of the second image derivatives: Harris operator [20] and its affine-invariant version [107], Tomasi-Kanade operator [25]. Wang and Brady [40] introduced an algorithm that looks for the points where the edges change direction rapidly. The most computationally efficient feature point detectors available [54, 140] directly test whether a patch under a pixel is self-similar by examining the nearby pixels. An integrated keypoint extraction algorithm, SIFT, has been proposed by Lowe [112].

This method enables extraction of distinctive invariant features that can be used for reliable matching between different views of an object or scene. Stability and robustness to various image transformations have further been improved by Ke and Sukthankar [114] in the extension of the original SIFT procedure, PCA-SIFT. Kadir and Brady [78] on the other hand proposed a salient region detector based on the principle of entropy maximisation. This method was further generalised to affine invariance in [117]. A number of other affine covariant region detectors have been developed. A good review of these methods is given by Mikolajczyk et al. [126].

Apart from easily perceivable low-level features like colour, edges ot corners, that are extracted via an appropriate filtering of the image, other, more complex local descriptors can be constructed using various sorts of transformations. For example, colour histograms [76, 83] are descriptive features that provide a useful characteristics of the image. They are frequently used as global descriptors for image matching. Histograms of oriented gradients (HOG) [132] are more often used as local features. They appear to be a good alternative to colour histograms as they very well capture the local spatial relationships between the image patches and are robust to illumination changes. HOG descriptors have been successfully used in a number of applications, e.g. for human detection [139], and are also employed as an underlying representation of the abovementioned SIFT features [112]. Furthermore, there were attempts to handle colour and edge information jointly in histogram construction in order to improve the detection accuracy [84].

Detecting more complex structures, such as arbitrary shapes or textured objects, is in general a more demanding task than discovering primitive features. One elegant way of capturing shapes is by using Hough Transform (HT) [3, 9, 29]. This method is suitable for both analytically and non-analytically described shapes. The principle of Hough Transform is that each pixel "votes" for each feature that it could possibly be a part of. This approach is inherently robust, as gaps, noise, and partial occlusion are ignored, but appear as a decreased strength of the feature. However, there are several major limitations of HT. First, it only utilises the object's contour information and is highly dependent on the quality of the edge/gradient image from which it is calculated. Second drawback is a fast-increasing demand for resources with the increase in the dimensionality of the parameter space, which considerably complicates a real-time implementation. Alternative methods are available for detecting narrower shape classes. For instance, in [93] a local radial symmetry property is utilised to detect the equiangular polygons. This approach is reported to be invariant to in-plane rotation and is sufficiently fast for real-time execution. A different direction is followed by Schneiderman and Kanade [106] who localise structured and textured objects like faces or cars in the scene. They introduce a trainable object detector which utilises multiple classifiers, each exhaustively scanning the input image in search of the target in a specified scale and orientation ranges. These classifiers are based on the class-conditional statistics of localised parts designed to capture various combinations of locality in space, frequency, and orientation.

Local features in natural images are usually well-localised in the space and frequency domain, and often also well-oriented, e.g. directional textures, or linear discontinuities (edges). Furthermore, any image patch can be represented as a superposition of some base functions. These observations provide rationale for a whole family of feature generation methods known as *multi-resolution analysis*. In the object detection domain they are used to decompose an image in multiple scales using a chosen class of basis functions. The resulting coefficients are used for target/background separation. A number of image descriptors associated with different function bases have been proposed in the state-of-the-art-literature: Fourier-based filters, wavelet-based filters, curvlets, ridglets and many other. An interesting review of the wavelet-based detectors operating on noisy images is given by Abdelkawy and McGaughy [91]. In [87] an efficient directional wavelet-based filter is used for ridge-line detection. Recently, Viola and Jones [111] designed an extremely fast algorithm for object detection using very simple basis functions – Haar wavelets. They showed that an efficient integral image representation, powerful key feature extraction algorithm based on AdaBoost [60], and a cascaded classifier design, provide an ample compensation for this simplicity. The algorithm has found a number of applications [99, 111, 131, 127].

In the recent years a rapid increase in the number of dynamic object detection and recognition applications has been observed, particularly in the areas of surveillance and visual driver assistance. Dynamic detection and recognition systems operate on live video. Therefore, they require fast algorithms to meet the real-time processing requirements. In such a scenario, the exhaustive exploration of the entire scene in search of the target object candidates is prohibitively expensive. Moreover, processing of each video frame independently does not take into account possible temporal dependencies between the consecutive frame observations. To overcome these limitations, various object tracking schemes are introduced. The simplest possible scheme consists in merely reducing the local search region at time t to the neighbourhood of the target object at time $t-1$. Obviously, this method does not model any correlations between the appearance of the target at different time points. It only makes a somewhat wrong assumption on the constancy of the target position in order to reduce the search region. However, if the motion of the target in the image plane is slow, which is the case in many applications, e.g. when the target is at a large distance from the camera or/and moves in the direction parallel to its optical axis, this approach might be perfectly adequate. At least in the situations when the actual detection method is sufficiently fast to allow the detector to be run in every frame of the input video in real time.

The true visual tracking system realises two tasks: target representation and localisation, and filtering and data association. Probably the simplest method of target representation and localisation is by storing its basic geometrical properties, e.g. centroid's position and scale, and computing their corresponding differences. This suffices for tracking of objects that are either physically small or are seen at a sufficiently large distance from the camera to consider them blobs. For closer and/or larger objects the geometrical representation must be adapted to the motion model that best describes how they move through the scene over time. Therefore, tracking of the planar objects typically requires the entire 2D geometrical transformation to be modeled, either affine, or homography. Motion model of a rigid 3D object must define its aspect depending on its 3D position and orientation.

Another common target representation and localisation method is through the interest points or regions, or more generally – features. These can be captured using a range of methods discussed earlier in this section. Parameters of the object's motion are estimated from the interframe correspondences between the features, depending on the actual representation used. For estimating the image correspondence from interest points, the iterative RANSAC algorithm is often used [6, 85]. Alternatively, the motion of an object can be inferred based on the optical flow field. A number of optical flow computation algorithms are available, e.g. [7, 8]. Object's motion can also be estimated using the block matching technique [47, 80], frequently used in video compression. Other well-established approaches to representation and localisation of the target include mean-shift tracking and contour tracking. The former is an iterative, kernel-based method that consists in finding maxima of a similarity measure between the density estimates of the model and the target image. The most typically used similarity measures are Bhattacharyya coefficient and the Kullback-Leibler divergence. Robust kernel-based tracking has recently undergone a rapid development [100, 133]. Contour trackers are generally referred to as *active contours* or *snakes* and are particularly useful for tracking the boundary of deformable objects that are clearly separable from the background [30].

Filtering and Data Association is mostly a top-down process. It involves estimating the true state of a dynamical system, either linear or nonlinear, given only a series of noisy observations of that system. In our case the hidden state of the system encodes the object's appearance or its motion parameters. One of the most popular filters is a Kalman Filter (KF) [1, 13]. Kalman filters are based on linear dynamical systems discretised in the time domain. The state of the system is assumed to be an unobserved Markov process, which implies that the current state is conditionally independent of all earlier states given the immediately previous state. At each discrete time increment, a linear operator is applied to the state to predict the new state, with some noise mixed in, and optionally some information from the controls on the system, if they are known. Then, another linear operator mixed with more noise generates the visible outputs from the hidden state. The Kalman Filter is optimal in a Bayesian sense for Gaussian random variables an can be considered analogous to the Hidden Markov Model (HMM) [17], with the key difference in that the hidden state variables take values in a continuous space (as opposed to a discrete state space in HMMs). However, the basic KF is limited to a linearity assumption. The Extended Kalman Filter (EKF) [13] is a nonlinear version of the Kalman Filter which approximates the optimality of Bayes' rule by linearization about the current state's mean and covariance. It is considered a standard in many nonlinear estimation applications, e.g. navigation systems and GPS. Unscented Kalman Filter (UKF) [46] is an improvement to the EKF which approximates the state probability density with a set of sample points. This guarantees better estimation when the transition and observation models are highly nonlinear.

More sophisticated filtering techniques are also available in computer vision. Particle filter (PF) [32] is a model estimation method based on simulation and can be considered a sequential (on-line) version of Markov chain Monte Carlo (MCMC) methods. It aims to estimate the sequence of hidden state parameters x_k, based only on the observed data y_k, where x_k follow the posterior distribution $p(x_k|y_1, \ldots, y_k)$ and the observations are assumed conditionally independent. The estimation is made by modelling the filtering distribution with a weighted set of samples. Their weights are sequentially updated using an on-line version of importance sampling technique [18]. Additionally, in order to avoid the degeneracy problem, resampling is performed when the effective number of particles drops below a predefined threshold. The actual performance of the PF tracker mostly depends on the number of samples and the choice of the importance function. The transition prior is frequently used in this role. A popular application of this variant of PF to contour tracking is known as *Conditional Density Propagation*, or for short a *Condensation* algorithm [52]. A number of other particle filter tracking applications emerged in recent years, e.g. [119, 134, 147, 148].

2.2 Object Recognition

Detection can be considered a special case of a classification problem with only two classes: the target and the background. In certain applications the problem to solve is defined such that only the anonymous instances of the target class need to be captured. A good example is a traffic monitoring system for vehicle counting or a human detector. In such systems the determination of the exact type or identity of the object is explicitly considered irrelevant, or for objective or technical reasons it cannot be found, e.g. because all instances of the target class look the same or the target is too far away from the camera. However, other problems clearly separate the detection from the classification. For example, a traffic sign recognition system operating on board of a moving vehicle must not only capture the sign instances in the scene, but also their exact types have to be determined in order to issue correct signals to the driver. Similarly, the contemporary visual access control systems not only detect human faces, but further, for verification, also match them to the database-stored pictures of the individuals with granted access to the guarded premises.

In general, visual object recognition approaches can be divided into discriminative and generative methods. In brief, the latter family of methods can be described as those that model the distribution of the image features, while the discriminative methods do not. More formally, the discriminative recognition models are a class of methods used for modelling the dependence of a target class variable c on an observed vector of features $\mathbf{x}$. Within a statistical framework, this is done by parametrical modelling of the conditional probability distribution $p(c|\mathbf{x})$, which can be used for predicting class c from the observation x. The values of the unknown parameters are usually inferred from a set of labelled training data. This may be done by making maximum likelihood estimates of the parameters, or by computing distributions over the parameters in a Bayesian setting. Generative models on the other hand model the joint distribution $p(c, \mathbf{x})$ of image features and class labels. This is typically done by learning the class-conditional densities $p(\mathbf{x}|c)$ and prior class probabilities $p(c)$, if they are not explicitly available. The required posterior probabilities required for classification are obtained using the Bayes' theorem:

$$p(c|\mathbf{x}) = \frac{p(\mathbf{x}|c)p(c)}{\sum_i p(\mathbf{x}|c_i)p(c_i)} \tag{2.1}$$

The generative and the discriminative approaches to object recognition have different properties and complementary strengths and weaknesses. A good discussion of those is given by Ulusoy and Bishop [125]. Let us first introduce the state-of-the-art discriminative recognition methods.

One of the most common techniques of discriminative visual object recognition is template matching. This approach associates a real-valued function with each class and assigns such a label to the unknown observation that maximises value of this function. Naturally, this function approximates the posterior probability $p(c|\mathbf{x})$ required for classification and the equal class priors $p(c)$ are implicitly assumed. Matching is a very generic term, and in practice it may proceed in many ways, depending on the chosen matching criterion, image representation, and matching control strategy. The most popular criterion used is a normalised cross-correlation between the tested image and the template, understood as some function (e.g. inverse or exponential) of the image distance. The latter may be expressed with different metrics e.g. Euclidean, Manhattan, or Chamfer, and over different image representations, e.g. raw RGB pixels, gray-level values, or binarised values. In certain applications other, more complex image representations and distance metrics may be suitable for matching. For example distance transform maps [16, 41] are convenient for contour matching. Haussdorf distance is particularly useful for shape comparison [31]. Also Fourier matching in frequency space is possible and sometimes very useful, e.g. for image mosaicing. Unfortunately, pixel-wise image comparison is very sensitive to pose changes and computationally expensive. One possible solution to this problem is via a block-wise comparison with content averaging or histogramming. A more robust approach involves comparing the images in a pyramidal fashion. Such a hierarchical, multi-scale image analysis dramatically reduces the computational load as the unlikely class templates can be discarded at a very early stage. An example of this approach using Distance Transforms is given by Borgefors and Gavrila [19, 55]. Their algorithms are reported to be fast and insensitive to noise and other disturbances.

An alternative matching approach that is insensitive to geometrical distortions utilises the abovementioned SIFT features [112], which are known to be resistant to a whole family of common image deformations as well as illumination variations and noise. For example, scale invariance is achieved by multi-scale image analysis and extraction of only those points which are significant across all scales. Rotation invariance on the other hand is determined at the representation level: around a given point a quadruple of Gaussian-smoothed gradient orientation histograms are calculated relative to their dominant orientation. More recently a significant improvement to SIFT was proposed: PCA-SIFT [114]. In this method image saliency is encoded again with the SIFT-like descriptors, but instead of smoothed, weighted histogram computation, the Principal Component Analysis (PCA) is applied to the normalised gradient paths. PCA-SIFT features are reported to be even more stable, distinctive, and robust to image transformations than the original SIFT. Robust image matching is also performed within the salient regions. A number of different approaches for the extraction of such regions have been proposed. For example, Matas et al. [118] define the salient region by an extremal property of the intensity function in the region and on its outer boundary. Kadir et al. [78, 117] relate the region saliency with the entropy of the contained intensity values. Alternative approaches to salient region extraction and their evaluation in image matching under various image transformations are reviewed in [126].

In certain situations feature point/region matching is unsuitable. First, feature extraction step may be computationally too costly, especially when it involves multiscale image analysis. Second, the target objects may be too small or have locally poor distribution of the underlying feature values, e.g. uniform colour, unidirectional gradients. This often results in an insufficient number of key features for matching. In this case global region descriptors may prove to be more adequate. Examples of such descriptors are abovementioned colour or gradient orientation histograms, popular global statistics like variance, energy, entropy, various moments and their invariants [53, 10, 2, 113]. Other possible representations may involve computation of shape-oriented descriptors like area, eccentricity, compactness or various contour signatures [53, 10]. Alternative solution to the problem of lacking discriminative object features is constructing the feature representation of the target classes from a large number of weakly discriminative image descriptors and combining them in a robust way. This principle underlies the boosting classification algorithms, e.g. AdaBoost, which have been proved extremely successful in a large number of applications, e.g. [99, 111, 131, 127, 151].

Other discriminative object recognition techniques feature a more complex classifier design. Linear discriminant analysis (LDA) are a family of methods that attempt to find the linear combination of features which best separate the target classes. AdaBoost [60, 61] is a supervised learning algorithm that also finds an optimal linear combination of weakly discriminative classifiers to construct a robust strong classifier. The strength of this method lies in its adaptivity to the data. The subsequent classifiers built are tweaked in favor of those training instances misclassified by the previous classifiers. AdaBoost algorithm, originally designed for solving the binary classification problems, has been extended to handle multi-class problems. However, its numerous adaptations are mostly based on various schemes for reducing the multi-class classification to multiple two-class problems, e.g. Guo et al. [77]. One of the exceptions is the recent study of Hao and Luo [146] where the authors derive a generalised form of AdaBoost based on the multi-class exponential loss function, as opposed to the two-class version of the loss function used in the original algorithm.

Support Vector Machines (SVMs) [42] are a set of related supervised learning methods used for binary classification and regression. SVM constructs a $n-1$-dimensional hyperplane optimally separating the n-dimensional data points. This hyperplane is chosen such that its distance from the neighbouring data points of both classes is maximised, i.e. the class separation margin is the largest possible, and in the same time the penalty function characterising the misclassified points is minimised. SVMs are suitable for solving both linear and non-linear classification problems, the latter requiring kernelisation [26]. As a result, the maximum-margin hyperplane is fit to the data in the transformed feature space. In order to adapt the Support Vector Machine to solving the multi-class problem, it is typically reduced to multiple binary problems, each associated with a dedicated binary classifier. Two common strategies of constructing such partial classifiers are *one-versus-all* and *one-versus-one*. The former involves training a separate classifier for discriminating between each class and all other classes. The latter method builds separate classifiers to distinguish between every pair of classes. The ultimate classification is done by picking the label associated with the maximum number of votes accumulated. Multi-class SVMs can also be implemented based on the generalised notion of the margin [88], which avoids the abovementioned problem binarisation. Support Vector Machines have a number of applications in computer vision, e.g. [57, 71, 75].

A yet another broad family of discriminative visual object recognition approaches utilise the feed-forward neural networks (FFNNs). Such networks typically contain an input layer with the number of neurons reflecting the dimensionality of the image feature vectors, an output layer with one neuron per class, and at most several hidden layers. They are trained in a supervised iterative way using the labelled data. Specifically, each sweep forward through the network results in the assignment of a value to each output neuron. This output is compared to the desired output: “1” for the node representing the correct class, and “0” for any other node. Error terms computed from that comparison are then used to adjust the weights in the hidden layers of the network so that, hopefully, when the next observation is presented on input of the network, the output values will be closer to the correct values. Neural network is trained until no further reduction in the classification error rate is observed. The most common FFNN training algorithm is back-propagation [37]. Many alternative network structures and learning strategies, but discussing those is far beyond the scope of this thesis. Selected studies where the applications of neural networks to visual object recognition are discussed are: [22, 92, 108].

In many applications recognition of the target must be performed based on a sequence of related observations. Various probabilistic graphical models are particularly suitable for addressing the recognition problems where dependencies between such observations need to be modeled. A conditional random field (CRF) is one type of discriminative probabilistic model often used for labelling of the sequential data [79]. In this model, the distribution of each random variable Y_i, associated with the label of the i-th observation in the sequence, is conditioned on the entire input sequence X, which avoids the need for independence assumptions between the observations. The label variables, Y_i, are usually arranged as a chain, with an edge between each Y_{i-1} and Y_i. This layout admits efficient algorithms for model training and inference. Conditional Random Fields have been recently extended to allow hidden states in the model (so-called Hidden Conditional Random Fields - HCRF). Quattoni et al. [109] model the objects as flexible constellations of parts conditioned on local observations found by an interest operator, where parts are hidden variables. For each object class the probability of a given assignment of parts to local features is modeled by a CRF. Wang et al. [137] applied a similar HCRF framework to model temporal sequences for gesture recognition. Graphical inference model for object recognition over time is also used by Yun et al. [159]. In this study the conditional probability of one object's existence at time t, given the existence of other objects at time $t-1$ is represented by a trained transition matrix.

An entirely different object recognition approach is represented by a generative paradigm. Generative object recognition methods are particularly useful for solving massively multi-class problems where the intra-class variation is high. This is often referred to as object categorisation, rather than classification. One of their great advantages of the generative methods is that they can utilise both labelled and unlabelled data, which minimises the human supervision in the model learning process. Secondly, generative models admit easy, incremental addition of new classes for which the class-conditional density can be learned independently of all classes already existing in the model. Finally, these models naturally handle objects' compositionality. As already mentioned, generative object recognition methods model the likelihood and the prior rather than the posterior directly. One of the two most popular models following this strategy is a *bag-of-words* (BoW) model originating from the natural language processing.

To represent an image using BoW model, an image is treated as a document composed of visual words. They are usually constructed in three steps: feature detection, feature extraction, and codebook construction. The first two tasks are usually done by extracting regularly-spaced regions [122] or by applying a chosen interest point detector on a number of example images and constructing robust descriptors around the extracted salient points [123, 122]. The codebook is typically built via k-means clustering over all feature descriptors obtained. Therefore, the resulting words can be considered representatives of multiple similar image patches. With the learned visual vocabulary available, an image is modeled as a geometrically unconstrained collection of local patches, each represented by a codeword from the vocabulary. In the training stage the goal is to learn the model that best represents the distribution of codewords in each category of objects or scenes. To achieve that goal, Bayesian inference is typically applied, e.g. naïve Bayes model [121] or more complex hierarchical Bayesian text models [123, 124, 122]. In order to categorise an unknown image, all the codewords are first identified in it. Then the category model is found that best matches the distribution of codewords in this particular image.

The bag-of-words models do not encode rigorous geometric information about the structure of an object. Therefore, usually a better choice for object categorisation is to adopt *constellation* models which explicitly assume that the objects are made of parts [56, 73]. These models encompass the important properties of an object: its shape, i.e. the relative locations of its parts, as well as the appearance of parts, both in a probabilistic (usually Gaussian) way. An image is typically treated as a collection of regions obtained by picking maxima of the interest operators like multi-scale Harris [20] or saliency [117]. Since the number of image features is typically much greater than the number of parts in the model, a hidden indexing variable is introduced which allocates features to parts. The model parameters can be learned for example via Maximum Likelihood (ML) approximation [**?**, 102].

However, this approach assumes a well-peaked parameter distribution, which relies heavily on the sufficient statistics of data. Therefore, a large number of training images are required in this case. Variational Bayesian Expectation-Maximization (VBEM) algorithm proposed by Fei-Fei et al. [103] is a different approach which utilises class priors (average shape/appearance means and variances) assembled from the previously learned unrelated object categories. This algorithm is superior to the ML approach in that it requires very few training images of each new class being added to the model.

2.3 Background of Traffic Sign Recognition

Road signs are an inherent part of the traffic infrastructure. They are designed to regulate flow of the vehicles, give specific information to the traffic participants, or warn against unexpected road circumstances. Road signs are always mounted in places that are easy to spot by the drivers without distracting them from manoeuvring the vehicle, e.g. on posts by the roadside or over the motorway lanes. Besides, their pictograms are designed in a way that admits easy discrimination between multiple signs, even from the considerable distance and under poor lightning and weather conditions. These properties of traffic signs have not changed for decades, with the exception of better, more endurable and more reflective materials being used in the production process. In addition, new sign types are constantly introduced to reflect an inevitable technological advance in the traffic infrastructure and road safety standards.

An interest in automatic road sign recognition arose naturally with the introduction of the machine vision systems and can be traced back to the research on vision-based mobile robots in the late 1960's and early 1970's. However, at that time the computational power available on a mobile platform was very limited and due to this technological gap only very few research groups ventured to work in this area. More attention to the problem was given only later, in the beginning of the digitisation era, when video processing became more attainable on a computer machine and in the same time the demand for driver comfort and safety increased. At that time the traffic sign recognition problem was considered in a much broader context of multi-aspect driver support, which set foundations for the term *Driver Support Systems* (DSS). This has not changed until now. Such systems are generally referred to as hardware and software tools providing continuous assistance in the routine driver's activities, giving information about the upcoming major navigation decisions and potential dangers, as well as monitoring the vehicle's state and the level of driver's safety.

Significant advances in the area of DSS were made in late 1980's and 1990's when numerous large-scale projects were developed in the USA, e.g. DARPA *ALV* programme [1], *IVHS* project [2], Europe, e.g. *PROMETHEUS* project [3], and Japan. The main contribution of these projects was popularisation of the intelligent vehicle concept, which gave birth to the numerous academic and industrial automotive research groups. It would have probably never happened if it had not been for the technological progress of the 1990's which made it possible to operate a visual driver assistance system on board of a vehicle with the use of the off-the-shelf video cameras and cheap mobile computers. In parallel, numerous research papers were published and many international discussions on DSS were initiated. One of such discussions in Summer 1990 in Tokyo not only resulted in a valuable book by Masaki et al. [27] on vision-based vehicle guidance, but also started a series of annual *Intelligent Vehicles Symposia'* sponsored by the *IEEE Industrial Electronics Society.*

At present, visual driver assistance is one of the hot topics in the machine vision and the intelligent vehicles communities. Many groups around the world conduct research in this area and the most innovative solutions are given opportunity to be tested in the realistic traffic situations in the famous DARPA Grand Challenge [160]. Large progress has also been made recently in traffic sign detection and recognition which resulted in the first successful, large-scale commercial applications, e.g [161, 162]. Although the existing solutions expose numerous limitations, the future of this technology is bright. With the current pace of development kept, in the coming years we are likely to see millions of new cars with the automatic visual traffic sign recognition systems operating on board. Let us below briefly review the published research that led to the success of this technology we are observing today.

First published studies related to the problem of traffic sign recognition (TSR) are dated mid 1980's. They discussed different computer vision methods for sign detection and recognition involving colour segmentation, edge analysis, various forms of template matching, classification based on the neural networks and many other. An excellent compilation of these studies can be found in [39]. Since traffic signs have *a priori* known shapes and colours, in a vast majority of the early approaches to the TSR problem both pieces of information were exploited. The typical setup involved three-stage processing: detection, analysis and classification. In this model a global scanning of the scene was first performed in order to identify a small number of subregions which should be tested for compatibility with the hypothesis of representing a traffic sign. This preliminary detection was usually driven by colour cue analysis. Traditionally, in the second step the extracted interest regions were more carefully analysed in order to be mapped into the appropriate semantic class categories or discarded. The more accurate classification of the positively verified hypotheses was done in the final step.

[1] *Autonomous Land Vehicle* programme led by *Defense Advanced Research Projects Agency*; focused on key technology issues leading to a new generation of intelligent, autonomous vehicles

[2] *Intelligent Vehicles Highway Systems*; a number of projects focused on improving safety and efficiency of Virginia state transportation network, and addressing various problems from traffic management and vision-based traffic monitoring to vehicle control

[3] Programme for an European Traffic with Highest Efficiency and Unprecedented Safety; 1986 – 1994, launched by the European automotive industry to develop intelligent vehicles able to self-organise their local driving manoeuvres in a co-operative manner without requiring minute guidance and control from the traffic management facilities

Such a classic approach was adopted in many early works, for example those related to the abovementioned *PROMETHEUS* project [28, 34]. In these studies the processing started with pixel-wise colour classification using neural networks to detect regions of interest (RoI). Then, a connectivity analysis was applied to the extracted regions to yield their symbolic description consisting mostly of colour, contour code, neighbourhood relations and the relations to the enclosed subregions. Finally, based on the *a priori* knowledge and the available symbolic descriptions of RoIs, the latter were filtered and for the remaining regions most likely containing traffic signs the appropriate class-membership hypotheses were generated. The final classification was reduced to answering a question about inclusion of certain coloured geometrical primitives in other primitives.

A similar strategy was chosen by Piccioli et al. [49]. They reduced the initial search region to those parts of the scene where the signs are most likely to emerge, assuming a forward-looking in-vehicle camera. Further localisation was obtained through the colour discretisation in a Hue-Saturation-Value (HSV) colour space [97, 129], granulation, and finally clustering. Once extracted, RoIs were analysed in terms of the presence of straight edge segments. Further, these segments were checked for having appropriate length and slope and being in appropriate relations to one another so as to filter out those which could not be part of traffic signs. Finally, the normalised cross-correlation (NCC) between the candidate image and the database-stored templates was calculated and the classification was made by picking the label of the best-scored model image. Frame-based classifications were additionally fused over time to make the final decision more reliable.

The algorithm of Escalera et al. [50] is an another example of the typical three-stage design. In the detection stage the colour information was utilised via fast image filtering in a colour space defined by ratios of the individual RGB channel intensities to the sum of all three channel intensities. In order to detect corners in such filtered images, authors used convolution masks optimally approximating the angles between the desired triangular and rectangular signs' edges. Analysis of the geometrical relationships between the found corners was in turn performed to finally verify the hypotheses of presence or absence of particular sign categories. For ultimate classification, the raw image regions cropped from the found candidate locations were passed on input of a multilayer perceptron neural network.

Several approaches to traffic sign detection did not take the colour information into account. Gavrila [55] proposed a hierarchical, multi-feature template matching technique based on Distance Transforms (DT) [16]. In this framework low-level features, such as oriented edge or corner points, were extracted and a DT was computed for each resulting binary feature map. The input image was matched in a coarse-to-fine fashion to the DT images of multiple road sign templates which were additionally translated, rotated and scaled in order to achieve the geometrical transformation invariance. The template DT images were organised in a tree structure which decomposed the entire set of traffic signs into subgroups according to their similarity in a feature space. At each tree node the test image was matched to a prototype template representing all similar templates at this node, rather than to each template separately. This dramatically sped up the matching process.

An entirely colourless approach to sign detection and recognition was also presented by Paclík et al. [66, 67]. In these studies the input edge map was scanned at different scales to enable comparison of the detected geometrical objects' shapes with the template road sign shapes. This resulted in a preliminary categorisation of the detected sign candidates. Further classification was arranged as a hierarchical process based on a decision tree with a structure reflecting the semantic decomposition of the road sign family. The rationale for this hierarchical classification scheme is that at each tree node the local classifier may operate in a feature space spanned by the image features that are the most discriminative for a particular subgroup of signs. Paclík et al. used the Parzen window classifier with product Laplace kernel and the features based on global image statistics, like mean, energy, entropy, moments and Hu's moment invariants [2]. The proposed multi-stage classifier demonstrated good recognition accuracy and was reported to quickly reject false alarms of the detection layer. An additional advantage of this approach is that it admits partial, coarse-level sign classifications. Note that when the decision at the finest level of the hierarchy is of little confidence, such a partial labelling, e.g. "speed limit" instead of "40 kph speed limit" or "50 kph speed limit" may still be valuable to the driver.

Prior knowledge about the colour and shape of traffic signs was utilised at the detection stage by Escalera et al. [120] who developed a deformable model of sign. In this framework an initial model represented a sign at a fixed distance, perpendicular to the optical axis of the camera, and located at the centre of the image. A deformable model was defined with respect to the deformation parameters $\mathbf{Z}$ which encoded the sign's displacement, scale changes and the in-plane rotation. It was iteratively updated based on the previous model parameters. Their optimal values were obtained via minimisation of the energy functions which related $\mathbf{Z}$ to the observable candidate sign's features: boundary colour, chromaticity of the sign's interior, gradients and edges. Two methods for searching of these parameter values were proposed: genetic algorithm and simulated annealing [11]. Classification of the captured road sign candidates was done using normalised cross-correlation template matching. The approach of Escalera et al. was tested using a number of images captured in realistic traffic scenes. It was able to successfully detect traffic signs under minor geometric transformations, varying illumination and occlusions. However, in this study a fixed affine motion model was used, making the approach inflexible. In addition, the detector was computationally too expensive for real-time implementation.

A joint treatment of colour and shape was proposed for example by Fang et al. [92] who exploited the known properties of the model signs in their detection framework. They built separate two-layer neural networks (NN) to extract colour and edge features from the input map of raw RGB pixel values. The "colour" networks were designed in such a way that the output neurons, which correspond to the potential road sign centres, got very excited only by the signal passed from the input neurons representing pixels of certain colour and satisfying appropriate geometric relationships defined by the model signs. In the similar way the "edge" networks worked, but they represented a mapping between the appropriately coloured edges on input and the possible sign centroids on output. The soft responses of both output layers for each pixel were combined in a fuzzy way, yielding an integrated sign-membership map. Apart from the novel NN approach to traffic sign detection, Fang et al. proposed a simple yet mathematically solid road sign tracking framework. With an assumption of known motion model and size of the real signs, they built a Kalman Filter (KF) tracker for prediction of the position and size of the detected sign candidates. It enabled substantial reduction of the local search space in the consecutive frames of the input video.

Formalised tracking algorithms were introduced in other, earlier published papers. A similar filtering scheme to the one of Fang et al. was proposed by Piccioli et al. [49]. Their tracker predicted the 3D position of a sign, based on its measured apparent location and size in the image plane, and assuming known physical sign's size and the straight, uniform (with known velocity) motion of the car along the optical axis of the camera. Extended Kalman Filter (EKF) was adopted to linearise the measurement model equations around the predicted state. Miura et al. [65] introduced a more sophisticated, but still KF-based, geometrical sign tracker. It was composed of two separate cameras, one wide-angle and one telephoto. While the wide-angle camera was used for capturing and tracking of the likely traffic signs in the scene in a similar way as in the abovementioned study of Piccioli et al., the actual recognition was performed from the images captured by the telephoto camera. It was directed to the predicted position of each prominent tracked candidate in order to capture the sign in more detail (zoomed), which improved the classification accuracy. The main limitation of the two-camera system of Miura et al. was its inability to focus on more than one sign at a time.

Several more recent studies deserve attention. Gao et al. [142] addressed a static sign recognition problem by employing a biologically-inspired model of vision. The segmentation component utilised *CIECAM97* colour appearance model to extract colour information from the input image and, based on it, capture the likely traffic signs. This process was facilitated by prior determination of the appropriate colour vector ranges under different viewing conditions, e.g. sunny, cloudy, using the representative training images. At the classification stage the shape features describing edge orientations around the radially distributed 49 sensor points were extracted from the candidate image regions. Concatenated feature vectors from all sensor points were compared to the vectors stored in the template database. Up to 98% correct match rate was reported for this method in the experiment involving images affected by substantial noise and perspective transformations. However, a long processing time of 0.2–0.7 s per image makes this method unsuitable for real-time applications.

A computationally more efficient algorithm was devised by Bahlmann et al. [127] who utilised the abovementioned Viola and Jones's rejection cascade at the detection stage [111]. Their cascade contained multiple, increasingly specialised boosted classifiers, each combining responses of the Haar wavelet filter masks convolving the input image at all possible locations and scales. To train the cascades, a large number of natural road sign and background images were used. Novelty of this work lies in the joint colour and shape modelling. Unlike in the previous Haar cascade implementations in which the wavelet filters were evaluated on the gray-scale images, Bahlmann et al. additionally parametrised each filter with colour. This extra parameter denoted the colour space to which the input image should be mapped prior to evaluating the filter. Instead of adopting an arbitrary colour appearance model, they considered different possible colour representations and allowed the most discriminative ones to be automatically selected in the boosted training process. Having detected the promising sign candidates, each candidate was tracked using a simple motion model and temporal information propagation. A Bayes Normal classifier was used for recognition, where the class-conditional feature distribution parameters, mean and variance, were estimated in the training process. Prototype implementation of the system introduced by Bahlmann et al. demonstrated very good performance, giving 1.4%, 6.0% and 15% detection, classification, and overall recognition error rate respectively and very few false alarms. However, the target problem considered was relatively simple, especially in terms of the detection, because it involved a single, very specific category of signs.

Another promising, road sign detection algorithm was presented by Barnes and Zelinsky [110]. Their method is based on the concept of regular polygon transform. In the language of statistics, it can be formulated as a problem of finding a posterior probability of a mixture of equiangular polygons of which the entire image is composed. In [128] Barnes et al. proved that given the edge structure of the image, an original 5-dimensional problem with unknown parameters: polygon centroid (c_x, c_y), radius r of the circle it is circumscribed around, orientation γ and the number of sides n, can be efficiently solved using a generic Hough Transform-like voting procedure. This problem becomes even easier in the case of traffic signs as their polygon types, sizes and orientation ranges are *a priori* known. Such a constrained version of the regular polygon detection algorithm was tested on a number of still camera road sign images, giving not more than 5% detection error rate. A real-time system for video-based traffic sign detection, combined with cross-correlation template matching classifier was reported to run as fast as at 20 Hz.

2.4 Evaluation of Previous Studies

Although the issue of traffic sign detection and recognition has been addressed for more than three decades, the number of recognised publications in this area is small compared to the number of those addressing many other computer vision problems like human detection or face recognition. This difference becomes even more striking when the numbers of commercial applications are to be compared. Whereas we can can think about numerous worldwide operating access control systems based on face or fingerprint analysis, visual driver assistance only very recently went beyond the stage of a prototype, e.g. [161, 162]. Moreover, the existing industry-scale applications expose certain limitations, e.g. only restricted categories of traffic signs are handled.

There are various reasons of why the existing approaches to road sign detection and recognition are imperfect. The main limitations seem to concern four different aspects of the problem: representaion and its invariance issues, processing pipeline design, temporal observation modelling, and the classifier design. Let us now discuss these problems in the light of the TSR state of the art.

In a vast majority of the existing road sign detection algorithms a sequential processing is adopted. The scene is typically segmented in order to identify the regions of interest (RoIs) and further analysis is carried out within the detected RoIs, e.g. [28, 34, 49, 50]. In such a sequential processing scheme the primary cues, colour and shape, are extracted and analysed independently. Usually, colour is utilised for preliminary scene segmentation. It is reasonable as road signs contain distinctively bright colours compared to the background, unless the adverse illumination destroys this valuable information. However, as the colour analysis does not take into account other image features like edges or gradient orientations, in certain situations it may yield suboptimal, inaccurate or even incorrect results. For example, a blue circular road sign will usually stand out from the background blue sky in a feature space spanned by joint colour-gradient features, but may appear completely indistinguishable when the colour information is analysed alone. A real-time processing requirement, which is common in a majority of practical applications, imposes further constraints on the design of the sign detectors. In the absence of computationally efficient multi-feature extraction methods, the sequential scheme may appear a decent solution.

Sequential processing often excuses the failures of inaccurate segmentation because it incrementally reduces the region of attention, each new feature type analysed refining the estimation obtained through the previous feature type analysis. This principle underlies for example the approach of Ritter [28] who first used neural network for identification of the characteristic colour patches in the scene, but then ran a connectivity analysis to filter out the patches not being part of the traffic signs. In a similar fashion Piccioli et al. [49] reduced the search area based on the *a priori* known ranges of image coordinates defining the region where the new traffic signs must occur in the scene. To identify the potential sign regions at a finer level, geometrical analysis of the edges extracted in the initially reduced image region was performed. Oftentimes, sequential analysis is done over time for consistency checking of the already detected candidate road signs. For example, in [128] a sign-like shape must appear in the scene for at least several concurrent frames, its radius must not have changed greatly during that time, and it must not move far in the image in order for the sign hypothesis to be accepted.

The main deficiency of the abovementioned algorithms is a lack of joint feature modelling. As analysis of different information cues is arranged as a processing pipeline, at each stage of an analysis a suboptimal image segmentation is obtained. This leads to expensive computation and hence makes a real-time implementation difficult. Besides, this design pattern introduces multiple potential bottlenecks and undesirable dependencies between different parts of the algorithm. This implies, that a failure in capturing a traffic sign region at an early stage of the analysis cannot be recovered from. Similarly, even a small inaccuracy introduced while estimating the position or scale of a traffic sign at the detection stage may result in a completely spoiled classification. An additional common limitation of many sequential road sign detection and recognition approaches is their high parametrisation. For example, the colour-based segmentation methods frequently depend on fixed thresholds which partition the colour space into regions corresponding to the perceivalble named colours, e.g. [67]. These thresholds are however not suitable for colour classification in presence of significant illumination changes, unless some sort of dynamic colour/contrast correction is made. Similarly, the edge-based methods, e.g. [49, 55, 92, 128] highly depend on the quality of the input edge images, which is in turn again affected by lightning conditions and by the distance to the target. This often causes unstability of the detector which in one conditions produces satisfactory results, but in other yields numerous false positives or does not capture signs at all.

The above arguments do not necessarily show that joint, non-parametric treatment of different image features to avoid sequentiality is not possible. As mentioned in section 2.3, two methods of simultaneous colour and shape modelling for traffic sign detection were proposed by Fang et al. [92] (integrated colour and shape neural networks) and Bahlmann et al. [127] (colour-parametrised Haar wavelet features). However these approaches are either very application-specific (Fang et al.) or introduce other problems, e.g. a large amount of training data required (Bahlmann et al.). More importantly, in the approach of Bahlmann et al. the impressive detection performance, i.e. high true positive rate and very low false alarm rate, was possible to achieve only because a narrow subcategory of very similar signs (speed limit signs and "no passing" signs) were targeted. Training a single Haar cascade on a more diverse gamut of signs representing multiple semantic categories leads to significant detection accuracy degradation. In other words, this method is only suitable to handle very specific sign detection problems and could be potentially used in the in-vehicle visual driver assistance if multiple detectors like this were operated simultaneously. Other, more robust methods for integration of multiple different-nature features are available in computer vision, e.g. covariance descriptors [145], but their application to traffic sign detection or recognition has never been attempted.

Lack of temporal, probabilisic description of the target tracking process is an another deficiency of the state-of-the-art traffic sign recognition methods. The most common strategy involves running the detector independently in each frame of the input video within a localised search region around the previously recorded position and scale of a sign. This approach is often sufficiently fast and accurate, as long as a robust but computationally efficient detector is available. However, its accuracy may substantially deteriorate or the track of a sign may be lost when the motion of a vehicle is not smooth, when the appearance of the target significantly changes over time, e.g. through varying illumination or occlusions, or when the tracked sign is subject to geometrical deformations. Note that the latter situation usually takes place when the target is close to the camera, i.e. when it carries the most valuable discrimnative information from the recognition point of view.

In the previous approaches to dynamic road sign recognition the tracking aspect was surprisingly seldom addressed. In most studies only the changes of the basic geometrical sign properties, 2D or 3D position and scale, were modelled, assuming known velocity of the vehicle. This modelling was most often based on a Kalman

Filter (KF) [49, 65, 92]. It is known to provide an optimal posterior state estimation when both the process model and the observation model are strictly linear. Therefore, for highly non-linear vehicle motions, e.g. when the car suddenly accelerates or brakes, a basic KF is no longer an adequate method. Adopting a more robust filtering technique like Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) or Particle Filter (PF), discussed in section 2.2, usually eliminates the nonlinearity problem completely. An example of an observation model linearisation achieved by adopting EKF is given by Piccioli et al. [49]. Surprisingly, in no widely recognised study such filters were used for temporal modelling of the apparent affine deformations of traffic signs. Such modelling would enable accurate pose estimation and pictogram recognition even at small distances from the camera because the frontal view of a distorted sign could be reconstructed at any time point based on the current affine transform parameter estimates. In addition, it should be noted that the existing sign trackers have purely geometrical character. In other words, they do not offer any probabilistic description of the temporal appearance changes on an individual pixel or feature level.

The contemporary traffic sign recognition algorithms could be more successful if they were able to discriminate between heterogenous sign categories and multiple classes more reliably. In practice, the commercial systems are limited to recognising the signs representing only narrow subcategories, e.g. prohibition signs in the case of the *Opel Eye* system [162]. Addressing the road sign classification problem in a more generic way, allowing hierarchical, multi-category sign interpretation invariant to the cross-country pictogram variations, requires a more robust classifier design. Ideally, it should be possible to train such a classifier based on the idealised highway code sign prototypes or relatively few realistic images as these, especially in the case of naturally very rare signs, are often difficult to acquire.

On one hand, construction of a robusts road sign classifier requires a more discriminative feature representation to be constructed. To improve the classifier's capability of distinguishing between very similar sign types, e.g. speed limits or road narrows, such a representation could be learned from the training data via exploitng and emphasising differences between the individual, particularly resembling classes. On the other hand, more robust classifiers may be required than those based on a simple template matching in a uniform feature space and the nearest neighbour rule, e.g. multi-class Support Vector Machines [88] or multi-class AdaBoost [146]. A promising direction was followed by Paclík et al. [144] who postulated representing each sign with a set of similarities to the stored class prototypes, each individually learned according to the *one-vs-all* paradigm. Hierarchical classifier structure was proposed by the same authors in [67]. However, the semantic decomposition of the traffic signs family proposed in this study was arbitrary. Unfortunately, no automatic road sign problem decomposition, e.g. via clustering, can be found in the state-of-the-art literature. Perhaps, a generative approach to recognition could more comprehensively address the semantic relationships between traffic signs, but this direction has also not yet been followed.

Chapter 3

Traffic Sign Detection

Visual object detection in large, potentially cluttered scenes is always a very difficult task. Detection of traffic signs is a good example of such a challenge. A successful detector should be able to efficiently address several application-specific problems, e.g. a large and diverse gamut of signs to be captured, insufficient figure-background contrast of signs that are seen in adverse illumination, or blur caused by the vibrations of the camera. However, in many aspects road sign detection resembles other, more general visual object detection problems. Therefore, when addressing it, the same, common, fundamental questions need to be answered.

- What is the object of interest? Is it unambiguously defined or has variable appearance?
- How to learn the object, if the model is not available? Is it possible to obtain a sufficient amount of training data?
- What makes the target object distinctive? How to extract its discriminative features?
- How to efficiently manage false positives and false negatives of the detector in runtime?
- How to efficiently scan a large region of the input image in search of the target object candidates?

In the case of traffic signs, the choice of the detection strategy largely depends on how broad class of signs is targeted. When a single type of sign is focused on, it is relatively easy to develop a fast and robust detector as the target object is uniquely or nearly uniquely defined. On account of this fact, the detector can operate on a dedicated discriminative feature representation that can be derived from real-life sign images or their highway code prototypes. The task becomes much more complicated when multiple diverse road signs are to be detected, which is the case in most practical applications. Usually, taking the real-time performance requirement into account, the existing TSR systems of this kind cannot offer a general-purpose detection algorithm capable of handling the entire traffic sign diversity in a uniform way. They must strike a balance between the flexibility of the solution and its computational efficiency. Therefore, they usually specialise in detecting only narrow sign categories that exhibit many common appearance characteristics. For example detection on the European prohibition signs is facilitated by the fact that all of them have a circular shape with red rim, white interior, and possibly some black symbols in it.

Apart from ensuring a flexibility of the detector and its high discriminative power in real-life traffic scenarios, several other issues need to be addressed. First, it must be decided how to efficiently scan potentially large regions of the scene with the detector and how to optimally reduce these search regions at no or little cost in the detection rate. Second, if the detector operates through dense image scanning, it is necessary to ensure that no multiple, redundant, i.e. nearly entirely overlapping hypotheses are generated around the true target locations and scales. Third, the invariance issues must be addressed. Ideally, although it is almost always neglected, the road sign detector should be pose-invariant in order to enable robust pictogram recognition even when the target appears distorted in the image. To achieve insensitivity to lightning changes, it should also be able to discriminate a sign from the background even when its colours look pale, e.g. as a result of shade or faded dye on the sign's surface. Other challenges involve dealing with motion blur, occlusions and other temporal aspects of the detection. They will be addressed in more detail in Chapter 4, which is devoted to road sign tracking.

The rest of this chapter is devoted to the sole problem of candidate sign detection in real traffic images which conceptually corresponds to the detection component shown on the upper right-hand side of the diagram in Figure 1.3. In Section 3.1 the previous studies on traffic sign detection are reviewed. Section 3.2 discusses the methods of fast interest region localisation which are practical in the context of video-based TSR. In section 3.3 we investigate the two practically useful approaches to discriminative traffic sign detection, one based on on a Hough Transform-style regular polygon transform and the other based on a trainable boosted classifier cascade. Certain limitations of these detectors are identified. In section 3.4 these limitations are addressed by introducing a *Confidence-Weighted Mean Shift* clustering algorithm for the detector's response refinement. Section 3.5 gives a summary of this chapter.

3.1 Background

In many early approaches to traffic sign detection, the prominent sign candidates were captured in the input image in a two-stage procedure. It involved an initial scene segmentation followed by a more detailed colour and/or shape analysis performed within the established interest regions. This strategy is generally justifiable as an exhaustive search for the target object in the entire image would be impractical and computationally expensive. Example studies where this direction was followed are the ones related to the already mentioned *PROMETHEUS* project. In the study of Ritter [28] the pixel-wise colour segmentation of the incoming image was first performed with a high-order neural network. On output of this step each image was assigned one of the three recognised sign category labels, depending in its dominant colours, or a background class. Based on the colour-segmented image and using available *a priori* knowledge, hypotheses on image regions containing traffic signs were generated. Connectivity analysis of these regions was further conducted in order to discard those where the probability of sign existence was too low. Priese et al. [34] replaced the colour region connectivity analysis with the analysis of the object's boundary, which enabled preliminary shape-based recognition.

Two-step detector design was also utilised in later studies. Piccioli et al. [49] sequentially applied two initial localisation methods. First, the search for the new sign candidates was simply limited to the a priori defined region of an image outside of which the signs are unlikely to occur. The second step resembled those described above. A pixel-wise colour classification in Hue-Saturation-Value space was first performed. Then, the scene was subdivided into 16 × 16 blocks, each classified as 1 or 0 according to whether the number of feature pixels exceeded a certain threshold. The regions of interest (RoIs) were constructed by clustering the blocks labelled as 1. The circular and triangular road sign candidates were then found within each interest region through a geometrical analysis of the edge segments contained in it. Escalera et al. [50] again used the colour information to segment the scene. It was done via fast image filtering in a colour space defined by ratios of the individual RGB channel intensities to the sum of all channels' intensities. Then, similarly to Piccioli et al., they conducted a geometrical analysis of each RoI's content. However, rather than edges, corner point information extracted with a set of appropriate convolution masks was exploited.

In the more recent approaches the abovementioned detection scheme was still often used, for example by Miura et al. [65]. However, the need for a comprehensive, more robust, and less parameter-dependent method quickly became commonly understood. An interesting method for simultaneous detection and recognition of traffic signs utilising Distance Transform (DT) was proposed by Gavrila [55]. The algorithm presented in this work was generic in that it could utilise many different low-level image features in a uniform way. Specifically, features like oriented edge or corner points were extracted from an input image and a DT was computed for each resulting binary feature map. The input image was matched in a coarse-to-fine fashion to the DT images of different-shape road sign templates, grouped by similarity within a tree structure. At each tree node the test image DT was matched to a prototype template DT representing all similar templates at this node, rather than to each template separately. This dramatically sped up the matching process. A colourless approach to road sign detection was also adopted by Paclík et al. [66, 67]. They scanned the input edge map at different scales to enable comparison of the detected geometrical objects' shapes with the template road sign shapes, which provided a preliminary categorisation.

A comprehensive approach to traffic sign detection was presented by Fang et al. [92]. They used two-layer neural networks (NN) to estimate the likelihood of road sign presence at each pixel location. Both networks accepted RGB pixel values of the original image on input. However, while one network was used to extract colour hue values from the input image, the other network extracted edges. Connections between the neurons were designed in such a way that the excitement of the output neurons was high only when a signal from the input layer was coming from pixels of sign-specific colours and satisfying appropriate geometric relationships defined by the model signs. The soft responses of both networks for each pixel were combined in a fuzzy way, yielding an integrated sign-membership map. In this map the centroids of the likely signs were identified at the locations associated with the above-threshold NN responses. Barnes and Zelinsky [110, 128] proposed an another promising approach to traffic sign detection based on the so-called *regular polygon transform*. This method is based on an assumption that the observed edge structure in the image is drawn from a Gaussian mixture of equiangular polygons of which the entire image is composed. An efficient Hough Transform-style voting procedure was proposed to non-parametrically approximate the regular polygon likelihood. It was shown suitable for detecting many types of popular, regularly-shaped symbolic road signs: circles, equiangular triangles, squares and octagons. However, no method of combining this promising approach with colour information was proposed.

Among the most recent approaches to traffic sign detection, the study of Bahlmann et al. [127] deserves particular attention. They adopted an automatic, parameter-free object detection technique based on a trainable, attentive cascade of boosted classifiers known from many previous successful applications, e.g. face detection [111], pedestrian detection [99], or hand tracking [131]. In this approach the sign detector was learned from a large number of training target and background images. In each layer of the cascade the AdaBoost algorithm [60] was used to determine some number of Haar wavelet features that the most accurately discriminated the signs from the background. This number, and the number of cascade layers were determined in the course of the training process, based on the desired cascade performance characteristic given on input. It was expressed in terms of the maximum acceptable detection rate reduction per layer added, minimum acceptable false positive rate reduction per layer added, and the maximum overall cascade's false positive rate. The original contribution of Bahlmann and his colleagues was a joint colour and shape modelling. It was achieved by parametrising each rectangular Haar wavelet filter with colour, along with the filter type, location and scale. This extra parameter denoted an actual colour representation of the underlying image over which a given filter was to be evaluated. This approach was reported to yield very good detection results and low false alarm rate in the experiments involving real-life traffic video. However, it seems that this exceptional performance was only possible to achieve because a very narrow category of traffic signs exhibiting similar appearance characteristics was considered by the authors.

The major limitation of the early traffic sign detection approaches was their high parametrisation and the lack of theoretical foundations. They often utilised application-specific heuristics rather than solid, analytically justified algorithms. Scene segmentation based on a manual colour space partitioning is sensitive to illumination changes. On the other hand, learning the sign-specific colour value ranges from the natural images is often difficult. It is because sufficient statistics can be constructed from a large number of such images and they are often difficult to acquire. All edge-based detection methods are also prone to failure in presence of substantial illumination or contrast changes. As they typically depend on fixed thresholds, e.g. threshold in the vote space in the case of Hough-style detection [128], such methods cannot generalise well to the other lightning conditions than those present while capturing the images from which they were trained. As a result, an edge-based sign detector may work satisfactorily in a typical, relatively constant illumination, but miss many true signs that appear in shade, or yield numerous false alarms in particularly well-lit scenes. Adaptive gradient thresholding dependent on the global scene contrast statistics could help resolve this problem. However, such techniques seem to have not been used in the previous studies on TSR.

Several interesting directions were followed in the more recent studies. First of all, attempts were made to jointly model the shape and colour cues of road signs [92, 127]. However, in the neural networks of Fang et al. [92] these two pieces of discriminative visual information were still extracted independently. Only at the feature fusion stage were they integrated. From this perspective the approach of Bahlmann et al. [127] is more advanced as the joint shape and colour modelling was used as early as in the feature description step. It is yet questionable whether their cascaded detector can offer stable performance in all kinds of traffic situations, especially in high clutter. Because this detector operates via dense scanning of the input image, multiple overlapping candidate location-scale hypotheses are likely to be obtained in the vicinity of the ground-truth sign locations. It was not mentioned how to eliminate such redundancy. Besides, dense image scanning is computationally expensive. The reported 10 fps runtime processing puts the detector proposed by Bahlmann and his colleagues, along with many other previous algorithms, below the expectations to be met in real-life visual driver assistance systems.

An another deficiency of the existing road sign detectors is their pose dependency. In practice, it is too often assumed that the signs appear undistorted, i.e. ideally front-looking, and hence only the in-plane rotation and scale changes are modelled. This assumption is generally valid provided that the sign is detected at a large distance from the camera and is not physically dislocated, e.g. tilted. However, an explicit modelling of the 3D orientation of the target by the detector would be a large step forward in TSR. Certain attempts to model the geometrical sign's distortion in the tracking step were made recently, e.g. in the work of Escalera et al. [120]. However, even in this case the abovementioned assumption of the initial lack of distortion was kept. The other challenging problem that has not yet been properly addressed in the TSR literature is related to occlusions. In the threshold-based methods, e.g. [128, 127], whether or not an occluded sign is detected depends on the strength of the image features that remain visible outside of an occluded part. In a video context, the occlusions lasting for more than one frame are resolved by the tracker, as long as it exists. A sparse feature representation of traffic signs, e.g. the one based on keypoints, could probably facilitate partially detection of the occluded signs. However, keypoint extraction proves to be unsuitable for relatively small, sparse-colour man-made objects.

3.2 Interest Region Localisation

In all practical applications of TSR the *a priori* position and scale of traffic signs in the scene are unknown. It is the detector's responsibility to capture this information. However, compared to the frequency of occurrence of a sign in the incoming video and the size of regions the traffic signs normally occupy in the image captured by a typical camera, the size of the entire scene to be explored is big. This necessitates devising an independent preliminary detection algorithm which aim is only to discard those regions of the input image where the probability of sign's presence is avoidably small. Rationale for using such an algorithm is that discarding non-target regions is computationally and algorithmically less expensive than capturing the instances of the interest object. Therefore, a practical approach, which we also adopt in this work, is to use a dedicated technique for reducing the local search region for the actual object detector. Then, the detector is run only in this region. It is further referred to as a region of interest (RoI).

To detect the prominent traffic sign candidates in the scene, the colour and shape is known to provide the most informative cues to be exploited. Therefore, in the RoI extraction step these pieces of information are also often used. In the previous studies usually the regions of sign-specific colours were first located in the image. It was followed by further analysis of image gradients or edges and their spatial relationships in order to further filter out the regions unlikely to be containing the traffic signs. If certain assumption on the vehicle's motion model can be made, which implies that the traffic signs can only appear in a specific portion of the scene, an instant removal of the remaining part from further analysis naturally becomes the first step in RoI extraction.

There seems to be no sufficiently robust algorithmic solution for rapid interest region localisation. A potentially cluttered traffic scene may contain many objects resembling road signs. These objects may appear in spatially scattered locations of the image and be of different scales, which makes the problem even more challenging. On one hand the RoI extractor must filter out as many false signs as possible. On the other, it must be conservative enough not to miss any true signs that are sufficiently well visible to consider them detectable. It is difficult to maintain a consistent tradeoff between these two desirable properties when the content of the scene dynamically changes, especially if this tradeoff is controlled parametrically, e.g. by thresholds on the intensity, colour or strength of the gradients. A weak design of the RoI extraction algorithm often leads to missing signs that should not be missed, or causes explosion of false alarms which in turn increase computation and degrade the system's capability of frame-rate operation.

In Section 3.2.2 we describe a fast method of finding the interest regions in the image, with application to coarse traffic sign localisation. It is based on a quad-tree technique which ensures fast recursive image processing. Prior to discussing the details of this method (Section 3.2.1), an outline of the integral image technique [111] is given and a suitable low-level image feature to be represented in an integral form is presented. This integral map of image features is used by our quad-tree focus operator evaluated experimentally in Section 3.2.3.

3.2.1 Low-level Features and Integral Image

Colour and gradient information are these low-level cues that can be efficiently exploited to quickly locate the regions of interest in the image. They jointly fully encode the appearance of traffic signs and, unlike higher-level cues, such as shape, are easy to extract. Integral image [111] is an intermediate representation of the image which encodes its pixel-based low-level features in an extremely useful way. Namely, upon construction of an integral map of the image features, the total "amount" of feature contained in any rectangular region of the image can be computed without pixel-wise scanning of the contents of this region. Combination of the appropriate low-level image features and the integral image technique provides the target representation on which our RoI extractor works.

In many road traffic scenes solely the colour of signs stands out sufficiently to spot them. However, in certain situations there may be relatively large fragments of the scene where the dominant colours resemble those used to cover the surface of traffic signs. An example may be a vivid blue sky or a large, brightly-coloured building. Therefore, we consider the colour gradients, instead of colour alone, as an even more informative piece of visual information for RoI extraction. Because the targeted signs have either red, blue or yellow rim, we want to amplify these particular colours and suppress any other colour in each input image. Then, using an integral image, the regions where the density of sign-specific colour gradients is above threshold can be rapidly extracted. The following set of linear colour transformations for each RGB pixel $\mathbf{x}$ is proposed to filter the image in order to achieve the abovementioned sign-specific colour amplification effect:

$$\begin{array}{l} f_R(\mathbf{x}) = \max(0, \min((x_R - x_G)/s, (x_R - x_B)/s)) \\ f_B(\mathbf{x}) = \max(0, \min((x_B - x_R)/s, (x_B - x_G)/s)) \\ f_Y(\mathbf{x}) = \max(0, \min((x_R - x_B)/s, (x_G - x_B)/s)) \end{array} \quad . \tag{3.1}$$

Transforms defined in (3.1) do not involve colour space conversion and are hence fast. They effectively extract the red, blue, and yellow image fragments. The effect they have on the example road traffic images is shown in Figure 3.1. In each resultant image the gradient magnitude for each pixel (i, j) is calculated using an approximative formula:

$$\bar{g}(i,j) = |g_x(i,j)| + |g_y(i,j)| \ , \tag{3.2}$$

where g_x, g_y denote the directional gradients. The above gradient magnitude computation method is adequate to our problem as at the RoI localisation step the exact pixel-wise magnitude values are not important and with equation (3.2) expensive multiplications and square root computations required by an Euclidean norm can be avoided.

For each colour-specific gradient magnitude map, G, an integral image is constructed. At each location (i, j) it contains the sum of the pixels above and to the left of (i, j), inclusive, as shown in Figure 3.2a:

$$I_G(i,j) = \sum\ _{y \le j}^{x \le i} G(x,y) \ . \tag{3.3}$$

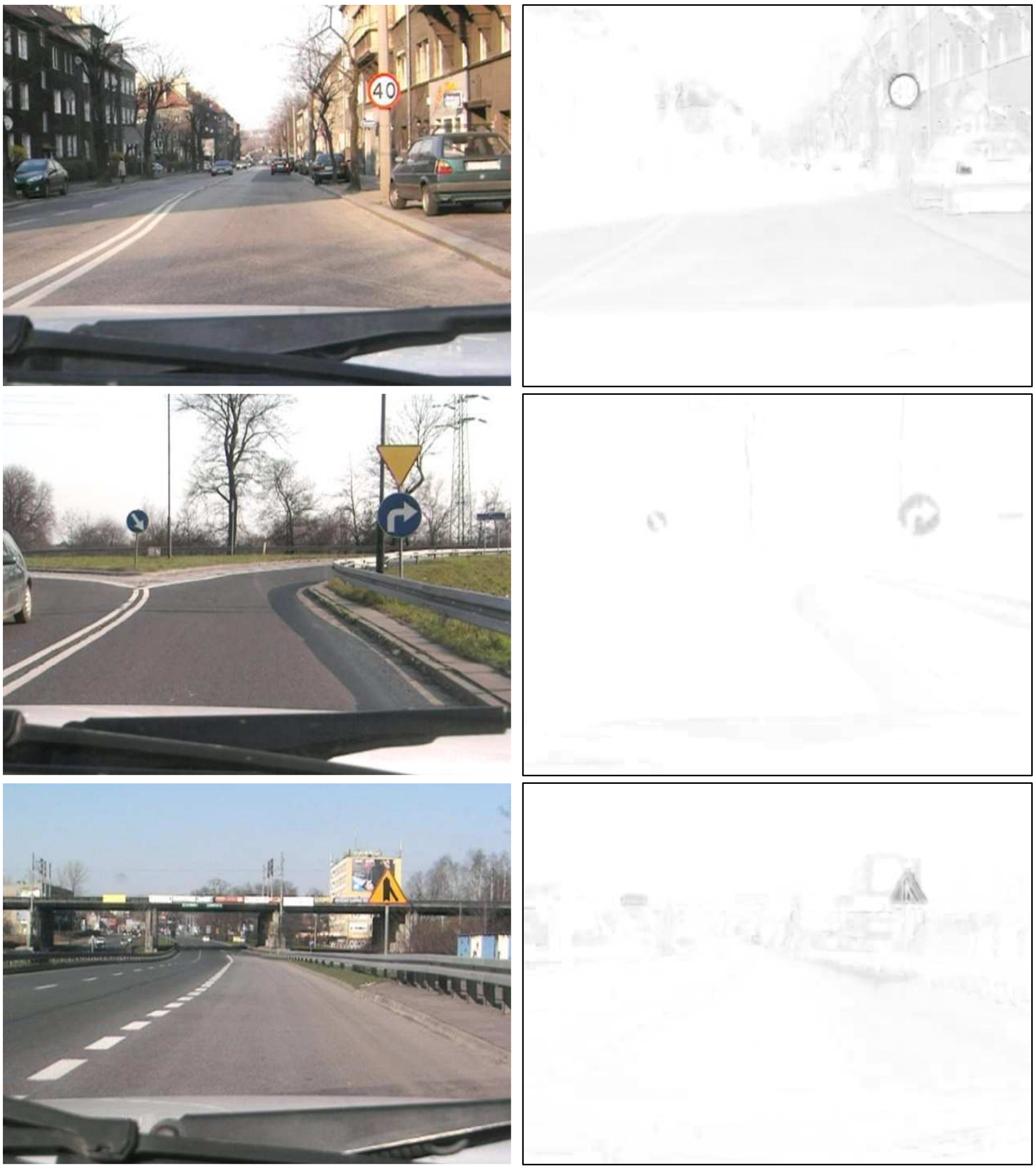


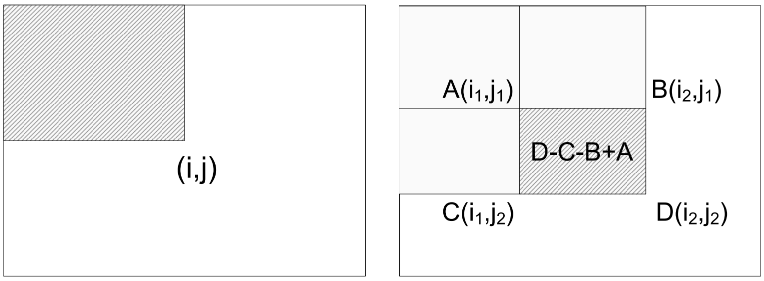
Figure 3.1: The effect of filtering the original RGB images (left) using the transforms defined in (1). Red, blue and yellow colour enhancement respectively are shown in the right column, from top to bottom. This figure is best viewed in colour.

The integral image can be computed in one pass over the original gradient magnitude image using the following recursive formulas:

$$\begin{aligned} S(i,j) &= S(i,j-1) + G(i,j) \\ I_G(i,j) &= I_G(i-1,j) + S(i,j) \end{aligned} \quad , \tag{3.4}$$

where $S(x,-1) = 0$ and $I(-1,y) = 0$. Once the integral gradient magnitude maps have been constructed, the cumulative colour-specific gradient contained in any rectangular region of the image can be computed using only four array referencing operations and three additions/subtractions, as shown in Figure 3.2b:

$$I_G(i_1,j_1,i_2,j_2) = I_G(i_2,j_2) - I_G(i_1,j_2) - I_G(i_2,j_1) + I_G(i_1,j_1) \ . \tag{3.5}$$



(a) (b)

Figure 3.2: Integral image: (a) the value at point (i,j) is the sum of all values at pixels above and to the left of (i,j), inclusive; (b) computation of the total amount of feature contained within a specified rectangular region.

3.2.2 Quad-tree Focus Operator

The proposed method for fast RoI extraction aims at locating the regions where the density of the red, blue and yellow colour gradients is high as this might be an indication of the presence of traffic signs. On the other hand, the chance of missing the true signs in the regions where this density is close to zero, is minimal. On input of our attentive operator the three integral images introduced in Section 3.2.1 are passed. In addition, two other quantities have to be defined: a minimum size of local regions to be considered and a minimum "amount" of feature contained in a region to consider it relevant for further analysis. A general idea is as follows. First, a region corresponding to the entire image is checked against the total colour gradient contained using a maximum of the values picked from all three colour-specific integral images. As it is typically far above the predefined threshold, the image is subdivided into four quarters and each quarter is recursively processed in the same way. The process is stopped either when the current input region contains less cumulative gradient than the predefined threshold or upon reaching the minimum region size. The above-threshold lowest-level regions are clustered and the ultimate RoIs are constructed as bounding rectangles of the found clusters. This procedure is formally described in Algorithm 1 and its top-down part is illustrated in Figure 3.3.

Algorithm 1 Quad-tree RoI extraction.

```
input: image $I_{W \times H}$, minimum “amount” of feature contained in a region to be considered relevant, $t_{min}$, maximum RoI diameter, $d_{max}$, minimum region size, $s_{min}$
output: set of RoIs, $S$
build a feature map $\mathbf{V}_I$
build an integral feature map $\mathbf{\Sigma}_I$ from $\mathbf{V}_I$
initialise an empty set of relevant smallest-scale regions $X = \emptyset$
call $ProcessRegion(R(1, 1, W, H), \mathbf{\Sigma}_I, t_{min}, s_{min}, X)$
call $ClusterRegions(X, d_{max}, S)$
inflate each RoI in $S$ by $d_{max}/2$ on each side
```

Algorithm 2 Procedure *ProcessRegion*.

```
input: region $R_{w \times h}$, integral feature map, $\mathbf{\Sigma}_I$, minimum “amount” of feature contained in a region to be considered relevant, $t_{min}$, minimum region size, $s_{min}$, a set of relevant smallest-scale regions, $X$
compute the amount of feature in $R$
if $\min\{w, h\} \geq s_{min}$ then
  if $\mathbf{\Sigma}_I(R) > t_{min}$ then
    set $w = w/2$, $h = h/2$
    for each quarter $Q_j$ of $R$ do
      call $ProcessRegion(Q_j, \mathbf{\Sigma}_I, t_{min}, s_{min}, X)$
    end for
  end if
else
  add $R$ to $X$
end if
```

Algorithm 3 Procedure *ClusterRegions*.

```
input: a set of relevant smallest-scale regions, $X$, maximum RoI diameter, $d_{max}$, an empty set of RoIs, $S$
randomly choose region $R_1$ from $X$
initialise new cluster $C_1 = \{R_1\}$ and add it to $S$
for each $R_i \in X$, $i = 2, \ldots, |X|$ do
  find the region $R_j$ and its cluster $C_j$ which is closest to $R_i$ among all regions in all clusters from $S$
  if $d(R_i, R_j) < d_{min}$ then
    put $R_i$ in $C_j$
  else
    initialise new cluster $C_i = \{R_i\}$ and add it to $S$
  end if
end for
```

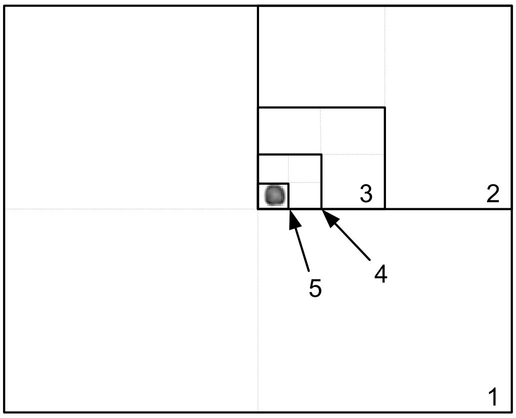


Figure 3.3: Top-down analysis of the image using the proposed quad-tree focus operator. The consecutive numbers correspond to the order of quarters being processed.

The threshold t_{min} is fixed and does not depend on the actual scale of the input region analysed in the *ProcessRegion* procedure. It must be set to an appropriately low value that can be used to reliably discriminate between the potentially relevant and irrelevant fragments of the scene at the smallest considered scale. It means that the goal of the top-down part of the algorithm is only rapid elimination of these regions of the image that do not contain the interest colours at all, or contain at least one of them but are nearly uniform, i.e. do not contain contrasting patches. For instance, they include fragments of sky or asphalt.

The following bottom-up part is intended to group the remaining relevant smallest-scale regions $R_i \in X$ into clusters which might correspond to multiple road sign candidates. The clustering scheme used is based on a simple nearest neighbour rule applied in one pass to X. Specifically, the first, randomly selected region R_1 from X is put into the initial cluster. Then, for each remaining region R_i the nearest region R_j among those assigned to any of the already existing clusters is found. If the distance between R_i and R_j is smaller than the predefined maximum distance, region R_i is added to the cluster in which R_j was found. Otherwise, a new cluster is initialised with R_i. The abovementioned cut-off distance corresponds to the maximum span of the cluster elements and is set to the apparent diameter of the largest signs the detector is able to capture. The ultimate regions of interest are constructed from the bounding rectangles of each cluster, inflated on each side by a fixed margin so as to eliminate the risk of accidentally capturing a sign lying on the edge of a cluster. The width of this margin corresponds to the maximum considered sign's radius. Verification of the hypotheses on sign' presence in each RoI is done by the actual traffic sign detector at the next stage of the processing.

3.2.3 Evaluation

The proposed focus operator is not expected to yield only true road sign regions as this is not possible based on such simple evidence as cumulative colour gradient. In cluttered urban scenes this algorithm is likely to produce multiple false RoIs or single overly large RoIs. However, at this stage of the processing the number of false alarms cannot be reduced at the cost of not detecting the regions containing true road signs. Therefore, it is the most essential to capture as many true positives as possible for given input data, even when uninformative fragments of the image have to be analysed too. In practice, if the threshold t_{min} is set correctly, which is done based on a number of training images with ground truth positions and scales of signs available, the algorithm captures a vast majority of signs in the incoming video, but the reduction of the computation involved is still huge.

In order to quantify the capability of our focus operator of localising road signs, we have performed an experiment involving realistic traffic images captured with a wide-angle in-vehicle camera in crowded street scenes. The total of 70 high-resolution images depicting 100 road signs were used, each of dimensions 1920×1088 pixels. Several examples are shown in Figure 3.4. In order to find the optimal threshold t_{min}, our algorithm was run for the increasing value of this threshold on these 70 images and the percentage of true signs covered by the produced RoIs was maximised. In Figure 3.5 we have illustrated this quantity, as well as the percentage of the total image area covered by the found RoIs, as functions of t_{min}.

The obtained results demonstrate the usefulness of our search region reduction algorithm. While it captures a vast majority of traffic signs emerging in the scene, the average area of the image to be analysed is only a small fraction of the entire image's area. The signs not covered by the found interest regions in this experiment were significantly shaded. Therefore, their distinctive colours appeared too pale and the resulting low figure-background contrast was insufficient.

3.3 Discriminative Traffic Sign Detectors

Detection of traffic signs, even in a previously reduced search region, is a challenging task. In a practical application the detector must be able to scan a potentially large portion of an input image in real time. Therefore, a reasonable balance between the detector's accuracy and its computational complexity must be struck. Probably the most accurate localisation of the target would be achieved by convolving the appropriately filtered RGB image with masks encoding the coloured sign-like shapes: circles, equilateral triangles, squares and octagons. However, this solution would be computationally intractable, even if a single apparent scale of signs was considered and no perspective distortion was assumed.

A bottom-up sign detection approach dominated the previous studies in TSR. It involved detection of the lowest-level image features like colour, edges, or gradient orientations. From this evidence the hypotheses on signs' presence were constructed and incrementally verified, based on the *a priori* known geometrical relationships between the signs' parts. The main limitation of a vast majority of these methods is that they are not generalisable, they lack convincing theoretical foundations, and are usually highly parametrised. To certain extent these limitations cannot be avoided. Very strict, industry-level performance requirements that the contemporary visual driver assistance systems must meet enforce application-specific solutions and necessitate fine-tuning.

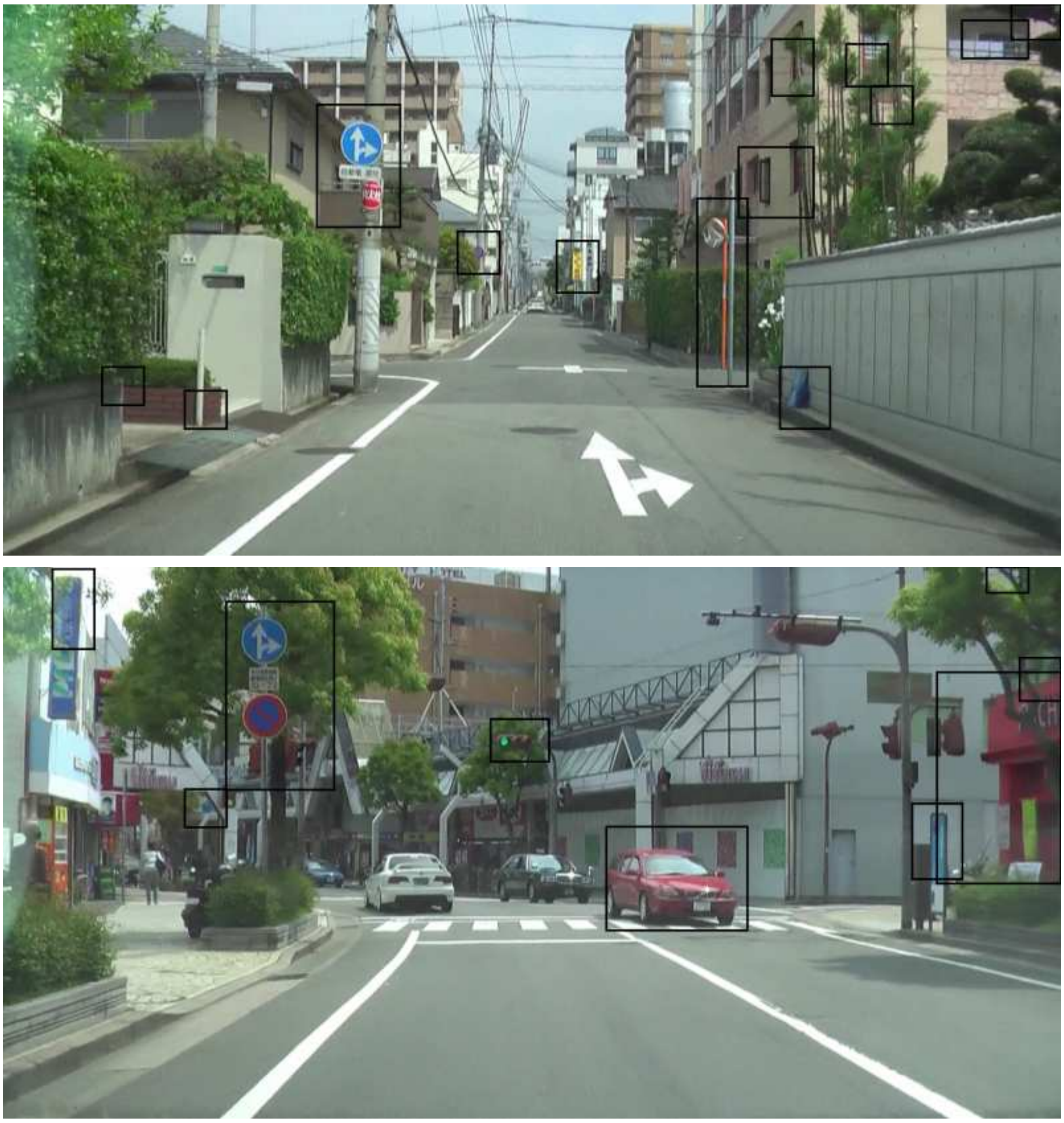
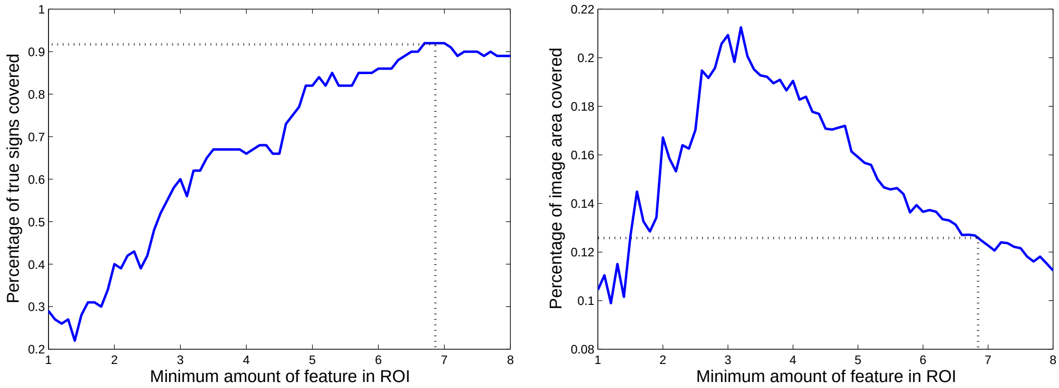


Figure 3.4: Example urban traffic images captured with a wide-angle camera mounted on board of a moving vehicle. On each image the regions of interest found by the proposed quad-tree focus operator are marked. The threshold t_{min} was set to the optimal value found in the training stage.



(a) (b)

Figure 3.5: Determination of threshold t_{min}: (a) the optimal value, $t^{\star}_{min} \approx 6.9$ corresponds to the maximum percentage of true signs covered by the found RoIs as a function of t_{min}, (b) relationship between the percentage of image area covered and the same values of t_{min} as in the left plot. $t^{\star}_{min}$ is marked with dotted lines in both plots.

A general concept underlying our approach to traffic sign detection is based on the assumption that no single object detector can simultaneously offer perfect accuracy, yield no false positives, and operate in real time with a noisy traffic image on input. Therefore, the primary focus is put on the postprocessing stage where the responses of even a relatively weak detector, that is used to scan an input image region, can be refined to produce a more accurate result. The entire Section 3.4 is devoted to this issue. In the following sections two object detectors are described that exhibit desirable characteristics in the context of video-based TSR: are relatively accurate and computationally inexpensive. The first detector is based on a generalisation of circular Hough Transform [3, 9] aimed at capturing the coloured regular polygons. It is discussed in Section 3.3.1. The second detector implements a fast attentive rejection cascade of Viola and Jones [111] for different low-level image features. We discuss this detection method in Section 3.3.2. In Section 3.3.3 experimental evaluation of these detectors using different datasets is given and their common limitations are pointed out.

3.3.1 Hough Transform and Its Extensions

One distinctive property of the most popular traffic signs is that their shapes are regular (see Appendix B). Although the bright colours of signs seem to draw the primary attention of human drivers, in the automatic sign detection system this radial symmetry might be a critical cue to be exploited. The hereby proposed traffic sign detector is based on an extension of circular Hough Transform proposed by Barnes et al. [128] and aimed at capturing instances of regular polygons in the image. As the original algorithm takes no account of the colour information at all, we combine it with the appropriate preprocessing of an input image based on the colour filters defined in equation (3.1). Further extension to this algorithm will be discussed in Section 4.3 where we implement a sign contour's tracker. The detailed probabilistic derivation of the detector of Barnes and his colleagues is given in [128]. Below, only the basics of its implementation are outlined.

The algorithm of Barnes et al. operates on a gradient map of the original grayscale image. It is thresholded with respect to gradient magnitude so as to discard the uninformative image pixels. Each remaining gradient element votes for a potential regular polygon's centroid a distance r away along the direction of the gradient vector, where r is the radius of a circle the targeted regular polygon is circumscribed around [1]. The number of votes cast depends on the predefined gradient orientation tolerance, k, and on the actual type of polygons being searched for. This is explained below. The resulting vote space can be treated as a map of likelihood of the presence of particular regular polygons in the original image.

Due to image sampling there is some residual uncertainty of directional gradient information in each image pixel (x, y). Therefore, instead of a single gradient direction, multiple directions $\theta_j(x, y) = \theta(x, y) \pm k^\circ$ around the measured orientation $\theta(x, y)$ are analysed for voting. In addition, in order to correctly estimate the likelihood of shapes other than circles, multiple votes must be cast along the line perpendicular to each checked gradient direction, and at distance r from the line extending the edge at (x, y). Each of these votes accounts for a possible centre of an equiangular polygon the analysed pixel could be part of. It is illustrated in Figure 3.6. Formally, the length of a vote line for a n-side regular polygon and given r is equal to:

$$w = 2\text{round}\left(r \tan \frac{\pi}{n}\right) \quad . \tag{3.6}$$

In addition to $w/2$ positive votes cast at distance r from pixel (x, y) on both sides of the central affected pixel, another $w/2$ negative votes are cast to suppress the responses generated by straight line segments which are too long to belong to the targeted regular polygon. These votes are marked black in Figure 3.6.

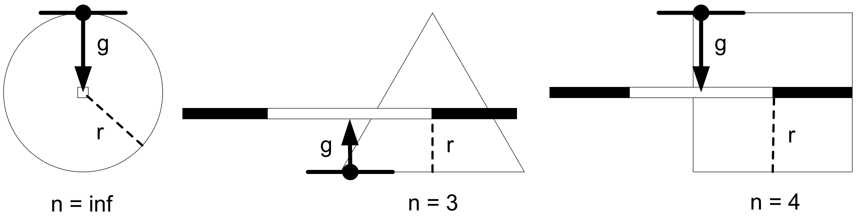


Figure 3.6: Voting lines associated with a given gradient element, depending on the type of regular polygon being targeted.

If there is a road sign in the input image, its apparent centroid will receive significantly more votes than any other image point, as shown in Figure 3.7. Therefore, filtering the vote space using a predefined threshold can be used to capture the position, the scale and the type of the regular polygon detected. However, up to that point the orientation of a sign still cannot be resolved. To do this, an another

[1] A circle can be regarded as a regular polygon with an infinite number of sides.

interesting property of regular polygons is exploited. Namely, note that if $\theta(x_i, y_i)$ is an angle between the gradient vector at point (x_i, y_i) lying on a regular polygon's boundary and the positive part of the x axis, i.e. $\theta(x_i, y_i) = \angle\vec{\mathbf{g}}(x_i, y_i)$, then for all contour points (x_i, y_i) multiplication of $\theta(x_i, y_i)$ by n yields the same angle, when folded by 360° or its multiplicity.

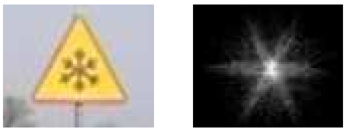


Figure 3.7: Output of a regular polygon detector with $n = 3$ for an example image depicting a cautionary traffic sign.

To achieve the rotational invariance of the regular polygon transform, an n-angle vector field $\vec{\mathbf{v}}$ is defined, where $\|\vec{\mathbf{v}}(x_i, y_i)\| = \|\vec{\mathbf{g}}(x_i, y_i)\|$ and $\angle\vec{\mathbf{v}}(x_i, y_i) = n\theta(x_i, y_i)$. Now, instead of a unit vote cast from each gradient element of the image at point $P(x_i, y_i)$ to the associated possible polygon centroid's locations (c_x, c_y), its n-angle gradient vector is assembled. Then, a vector addition is done to determine the likelihood of a regular polygon's centroid at each (c_x, c_y), and its orientation:

$$\begin{aligned} l(c_x, c_y) &= \left\|\vec{\mathbf{V}}(c_x, c_y)\right\| \\ \theta(c_x, c_y) &= \angle\vec{\mathbf{V}}(c_x, c_y) \end{aligned} \quad , \tag{3.7}$$

where:

$$\vec{\mathbf{V}}(c_x, c_y) = \sum_{P(x_i, y_i)} \vec{\mathbf{v}}(x_i, y_i) \ . \tag{3.8}$$

In order to incorporate the colour information into the regular polygon detector, each found interest region is first filtered using the transformations defined in equation (3.1). Afterwards, for each filtered region image the colour-specific edge maps are extracted by a simple filter which for a given pixel picks the highest difference among the pairs of neighbouring pixels that could be used to form a straight line through the middle pixel being tested. Obtained values are thresholded and only in the resulting edge pixels values of directional and magnitude gradient are calculated. This technique is adequate to our problem as it enables a quick extraction of edges and avoids expensive computation of the whole gradient magnitude map which, with the exception of the sparse edge pixels, is of no use to the detector. Also, the produced edges are thick, which provides more support for Hough voting compared to the single-pixel edges obtained by a Canny algorithm. In the preliminary experiments it has been shown to increase the accuracy of the regular polygon detector.

For a given pair of gradient and edge images associated with colour c, appropriate instances of the regular polygon detector are run to yield a set of possible sign shapes. For instance, for a "blue pair" a circular shape detector is triggered to search for the blue instruction signs, e.g. "turn left" or "turn right", and a square detector is run to detect potential square information signs, e.g. "pedestrian crossing" or "parking place". If two or more prominent overlapping shapes are detected in the same RoI by different detector instances, only the one receiving the maximum number of votes is retained. As each found candidate has known shape and border colour c, the detector serves as a pre-classifier reducing the number of possible templates to analyse at the later stage of the processing to the ones contained in the respective category. The optimal thresholds in the vote space are determined individually for each category at the training stage, based on a set of realistic traffic sign images.

3.3.2 Attentive Classifier Cascade

In this section an another approach to traffic sign detection is presented. It utilises a trainable attentive cascade of boosted classifiers introduced by Viola and Jones [111]. Unlike the Hough-based detector introduced in Section 3.3.1, this technique does not require any prior knowledge about the appearance of the target. Therefore, theoretically it can be used to learn any object and hence solve much more generic object detection problems than the one we currently focus on in this work.

An attentive classifier cascade revolves around an idea of building a multi-layer classifier in which at each new layer the layer-specific binary classifier is trained in a supervised way using all available true positive images and only these negative (background) images that were misclassified in the previous layer. This way the cascade is arranged such that in runtime the most top-level classifier can quickly reject a majority of irrelevant parts of the scene, leaving the more ambiguous regions to process by the classifier in the next layer. This recursive delegation process is further continued for the increasingly "hard" regions and only the regions successfully passing the last layer are retained.

Let us discuss our cascade learning process in more detail. It is presented in Algorithms 4 and 5. The critical parameters of the cascade are provided on input by the user. These include f, the maximum acceptable false positive rate per layer, d, the minimum acceptable detection rate per layer, and the target overall false positive rate, F_{target}. The first two parameters are defined as percentages of the respective quantities obtained at the previous cascade layer. In other words, f expresses a minimum desirable reduction in the cascade's false positive rate to be achieved upon adding a new layer to it. Similarly, $1 - d$ expresses a maximum acceptable reduction in the cascade's true positive rate when a new layer is added. The third parameter, F_{target} controls when the training stops. Note that in our application the cascade is used to scan an input image region pixel by pixel. Therefore, the overall false positive rate must be very small to prevent the cascade from detecting a large number of distinctive non-sign objects. We set the value of this parameter to $F_{target} = 0.1\%$. The other two parameters are set as follows: $f = 50\%$, $d = 99\%$.

To train the classifier in each layer of the cascade, we follow the similar approach as the one adopted by Viola and Jones [111]. Specifically, the AdaBoost feature selection scheme is used (Algorithm 5), where each weak classifier in a boosting combination is associated with a single scalar-valued image descriptor. The trained

Algorithm 4 Attentive cascade learning.

```
input: maximum acceptable false positive rate per layer, $f$, minimum acceptable detection rate
    per layer, $d$, target overall false positive rate, $F_{target}$, a set $P$ of $N_P$ positive training examples,
    a number defining the proportion of negative examples to positive examples in the training
    dataset, $\alpha$, a validation set $V$ containing both positive and negative examples, a set $I_N$ of
    images not containing the traffic signs, a set of $k$ image features $\Phi = \{\phi_j : j = 1, \ldots, k\}$
output: a cascaded road sign detector $C$
set $C = []$, $F_0 = 1.0$, $D_0 = 1.0$, $i = 1$
populate the negative example set $N_1$ with $\alpha N_P$ background regions cropped at randomly
    selected locations from randomly chosen images in $I_N$
while $F_i > F_{target}$ do
  set $n_i = 0$, $F_i = F_{i-1}$
  while $F_i > fF_{i-1}$ do
    set $n_i = n_i + 1$
    call $h_i = TrainLayerClassifier(P \cup N_i, \Phi, n_i)$
    add the trained classifier to the cascade: $C = C + h_i$
    evaluate the cascade on a validation set $V$ to determine $F_i$ and $D_i$
    decrease the threshold of classifier $h_i$ until the cascade satisfies $D_i \geq dD_{i-1}$
  end while
  if $F_i > F_{target}$ then
    generate random negative examples from $I_N$ and put each example misclassified by the
        cascade in $N_{i+1}$ until $\alpha N_P$ examples are generated
  end if
  set $i = i + 1$
end while
```

weak classifier h_j labels each input image $\mathbf{x}$ by comparing the value of the associated feature ϕ_j to the learned threshold t_j:

$$h_j(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad p_j\phi_j(\mathbf{x}) < p_j t_j \\ -1 & \text{otherwise} \end{cases}, \tag{3.9}$$

where p_j is a sign determining the direction of the inequality. In the original work of Viola and Jones four types of Haar wavelet filters were used to populate the pool of features on input of the AdaBoost. However, this pool can be extended by considering any scalar-valued features. This kind of features may be constructed based on the differences between spatially separated image elements or between the corresponding elements of two different images. Alternatively, vector-valued features can be incorporated into the AdaBoost framework. However, this also necessitates change of a weak classifier so that the discriminant function underlying it can cope with multidimensional data.

For traffic sign detection a set of scalar-valued rectangular filters (RF) shown in Figure 3.8 are considered. They extend the basic Haar wavelets and, apart from horizontal and vertical patterns, also capture certain oblique structures of the underlying image. Each rectangular filter $\phi_j(x, y, w, h, c)$ is parametrised by the x and y coordinates of the top-left corner (reference point), width w and height h of each rectangular component. The last parameter, c is colour. It determines the type of colour transformation to be applied to the input image before the descriptor is evaluated. Incorporating the colour information this way allows AdaBoost to automatically select the colour representation of the most informative local image

Algorithm 5 Function *TrainLayerClassifier.*

```
input: a set of $N = N_P + N_N$ training examples $D = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, N\}$, where $y_i = -1$ for
    negative examples, $y_i = 1$ for positive examples, and $N_P$ is the number of positive examples
    and $N_N$ is the number of negative examples, set of $k$ image features, $\Phi = \{\phi_j : j = 1, \ldots, k\}$,
    the number of image features to incorporate in the classifier, $n$
output: a boosted classifier $H$
initialise example weights $w_{1,i} = \frac{1}{2N_P}$ or $w_{1,i} = \frac{1}{2N_N}$ for $y_i = 1$ and $y_i = -1$ respectively
for $t = 1, \ldots, T$ do
    normalise example weights
        $$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{N} w_{t,j}}$$
    for $j = 1, \ldots, k$ do
        train a weak classifier $h_j$ based on feature $\phi_j$
        determine the error rate of $h_j$:
            $$e_j = \sum_{i=1}^{N} w_{t,i} |h_j(\mathbf{x}_i) - y_i|$$
    end for
    choose the classifier, $h_t$ with the lowest error rate
    update example weights:
        $$w_{t+1,i} = w_{t,i} \beta_t^{1-r_i} \quad ,$$
    where $\beta_t = \frac{e_t}{1-e_t}$ and $r_i = 0$ if $h_t$ classifies example $\mathbf{x}_i$ correctly, $r_i = 1$ otherwise
end for
assemble strong classifier:
    $$H(\mathbf{x}) = sgn \left[ \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right] \quad ,$$
where $\alpha_t = log(\frac{1}{\beta_t})$
```

features that will constitute the final detector. The value of each RF filter is determined by subtracting the sum of pixels which lie within the white rectangles R_w from the sum of pixels lying within the black rectangles R_b. Formally:

$$\phi_{x,y,w,h,c}(\mathbf{x}) = \sum_{R_b} A(R_b(c(\mathbf{x}))) - \sum_{R_w} A(R_w(c(\mathbf{x}))) \ , \tag{3.10}$$

where A denotes the area (the number of pixels) of a given rectangle and $c(\mathbf{x})$ is an appropriate colour transformation of the input image. Three possible values of the colour parameter are considered: $c = \{red, blue, yellow\}$ which correspond to the colour transformations defined in equation (3.1). Colour parametrisation significantly increases the discriminative power of the weak classifiers based on the rectangular filters from Figure 3.8.

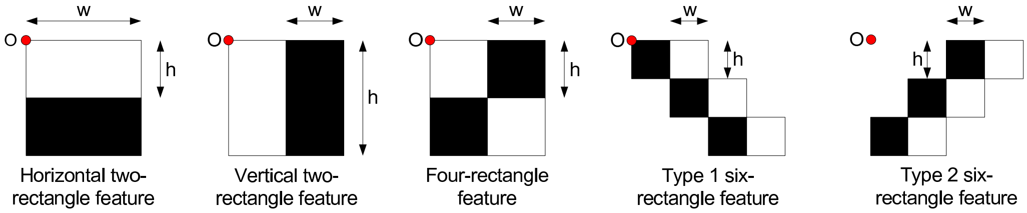

w
O
h
Horizontal two-rectangle feature
w
O
h
Vertical two-rectangle feature
w
O
h
Four-rectangle feature
w
O
h
Type 1 six-rectangle feature
w
O
h
Type 2 six-rectangle feature

Figure 3.8: Rectangular filters used to train the traffic sign detector. In each image the position of the origin (the red point) as well as width and height of a single rectangular component are marked.

In Section 3.3.3, apart from traffic sign detection, we also demonstrate the performance of a boosted classifier cascade in low-resolution pedestrian detection, which is an another traffic-related problem often considered in visual driver assistance. Unlike in the case of traffic signs, for which the colour information is critical and the direction of the colour-specific gradients is known, there is no need to parametrise rectangular filters with colour while learning a pedestrian detector. In general, it cannot be predicted of what colour the human's clothing might be and whether the human figure will appear darker or lighter than the background. Therefore, in human detection the previously discussed rectangular filters are evaluated on gray-scale images and with the sign of the area difference ignored:

$$\phi_{x,y,w,h}(\mathbf{x}) = \left| \sum_{R_b} A(R_b(\mathbf{x})) - \sum_{R_w} A(R_w(\mathbf{x})) \right| \ , \tag{3.11}$$

In addition to the aforementioned rectangular image filters, also their "floating" variants are considered (FRFs). They are defined in the same way as standard RFs, but each takes an additional parameter δ allowing it to float around its reference position (x, y) within a square area bounded by $x \pm \delta$ and $y \pm \delta$. Value of FRF is determined as a maximum of the corresponding RF values over all its possible locations. This helps achieve a translational invariance of the filters. Several other distance-based features used are characterised below:

- **HOG Distance (HOGD)** - it quantifies the difference between how much the locally evaluated gradient orientation histogram differs from its mean computed over all target and non-target examples in the training dataset. Specifically:

$$\phi_{x,y,w,h}(\mathbf{x}) = \|\mathbf{g}_{x,y,w,h} - \bar{\mathbf{g}}^P_{x,y,w,h}\| - \|\mathbf{g}_{x,y,w,h} - \bar{\mathbf{g}}^N_{x,y,w,h}\| \ , \tag{3.12}$$

where $\mathbf{g}_{x,y,w,h}$ stands for the HOG computed at the region of size (w, h) with the top-left corner at (x, y), and $\bar{\mathbf{g}}^P_{x,y,w,h}$, $\bar{\mathbf{g}}^N_{x,y,w,h}$ are mean HOGs determined from all positive and negative training images respectively.

- **Mean Gradient Distance (MGD)** - this feature measures a difference between the average pixel-wise gradient distance from the positive class mean and the average pixel-wise gradient distance from the negative class mean, calculated within a given rectangular region of the image specified by x, y, w, h. The gradient used may be directional or magnitude. If we let $\mathbf{g}_{x,y,w,h}$ be a $w \times h$-dimensional portion of a given-type gradient map, originating at (x, y), and $\bar{\mathbf{g}}^P_{x,y,w,h}$, $\bar{\mathbf{g}}^N_{x,y,w,h}$ denote the means of this 2D vector over all available training examples from the positive and negative class respectively, then the MGD feature evaluates to:

$$\phi_{x,y,w,h}(\mathbf{x}) = \frac{1}{wh}\left(\sum_{s=x}^{x+w}\sum_{t=y}^{y+h} |\mathbf{g}(s,t) - \bar{\mathbf{g}}_P(s,t)| - \sum_{s=x}^{x+w}\sum_{t=y}^{y+h} |\mathbf{g}(s,t) - \bar{\mathbf{g}}_N(s,t)|\right) \ . \tag{3.13}$$

- **Combined Mean Gradient Distance (CMGD)** - an extension of MGD in two aspects. First, different average gradient distances are combined together into a vector of length $2n$ where n is the number of gradient maps used. Each $2i$-th slot of such a feature vector is occupied by an average pixel-wise distance from the positive class's mean $\Delta\mathbf{g}^{(i)}_P$ and each $2i{+}1$-th slot by the same distance from the negative class mean $\Delta\mathbf{g}^{(i)}_N$, where:

$$\begin{array}{ll} \Delta\mathbf{g}^{(i)}_P & = \sum_{s=x}^{x+w}\sum_{t=y}^{y+h} |\mathbf{g}^{(i)}(s,t) - \bar{\mathbf{g}}^{(i)}_P(s,t)| \\ \Delta\mathbf{g}^{(i)}_N & = \sum_{s=x}^{x+w}\sum_{t=y}^{y+h} |\mathbf{g}^{(i)}(s,t) - \bar{\mathbf{g}}^{(i)}_N(s,t)| \end{array} \ . \tag{3.14}$$

Second, the weak classifier operating in the CMGD feature space is associated with the Gaussian Mixture density estimator. It is used to model the feature distribution for the pedestrian and non-pedestrian classes separately. The induced classifier assigns a label to the unknown example $\mathbf{x}$ by choosing the mixture yielding higher value of PDF for it.

3.3.3 Application 1: Traffic Sign Detection

In order to evaluate the performance of the detectors presented in Section 3.3.1 and 3.3.2, a series of experiments have been conducted involving both static images of road signs and realistic traffic video captured from a moving vehicle. In the first experiment we evaluated the regular polygon transform. The goal was to detect four categories of Polish traffic signs in each frame of the input video captured with a standard, car-mounted DV camcorder. These categories are shown in Figure 3.9. Minimum radius of the targeted signs was set to 15 pixels. The Hough-based detectors were used as described in Section 3.3.1. Specifically, each detector used the variant of regular polygon transform and the colour image filter from equation 3.1 corresponding to the shape and the rim colour of the signs in the targeted category. For instance, to detect the cautionary signs, the parameter n of the regular polygon transform was set to 3 (equiangular triangles), and the detector operated on the original RGB image transformed using the third filter from equation 3.1. Each category-specific detector was also independently tuned, based on a number of training images, to determine the optimal threshold in the Hough vote space. The multiple spatially close, above-threshold detector's responses were filtered by simply picking maxima over circular local regions of a predefined radius.

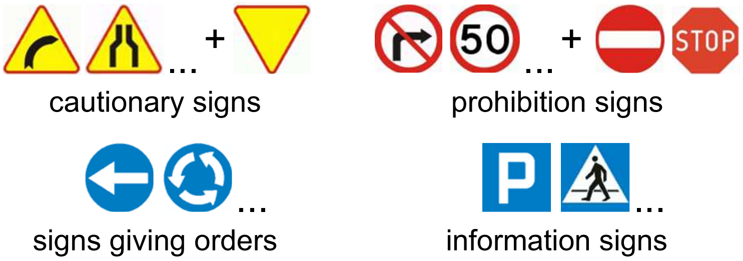


Figure 3.9: Four categories of Polish traffic signs captured by the Hough-based regular polygon detectors. The octagonal "STOP" sign is put into the category of red circular prohibition signs because in the low resolution imagery the noisy contours of octagons seen at a substantial distance from the camera are well approximated by circles.

Table 3.1 illustrates the experimental results obtained for different sign categories. The overall detection rate reached nearly 93% at an average processing speed of 25-30 fps. Most failures were caused by the insufficient chromaticity contrast between a sign's boundary and the background, especially for pale-coloured and shady signs. In a few cases this low contrast was caused by the poor quality of the physical target objects rather than their temporarily confusing appearance. Also, even if the sign was captured, the accuracy of the contour fit provided by the Hough detector was at times unsatisfactory. It was caused by low image resolution (when the distance from the camera was large), presence of confusing background objects' edges, or motion blur, which in turn generated blurry peaks or multiple peaks in the Hough vote space. At the detection stage, which is of little practical use, it may not be critical. However, for recognition, where a fine alignment of the target is highly desirable, this problem must be addressed. A convenient method of refining the responses in the Hough space will be presented in Section 3.4.

The second experiment was intended to quantitatively illustrate the problem of insufficient accuracy of the available object detectors. In this experiment we only

	RC (55)	BC (25)	YT (42)	BS (13)	overall (135)
detected	51/60 (85.0%)	16/16 (100.0%)	85/89 (95.6%)	43/48 (89.6%)	195/213 (92.9%)

Table 3.1: Detection rates obtained in the video-based experiment. The number of classes in each sign category: red circles (RC), blue circles (BC), yellow triangles (YT), and blue squares (BS) is given in parentheses.

considered the images of circular signs that were cropped from the individual frames of a video captured in Japan. Generally, as in most images the road signs appear in high visual clutter, this dataset is harder than the one used in the previous experiment. Both detectors discussed in Sections 3.3.1 and 3.3.2 were tested using a total of 8175 images, a few of which are shown in Figure 3.10. Another 4218 images were used for cascade training. Each image contained a sign in the centre and its close neighbourhood. The width and height of each region was equal to 3 × sign's diameter. For each image a ground truth centre position and the radius of a sign was given by a triple: (x_c, y_c, r). The experiment was repeated for: 1) varying threshold in the Hough vote space, and 2) varying threshold of the classifier in the last cascade layer. Quantities measured were: 1) mean number of candidates detected per image, 2) mean distance between a detected circle and the ground truth circle expressed with a Euclidean metric over the abovementioned position-scale triples, and 3) miss rate, i.e. the percentage of images where no sign was detected. Relationship between the miss rate and the two other quantities is illustrated in Figure 3.11.

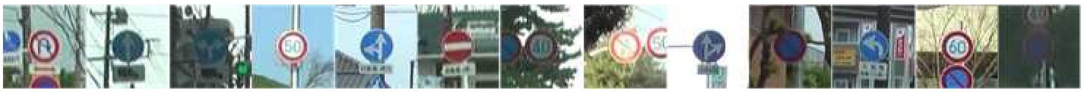


Figure 3.10: Example Japanese traffic sign images from the dataset used for quantitative evaluation of the accuracy of our road sign detectors.

The experiment showed that the Haar cascade is a slightly more accurate road sign detector than the circular Hough transform in the entire range of practically useful operating points. However, this advantage was achieved at the cost of much higher sensitivity, and hence more computation. While the average processing time of a single image was approximately 10ms for a Hough detector, this time increased to over 20ms for a Haar cascade, when scanning the input image with a 2-pixel step. For a pixel-by-pixel scanning, the cascaded classifier was approximately 7 times slower than the Hough detector. Regardless of the results of this comparison, both techniques, when used alone, appear to be impractical due to the multiple redundant location hypotheses produced for single true target objects. For further processing, e.g. pictogram classification, a single accurately fit candidate region is required for each true sign seen in the scene. To address this problem, an appropriate postprocessing of the detector's responses is proposed in Section 3.4.

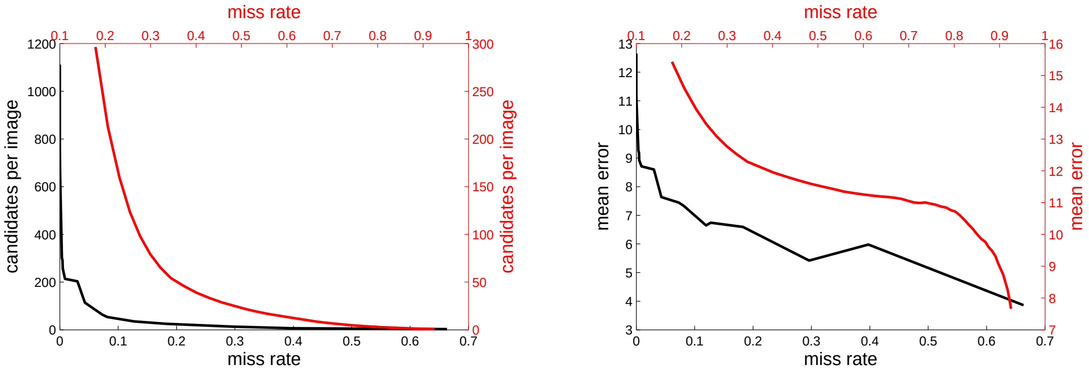


Figure 3.11: Relationship between the mean number of candidates per image detected and the miss rate (left), and between the mean distance of the detected candidates from the ground truth locations and the miss rate (right). Black lines and axes correspond to the Haar cascade and the red lines and axes correspond to the regular polygon detector. This figure is best viewed in colour.

3.3.4 Application 2: Human Detection

The object detection approach discussed in Section 3.3.2, together with the quad-tree RoI extraction technique introduced in Section 3.2, can be adopted to address other challenging traffic-related problems, e.g. moving pedestrian detection. In this case the goal is to possibly the best discriminate between a moving human figure and anything else (the background). Potential applications of such a detector could be intruder detection or it could serve as a prerequisite for some higher-level human behaviour analysis, e.g. gait recognition or more general action recognition.

To quickly reduce the search region of the image, our quad-tree focus operator is utilised in a similar way as for crude road sign localisation, but the inter-frame motion is considered the key low-level image feature to be exploited. This approach is valid as long as the camera is stationary, which is inherently assumed, or when its ego-motion can be compensated. In each frame of the input video, instead of measuring the spatial image gradient in each image pixel, the temporal gradient is computed:

$$g_t(i,j) = |I_t(i,j) - I_{t-1}(i,j)| \ , \tag{3.15}$$

where I_{t-1} and I_t are the consecutive frame images. Upon construction of the integral map of temporal gradient image G_t at time t, the total amount of inter-frame motion observed within any rectangular region of the image can be rapidly measured. For static street surveillance this method instantly provides a fast and robust RoI extraction. To use it on board of a moving vehicle, the ego motion of the camera has to be compensated. Within each found RoI the human figure detection can be done using a trained cascade from Section 3.3.2 and the gray-level scalar-valued image descriptors listed at the end of that section.

We have evaluated the detection accuracy of a boosted cascade using the low-resolution images of humans and non humans from the Daimler-Chrysler dataset [163]. 1225 18 × 36 labelled images were used to train the cascade. 2450 same-size labelled images served as a validation dataset for adjusting the thresholds of boosted classifiers at each cascade layer. Another 6125 unlabelled images were used for testing. Example images from the Daimler-Chrysler dataset are shown in Figure 3.12. In Figure 3.13a the detection rates of the cascades trained using the standard rectangular filters (RF) from Figure 3.8 and their "floating" versions (FRF) have been compared. Refer to the end of Section 3.3.2 for details of these features. In Figure 3.13b the comparison of the Receiver Operating Characteristic (ROC) curves obtained using more image descriptors and their mixtures is provided.

Figure 3.12: Example pedestrian and non-pedestrian images from the Daimler-Chrysler dataset [163].

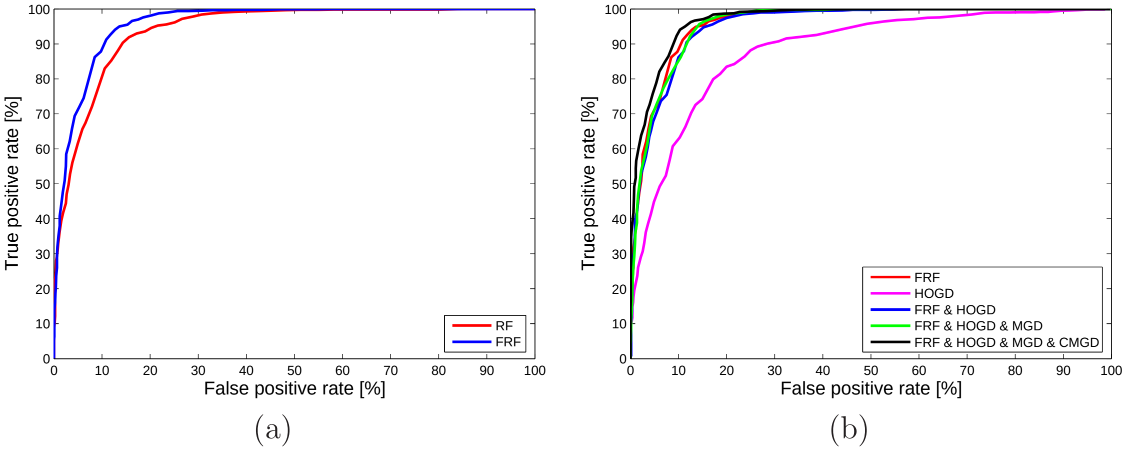


Figure 3.13: Detection accuracy of a boosted classifier cascade trained to discriminate between the pedestrians and the background using the Daimler-Chrysler dataset [163]: a) comparison of the results obtained in the situations when only RF or only FRF features were allowed; b) comparison of the detection accuracy observed for different combinations of image features discussed in Section 3.3.2 incorporated within the AdaBoost framework. This figure is best viewed in colour.

As seen, adding alternative features to the cascade training framework does not always have a positive effect on the detection accuracy. Actually, only replacing a set of the earlier discussed rectangular filters with their "floating" variants improves the results noticeably. Further enlargement of the area under ROC in Figure 3.13b was achieved by adding the CMGD features to the input feature pool. In practice, only one, global CMGD feature, corresponding to the entire 18 × 36 pixels region, was found highly discriminative and was therefore incorporated in the cascade. The best obtained results in this experiment are very close to the limits reported for this dataset [138, 154].

Note that despite the seemingly good shape of the ROC curves reported in the experiment involving static images, the usefulness of the cascaded human detector in a realistic scenario is very limited. Even when only one out of ten true pedestrians are misclassified by the cascade, the false positive rate is still at approximately 10% level. If a large area of a realistic traffic image had to be scanned in real time, it would correspond to a large overall number of false alarms, making such a detector impractical. For instance, for a 640 × 480 video resolution 10% false alarm rate corresponds to over 30,000 positive hypotheses on average when pixel-by-pixel image scanning id done and over 7,500 positive hypotheses on average where the image is scanned with a 2-pixel step. This fact illustrates that a reliable discriminative human detection in such a low resolution is probably beyond the capabilities of the available algorithms unless further steps are taken to refine the output of the insufficiently discriminative detector. This could be done by finding maxima in the detector's response space, which is an idea underlying the technique proposed in Section 3.4.

3.4 Confidence-Weighted Mean Shift Detection Refinement

Detection of small or distant objects in low-resolution imagery is in general a challenging problem. Because such objects look small and are often noised and blurred in the image, successful detection of the characteristic parts of the target through key-point or salient region extraction is impossible. Consequently, as the information about the object's compositionality is sparse and of little confidence, the generative object detection methods cannot be applied. This justifies the use of discriminative detectors for capturing objects like traffic signs or humans in mid or far plane. However, as our experimentation in Sections 3.3.3 and 3.3.4 shows, the available detectors of this kind are often insufficiently accurate and produce numerous redundant positive responses around the true target locations/scales while scanning an input image. In the case of TSR, these responses must be processed further in the processing pipeline by the road sign classifier. From this perspective, analysis of each candidate sign location and scale separately, where the locations and scales greatly overlap with one another, is impractical and computationally intractable. Instead, the redundancy may be eliminated by finding good representatives in the clouds of positive detector's hypotheses.

One possible way of increasing the accuracy and selectivity of an over-sensitive detector is to consider its response space a probability distribution with modes to be found. Mean shift algorithm [82] is a well-established non-parametric kernel density estimation technique that can be used for this task. However, the original mean shift formulation does not account for the possibly varying relevance of the input points.

As this situation takes place here – the relevance of the sign location/scale hypotheses varies, we propose a simple modification of the mean shift algorithm, called *Confidence-Weighted Mean Shift*. It alleviates the aforementioned shortcoming by incorporating the confidences of the detector's responses into the mode finding procedure. It is shown that such a refinement procedure can be applied to the output of any detector that yields a soft decision or can be modified to do so.

3.4.1 Definition

Assume that a traffic sign detector, regardless of which actual discrimination method is used, yields a number of positive location hypotheses $\mathbf{x}_j$ for a given image. By "positive" we mean those hypotheses that receive the above-threshold response, where the threshold is determined at the training stage. In the case of the detectors discussed in Sections 3.3.1 and 3.3.2, this threshold is set in the shape parameter space (Hough-based regular polygon detectors), or is used to determine the minimum combined response of the weak classifiers for the tested location to be considered as containing the target (boosted classifier cascade). Each positive hypothesis of the detector is characterised with a vector, $\mathbf{x}_j = [x_j, y_j, s_j]$, encoding the object's centroid position and its scale. In addition, $\mathbf{x}_j$ is assigned a confidence value, q_j, which is related to the soft response of the detector. In the case of a single binary boosted classifier associated with each layer of the cascade, such a confidence measure can be naturally related to the distance of the response from the linear decision boundary:

$$q_j = q(\mathbf{x}_j) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}_j) \ , \tag{3.16}$$

where $h_t(\mathbf{x}_j)$ denote the weak classifier responses, $\alpha_t = \log(\frac{1-e_t}{e_t})$, and e_t are the error rates of the weak classifiers. In the case of an entire cascade, the confidence formula can no longer be treated as a distance from the decision boundary, which is now non-linear. However, it can be approximated by a sum of $q_j^{(k)}$ terms over all K cascade layers, taking the modified thresholds t_k in each layer into account:

$$q_j = \sum_{k=1}^{K} q_j^{(k)} = \sum_{k=1}^{K} \sum_{t=1}^{T_k} (\alpha_t h_t(\mathbf{x}_j) - t_k) \ . \tag{3.17}$$

In the case of Hough-based regular polygon detectors, the confidence of each above-threshold hypothesis on the desired shape's presence can be simply measured with the normalised number of votes accumulated for this hypothesis in the voting space. Normalisation can be made relative to the maximum number of votes recorded in a given image. In general, confidence q_j of the detector's response can be expressed with any quantity that evaluates to a numerical, comparable value, and is indicative of the likelihood of the target object's presence at a given image position and for a given scale.

Let $f(\mathbf{x})$ be the underlying distribution of $\mathbf{x}$. Our goal is to find the maxima of this distribution. Given n data points, $\mathbf{x}_j$, $j = 1, \ldots, n$ on a 3-dimensional space $\mathbb{R}^3$, the multivariate kernel density estimate of $f(\mathbf{x})$ obtained by kernel $K(\mathbf{x})$ is given by:

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{j=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x}_j}{h}\right) \quad , \tag{3.18}$$

where h is the bandwidth parameter determining the scale of the estimated density, and $d = 3$. If a radially symmetric kernel is used, it suffices to define a function $k(\mathbf{x})$, called profile of the kernel K, satisfying:

$$K(\mathbf{x}) = c_{k,d} k\left(\|\mathbf{x}\|^2\right) \quad , \tag{3.19}$$

where $c_{k,d}$ is a normalisation constant which assures $K(\mathbf{x})$ integrates to 1. Introducing the kernel profile in (3.18) yields:

$$\hat{f}(\mathbf{x}) = \frac{c_{k,d}}{nh^d} \sum_{j=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_j}{h}\right\|^2\right) \quad , \tag{3.20}$$

To find, modes of the above density function, zeros of its gradient have to be found, i.e. $\nabla\hat{f}(\mathbf{x}) = 0$. The gradient of the density estimator is given by:

$$\nabla\hat{f}(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{j=1}^{n} (\mathbf{x}-\mathbf{x}_j) k'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_j}{h}\right\|^2\right) \quad . \tag{3.21}$$

Substituting $g(\mathbf{s}) = -k'(\mathbf{s})$ and introducing g into (3.21) yields:

$$\nabla\hat{f}(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{j=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_j}{h}\right\|^2\right)\right] \left[\frac{\sum_{j=1}^{n} \mathbf{x}_j g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_j}{h}\right\|^2\right)}{\sum_{j=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_j}{h}\right\|^2\right)} - \mathbf{x}\right] \quad . \tag{3.22}$$

The first term in equation 3.22 is proportional to the density estimate at $\mathbf{x}$ computed with kernel $G(\mathbf{x}) = c_{g,d} g\left(\|\mathbf{x}\|^2\right)$, where $c_{g,d}$ is the corresponding normalisation constant. The second term, called *mean shift*, is a vector that always points towards the maximum increase in the density. It is modified by incorporating the point confidence terms $q_j(\mathbf{x})$, as follows:

$$\mathbf{m}_{h,g}(\mathbf{x}) = \frac{\sum_{j=1}^{n} \mathbf{x}_j q_j(\mathbf{x}) g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_j}{h}\right\|^2\right)}{\sum_{j=1}^{n} q_j(\mathbf{x}) g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_j}{h}\right\|^2\right)} - \mathbf{x} \quad . \tag{3.23}$$

The modified mean shift clustering algorithm is called *Confidence-Weighted Mean Shift*. It iteratively steps towards the stationary points of the estimated density via alternate computation of the mean shift vector, $\mathbf{m}_{h,g}(\mathbf{x}^{(t)})$, and translation of the current kernel window by this vector, $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \mathbf{m}_{h,g}(\mathbf{x}^{(t)})$, until convergence. Plugging the confidence weights into the mean shift formula is equivalent to amplifying the density gradients pointing towards the more reliably detected instances of the target class. Therefore, the mode finding procedure is prevented from getting locked in the regions containing dense positive yet weak hypotheses. The found modes of $f(\mathbf{x})$ can be treated as the ultimate output of the detector. From these modes the location and the scale of the object candidates to track and/or recognise can be readily extracted.

3.4.2 Discussion

In order to evaluate the influence of the *Confidence-Weighted Mean Shift* clustering algorithm on the responses of the object detectors presented earlier in this chapter, we have repeated the experiment from Section 3.3.3, but applying the refinement procedure to the output generated by each detector. Obtained results are shown in Figure 3.14. It can be noticed that in the case of the circular Hough detector, the mean number of detected candidates per image roughly corresponds to the percentage of the images where any candidate was detected. This implies that the proposed detection refinement scheme most likely collapses the multiple positive responses of the detector into a single candidate, which is an intended outcome. The same effect is achieved by a refined cascade of classifiers only for a relatively high threshold of the last layer, when the miss rate of the detector is considerable.

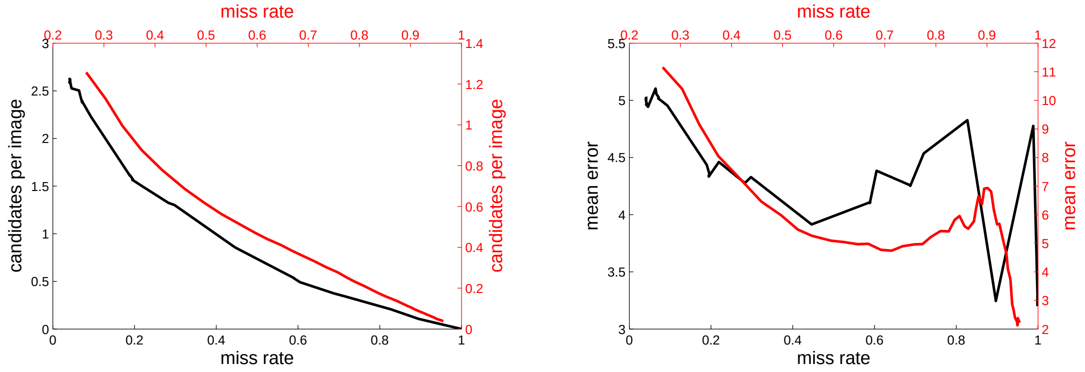


Figure 3.14: Relationship between the mean number of candidates per image detected and the miss rate (left), and between the mean distance of the detected candidates from the ground truth locations and the miss rate (right). Black lines and axes correspond to the refined Haar cascade and the red lines and axes correspond to the refined regular polygon detector. This figure is best viewed in colour.

The mean error of both detectors is lower with the refinement procedure enabled, with the Haar cascade being by 25-50% more accurate. Interestingly, the improvement in the accuracy of the Hough detector is dramatic, while the refined Haar cascade merely eliminates redundancy, but does not reduce the error significantly. Moreover, the difference between the average processing time of a single image using each detector became even more apparent (see Section 3.3.3 for comparison). It stayed at the 10 ms level for the Hough detector, but increased to 50-100 ms for the cascade scanning the input images with 2-pixel step. Overall, no or little advantage over the much simpler Hough Transform-based detectors, which do not require any training, does not provide convincing justification for using the boosted cascade approach to accurately detect multiple diverse types of road signs in the traffic video. However, without the mean shift clustering, it can be used as a method of RoI extraction, complementary to the quad-tree operator presented in Section 3.2.2.

In a separate experiment we trained a low-resolution binary human classifier using thousands of 18×36 pixels images from the Daimler-Chrysler dataset [163], as described in Section 3.3.4. The trained cascade was evaluated on a number of short video sequences depicting different types of human gait, where humans were usually seen at distance 20-50 metres from the camera. These sequences were captured from a camera mounted on board of a parked vehicle. Therefore, the problem can be classified into static surveillance. To make our human detector scale-invariant, each input frame was decomposed into a pyramid of downscaled copies of the same image, each two consecutive copies differing by a scale factor of 0.9. While all images in the pyramid were scanned, the operational scale of the detector was kept constant and equal to the size of the images used in the training stage.

Similarly to what was observed in the experiment involving traffic sign images, the cascade in runtime produced clusters of overlapping candidate object bounding boxes. As long as the search region can be efficiently reduced based on the inter-frame motion, as described in Section 3.3.4, the locally applied mean shift mode finding algorithm successfully integrates multiple hypotheses and hence increases the overall accuracy of the detector. However, if the condition on the stationarity of the camera does not hold, e.g. when the vehicle is in motion which cannot be compensated, and no other RoI extraction technique is available, the cascaded classifier behaves as the ROC curve from Figure 3.13 suggests. Undesirable false positives are generated around the objects with dominant vertical gradients which resemble humans seen at a considerable distance from the camera, e.g. traffic sign poles or tree trunks.

To visualise how the *Confidence-Weighted Mean Shift* clustering algorithm refines the responses of the Hough-based and boosted cascade detectors, we have included several test images used in the above discussed experiments. They are shown in Figures 3.15 and 3.16. Each pair of images in the former figure contains marked estimated circular signs' contours detected by the HT before and after applying the refinement procedure. In Figure 3.16 clouds of the cascade's responses are shown together with the modes found in the response space using the standard and the *Confidence-Weighted Mean Shift* clustering.

3.5 Summary

Detection of traffic signs in the traffic video is in general a challenging problem. By many it is considered the most difficult task in video-based TSR, compared to tracking and recognition. There seem to be two major difficulties related to the detection of road signs. First, in the dynamically changing scene, it is hard to localise the search region in both rapid and reliable way. The second limitation of the available road sign detectors is that they are insufficiently discriminative,

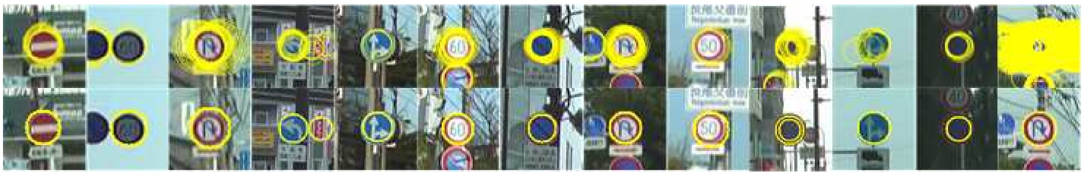


Figure 3.15: Output of the Hough circle detector before (upper row) and after (lower row) applying the *Confidence-Weighted Mean Shift* refinement procedure. The transparency of the detected circles in the upper row images correspond to their confidences expressed with the scaled number of votes picked from the Hough voting space.

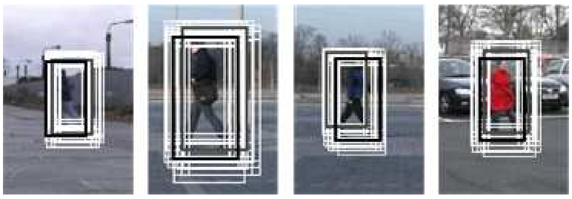


Figure 3.16: Typical output of an attentive boosted classifier cascade applied to moving pedestrian images. Clouds of hypotheses marked white are visible around the human figures. Gray boxes correspond to the modes found by the standard mean shift algorithm. Black boxes denote the modes found by the proposed *Confidence-Weighted Mean Shift*.

partially because the crucial pieces of information, colour and shape of the signs, are not fully and simultaneously exploited. As a result, the TSR systems that are currently released on the market, e.g. [162], specialise in detecting only certain types of traffic signs which have similar appearance and are therefore easier to learn.

In this chapter we attempted to address the above limitations in order to improve the road sign detection process. Our approach is based on an intuition that a separate preliminary localisation step, made to optimally reduce the interest region of the scene, is necessary to make a dynamic recognition system computationally efficient. Such a preliminary location algorithm has been introduced in Section 3.2. This way we have clearly separated the rough localisation step from the actual detection step. In the former step, detection accuracy is only optimised with respect to minimising the number of missed true signs, even at the cost of large increase in the number of false positive interest regions. The true sign-background discrimination, with false positive rate being optimised too, is made by the actual object detector. Its design is much more sophisticated than that of RoI extractor. It more fully exploits the *a priori* information about the traffic signs. Finally, it is aimed at accurate pixel-level estimation of the position and the scale of the target as on the quality of this estimation the success of recognition depends.

Reducing the area of the input scene to be analysed by the detector is carried out using a quad-tree focus of attention operator. This technique is very simple to comprehend and to implement. It merely attempts to optimally narrow down the analysis region to those fragments of the input image where there is sufficient cumulation of pixels lying on the edges of sign-specific colour regions. Strength of our focus operator lies in its computational efficiency. First, it processes the image recursively, in a greedy, quad-tree fashion. At each level of recursion it measures the amount of sign-specific colour gradients contained in the currently analysed window. The window dimensions are halved with each step down the recursion and the recursion is broken whenever a minimum amount of colour gradient is not exceeded. Because bright colours of traffic signs are generally very rare in reality, in practice this recursion is very often broken, which dramatically reduces the analysis time. This exceptionally fast processing would have not been possible to achieve, if it had not been for the usage of integral images in the colour gradient measurement process. Integral imaging makes the computation of the cumulative feature contained in any given rectangular region independent of the size of this region. Apart from for traffic sign localisation, the quad-tree is useful in solving other problems, like localisation of moving targets, which we have demonstrated in Section 3.3.4.

At the sign detection stage the main emphasis has not been put on maximising the discriminative power of the binary object classifier, where the limitations resulting from low image resolution, adverse illumination, noise and other factors are significant. Instead, we consider two practical, decently accurate and fast sign detectors, but put a special effort on refining their responses. The object detectors investigated in the context of TSR are: a Viola and Jones's cascade of boosted classifiers [111] and a colour-aware, regular polygon detector borrowed from the work of Barnes et al. [128]. The experimental evaluation of these techniques made in Sections 3.3.3 and 3.3.4 has shown that both detectors are insufficiently discriminative in the task of capturing low resolution objects: traffic signs and humans. As a result, when their operating points are chosen such that practically all true instances of the target are detected, a number of redundant positive responses are also produced in their vicinity.

Confidence-Weighted Mean Shift clustering algorithm has been proposed to refine the output of the object detectors considered in this chapter. This refinement is aimed at collapsing clouds of redundant sign location/scale hypotheses into single bounding boxes, accurately delimiting the true target instances. The idea of this kernel density estimation technique is centred around finding maxima in the detector's response space, which is treated as a sampled probability distribution, modulated by the response confidences. For this method to work, the detector must yield soft responses which are indicative of the likelihood of the presence of the target in a given scale and at a given image location. Our detectors satisfy this condition, which is justified in Section 3.4.1. The proposed mean shift clustering algorithm pursues the desired modes of the confidence-weighted response distribution in an iterative fashion, by shifting the kernel window in the direction of the steepest increase in the density of this distribution. Novelty of this algorithm, compared to standard mean shift, lies in that it accounts for the varying relevance of the input points, which leads to an improved detection accuracy.

Chapter 4

Target Tracking

Object tracking is a common task in many video-based machine vision applications, e.g. street surveillance or military target tracking. It is essential in order to maintain the state of an object as it evolves over time by changing its 3D position, apparent size, colour, texture, or when it is temporally invisible. It should be noted that such applications naturally require the tracker to work in real time which allows it to capture the target as new observations are acquired live from the input video. A natural question to ask is whether maintaining a specialised tracking component in a dynamic object recognition system like TSR is necessary. In other words, cannot the detector alone achieve the same goals as separate detector and tracker would do?

For several reasons it can not. First, the possibility of achieving an invariance to general pose and appearance changes at a feature representation level is limited. Therefore, one possible role of a tracker involves probabilistic modelling of these changes. In this scenario the parametric description of the target state is propagated over time, which involves alternate predictions of the state parameters and their updates, driven by the consecutive frame observations. The second critical role of a tracker is modelling the temporal dependencies between these observations. Since in the absence of a tracker the detection must be performed independently in each frame of an input video, these relationships are invertibly lost and hence the target state estimation over time is less stable and less accurate. Finally, spatial tracking of the moving objects dramatically reduces the computation involved, compared to the individual frame-based detections. Depending on the problem and the tracking scheme adopted, this reduction can be achieved in two ways. First, the tracker can be used to merely restrict the boundaries of a search region which is locally scanned by the detector in every image of the input sequence to register the new position/scale of the target. In the second scenario using a dedicated detector for spatial localisation of the existing object candidates is unnecessary as their actual location and 3D pose in the scene can be retrieved from the tracker's parametric state estimates. In this case the role of the detector is reduced to merely registering the observable features at a fixed image location, predicted by the tracker.

Tracking road signs from a car-mounted camera is difficult for several reasons. The vehicle may undergo a generally non-uniform motion, e.g. it may rapidly accelerate, brake, or change the driving direction at the crossroads or while overtaking an another vehicle. This makes it risky to *a priori* assume an arbitrary motion model, e.g. constant direction and speed. Further difficulties in road sign tracking are related to the instability of the mobile camera mount. Whenever the car moves on an uneven road surface, its vibrations are propagated to the camera, which is reflected in a blur reducing the contrast in the captured video. This phenomenon is generally difficult to model. Another factor severely degrading the performance of road sign trackers is insufficient and often varying illumination. Traffic signs are very distinctive owing to their characteristic shapes and bright colours. However, in adverse illumination both properties cannot be exploited fully as the contrast around the sign's boundary and its pictogram symbols is much less sharp and the normally distinctive colours appear pale. Additionally, shadows and incident light reflections, which often occur in traffic scenes, may affect the consistency of the track.

In many realistic situations, especially in urban scenes, the traffic signs must be detected and recognised in presence of high visual clutter. Although the appearance of a sign itself does not change much while being approached by a vehicle, its neighbourhood in the image may change rapidly. It increases a risk of confusions between the true signs and the other objects or the background and hence makes the temporal tracking of signs, particularly their contours, challenging. Finally, the computational aspect of tracking must be pointed out as this the most intensive step in TSR, assuming that the full exploration of the scene in search of the newly emerged candidates does not need to be performed in every frame of the input video. With potentially up to several different signs present in the scene, the total time needed to estimate the state of each candidate in each frame should not exceed several milliseconds. Therefore, the tracker, instead of being confined to merely reducing the local search region for the detector, should also operate on a low-dimensionality feature representation of the target, so that the update of the track's state based on the current-frame observation can be made sufficiently fast.

In the following parts of this chapter we will more exhaustively discuss the above-mentioned problems and, by presenting our contributions, address some of them. Conceptually, the content of the following sections corresponds to the implementation of a tracking component of a TSR system, shown on the upper left-hand side of the diagram in Figure 1.3. Section 4.1 contains discussion of the state-of-the-art approaches to road sign tracking. In Section 4.2 the baseline, Kalman Filter/Particle-Filter-based method is described. In Section 4.3 a contour tracking framework for temporal localisation of a sign's boundary is presented. Usefulness of this method in presence of visual clutter is demonstrated experimentally. Section 4.4 is devoted to a discussion and evaluation of a robust affine motion tracker which is capable of reconstructing a frontal view of the target, irrespective of the camera's viewpoint. Finally, conclusions of this chapter are drawn in Section 4.5.

4.1 Background

The tracking aspect of a dynamic traffic sign recognition has been relatively rarely addressed in the previous studies on TSR. It is partly due to little experimentation done outside the car industry, involving real vehicles. In many early studies the need for using a tracker to facilitate the detection of road signs over time was overlooked. Frequently, it can be reasonably justified. For example, when a sign is seen at a substantial distance from the camera and against relatively consistent background, the changes in the apparent sign's motion and appearance are small. In that case a good localisation of the target can be achieved by running the detector in each video frame within a small neighbourhood of the previously observed sign's location and scale, without recourse to sophisticated temporal state modelling.

The most widely adopted approach to temporal road sign traffic is based on Kalman filtering (KF) [1]. This technique was used to predict the position and scale of the target in the consecutive frames of the input video and hence to reduce the region to be scanned by the detector. The geometric distortions of signs were always considered negligibly small and therefore the full affine motion was not modelled. The Kalman Filter is inherently linear, i.e. it yields an optimal (in a Bayesian sense) estimate of the state only when the equations describing the process and the observation model are linear. Therefore, the previous KF-based road sign trackers assumed a uniform, constant-direction motion of the vehicle, e.g. [49, 92], usually with *a priori* known velocity and the size of the real sign plates. The corrections to the assumed straight and uniform motion were implemented in the state equations as errors of the model.

A representative example of a Kalman Filter-based road sign tracker can be found in the study of Piccioli et al. [49]. They assumed that the car was moving with approximately constant speed $\mathbf{v} = (0, 0, v)$ in the Z-direction and on an approximately straight road with constant slope. In their model the state variables defined a 3D position of the road sign's centroid and the vehicle's velocity. A linear state transition model simply described the constancy of the state variables (up to the model errors), with an exception of the Z coordinate which changed proportionally to the vehicle's displacement. The measured quantities included the location (x_c, y_c) of the detected sign in the image and its apparent size σ. They were related to the state variables through the known focal length of the camera, and assuming fixed 2-pixel measurement errors. Nonlinearity of the resulting measurement model was eliminated by adopting the Extended Kalman Filter (EKF) which linearised the measurement equations around the predicted state.

Fang et al. [92] gave the same rationale for using a Kalman Filter as Piccioli and his colleagues. KF was primarily utilised in this work to predict the position and scale of the detected traffic sign candidate in each new frame of the input video, as well as to reduce the local analysis region. Authors again assumed an uniform, straight-line motion of a vehicle, with known car velocity, physical size of the signs and the focal length of the camera. However, both: the state transition model and the measurement model related together the same, apparent position and scale parameters of the target in the image: x, y and $1/r$, where the latter quantity was an inverse of the radius of the supposed road sign in the image plane. In this variant of the KF both models were linear and the measurement model was trivial, i.e. it was described with an identity matrix.

The main disadvantage of Kalman Filter is its assumption of the linearity of the state transition and the observation models, which to certain extent can be compensated by adopting the EKF technique. An another extension to the original KF, an Unscented Kalman Filter (UKF), is even more suitable to model the nonlinear target motions. However, this method seems not to have been used in any previous recognised studies on TSR. The existing KF-based trackers do not model the affine motion of the target. They are also dependent on the external parameters: vehicle's velocity, physical sign dimensions, and the focal length of the camera. While the value of the last parameter is readily available, the physical size of traffic signs, even those containing the same pictogram, may vary. Regarding the car velocity, the Kalman filtering discussed above is practically useful if the information from the speedometer is provided on input of the TSR system. Otherwise, their applicability becomes limited by their own assumptions on the nature of the motion. However, Miura et al. [65] proposed an interesting two-camera tracking system in which the position of a sign in the image was predicted without the vehicle's speed and the focal length information. In their approach this prediction was done in the images captured by a front-looking, wide-angle lens camera. The other, telephoto camera changed its viewing direction to focus the attention on the predicted sign's location, so that a zoomed image of the target could be acquired for further analysis. This tracking system was more advanced compared to the previous approaches. However, its apparent limitation was that it could only analyse one candidate sign at a time. Besides, switching the focus between different candidate signs was slow due to the mechanical limitations of the hardware.

As road signs are solid, planar objects, they essentially look the same to a human driver approaching them, unless the vehicle is already very close to the target, substantial illumination changes occur, or they are temporarily occluded. In the same time it is practical to recognise signs as early as possible, in order to notify the human of the incoming dangers or suggest appropriate actions/manoeuvres such that sufficient amount of time is left for analysis of that information and a correct reaction to it. Because the geometrical distortion of traffic signs is at that time usually avoidably small, temporal modelling of their appearance and 3D pose was not found critical and hence has not been so far seriously considered in TSR. What still lacks is a robust probabilistic framework for estimation of the track's state in the extraordinary circumstances, e.g. suddenly changing illumination, or in presence of occlusions, when the target or its parts remain temporarily invisible. Also, it would be practical to model over time, apart from the position and scale of the signs, also the remaining parameters of their apparent motion: rotation and shear, preserving their relationships within an affine matrix structure. It would enable pose-invariant sign recognition.

Among the previously used tracking methods the one presented by Escalera et al. [120] was in the minority where probabilistic temporal appearance modelling was attempted. In this approach a deformable model of sign was developed which enabled detection of perspectively distorted, poorly illuminated, noised and occluded signs. However, in this work only the search strategies for obtaining the optimal deformation parameters based on matching the observable image features to the deformable model were outlined. This search was driven by an indeterministic minimisation of a sum of relatively complex energy functions encoding the information about the colour and the shape of the targeted traffic signs. As a result, the proposed detector was found far too slow to be run in real time. Unfortunately, authors seem to have assumed an arbitrary motion matrix relating the deformation parameters between consecutive time points. The true challenge is to devise a method for robust dynamic estimation of this motion matrix.

4.2 Kalman and Particle Filtering

In this section a baseline road sign tracking method is presented. It is based on the Kalman-Bucy filtering technique [1] commonly adopted in many previous studies on TSR, e.g. [49, 92]. Our tracker makes an assumption of straight-line, constant-velocity motion of the vehicle. It attempts to model only the geometry of the target over time and is composed of two parts: Kalman Filter (KF) for position and scale prediction, as well as a local search region estimation, and a Particle Filter (PF) used to correct the anisotropic scaling of the detected sign. The aforementioned components of the proposed geometrical tracker are detailed in Sections 4.2.1 and 4.2.2. In Section 4.2.3 the tracking results obtained using the combined Kalman Filter-Particle Filter method are discussed.

4.2.1 Kalman Filter

The Kalman Filter addresses a general problem of estimating the state $\mathbf{x} \in \mathbb{R}^n$ of a stochastic linear process:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t \ , \tag{4.1}$$

given the observations $\mathbf{z} \in \mathbb{R}^m$:

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \ . \tag{4.2}$$

$\mathbf{A}$ is the state transition matrix describing how the process advances in time, $\mathbf{B}$ relates the optional control input $\mathbf{u}$ with the state, and $\mathbf{H}$ is the measurement matrix relating the hidden state variable $\mathbf{x}$ with the vector of observations $\mathbf{z}$. The state and the observation dynamics are assumed to be linear. Besides, the random variables $\mathbf{w}_k$ and $\mathbf{v}_k$, representing the process and measurement noise respectively, are assumed to be independent of each other, white, and normally distributed with zero mean:

$$\begin{array}{ll} p(\mathbf{w}) & \sim N(0, \mathbf{Q}) \\ p(\mathbf{v}) & \sim N(0, \mathbf{R}) \end{array} \ . \tag{4.3}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are the process and measurement noise covariance matrices respectively. The assumptions of the linearity of the system dynamics and the normality of the noise processes imply that KF can be considered a special-case filter. This is due to the fact that the conditional probability density function

$p(\mathbf{x_t}|\mathbf{y}_1,\ldots,\mathbf{y}_t,\mathbf{u}_0,\ldots,\mathbf{u}_{t-1})$, is Gaussian for every time step t. Therefore, the filter does not need to evaluate and propagate the entire conditional PDF, but only its mean and variance which completely describe the Gaussian distribution.

The Kalman Filter operates in a prediction-correction fashion governed by the following equations:

$$\begin{array}{ll} \hat{\mathbf{x}}^-_{t+1} & = \mathbf{A}_t\hat{\mathbf{x}}_t + \mathbf{B}_t\mathbf{u}_t \\ \mathbf{P}^-_{t+1} & = \mathbf{A}_t\mathbf{P}_t\mathbf{A}_t^T + \mathbf{Q}_t \end{array} , \tag{4.4}$$

$$\begin{array}{ll} \mathbf{K}_{t+1} & = \mathbf{P}^-_{t+1}\mathbf{H}^T_{t+1}(\mathbf{H}_{t+1}\mathbf{P}^-_{t+1}\mathbf{H}^T_{t+1} + \mathbf{R}_{t+1})^{-1} \\ \hat{\mathbf{x}}^+_{t+1} & = \hat{\mathbf{x}}^-_{t+1} + \mathbf{K}_{t+1}(\mathbf{z}_{t+1} - \mathbf{H}_{t+1}\hat{\mathbf{x}}^-_{t+1}) \\ \mathbf{P}^+_{t+1} & = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H}_{t+1})\mathbf{P}^-_{t+1} \end{array} . \tag{4.5}$$

In equations 4.4, known as *time update*, the filter first projects the process state $\mathbf{x}$ and error covariance $\mathbf{P}$ estimates ahead from time t to $t+1$, to obtain new *a priori* estimates for the next time step. Estimation is made according to the state transition matrix $\mathbf{A}$ and optional matrix $\mathbf{B}$. During the so-called *measurement update*, expressed with equations 4.5, the filter incorporates the new measurement $\mathbf{z}_{t+1}$ into the *a priori* state estimate at time $t+1$ to obtain an improved, *a posteriori* state and error covariance estimates. K_{t+1} is a so-called *optimal Kalman gain* matrix that minimises the *a posteriori* error covariance.

The geometry of the model underlying our implementation of the Kalman Filter introduced above is illustrated in Figure 4.1. The quantities we want to track over time are the apparent sign's centroid (x, y) and its radii along both axes, r_x, r_y). We are not interested in estimating the real-world 3D position of the target. It is assumed that the vehicle's velocity, $\mathbf{v}$, the physical radius of a sign, R, and the focal length of the camera, f, are known [1]. At first it can be noticed that:

$$\begin{array}{lll} \frac{R}{r_x(t)} & = \frac{d(t)}{f} & \\ \frac{R}{r_x(t+1)} & = \frac{d(t+1)}{f} & = \frac{d(t)-v\Delta t}{f} \end{array} , \tag{4.6}$$

from which we can derive:

$$r_x(t+1) = \frac{fRr_x(t)}{fR - v\Delta t r_x(t)} = F_r(r_x(t)) \ . \tag{4.7}$$

As a result, we have related the sign's radius at time $t+1$ to its radius at time t using only the known quantities: the vehicle's velocity v, the camera's focal length f, and the real sign's radius R, which is usually standarised.

Furthermore, also the following relations hold:

$$\begin{array}{lll} \frac{X}{x(t)} & = \frac{d(t)}{f} & \\ \frac{X}{x(t+1)} & = \frac{d(t+1)}{f} & = \frac{d(t)-v\Delta t}{f} \end{array} , \tag{4.8}$$

[1] In the case of non-circular signs by "radius" we mean the radius of the circle a given regular polygon is inscribed on.

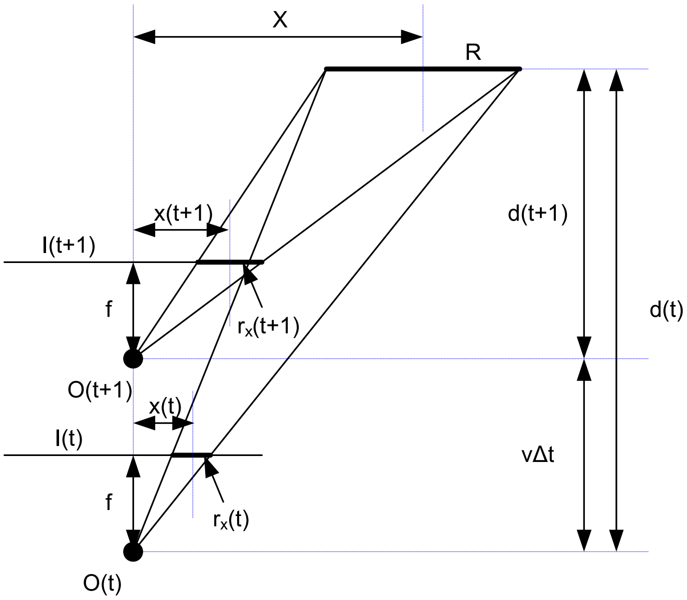


Figure 4.1: Geometrical arrangement of the dynamic road scene. A car at point $O(t)$ at time t advances by $v\Delta t$ to reach point $O(t+1)$ at time $t+1$. The sign and its projections onto the image planes $I(t)$ and $I(t+1)$ have been marked. f denotes camera's focal length.

where x denotes the horizontal image coordinate of the sign's centroid. Using (4.6) and (4.8) we obtain:

$$x(t+1) = \frac{r_x(t+1)x(t)}{r_x(t)} \quad , \tag{4.9}$$

and finally substituting $r_x(t+1)$ with (4.7) we get

$$x(t+1) = \frac{fRx(t)}{fR - v\Delta t r_x(t)} = F_x(x(t)) \quad . \tag{4.10}$$

The same derivation can be made for the vertical sign's radius r_y and y coordinate of its centroid. Using equation (4.10), the position of the sign's centroid in the image at time $t+1$ can be computed based on its position and size at time t.

With known relationships between the state and the observation variables, KF equations (4.4) and 4.5 can be concretised. First, note that both state equations, i.e. (4.7) and (4.10), are non-linear. Therefore, we adopt the Extended Kalman Filter (EKF) technique [13] to linearise the state equation around the previous state estimate $\hat{\mathbf{x}}_t$. Specifically, the state and the observation vectors in our model are defined as follows:

$$\hat{\mathbf{x}}_t = \begin{bmatrix} \hat{x}(t) \\ \hat{y}(t) \\ \frac{1}{\hat{r}_x(t)} \\ \frac{1}{\hat{r}_y(t)} \\ 1 \end{bmatrix} \quad \mathbf{z}_t = \begin{bmatrix} x(t) \\ y(t) \\ \frac{1}{r_x(t)} \\ \frac{1}{r_y(t)} \\ 1 \end{bmatrix} \quad . \tag{4.11}$$

The nonlinear process equation (4.10) is replaced with an approximate linear equation composed of the partial derivative of the nonlinear function:

$$x(t+1) = \left[\frac{\partial F_x(x)}{\partial x}\right]_{\mathbf{x}=\hat{\mathbf{x}}_t} x(t) = \frac{fR}{fR - v\Delta t \hat{r}_x(t)} x(t) \ . \tag{4.12}$$

The other equation, replacing (4.7), thanks to inverting r in the state vector, does not require linearisation:

$$\frac{1}{r_x(t+1)} = \frac{1}{r_x(t)} - \frac{v\Delta t}{fR} \ . \tag{4.13}$$

Full state transition matrix $\mathbf{A}_t$ can be written as:

$$\mathbf{A}_t = \begin{bmatrix} \frac{fR}{fR - v\Delta t \hat{r}_x(t)} & 0 & 0 & 0 & 0 \\ 0 & \frac{fR}{fR - v\Delta t \hat{r}_y(t)} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -\frac{v\Delta t}{fR} \\ 0 & 0 & 0 & 1 & -\frac{v\Delta t}{fR} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} , \tag{4.14}$$

and the measurement matrix $\mathbf{H}_{t+1}$ is a 5×5 identity matrix. It means that we measure directly the quantities of which the process' state vector is composed. As there is no control input in our model, the component $\mathbf{B}_t\mathbf{u}_t$ in equation (4.4) has been skipped. Also due to the lack of statistical knowledge about the distribution of the estimate errors, the process error covariance matrix $\mathbf{P}$ is initialised with the identity matrix, i.e. $\mathbf{P}_0 = \mathbf{I}_{5\times 5}$.

Finally, it should be noted that the above described Kalman tracker does not relieve us from using the detector. Actually, it is ran in each frame of the input video to obtain the measurements used in the update equations of the tracker. However, based on the knowledge about the predicted state at time t and the uncertainty of this prediction, the local analysis region that needs to be scanned at this time by the detector can be largely reduced. Specifically, it is centred at the predicted 2D location of a sign, and its size is given by the predicted 2D sign's size, augmented with the expected errors on location and size:

$$\begin{aligned} \Delta x_t &= \hat{r}_x^-(t) + \sqrt{(\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t)_{x,x}} + 2\left(\sqrt{(\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t)_{\frac{1}{r_x},\frac{1}{r_x}}}\right)^{-1} \\ \Delta y_t &= \hat{r}_y^-(t) + \sqrt{(\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t)_{y,y}} + 2\left(\sqrt{(\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t)_{\frac{1}{r_y},\frac{1}{r_y}}}\right)^{-1} \end{aligned} \ . \tag{4.15}$$

4.2.2 Particle Filter

The Kalman Filter introduced in Section 4.2.1, together with the discriminative object detectors discussed in Section 3.3, can be used to effectively track the road signs over time in the incoming video. However, neither of these detectors can retrieve the full 3D geometry of the target. The cascade of boosted classifiers only provides the information about the position of the object's centroid and its scale.

The regular polygon detector can additionally retrieve the in-plane rotation of traffic signs having non-circular shapes. For circular signs it is not possible. In both cases insufficient knowledge is available to determine the apparent rotation of traffic signs around the axes perpendicular to the optical axis of the camera, which may result from the changes of the camera's viewpoint relative to the target. The Particle Filter-based approach discussed in this section addresses this problem in that it enables correction of the anisotropic scaling which is the most common type of observable traffic signs' distortion. Although estimation of the complete set of parameters describing the affine motion of the target is possible using this technique, we confine ourselves to scale correction which is computationally tractable.

The proposed filter is initialised at the time the new likely traffic sign is detected in the scene. At this point the information about its centroid and scale $r_x = r_y$ is provided by the detector, be it the attentive classifier cascade or the Hough-based shape detector. The state at any time t is given by a pair of variables $\hat{\mathbf{s}}_t = (\hat{s}_{x,t}, \hat{s}_{y,t})$. They encode the horizontal and the vertical scaling factors of the affine transform that is to be applied to the symmetric shape found by the detector around the prior KF centroid and scale at time t. The aim of the PF at this point is to refine the previous-frame scale estimates such that the corrected shape better matches the currently observed image features. The refined scale estimates, together with the centroid's location found by the detector (as well as the in-plane rotation if the regular polygon detector is used) are used to update the KF state at time t. Note that in this tracking scheme the object detectors discussed in Section 3.3 are used even when the signs appear geometrically distorted, i.e. depart from the frontal poses for which the respective detectors are designed. However, with the small deformations that typically occur in real traffic situations, these detectors can still easily capture such signs.

Our Particle Filter generates a weighted set of N particles $\{(\mathbf{s}_{k,t}, w_{k,t}) : k = 1, \ldots, N\}$ that approximate the posterior distribution $p(\mathbf{s}_t | \mathbf{y}_0, \ldots, \mathbf{y}_t)$, where $\mathbf{y}_i$ denote the consecutive frame observations. The importance weights $w_{k,t}$ are approximations to the relative posterior densities of the particles such that $\sum_{k=1}^{N} w_{k,t} = 1$. Sampling Importance Resampling (SIR) technique [21] is adopted to approximate the expectation:

$$\int f(\mathbf{s}_t) p(\mathbf{s}_t | \mathbf{y}_0, \ldots, \mathbf{y}_t) d\mathbf{s}_t \approx \sum_{k=1}^{N} w_{k,t} f(\mathbf{s}_{k,t}) \ . \tag{4.16}$$

For a finite set of particles, the algorithm performance is dependent on the choice of the proposal distribution $\pi(\mathbf{s}_t | \mathbf{s}_{1,\ldots,t-1}, \mathbf{y}_{1,\ldots,t})$. The optimal proposal distribution is given by $p(\mathbf{s}_t | \mathbf{s}_{t-1}, \mathbf{y}_t)$. However, for simplicity of computation we use the transition prior:

$$\pi(\mathbf{s}_t | \mathbf{s}_{1,\ldots,t-1}, \mathbf{y}_{1,\ldots,t}) = p(\mathbf{s}_t | \mathbf{s}_{t-1}) \ . \tag{4.17}$$

It is modelled as a 2D Gaussian with mean at $(1, 1)$. In other words, in a new time step each particle is drawn such, that the expected scale of the associated hypothesised shape is the same as its scale in the previous time step.

The critical issue is how the observation process is carried out, and consequently, how the particles are updated. Below we only discuss this issue in detail with an assumption that the extension of the Hough Transform discussed in Section 3.3.1 is used as the shape detector. Namely, assuming that the k-th particle's state is $\mathbf{s}_k = (s_{x,k}, s_{y,k})$, and the type of the tracked shape has previously been determined, the hypothetical circle/regular polygon is constructed and its vertices (symmetry axes' endpoints in the case of a circle) are transformed through the scaling matrix:

$$\mathbf{S}_k = \begin{bmatrix} s_{x,k} & 0 & 0 \\ 0 & s_{y,k} & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{4.18}$$

Then, with the positions of the transformed points available, the shape's contour can be readily generated because under $\mathbf{S}_k$ the edges of a polygon remain to be straight segments spanned between the vertices. Moreover, the slope of each edge of a hypothesised polygon can be easily computed. A circle transformed through $\mathbf{S}_k$ becomes an ellipse which analytical description, together with the contour points and their slopes, can also be retrieved based on the coordinates of the transformed axes' endpoints.

Taking above into account, an error measure is associated with each particle. This error is calculated by following the circle's/polygon's contour and recording for each of its pixels the normalised difference between the slope of the hypothetical edge and the angle of the colour-specific gradient under that pixel. Assuming that C is a set of contour points P_i, the error for the entire contour is computed as a weighted average angular difference, where the weight associated with each contour pixel is related to the gradient magnitude at that pixel:

$$E_k = \frac{\sum_{P_i \in C} \|\mathbf{g}(P_i)\| (\angle \mathbf{g}(P_i) - \theta_{P_i})}{\Delta\theta_{max} \sum_{P_i \in C} \|\mathbf{g}(P_i)\|} , \tag{4.19}$$

where θ_{P_i} denotes the slope of the hypothetical edge to which point P_i belongs and $\Delta\theta_{max} = 180°$ is the maximum possible angular difference. The conditional probability $p(\mathbf{y}|\mathbf{s}_k)$ needed for updating the particles' weights is modelled as:

$$p(\mathbf{y}|\mathbf{s}_k) = \lambda e^{-\lambda E_k} \tag{4.20}$$

for some constant λ.

Note that a similar weight update scheme can be adopted with the boosted classifier cascade serving as a sign detector. However, in that case the entire image region under the hypothesised sign (not just its characteristic boundary points) would have to be transformed through the scaling matrix $\mathbf{S}_k$. Moreover, on the transformed portion of the image the cascaded classifier would have to be evaluated to yield a soft decision confidence measure introduced in equation (3.17). This measure is associated with the observation likelihood and hence, after a proper normalisation, it could finally substitute $p(\mathbf{y}|\mathbf{s}_k)$ in weight update equation (4.20).

The entire sequence of actions taken in a single step of the tracker incorporating the proposed particle filter is shown in Algorithm 6. Note that: 1) the ultimate state estimate at time $t+1$ is given by the posterior KF state $\hat{\mathbf{s}}^{+}_{t+1}$, 2) an additional resampling step is performed when the effective number of particles falls below a predefined threshold. This prevents degeneracy of the algorithm, i.e. domination of a single particle with nearly unit weight over all other particles with nearly zero weights.

4.2.3 Evaluation

The combined Kalman Filter-Particle Filter road sign tracker has been implemented and integrated with the regular polygon detectors discussed in Section 3.3.1. To observe how this tracker performs in the realistic traffic scenarios, we tested it on nearly 200 short video clips recorded using a front-looking DV camcorder mounted right in front of the car's windscreen. In this experiment the regular polygon detector was set to capture the initial candidate signs of radius in between 15 and 25 pixels. The tracker was intended to take over the detection process upon establishing each of these initial candidates. The tracking was carried out until the target reached the boundary of the visible fragment of the scene. In Figure 4.2 several image sequences illustrating the tracking process are shown.

In a majority of test sequences the accuracy of the tracker was high. In the cases where there is non-zero out-of-plane rotation of the target, the regular polygon detector alone may not be able to fit an accurate contour to the observed road sign. This problem is caused by the fact that the shape of a distorted sign is no longer radial-symmetric. As a result, the detector may produce only partially correct fits. For example, only 3 out of 4 sides of an apparent rectangle may be well-matched by a square detector. Particle Filter-based anisotropic scale correction proposed in Section 4.2.2 successfully resolves this problem and proves to be very useful in most traffic situations. In addition, it does not heavily degrade the computational performance of the tracker. It is so because searching for the values of only two scale parameters within a very small value range requires relatively few particles.

It is important to note that the proposed tracker also behaves reasonably well in presence of occlusions. Specifically, minor partial occlusions are addressed at the level of the detector which can still capture the occluded regular shapes, but appearing with lower strength in the vote space. More severe partial occlusions or full occlusions lasting for several frames are addressed by exploiting the predictive capability of the Kalman Filter. Namely, the KF state and the local search region are projected from time t to $t+1$. The detector is then run within this region. If, due to temporary occlusion or any other reason, the new shape cannot be detected at time $t+1$, the state estimate is simply not updated and the Particle Filter step is omitted. We set the maximum number of frames for which the position and the scale of the tracked sign cannot be measured and hence they are only estimated using the time update equations of the KF. Obviously, during that time the error of the state estimate grows from frame to frame, but the track is not lost. This increases the chance of re-capturing the sign in the first frame where it again becomes visible.

Algorithm 6 Single step of the combined Kalman Filter-Particle Filter tracker.

```
input: posterior Kalman Filter state estimate $\hat{\mathbf{x}}_t$ and error covariance matrix $\mathbf{P}_t$ at time $t$,posterior Particle Filter state estimate at time $t$, $\mathbf{s}_{k,t}$, $k = 1, \ldots, N$, where $N$ is the number of particles, image $I_{t+1}$ at time $t+1$
output: posterior Kalman Filter state estimate $\hat{\mathbf{x}}_{t+1}$ and error covariance matrix $\mathbf{P}_{t+1}$ at time $t+1$, posterior Particle Filter state estimate at time $t+1$, $\mathbf{s}_{t+1}$
using $\hat{\mathbf{x}}_t$ and $\mathbf{P}_t$, predict prior KF state $\hat{\mathbf{x}}^-_{t+1}$ and prior KF error covariance matrix $\mathbf{P}^-_{t+1}$ at time $t+1$ using the KF time update equations (4.4)
based on $\hat{\mathbf{x}}^-_{t+1}$ and $\mathbf{P}^-_{t+1}$, determine the local search region for the detector at time $t+1$, $R_{t+1}$
run the detector in region $R_{t+1}$ of $I_{t+1}$ to obtain the suboptimal measurement $\tilde{\mathbf{x}}_{t+1}$ of the location and scale of the sign
for $k = 1$ to $N$ do
    draw particle $\mathbf{s}_{k,t+1}$ from the proposal distribution $\mathbf{s}_{k,t+1} \sim p(\mathbf{s}_{k,t+1}|\mathbf{s}_{k,t})$
end for
for $k = 1$ to $N$ do
    based on $\tilde{\mathbf{x}}_{t+1}$ and the previous scale estimate, $\mathbf{s}_{k,t}$, re-scale the detected sign and compute $p(\mathbf{y}_{t+1}|\mathbf{s}_{k,t+1})$ using equations (4.19) and (4.20)
    update weight of the particle $\mathbf{s}_{k,t+1}$
        $$\hat{w}_{k,t+1} = w_{k,t} \frac{p(\mathbf{y}_{t+1}|\mathbf{s}_{k,t+1}) p(\mathbf{s}_{k,t+1}|\mathbf{s}_{k,t})}{\pi(\mathbf{s}_{k,t+1}|\mathbf{s}_{k,1,\ldots,t}, \mathbf{y}_{k,1,\ldots,t+1})} = w_{k,t} p(\mathbf{y}_{t+1}|\mathbf{s}_{k,t+1})$$
end for
for $k = 1$ to $N$ do
    normalise weight of the particle $\mathbf{s}_{k,t+1}$
        $$w_{k,t+1} = \frac{\hat{w}_{k,t+1}}{\sum_{l=1}^{N} \hat{w}_{l,t+1}}$$
end for
obtain the maximum likelihood estimate of the PF state at time $t+1$:
    $$\hat{\mathbf{s}}_{t+1} = \sum_{k=1}^{N} w_{k,t+1} \mathbf{s}_{k,t+1}$$
compute an estimate of the effective number of particles:
    $$\hat{N}_{eff} = \frac{1}{\sum_{k=1}^{N} (w_{k,t+1})^2}$$
if $\hat{N}_{eff} < N_{thres}$ then
    draw $N$ particles from the current particle set with probabilities proportional to their weights and replace the old set with the new set
    for $k = 1$ to $N$ do
        set weight of the particle $\mathbf{s}_{k,t+1}$ to $w_{k,t+1} = \frac{1}{N}$
    end for
end if
use centroid location information from $\tilde{\mathbf{x}}_{t+1}$, ML Particle Filter's state estimate $\hat{\mathbf{s}}_{t+1}$, and the KF measurement update equations (4.5) to estimate the posterior KF state $\hat{\mathbf{x}}^+_{t+1}$ and posterior KF covariance matrix $\mathbf{P}^+_{t+1}$ at time $t+1$
```

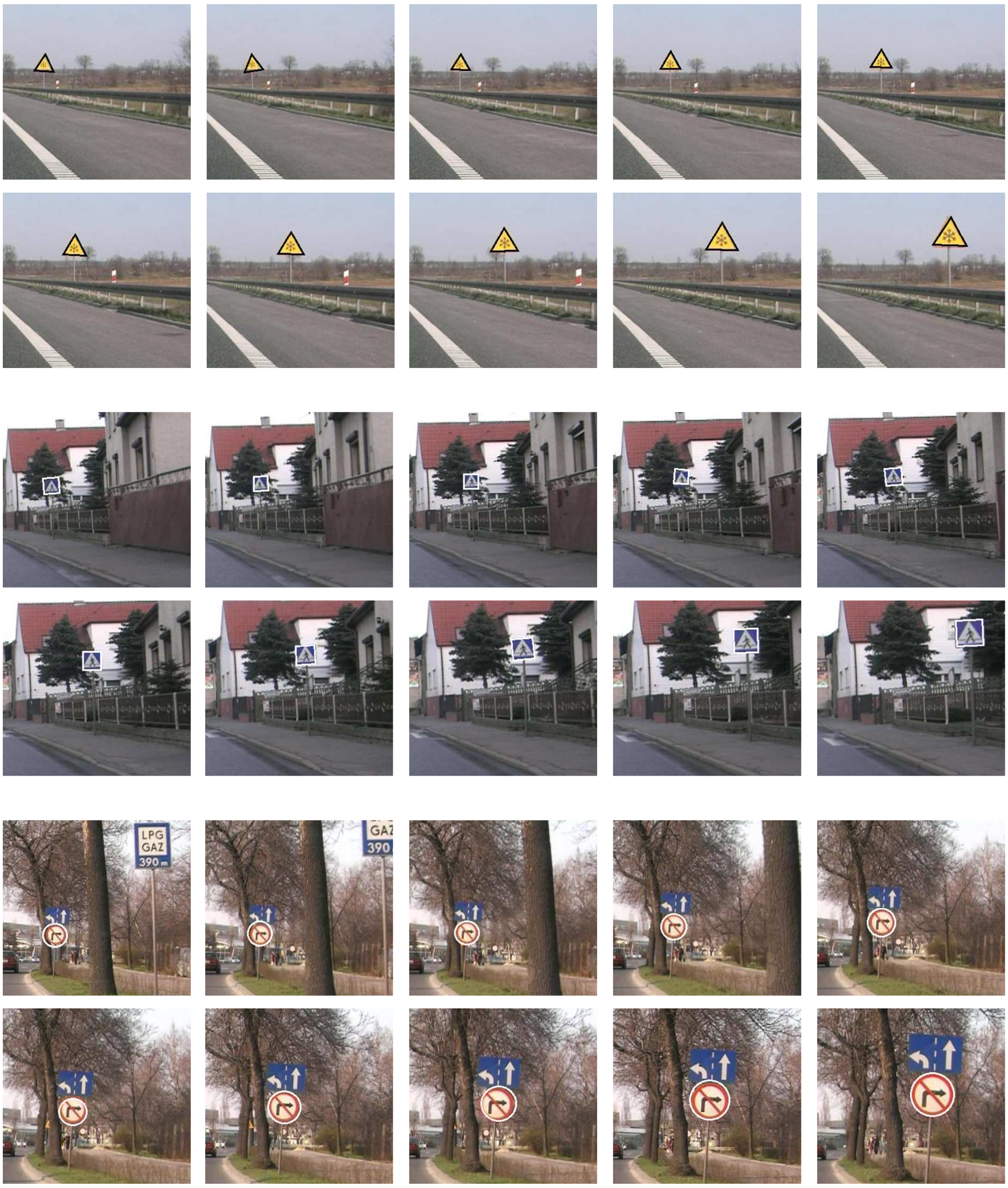


Figure 4.2: Three example image sequences illustrating the output of the KF-PF tracker. In each image the contour of a sign is marked black (first sequence) or white (second and third sequence). Every fourth frame from the original sequences is shown.

4.3 Contour Tracking

In section 4.2 a combined Kalman Filter-Particle Filter road sign tracker was introduced. In the typical traffic situations this tracker performs reasonably well as it generates accurate contour estimations, partially handles the affine sign distortion, and can recover from temporary occlusions. One shortcoming of this approach is that it merely employs the Kalman Filter and the Particle Filter to generate hypotheses on the geometry of the target, but these filters do not directly affect the observation process. This is best illustrated in the KF measurement equation matrix, which is identity. In consequence, the tracker is exposed to the same problems as the detector is. For example, if the input edge/gradient information is of low quality, or is contaminated with the accidental edges/gradients coming from the background clutter, the accuracy of tracker may significantly degrade.

The other, sometimes particularly severe problem is related to the specific properties of traffic signs. Namely, the colour layout and the pictogram symbols may be arranged in such a way that smaller regular polygons are formed inside the outer boundary, which is of primary interest to the detector and the tracker. As high-contrasting colours are dominating in the inner part of traffic signs, these extra undesired polygons are likely to receive more votes in the Hough space than the shape delimiting the entire sign. To support this claim, several examples of road signs containing self-similar regular shape structures have been illustrated in Figure 4.3.

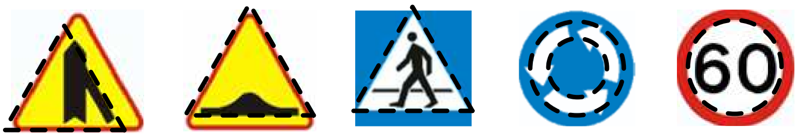


Figure 4.3: Examples of traffic signs where pictograms constitute extra regular shapes that are likely to confuse the circle/regular polygon detector. These shapes are marked with black dashed line.

In this section we introduce a robust contour tracking algorithm that overcomes the abovementioned limitations. It is based on the *Pixel Relevance Model* (PRM), introduced in Section 4.3.1, where the pixel relevance is defined as a confidence measure for a pixel being part of a sign's contour. The relevance of the hypothesised contour pixels is updated dynamically, according to a spatio-temporal voting scheme and within a small search region maintained by the Kalman Filter, which ensures fast computation. The map of pixel relevance is propagated over time such that the pixels likely belonging to the boundary of the tracked sign are amplified relative to those lying inside and outside of it. Such a contour-belonginess map serves as an input to the regular polygon detectors discussed in Section 3.3.1. The details of this algorithm are given in Section 4.3.2. The tracker incorporating the *Pixel Relevance Model* becomes robust to both background clutter, common in the urban traffic scenes, and the presence of misleading edges in the inner part of a sign's plate. We discuss these advantages in Section 4.3.3.

4.3.1 Pixel Relevance Model (PRM)

In this section the semantics of the *Pixel Relevance Model* is outlined. We call a pixel x_{ij} relevant if it belongs to the contour of a road sign and irrelevant otherwise. Formally, relevance r_{ij} of pixel x_{ij} is defined as a real number between 0 and 1, i.e. x_{ij} is completely irrelevant when $r_{ij} = 0$ and completely relevant when $r_{ij} = 1$. As the signs move through the scene and grow in size while being approached by the vehicle, relevance must change over time as well. The dynamics of this process is modelled using a spatio-temporal voting graph, fragment of which is shown in Figure 4.4. The graph encapsulates the relevance distribution, $\mathbf{r}(t)$, over an image region at time t in a set of state nodes. Evolution of a single pixel's relevance is assumed to be a first order stationary Markov process and the supposingly weak correlations between the same-slice pixels are ignored. Relevance of each pixel x_{ij} at time t, $r_{ij}(t)$, is dependent on the relevance of its neighbourhood in the previous frame, $r_{N(i,j)}(t-1) = \frac{1}{n}\sum_{x_{kl}\in N(i,j)} r_{kl}(t-1)$ (where n is the size of the neighbourhood), and the observable feature at time t, $f(x_{ij}(t))$.

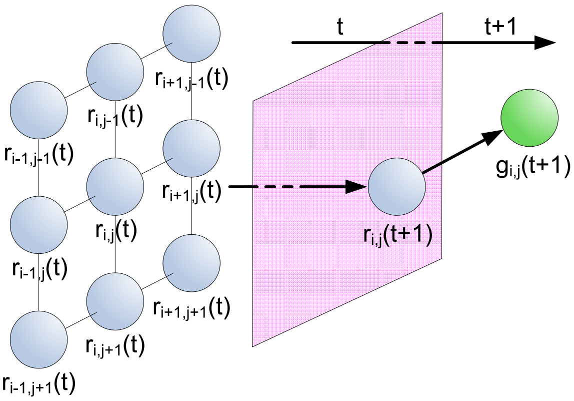


Figure 4.4: Fragment of the spatio-temporal voting graph structure used to model the dynamics of pixel relevance. Consecutive time slices are shown.

Our state transition model is defined by the following function supported on a $[0,1]$ interval:

$$\phi_{ij}(t) \sim \left(1 - e^{-k_T r_{N(i,j)}(t-1)}\right) \tag{4.21}$$

for some constant k_T. We are postponing the more precise definition of the transition function to Section 4.3.2. Relevance projected from time slice t to $t+1$ is further conditioned on the observed feature at time $t+1$, and this update process is defined by the same class of functions:

$$\psi_{ij}(t) = 1 - e^{-k_O f(x_{ij}(t))} \tag{4.22}$$

for another parameter k_O. Using the above definitions, evolution of the pixel relevance can be expressed as:

$$\begin{array}{l} r_{ij}^{-}(t+1) = r_{N(i,j)}(t)\phi_{ij}(t+1) \\ r_{ij}^{+}(t+1) = r_{ij}^{-}(t+1)\psi_{ij}(t+1) \end{array} . \tag{4.23}$$

In Section 4.3.2 we discuss how the above defined *Pixel Relevance Model* is integrated with the tracking scheme introduced in Section 4.2.

4.3.2 PRM-based Tracking

Our regular polygon detector is triggered every fixed number of frames to capture new candidates emerging in the scene. In each region of interest found using the focus operator introduced in Section 3.2.2, the colour-specific gradient maps are obtained, as described in Section 3.3.1. Extracted colour gradient maps provide the sufficient data for initialisation of our Pixel Relevance Model. Specifically, we consider the measured colour gradient magnitude at pixel x_{ij} to be an observable symptom of the unknown pixel relevance, and use it to initialise the first estimate of the state, i.e. $r_{ij}(0) = g_{ij}$. In this initial frame, where the candidate sign is for the first time observed, as well in all subsequent frames, the pixel relevance information will be consistently exploited by the regular polygon detector and in the particle re-weighting process (refer to Section 4.2.2 for details).

The road sign tracker incorporating the *Pixel Relevance Model* uses the Kalman Filter to maintain a localised search region around the expected position of the candidate sign. On the other hand, PRM is used to temporally update a belief on the relevance of the pixels contained in this region. Then, the appropriate regular shape detector is run on the current colour-specific edge, directional gradient, and posterior pixel relevance maps extracted within the interest region, so as to generate the temporarily best-matching contour of a sign being tracked. Next, the Particle Filter is employed to correct the out-of-plane rotation of the detected sign using the appropriately modified shape error formula (4.19). Finally, the KF update step is performed upon correcting the scaling factors of the candidate shape at time $t + 1$. A block diagram illustrating this process is provided in Figure 4.5 and the details are given below.

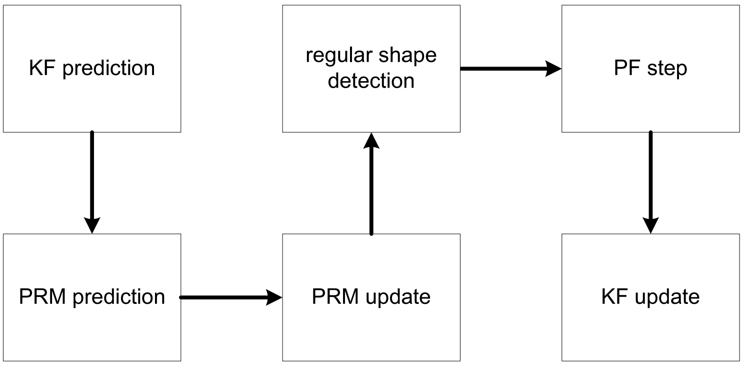


Figure 4.5: Block diagram of the road sign tracker incorporating the Kalman Filter, the Particle Filter, and the *Pixel Relevance Model*.

The most interesting feature of the PRM-based tracker is that the prior KF state estimate can be used to modulate the relevance of the pixels in the current search region. It is done by highlighting the pixels that lie on the predicted, motion-compensated contour of a sign or in its vicinity, and diminishing the importance of the remaining pixels. As a result, the unwanted edges inside and around the sign being tracked become suppressed and the more accurate shape fits can be obtained.

Knowledge of the previous-frame and the current-frame Kalman Filter state estimates provides the necessary data for the abovementioned motion compensation. Specifically, the apparent inter-frame velocity and scale change of the target in the image plane are approximated by the differences between the appropriate centroid coordinates and radii of the posterior shape at time t and the prior shape at time $t+1$. Consequently, the pixel relevance map from time t, together with the hypothesised shape with centroid at $(x(t), y(t))$ and radii $r_x(t)$, $r_y(t)$, are shifted by the found motion vector. Next, a distance transform (DT) [16] is computed in the current search region for the contour of this motion-compensated shape. The previous-frame pixel relevance information and the obtained DT map are used to obtain the new prior pixel relevance map at time $t+1$, according to the equation:

$$\phi_{ij}(t+1) = e^{-k_D d_{DT}(i,j)} \left(1 - e^{-k_T r_{N(i,j)}(t)}\right) \quad , \tag{4.24}$$

where $d_{DT}(i,j)$ denotes the appropriate distance value picked from the DT image, $k_D \sim \frac{1}{E^2}$ and E is the average of the prior variances of the KF state parameters at time $t+1$. The posterior relevance of the pixels contained in the search region is obtained using equation (4.22) via incorporating the magnitudes of the observed gradients, which act as observable features $f(x_{ij}(t))$.

4.3.3 Discussion

To better understand the advantage of the *Pixel Relevance Model*, several points need to be emphasised. First, the posterior pixel relevance map is passed on input of both the regular shape detectors and the Particle Filter. In both cases it plays the same role as the gradient magnitude information that would otherwise be used. Specifically, the Hough-based detectors use the relevance of the pixels to determine the strength of the votes cast for each potential road sign's centroid. It replaces the magnitude of the n-angle gradient vector in equation 3.7. The Particle Filter incorporates the pixel relevance values in the update step (line 9 of Algorithm 6), in equation 4.19, in which case they substitute the magnitude of the gradient elements along the tested sign's boundary. Using the pixel relevance instead of the raw gradient information means that apart from the current-frame information, also the past observations, encompassed by the spatio-temporal graph structure from Figure 4.4, are incorporated by the tracker. The effect of this replacement is a consistently better localisation of traffic signs in the consecutive video frames, particularly in the cluttered street scenes.

Secondly, the role of the Distance Transform in the above described tracking framework is very important. Note in (4.24) that the distance term d_{DT} effectively suppresses the estimated relevance of the pixels that are far from the predicted sign's contour. This relevance decreases exponentially with the increasing value of d_{DT}. It has a desired effect of attenuating the undesired gradients inside and outside of the sign's boundary, as shown in Figure 4.6. These gradients would otherwise increase the risk of inaccurate fitting of a regular shape's contour to the observed traffic signs, especially one of those shown in Figure 4.3.

The other point that requires attention is the additional role of the Kalman Filter – on-the-fly adjustment of the *Pixel Relevance Model* parameters that control the relevance transition function $\phi_{ij}(t)$. For a very accurately estimated KF search region, the pixel relevance is peaked at the expected contour of a sign and the unwanted edges, that with high probability are part of the sign's interior or cluttered background, become suppressed, as in Figure 4.6d. This in turn precisely directs the focus of the shape detector to the sign's centroid. However, when the uncertainty on the Kalman Filter state is high, our confidence of having captured the sign area well is lower accordingly. In this case the DT modulation factor is relaxed and the relevance model is to a larger extent allowed to "evolve on its own". Consequently, the relevance image is much more blurry and better reflects the currently observed gradients. This gives the detector an opportunity to recover from a likely poor contour fit.

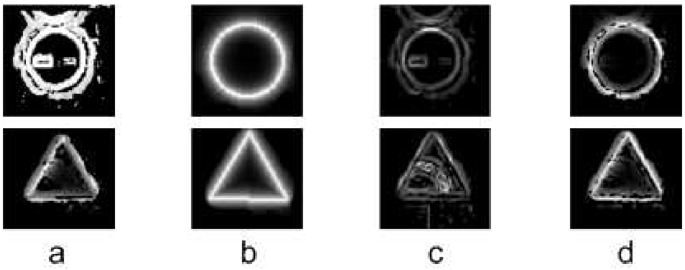


Figure 4.6: Different stages of pixel relevance processing in a single video frame around a tracked candidate road sign: a) relevance map at time t, b) prior relevance map at time $t+1$ (after KF-regularised projection), c) gradient magnitude map at time $t+1$ (observable evidence), d) posterior relevance map (after incorporating the gradient measurements). In all images the intensity is scaled to the range $[0, 1]$ for better visualisation.

Figure 4.7 shows examples of the two realistic image sequences where a sign is being tracked over time. Each upper row of images illustrates the momentary gradient magnitude map, whereas the images in each lower row depict the corresponding posterior pixel relevance maps. It can be noticed that in the pixel relevance images the signs' contours are clearly emphasised, but the other high-contrasting image regions, including the pictograms, tend to be suppressed. Thanks to the accurate motion compensation of the pixel relevance map provided by the Kalman Filter, this technique is very useful in combination with the regular shape detector based on the algorithm of Barnes et al. [128]. As the latter focuses on the contour of the object being tracked, possibility of inaccurate sign detection in each frame of the input video is greatly reduced.

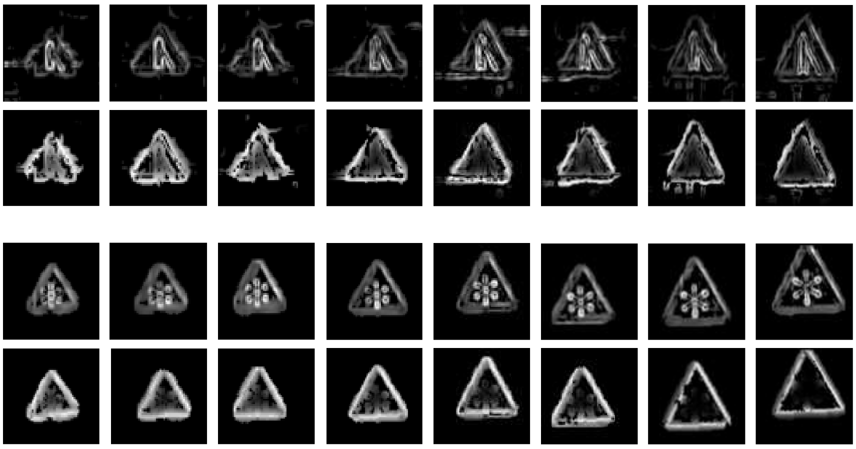


Figure 4.7: Example sequences of a traffic sign being tracked, from the perspective of gradient magnitude (upper rows) and pixel relevance (lower rows). While the gradient maps contain all high-magnitude regions, in the pixel relevance images the non-contour peaks of the gradient tend to be attenuated. Each image corresponds to the local search region at a given time point and for illustration purpose these images are scaled to equal size.

4.4 Affine Tracking

In majority of previous studies on TSR, the common assumption was made that the traffic signs ideally face the camera and hence appear unaffected by the perspective distortion. In most traffic situations it is desirable to detect and recognise the sign when its distance from the incoming vehicle is relatively large. This admits early reaction to the traffic condition described by the sign's pictogram and hence increases security of the human driver. Since signs seen at a distance usually indeed appear nearly front-looking, unless they are not physically dislocated (e.g. by vandals), their geometrical deformation can be considered avoidably small. This justifies usage of the radial symmetry-based detectors, such as those discussed in Section 3.3.1.

Recognising traffic signs at a distance has also disadvantages. When the sign is far away from the camera, it appears small in the image, unless a telephoto lens is used. Therefore, its pictogram in low resolution might appear noisy, blurry and, most of all, less discriminative. This reduces the confidence of the decision made by the road sign classifier. From this perspective, it is more adequate to classify the signs based on the observations made when the target is close to the camera. However, at this point the perspective distortion is often considerable and can no longer be neglected. Several pictures of traffic signs taken at small distance from the target are shown in Figure 4.8. As seen, such signs cannot be considered ideal circles or regular polygons, but rather ellipses and more general triangles/quadrangles. This calls for an appropriate temporal modelling of the affine geometry which would enable viewpoint-invariant sign detection and recognition.

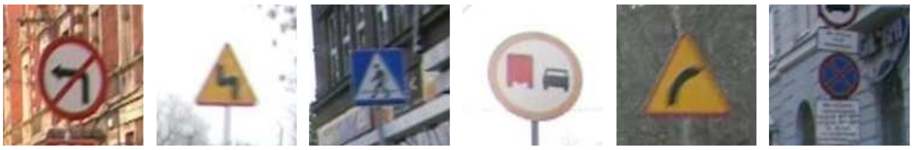


Figure 4.8: Examples of traffic signs that look geometrically distorted in the image plane and therefore cannot be considered perfect circles or regular polygons. All pictures were taken from a moving vehicle at a small distance from the target.

In the following sections the affine tracking model is developed. This approach has been adopted in several recent studies, e.g. by Bayro-Corrochano and Ortegón-Aguilar [150] or Tuzel et al. [156], but we are pioneering in applying it to road sign tracking. It poses the monocular object tracking as a regression problem where regression is used, based on the Lie group theory, to construct a robust function encoding the relationships between a unique feature representation of the target object and the affine distortions it is subject to in the image plane. We show that this function can be learned on-the-fly from random affine transformations applied to the image of the object in known pose. Secondly, we demonstrate its capability of reconstructing the full-face view of a sign seen from varying viewpoints. In Section 4.4.1 the theoretical fundamentals of the approach are given. Section 4.4.2 discusses the concrete realisation of the regression tracker in the TSR context. Discussion and experimental evaluation of the proposed affine tracking algorithm is presented in Section 4.4.3.

4.4.1 Tracking as a Regression Problem

Let $\mathbf{M}$ be an affine matrix that transforms a unit square at the origin in the object coordinates to the affine region enclosing the target object in the image coordinates:

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \ , \tag{4.25}$$

where $\mathbf{A}$ is a 2×2 nonsingular matrix and $\mathbf{t} \in \mathbb{R}^2$. Let $\mathbf{M}^{-1}$ be an inverse transform, that maps the object region from image coordinates back to the object coordinates, as shown in Figure 4.9. Our goal is to estimate the transformation matrix $\mathbf{M}_t$ at time t, given the observed images up to that point, $I_{0,\ldots,t}$, and the initial transformation $\mathbf{M}_0$. $\mathbf{M}_t$ is modelled recursively:

$$\mathbf{M}_t = \mathbf{M}_{t-1} \Delta \mathbf{M}_t \ , \tag{4.26}$$

which means that it is sufficient to estimate only the increment $\Delta\mathbf{M}_t$ corresponding to the inter-frame motion of the target from time $t-1$ to t in object coordinates. It is determined by the regression function f:

$$\Delta \mathbf{M}_t = f\left(\mathbf{o}_t(\mathbf{M}_{t-1}^{-1})\right) \ , \tag{4.27}$$

where $\mathbf{o}_t(\mathbf{M}_{t-1}^{-1})$ denotes an m-dimensional image descriptor applied to the previously observed image, after mapping it to the unit rectangle. The choice of this descriptor is postponed to Section 4.4.2.

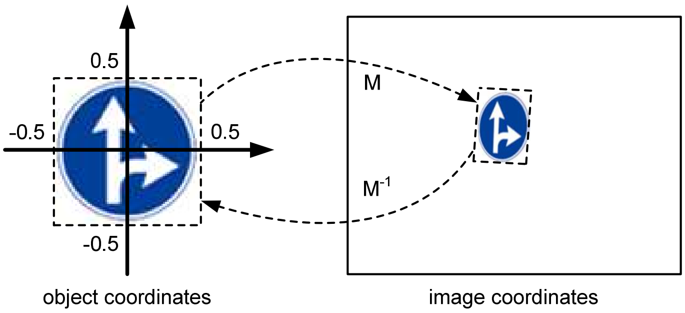


Figure 4.9: Affine transformation matrix and its inverse.

The regression function $f : \mathbb{R}^m \longmapsto A(2)$ is an affine, matrix-valued function, where $A(2)$ denotes a two-dimensional affine transformation. To learn its parameters, it is necessary to know the initial pose of an object, $\mathbf{M}_0$, and the image I_0 at time t_0. Training examples are generated as pairs $(\mathbf{o}_0^i, \Delta\mathbf{M}_i)_{i=1,\ldots,n}$, where $\Delta\mathbf{M}_i$ are random deformation matrices around identity, $\mathbf{o}_0^i = \mathbf{o}_0(\Delta\mathbf{M}_i^{-1}\mathbf{M}_0^{-1})$, and the number of examples is smaller than the dimensionality of the image descriptor used, i.e. $n < m$. In other words, for each training example the image coordinates of the object are multiplied on the left by $\Delta\mathbf{M}_i^{-1}$ and a new descriptor is computed in the resulting image. The optimal parameters of f are derived on the grounds of the Lie group theory by minimising the sum of the squared geodesic distances between the pairs of motion matrices: estimated $f(\mathbf{o}_0^i)$, and known $\Delta\mathbf{M}_i$. An outline of the training process, repeating the derivations of Tuzel et al. [156], is given in Appendix A.

4.4.2 Affine Tracker Architecture

Figure 4.10 illustrates how the regression tracker introduced in section 4.4.1 is utilised as a part of the entire traffic sign recognition system. Once a candidate sign has been detected for the first time, a new tracker is initialised with the region corresponding to the bounding rectangle of the found regular shape instance, assuming no distortion. This assumption is valid as road signs are detected for the first time at a considerable distance from the camera, where this distance is much greater than the distance of the sign from the camera's optical axis. The only exception is when the traffic sign is physically dislocated, e.g. rotated around the axis going through the pole on which it is mounted, or tilted to the ground. At this point a small number of random affine deformations are generated from the observed image and used for instant training [2]. A map of 6 × 6 regularly-spaced 6-bin gradient orientation histograms is used as an object descriptor. In practice, to minimise the contamination of the unit square to which the target image region is mapped by the

[2] $n = 50$ random training deformations were used in the experimental evaluation of our affine tracker.

peripheral background pixels, this descriptor is computed over the area shrunk by a several-pixel margin on each side.

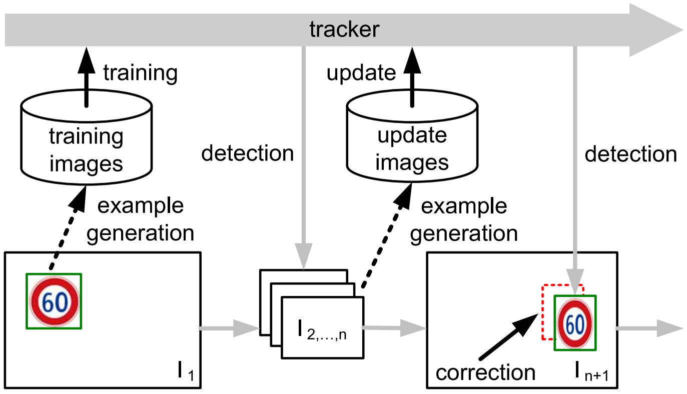


Figure 4.10: Operation of the affine road sign tracker over time. The period between the initial candidate detection and the first tracker update is depicted.

In a realistic traffic scenario the scene is often difficult and changes fast. Therefore, the accuracy of the tracker is likely to deteriorate very quickly. It is a result of the cumulated reconstruction errors caused by: 1) the aforementioned background contamination of the image in object coordinates, and 2) changing appearance of the target. To deal with this problem, we update the tracking function after every n frames of the input video. It means that the just-trained tracker is employed to detect the sign in n subsequent frames, each being used to generate and enqueue m new random deformations. Then, in the $n+1$-th frame the tracker is re-trained on the collected portion of $n \cdot m$ random training transformations.

The above update process is carried out in a similar way as the initial training, i.e. by minimising the sum of the squared geodesic distances between the estimated and the known, randomly generated motion matrices. However, another constraint is introduced on the difference between the current and the previous regression coefficients. It prevents function f from changing its parameters too much and hence ensures smooth tracking after the update. For details, refer to Appendix A. The re-trained tracker is used to re-estimate the pose of the observed sign and the space is allocated for a new portion of $n \cdot m$ random transformations.

Two implementation details require clarification. First, the optimal values of the regularisation parameters λ and γ described in Appendix A have been determined experimentally for the chosen feature representation of the candidate region in object coordinates, based on a separate set of training synthetic image sequences. These values are $\lambda = 0.02$, $\gamma = 0.05$. The abovementioned sequences were generated in the OpenGL framework [164]. In each such sequence a template image of one sign is shown in an empty 3D scene. The consecutive images depict the sign getting closer to the virtual camera and hence increasingly distorted. This simulates a realistic scenario of a car approaching a road sign mounted on the side of the road or above the road lane. The scenes were deliberately constructed without any background and with constant illumination so as to minimise the effects of possible contamination of the image regions enclosing the target and to ensure its consistent appearance. By learning λ and γ this way, we wanted to avoid potential bias that could be introduced to the tracker if it was trained using a limited number of real-life sequences recorded in specific traffic situations, and at specific lightning and weather conditions.

Second, we check the correlation-based condition to determine when to consider the track lost. Specifically, it takes place when the sign either gets out of the field of view of the camera, or when the normalised cross-correlation (NCC) between its image in object coordinates and the same image recorded at the time of last update falls below a predefined threshold. The longest acceptable period for which the correlation stays below this threshold is extended to several frames to prevent instant track losing due to the short-term occlusions. Also, note that an occluded target may confuse the normally operating tracker, i.e. destroy the structure of the affine motion by making the learned tracking function yield unpredictable values. To avoid this, the last motion matrix estimate, recorded before the NCC fell below the acceptable minimum, is restored and used. In other words, the last stable motion is frozen.

4.4.3 Experiments

We have conducted an experiment aimed at evaluating the ability of the road sign tracker presented in Sections 4.4.1 and 4.4.2 to retrieve the full-face view of a sign under affine transformations. This experiment was done in the following way. Five synthetic image sequences were prepared using the OpenGL framework [164] in the same way as for the sequences used to determine the regression regularisation parameters λ and γ (see Section 4.4.2 for details). One of these sequences is shown in Figure 4.12. To simplify the detection step, only the circular signs were considered. For each image sequence the circular Hough detector refined by the *Confidence-Weighted Mean Shift* clustering, introduced in Section 3.4, was set to capture the circle instances of radius in between 12 and 24 pixels. The tracker was triggered at the time of initial detection of a sign and updated every $n = 15$ frames. Upon the initial detection, the nearly undistorted image of a sign in gray-scale was recorded to serve as a reference image.

Robustness of the on-line learned tracking function to the affine distortions was measured by recording a normalised cross-correlation between the reconstructed full-face view of a sign in each frame and the reference image. The changes of this correlation over time for all five sequences are shown in Figure 4.11. In each plot the behaviour of NCC for a 6D regression function encoding all six 2D affine transform parameters is compared to the behaviour of NCC observed using three other trackers. These are: 1) a 4D regression function encoding only two rotation-shift parameters and both translation parameters, 2) a 3D regression function encoding only one isotropic scaling-rotation parameter and both translation parameters, and 3) a simple tracker which makes independent circle detections in each frame, but uses a Kalman filter (KF) [1], as described in Section 4.2.1, to predict the position and the scale of a sign. During the on-line training of the regression trackers, all non-translation parameters were randomly generated within the range $[-0.2, 0.2]$ and the translation parameters were randomly generated within the range $[-0.4, 0.4]$.

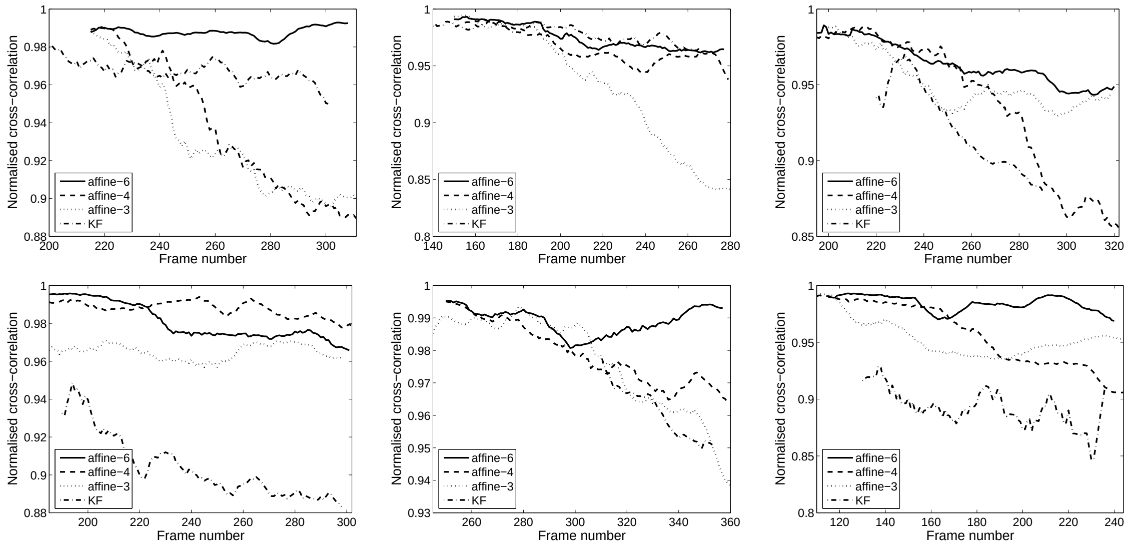


Figure 4.11: Normalised cross-correlation (NCC) between the reference image recorded at the time of initial sign detection and the reconstructed full-face view of a sign in each subsequent frame of the input sequences. Each sequence was generated in a synthetic empty 3D scene and simulates the typically sign distortions observed from a moving vehicle.

Based on the results of the above experiment, we conclude that learning the motion model based on the Lie algebra enables construction of a robust road sign tracker which is invariant to the affine transformations. In Figure 4.11 the 6D affine tracker outperforms the two other regression trackers and the KF-based tracker, which do not model the full structure of the motion. The correlation between the original frontal view of a sign and a view inferred from the current transformation parameter estimates remains high for the entire duration of the sequences. In the case of the 4D and 3D affine trackers, as well as the KF-based tracker, this correlation drops more quickly, particularly in the second part of each sequence. Note that because the model signs were used in this experiment and the scene was background-free and constant-illuminated, nearly entire contribution to the NCC decrease is from the cumulated errors of the tracker.

In another experiment the affine regression tracker was tested on real-life videos captured from a moving vehicle in crowded Japanese street scenes. The illumination of the scene was roughly constant in all test videos. In system runtime, a 720 × 540 pixels portion of the scene was cropped from the upper-central region of each frame of the input video, and further downscaled by 50%. The range of radii of the circles captured by the detector was again set to 12-24 pixels and the tracker updated itself every $n = 15$ frames, generating $m = 6$ new random affine transformations in each frame. The ranges of values of the random affine transformation matrix components were also the same as previously. In this experiment the tracker demonstrated its ability to rapidly correct small affine sign distortions, which enabled robust, frame-rate sign recognition at a later stage of the processing. Example videos demonstrating this ability are available as a supplementary material accompanying this dissertation. Images from two example sequences that illustrate the output of the tracker are shown in Figure 4.12.

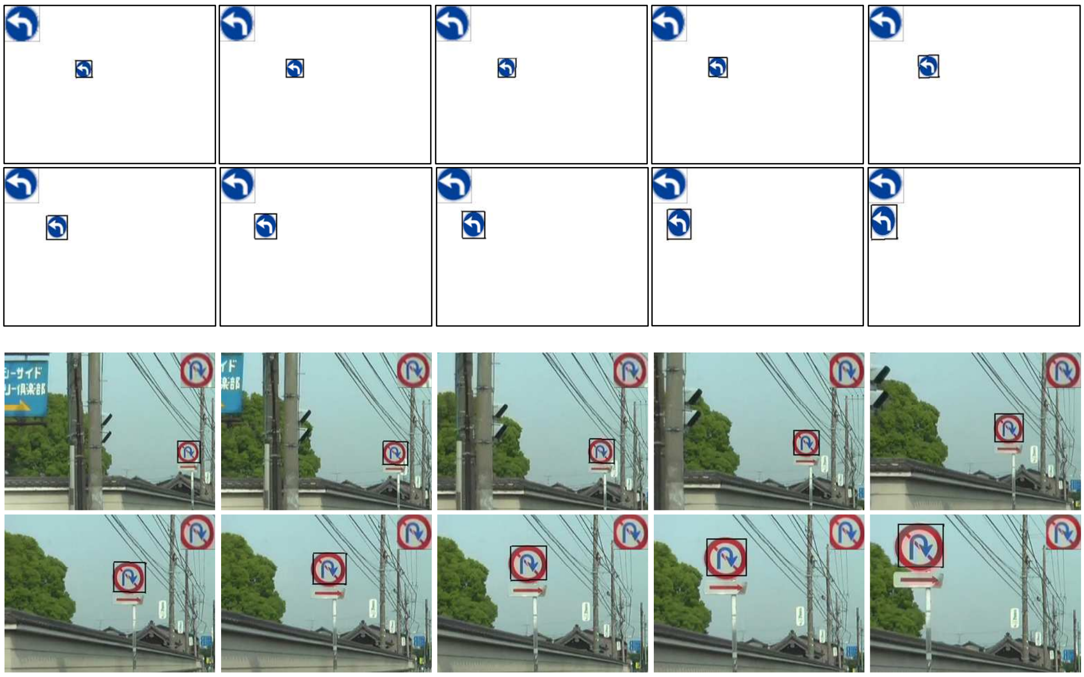


Figure 4.12: Example sequences illustrating the operation of the affine road sign tracker. The first sequence (every 11th frame is shown) was generated in OpenGL and illustrates a model sign tracked in a background-free scene. The second sequence (every 7th frame is shown) was recorded with a camera mounted on board of a moving vehicle. In each image a black frame is drawn around the sign in the currently estimated pose. Besides, in the corner of each image the reconstructed frontal view of a sign is shown.

To sum up, as long as the traffic sign is far away from the camera and hence remains relatively unaffected by the affine distortion, all tested methods provide a satisfactorily accurate track of the target. However, when the sign gets closer to the camera and thus becomes more substantially distorted in the image plane, only the fully-affine regression tracker remains capable of restoring the full-face view of the target with low error. From the point of view of the entire system this is a particularly useful property because the most informative frames of the input video, when the appearance of a sign is the least ambiguous, can be efficiently used for recognition. Note that even when only the translation components of the affine transformation matrix are non-zero, i.e. when the current location estimate of a sign is simply offset relative to its true position, the learned regression function compensates this offset in exactly the same way as any other, more complex affine deformation.

Finally, it is important to emphasise the critical role of the accuracy of initial candidate sign detection, which is a bottleneck of the affine tracker. As the regression function is learned in system runtime based on this initially detected instance, even small errors made in the position and scale estimation at this point may spoil the entire track afterwards. Such errors will be simply propagated over time because in subsequent updates the tracking function will be learned from an inaccurately cropped sign image contaminated with the background fragments. To minimise this effect, we do not record a new candidate sign right after it has been captured by the detector. Instead, a localised detection is repeated in the subsequent frames, assuming no affine distortion, until a decent stability of the captured shape's location and scale estimates is reached. Namely, the confidence of the detector's response, expressed in the way discussed in Section 3.4.1, for k consecutive frames must not change by more than a predefined threshold before a new affine tracker is initialised for the newly detected sign instance.

4.5 Summary

In a video-based traffic sign recognition system the target tracker is an important component that improves both detection and recognition. On one hand, road signs themselves do not normally change their appearance over time and their apparent motion in the image plane is slow and hence less challenging than in many other applications, e.g. those requiring modelling of an articulated human motion. This implies that the inter-frame pose changes of traffic signs are minimal and can easily be predicted. However, when the road sign is already at a small distance from the camera, it often becomes more substantially distorted. Moreover, the appearance of traffic signs might change unexpectedly, under the influence of extraordinary factors, such as suddenly changing illumination. All these phenomena call for appropriate temporal modelling. Because such a modelling incorporates past observations in the current state estimate, it provides more accurate track of the target.

The tracking is also required for maintaining an accurate estimate of the sign's pose over time even when the detector fails to capture the target, e.g. due to confusion with the background clutter or short-term occlusion. This prevents the TSR system from accidental track losing and hence helps develop a consistent prediction of the class label at the recognition stage. Finally, the road sign tracker radically reduces the computation required to process a single-frame observation. It is because the object detector is either not needed at all, or it needs to be run only in the local search region, predicted based on the previous location and scale estimates of the target.

Three different contributions to road sign tracking have been made in this chapter. At first, we have proposed a combined Kalman Filter-Particle Filter (KF-PF) tracker that performs reasonably well in a vast majority of realistic traffic situations. The Kalman Filter part is used for general location and scale prediction, as well as the local search region reduction. Non-linearity of the KF prediction equations is solved by adopting the Extended Kalman Filter technique which linearises the problematic equations around the current state estimate. The Particle Filter implements the Sampling Importance Resampling algorithm [21] where each particle's importance is updated based on how well the associated hypothesised shape matches the actual orientations of the gradient in the image. Particle Filter is responsible for anisotropic scale correction and is useful whenever the signs appear distorted enough to no longer look like perfect circles or equiangular polygons. Prototype implementation of the KF-PF tracker has demonstrated good performance on a number of realistic video sequences captured from a real moving vehicle.

An improvement to the abovementioned tracker is aimed at the temporal estimation of the traffic sign's contour, which is in our opinion the most reliable piece of information to be exploited for accurate target detection and recognition. The proposed contour tracker has been integrated with the KF-PF framework which, as previously described, is responsible for sign's centroid and scale prediction as well as for maintaining a localised search region. A *Pixel Relevance Model* has been introduced to update the relevance of the pixels contained in this region over time, where the relevance of a pixel is defined as a likelihood of it belonging to the sign's contour. At the prediction step the pixel relevance map is propagated to the next time point via spatio-temporal voting in the neighbourhood of each pixel. Then, it is motion-compensated, i.e. translated by the inter-frame 2D motion vector found based on the current and the previous Kalman Filter state estimates. In addition, it is smoothed by the distance transform computed around the boundary of a hypothesised motion-compensated sign. This effectively suppresses the uninformative gradient pixels inside and outside of this boundary. Finally, the relevance of the pixels within the interest region is updated based on the magnitude of the currently observed image gradients.

Advantages of pixel relevance are exploited when the Hough-style regular shape detector is run in the local RoI. It no longer uses the raw current-frame gradient information, but votes for the likely sign locations using the pixel relevances as weights. These relevances encompass the information cumulated over a series of past observations and are peaked only along the true sign contours. As a result, the traffic signs can be more accurately detected. Experimental evaluation of the contour tracker showed that this capability is particularly useful in the cluttered urban traffic environments where the risk of confusing a sign's contour with the accidental background objects or with the parts of its own pictogram is high.

Affine motion tracker borrowed from Tuzel et al. [156] has been finally considered as part of a TSR system, which is its first application in this area. This tracker maintains a matrix-valued function of the observation vector, evaluated using learned regression coefficients. When multiplied on the left by the previous-frame motion matrix, this function gives an accurate pose estimate of the target in the current frame. The abovementioned regression coefficients encode the relationships between a unique feature representation of the target and the affine transformations it is subject to while being approached by a car-mounted camera. Each such transformation defines a mapping between the sign in the image coordinates and the same sign in the normalised object coordinates. Regression function learning is carried out on the grounds of the Lie algebra, by minimising the sum of the squared geodesic distances between the estimated and the known motion matrices. The latter are randomly generated from the image of a sign in known pose, either arbitrarily assumed frontal (initial detection) or obtained as a result of previous estimations.

The architecture of the affine tracker is designed to address temporal appearance changes of traffic signs, e.g. caused by varying illumination. It is done via periodic re-training of the regression function. In addition, the accidental track losing is prevented by allowing the normalised cross-correlation between the current view of a sign in object coordinates and the same view recorded at the time of last update to stay below a predefined level only for a short period of time. The last reliable motion matrix is restored and used during that period, instead of performing a standard motion estimation with the learned tracking function. It ensures that the structure of the affine motion is not destroyed through the unpredictable behaviour of this function when learnt from the occluded views of the target.

Experimental evaluation of the regression tracker has demonstrated its usefulness in video-based traffic sign recognition. The tracker is fast, accurate and pose-invariant. It models properly the affine transformations of traffic signs, which was checked using natural and synthetic image sequences. Finally, as the parameters of the abovementioned transformations can be reliably estimated, the affine distortion is appropriately compensated in each frame of the input video so as to obtain a full-face view of a sign. This is particularly useful when the sign is already close to the vehicle passing it, and hence carries the most valuable information from the perspective of the classifier.

Chapter 5

Traffic Sign Recognition

Recognition of traffic signs is a hard multi-class problem. In practice, handling the entire gamut of pictograms is never considered in TSR. This would be impractical as the total number of signs is huge, they differ from country to country, and some of them are extremely rare. Therefore, the common approach adopted is to focus on a relatively narrow category of the most relevant signs within one country. This reduces the complexity of the classification task and is hence more suitable for in-vehicle application. Besides, many traffic signs are not standarised in terms of colour, shape and the symbols contained. A good example are the signs giving direction, which may vary in terms of size, shape, and the background colour of the plate. Such signs also contain various symbols, like arrows, as well as variable-font characters. With such a great appearance variability, only very coarse class definitions make sense, but this kind of categorisation is of little practical use. Alternatively, the contained textual information, if present, may be focused on. However, fast and reliable recognition of these kinds of patterns is not possible in low-resolution and noisy imagery captured with a wide-angle camera and therefore requires more advanced hardware, e.g. an additional telephoto lens.

Road sign recognition involves two critical tasks, each requiring special attention: feature extraction and classification. In many object recognition problems the feature extraction step is considered more important. A discriminative representation of an object provides a mapping from the high-dimensional original image space to the low-dimensional feature space in which the different-class patterns are more clearly separated from one another. With such a mapping available, even a simple classifier design will offer satisfactory performance. In the case of TSR, a major difficulty at this level seems to be a joint modelling of colour and shape of traffic signs, which are these features that make them appear so distinctive to the driver, even in high visual clutter and in adverse lightning and weather conditions. Note that neither of these cues alone guarantees a sign to appear sufficiently distinguishable from the background in the entire spectrum of possible traffic situations. For example, in the street scenes there are many circular structures and straight edge segments with appropriate slope that could potentially constitute signs' contours. Only in presence of appropriately arranged, highly contrasting colours the true sign boundaries can be readily distinguished from those belonging to the other, accidental objects. Another challenging problem regarding the feature extraction step is ensuring that such features are robust to various types of image transformations which often occur in realistic traffic scenarios: adverse illumination, geometric distortion, shadows, motion blur or partial occlusions.

At the classifier design step, assuming that discriminative sign features are identified, the major difficulty is handling a large number of pictograms, frequently very similar to one another. Often a satisfactory separation of multiple classes at the feature representation level is not possible. In that case a discriminative power of the classifier can be increased by further transformations of the feature space, as is done for instance in the case of multi-class Support Vector Machines by Crammer and Singer [88] or multi-class AdaBoost by Hao and Luo [146]. Another possibility is to combine different classifiers into trees [12] or ensembles, e.g. via bagging [44] or boosting [60], in which each classifier partially contributes to the ultimate decision. However, care should be taken to maintain balance between the classifier's complexity and its computational efficiency, especially when matching with many dozens of sign prototypes is involved.

An alternative to increasing the classifier's complexity is decreasing the complexity of the classification problem via its appropriate decomposition. This idea provides rationale for the strategies known as *one-vs-all* [115] and *one-vs-one* [77, 81]. The first technique attempts to build a discriminant function separating each particular class from all other classes. The second one constructs classifiers that can recognise all possible pairs of classes. In both cases, the ultimate decision is made by choosing the class label with the maximum number of votes accumulated across partial classifications. Unfortunately, for *one-vs-all* and *one-vs-one* approaches, scalability is an issue, i.e. the number of class-specific discriminator functions required grows linearly/quadratically with the number of classes. Specifically, in *one-vs-all* approach N classifiers are required for N classes and $N(N-1)/2$ classifiers are needed for the same N classes in *one-vs-one* approach. Although such class-specific discriminants are typically much simpler than a monolithic N-class classifier, it does not compensate the overall complexity and the training effort involved. To make matters worse, also a large number of training examples are required, but these, in the case of certain, naturally very rarely occurring signs, are often hard to acquire.

Compared to the other components of a TSR system, the detector and the tracker, the classifier performs the highest-level analysis of the visual information acquired from a vehicle, and yields the ultimate output desired by a human driver. While he/she is not interested in how the signs are detected and tracked, automatic interpretation of the observed road sign patterns has immediate applications in the intelligent vehicles. The classified signs detected in the scene ahead of a car can be displayed on the dashboard to make the driver aware of the incoming traffic situation. Also a natural language audio message interpreting the classified pictogram can be played so that the human's attention is not too much distracted from observing the road. In the future, traffic sign recognition functionality is likely to be fully integrated with other components of driver assistance systems. In particular, the sign interpreter will certainly be allowed to trigger certain mechanical actions to increase the driver's safety or to prevent a vehicle from breaking the traffic regulations.

In the rest of this chapter the background of traffic sign recognition is discussed (Section 5.1). It is followed by the presentation and evaluation of our original contributions in this area. In Section 5.2 we investigate a discriminative, class-specific representation of traffic signs which can be learned from sign prototypes according to *one-vs-all* strategy and used in system runtime within a class-specific template matching framework. In Section 5.3 the idea of a trainable sign similarity function learned from image pairs is developed. In this section the two different realisations of this idea are discussed. One is based on the *SimBoost* algorithm, a variation of the popular AdaBoost technique. The other similarity learning algorithm utilises a kernel regression tree framework. Both techniques are thoroughly evaluated on various image datasets by measuring the recognition rate of the nearest neighbour classifiers incorporating the learned similarity measure. Summary of this chapter is given in Section 5.4.

5.1 Background

A baseline approach for traffic sign classification involves a pixel-based cross-correlation template matching. This technique was used for example in [49, 65, 120]. It is robust to illumination changes. Unfortunately, it is useful only on condition that the objects in the tested image and in the template images are well aligned. Satisfactory alignment is frequently difficult to achieve by the automatic sign detection systems, especially when the incoming images are affected by the car vibrations, when the target is seen against cluttered background, or when it is geometrically distorted. Partial occlusions also severely degrade the performance of this method as they introduce regions of noisy pixels to the images of observed traffic signs.

Feature-based approaches to road sign classification are more popular. For example, Escalera et al. [50] utilised a multi-layer perceptron for this task. Their neural network (NN) was trained using the 30 × 30-pixels images of ideal signs affected by various transformations. For example, minor misalignments of the detected signs were accounted for by presenting on input of their network the slightly shifted and rotated sign prototypes. Contrast variations of the realistic sign patterns were simulated by adding different levels of Gaussian noise to the prototype images, followed by thresholding, in order to obtain the information located in the inner part of each sign. Nguwi and Kouzani [157] used the same type of neural network, but decomposed the multi-class classification problem into a series of one-to-one problems, each assigned with a separate network. Douville [72] fed a neural network with Normalised Gabor Wavelet Transform features from multi-resolution filters. He reported high recognition rate and real-time processing when classifying video traffic signs in background clutter and under geometric transformations. Neural network approach was also adopted in [35] (Radial Basis Function NN), [36, 45] (Backpropagation NN) and [98] (Adaptive Resonance NN).

Other feature-based methods for traffic sign recognition were proposed. For example in [34] the preliminary classification was done by measuring the probability of a convex hull of the candidate object representing regular sign shapes. The exact class of a sign was determined by colour histogram matching. Paclík et al. [66] used a simple decision tree to decompose the road sign family into subgroups of signs that are similar with respect to shape and colour. The test image was classified in two steps. First, the general category of a sign was found by propagating the image through the aforementioned tree. Then, the test image was described with several global shape descriptors, like moments and compactness, and the exact type of sign was found using a Laplace kernel classifier in such a feature space. Promising results were reported using this method, even in presence of few training images. However, the hierarchical decomposition of the problem was done manually and was hence unconvincing. A yet another feature-based approach was adopted by Hsu and Huang [74]. They trained a traffic sign classifier by inferring class-specific Matching Pursuit filter bases from the training data. The input image was labelled by projecting it to the found filter sets and determining the best-matching class. Gao et al. [142] employed the biologically-inspired vision models to represent both colour and shape features of the traffic signs. They achieved a good recognition rate for static images of signs affected by substantial noise and perspective transformations.

An interesting concept of a similarity representation of traffic signs was recently introduced by Paclík et al. [144]. In this study each sign was represented by a set of similarities to the stored classes' prototype images. The similarity measure was estimated individually for each sign class using a supervised learning algorithm based on sequential forward feature selection. This algorithm aimed at determining a compact set of local image regions where the target class was the best separable from all remaining classes with respect to the Fisher's discriminant ratio. An unknown sign image was assigned to the class of the most similar prototype. Good performance of the classifiers based on the trainable similarity was demonstrated experimentally for relatively easy classification problems, i.e. those involving small number of relatively dissimilar classes. A question whether other separability criterions, feature representations, or search strategies would not lead to better classification accuracy remains open and calls for investigation.

Apparently, in very few recognised studies the traffic sign recognition problem was investigated thoroughly, i.e. addressing a large number of diverse signs, their natural grouping, and the relationships between the groups. Usually, the TSR systems are not able to efficiently discriminate between as many as a few dozens of classes or more. Therefore a majority of previous recognition algorithms specialised in discriminating between the signs belonging to only one narrow category, exhibiting similar shape and colour properties, e.g. prohibition signs, or cautionary signs. The typical number of classes recognised in these studies was up to 30, e.g. 23 in [127], or up to 18 classes in [144]. In some studies more signs were considered, but the original classification problem was decomposed at an early stage of the processing within a decision tree into multiple simpler problems. Then, a separate classifier was trained for each distinguished category of signs. This reasonable strategy was chosen for instance in the study of Paclík et al. [66] (50 classes), or Escalera et al. [120] (165 classes).

Building a well-generalisable road sign classifier, that would admit easy, incremental incorporation of new pictograms without degrading the overall performance, is difficult. The discriminative recognition approaches that have been used in the state of the art do not provide any mechanism for such incremental learning. Another problem with the discriminative road sign recognition methods, which typically require substantial amount of training images for training, is related to data acquisition. Namely, certain types of signs occur very rarely in reality. Therefore, a sufficient number of training images representing such signs cannot be easily collected. Even if natural images of each pictogram are available, there must be necessarily a large quantitative imbalance regarding the proportions of different classes' representatives in the data. This, if no appropriate preventive measures are taken, increases the risk of biasing the classifier.

The abovementioned problem could most likely be addressed in two different ways. First, a discriminative feature representation of traffic signs, incorporating colour and shape information, as well as a robust distance metric for matching, could be constructed from idealised class prototypes available in a highway code. This seems intuitive as the traffic signs are perfectly well-defined objects. Therefore, it should be possible to recognise them based essentially on the discriminative information obtained from the clean template images. Apparently, no comprehensive, solely template-based discriminative feature extraction technique has so far been proposed in the previous studies on TSR. Another possible technique that could be adopted to address certain aspects of traffic sign recognition is based on a generative paradigm. It is particularly useful for solving problems involving multiple classes with high intra-class variability, and can learn the model from few training images, which could address the data acquisition issue. Generative methods also allow the new classes to be added incrementally by learning the class-conditional density, independently of all classes already existing in the model. These properties of generative models could possibly be exploited to automatically learn new sign categories from sparse representative images available. However, there are also known problems related to generative models, such as expensive computation which could hamper real-time TSR system implementation.

5.2 One-vs-all Class-specific Template Matching

Traffic signs are objects that are very well defined in terms of their shapes, colours and contained pictogram symbols. Even though there is some intra-class variability with regard to their appearance [166], it seems intuitive to exploit the prototype sign images for constructing a discriminative model of signs. The usual way of doing it is via incorporating the *a priori* knowledge of the target appearance into the arbitrary semantic rules governing the classification. We want to go one step further – devise an automatic framework for discriminative representation inference, based essentially on the template images of road signs. A successful algorithm accomplishing this task would have numerous advantages. Among others it would eliminate the time-consuming training data acquisition and labelling steps and hence dramatically simplify the classifier training process.

In the rest of this section the idea of discriminative local region representation of traffic signs and the class-specific template matching is developed. It is centred around an intuition that a good discrimination between a large number of potentially similar objects can be achieved by focusing on how each of them differs from all the remaining objects. This concept leads to the discriminative local region representation of traffic signs which we present in Section 5.2.2. Earlier, in Section 5.2.1 we describe the process of colour discretisation and introduce the *Colour Distance Transform*. They jointly enable robust comparisons between the pictogram regions in image pairs where one image contains the noisy observation coming from an input traffic video and the other is an idealised sign's prototype. Based on these prerequisites, we build a simple, temporally integrated nearest neighbour classifier which matches the image of an unknown sign with the template sign images with respect to their learned class-specific local region representations. It is discussed in Section 5.2.3. An extensive experimental evaluation of the presented approach is given in Section 5.2.4.

5.2.1 Image Representation

Selecting an optimal feature set for a large number of template sign images is a non-trivial task. The simplest choice is probably to extract the local image characteristics at the pre-defined, regularly distributed locations uniform for all signs, as for instance Gao et al. [142] did. Naturally, the drawback of this approach is that the feature locations are chosen manually, irrespective of how much discriminative information the corresponding image fragments carry. Another possibility is to describe each sign by some global numerical characteristics, e.g. moments. However, this technique proves useful when the number of classes to recognise is low and these classes are themselves significantly different. With the increase in the number and similarity of classes, moments and other global shape descriptors become less discriminative, which was confirmed in the preliminary experiments we made using the still camera images of traffic signs.

Several automatic feature selection techniques such as Principal Component Analysis (PCA) or AdaBoost are available. However, aiming at retrieving the global variance of a whole dataset and not taking the class labels into account, PCA is not capable of capturing features critical to the individual templates. On the other hand, AdaBoost framework is known to provide a way for extracting a compact representation and generating efficient classifiers. However, it is originally designed to solve binary problems and a generalisation to multi-class problems is not straightforward. Besides, a large amount of data is required for AdaBoost training. Acquisition and preprocessing of such data is a very time-consuming process, with an additional difficulty caused by the fact that certain road signs can be seen extremely rarely in reality. Finally, traffic signs are very well defined objects with only small intra-class variability and therefore can be unambiguously represented with clean prototypes. Therefore, a question arises whether there is sufficient justification for learning them from the real-life data.

A separate issue is the choice of the underlying image representation from which salient features could be generated, and the actual meaning of "saliency". As already mentioned, most of the high-level visual shape descriptors, such as moments, were found inadequate to represent the detected noisy candidate road sign images due to the insufficient discriminative power of such descriptors in presence of many similar classes in the pictogram database. On the other hand, low-level, pixel-based methods tend to suffer from high sensitivity to all kinds of geometrical distortion, e.g. shifts and rotations, as well as misalignments resulting from the imperfection of the detection process, which are to some extent unavoidable. Histogram-based methods partly overcome this limitation, but in return they lose the valuable information about the spatial arrangement of the individual pixels. Therefore, they often fail in the recognition problems involving large number of classes. This calls for more adequate sign representations.

We trialled a patch-based approach to describe an observable road sign pattern with a collection of salient local features. One of the well-justified meanings of visual saliency, proposed by Kadir and Brady [78], was considered at this point. However, three kinds of problems were encountered. First, different traffic signs cannot be represented by even a roughly equal number of salient regions as some in fact contain nearly nothing visually salient in a sense of large entropy, e.g. “no vehicles” sign or “give way” sign shown in Figure 5.1. Second, due to the very low number of sign-specific colours used in the model images, the meaning of entropy defined over the pixel intensity becomes vague. Different saliency formulations are also inadequate unless a large amount of training data is available. Finally, “salient” defined by the entropy within a single image does not necessarily mean “discriminative” among a group of signs, especially when these signs are very similar to one another.

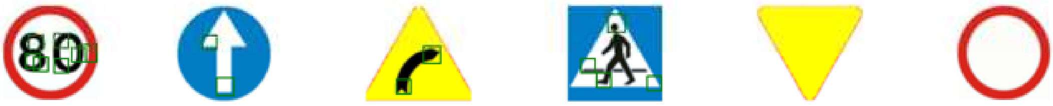


Figure 5.1: Examples of traffic signs with the high-entropy salient regions marked. These regions were extracted using a simplified version of Kadir and Brady’s algorithm [78] that allows only square regions. Obtained regions were clustered in the spatial domain. Note that in the case of the two last signs no salient regions were found.

Motivated by Paclík et al. [144], we propose in Section 5.2.2 an algorithm that extracts for each template sign a limited number of local image regions in which it looks possibly the most different from all other templates in the same category. This way, we define an alternative meaning of visual saliency to the one suggested by Kadir and Brady [78]. The extracted discriminative regions are further used for comparing the noisy video frame observations with the idealised road sign templates to make a reliable on-line sign classification. In the rest of this section we first outline the process of converting a raw RGB image into a more suitable discrete-colour representation and define a *Colour Distance Transform* (Section 5.2.1.1) upon which a robust discriminative region distance metric is built. Definition of local regions and the aforementioned dissimilarity metric, as well as a description of a region selection algorithm, are postponed to Section 5.2.2.

5.2.1.1 Colour Discretisation and Colour Distance Transform

The detected and tracked road signs in each video frame are passed on input of the recognition module as rectangular image regions containing the target object and, depending on its shape, also background fragments, as depicted in Figure 5.2. If there is any in-plane rotation or anisotropic scaling indicated at the detection/tracking stage, they are first compensated. In order to prepare the candidate for classification, the image is scaled to a common size, typically 60 × 60 pixels for circular and square signs, and 68 × 60 pixels for triangular signs. Undesirable background regions are then masked out using the information about the general object's shape provided by the regular polygon detector (for details, refer to Section 3.3.1). It is important to note that the full colour spectrum is far more than necessary to identify the pictogram. Real signs contain only up to four distinctive colours per category, where by category we mean one of those shown in Figure 3.9. Therefore, the candidate images are subject to on-line colour discretisation according to the category-specific colour models learned off-line from a set of training images, as described below.

For each category of signs a number of frames are picked randomly from a set of realistic traffic video sequences depicting the respective signs. From the region occupied by a sign in each image we manually pick several pixels representing known named colours and record their RGB values, which are further transformed to CIE XYZ values [1]. This is how the training data is constructed. Then, the Expectation Maximisation (EM) algorithm [4] is employed to estimate an optimal Gaussian Mixture Model (GMM) [15] for each colour specific to this category. The procedure is restarted several times for the increasing number of randomly initialised Gaussian components and the best model in terms of the mean data likelihood is saved.

In system runtime the appropriate off-line learned GMMs are used to classify each RGB pixel in the input sign region into one of the admissible colours by picking the model that has most likely generated this pixel. On the implementation side, it should be noted that a large speedup of this colour discretisation can be achieved at the cost of higher memory consumption. Namely, the colour models can be used in advance to assign the appropriate colour label to each possible RGB triple, yielding a look-up table with 255^3 entries for each sign category. This way, intensive computation can be avoided by picking the colour labels directly from the memory-stored look-up tables. Sample results of the on-line colour discretisation described above are illustrated in Figure 5.2.

Our ultimate goal is to enable robust comparisons between the realistic and the model images in a discrete colour representation. To this end we know how to rapidly obtain such a representation from the incoming RGB image regions enclosing the detected sign candidates. We also possess the template images where the colour palette is already sparse. To facilitate the aforementioned comparisons, a separate distance transform (DT) [16] is computed for each discrete colour, giving output

[1] CIE XYZ colour appearance model is used in preference to the raw RGB space because it is device-independent.

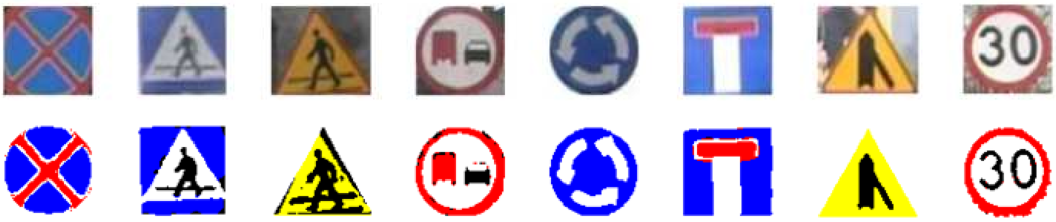


Figure 5.2: Example images obtained by the sign detector before (above) and after (below) background masking and colour discretisation; 2 bits suffice to encode colours in each image.

similar to this shown in Figure 5.3. In DT computation pixels of a given colour are simply treated as feature pixels and all the remaining pixels are treated as non-feature pixels. A $(3, 4)$ Chamfer metric [58] is used to approximate the Euclidean distance between the feature pixels. To emphasise a strong relation to colour, we call this variant of DT a *Colour Distance Transform* (CDT).

One practical problem with CDT emerges when a given colour is absent in an input image. Note that within each category of signs there are some that do not contain the colours present in the other. Many examples of such signs can be found in Figure B.1 in Appendix B. However, for the sake of comparisons, all colour distances must be sensibly represented in each template sign. To reflect the absence of a given colour in CDT, positive infinity is not appropriate as it causes numerical problems. Instead, a fixed maximum relevant colour distance value $d_{max} = 10\,\text{px}$ is introduced and the colour-specific DT images are normalised by assigning each pixel (x, y) in the image I a value defined as:

$$\tilde{d}_{CDT}(I,x,y) = \begin{cases} \frac{d_{CDT}(I,x,y)}{d_{max}} & \text{if} \quad d_{CDT}(I,x,y) \leq d_{max} \\ 1.0 & \text{if} \quad d_{CDT}(I,x,y) > d_{max} \end{cases} . \tag{5.1}$$

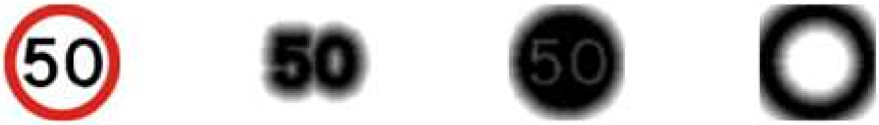


Figure 5.3: Normalised Colour Distance Transform images. From left to right: original discrete-colour image, black DT, white DT, red DT. Darker regions denote shorter distance.

5.2.1.2 Justification of CDT

Variable illumination and small viewpoint-dependent apparent deformations of traffic signs are among the main sources of their intra-class appearance variability. In other words, these factors often cause the same pictograms to look different from scene to scene. In the previous section it was explained how the proposed colour binning addresses the former problem. Hereby, we claim that the *Colour Distance Transform* enables modelling the distribution of the discrete-colour appearance of traffic signs under minor affine transformations without recourse to the massive volumes of natural data, and hence addresses the latter problem.

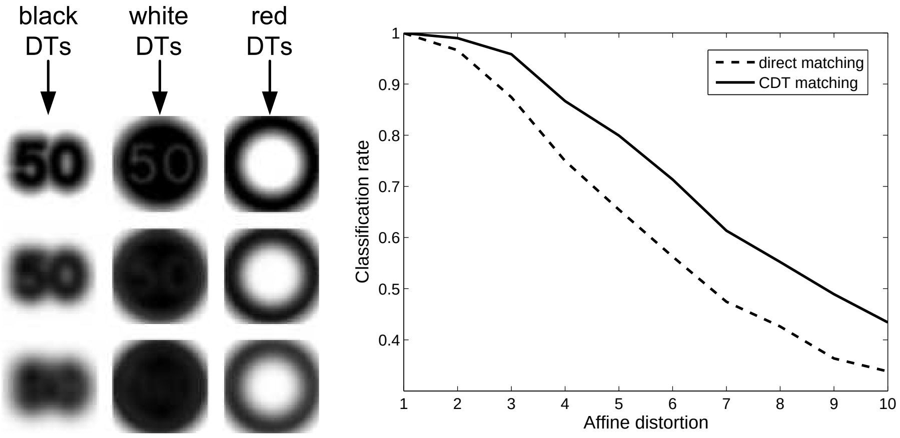
To provide evidence to support this claim, we have done two simple experiments. First, from each discrete-colour image of a chosen template sign $n = 1000$ random affine transformations were generated. All parameters of the affine transform matrices, translation, x, y, rotation around the centroid, θ, scale, s_x, s_y, and shear, h_x, h_y, were drawn from the clipped normal distributions with appropriately low standard deviations to ensure the generated distortions were realistically small. In this experiment an alternative method of constructing the colour distance maps was evaluated. Namely, for each pixel we counted how many times this pixel was not of a given colour in each distorted image and divided it by n. Figure 5.4a illustrates the resulting frequencies obtained for different values of standard deviation of the affine parameters. Apparently, the resulting images very much resemble the CDT images (see Figure 5.3 for comparison) when the distortion parameters are appropriately chosen.

In the second experiment we directly compared the accuracy of template matching under small affine transformations using: 1) a distance metric based on the co-occurrence of discrete colours in both compared images, and 2) a distance metric based on CDT. To clarify, in the first method simply a fraction of the spatially corresponding pixels having different colours in both discretised images was calculated. In this experiment each of 42 model cautionary signs was artificially distorted 100 times. Each such distorted image was compared to each undistorted model, i.e. to one template representing the same sign and 41 templates of the remaining signs. The standard deviations of the affine transformation parameter distributions, the same as those used in the first experiment, were gradually increased, starting from: $\sigma_x = \sigma_y = 0.5$ px, $\sigma_\theta = 0.5°$, $\sigma_{s_x} = \sigma_{s_y} = 0.005$, $\sigma_{h_x} = \sigma_{h_y} = 0.005$, and with step: $\Delta\sigma_x = \Delta\sigma_y = 0.5$ px, $\Delta\sigma_\theta = 0.5°$, $\Delta\sigma_{s_x} = \Delta\sigma_{s_y} = 0.005$, $\Delta\sigma_{h_x} = \Delta\sigma_{h_y} = 0.005$. Correct classification rates obtained are shown in Figure 5.4b.

Results of the above experiments suggest that the *Colour Distance Transform* is suitable for comparing pairs of discretised images of traffic signs affected by minor affine transformations. Smooth distance metric defined over CDT clearly outperforms simple pixel-wise discrete colour matching. Together with the proposed colour discretisation technique, CDT becomes a good alternative to the pose- and illumination-invariant traffic sign appearance modelling. It is superior to the data-driven methods in that, with the exception of the images needed for GMM colour classifier training, it does not require the realistic traffic sign images.

5.2.2 Feature Selection

With a CDT-based smooth distance metric available, discrete-colour images of traffic signs can be compared pixel by pixel, without risking serious recognition rate degradation caused by small image misalignments. However, our intuition is to reduce the dimensionality of the feature space not only by discarding the redundant colour information, but also by selecting only those patches in each pictogram that



(a) (b)

Figure 5.4: Experimental evaluation of *Colour Distance Transform.* (a) Colour distance maps: black (left column), white (central column), and red (right column), obtained for a discrete-colour image of a 50 kph speed limit sign in the experiment involving generation of 1000 random affine distortions. Pixel intensities correspond to their frequency of having a given colour in the transformed images. Each parameter of the affine transformation: translation, x, y, rotation around the centroid, θ, scale, s_x, s_y, and shear, h_x, h_y, was perturbed according to a clipped normal distribution. Standard deviations of these distributions were: $\sigma_x = \sigma_y = 1$ px, $\sigma_\theta = 1°$, $\sigma_{s_x} = \sigma_{s_y} = 0.02$, $\sigma_{h_x} = \sigma_{h_y} = 0.02$ (top row), $\sigma_x = \sigma_y = 3$ px, $\sigma_\theta = 3°$, $\sigma_{s_x} = \sigma_{s_y} = 0.02$, $\sigma_{h_x} = \sigma_{h_y} = 0.02$ (central row), $\sigma_x = \sigma_y = 5$ px, $\sigma_\theta = 5°$, $\sigma_{s_x} = \sigma_{s_y} = 0.05$, $\sigma_{h_x} = \sigma_{h_y} = 0.05$ (bottom row). (b) Correct classification rate for a 42-class Polish cautionary signs problem as a function of image distortion, obtained using two discrete image comparison methods. A simple nearest neighbour template matching classifier was used. In the first comparison method (dashed line), each distorted image was compared to the undistorted template signs by counting the numbers of spatially corresponding pixels having unmatching colours. In the second method (solid line), distance between each compared pair of images was measured pixel-wise using the CDT-based metric.

are really unique for the class it is representing. Below, it is presented how such discriminative image fragments are described and selected.

5.2.2.1 Discriminative Local Regions

An overcomplete space of local regions is obtained by considering all possible subwindows of the input image of size $m \times n$ pixels, where two neighbouring subwindows of the same dimensions half-overlap. Typically, we use $m, n = \{4\,px, 8\,px\}$. Within each region r_k dissimilarity between the images I and J can be calculated using the discrete-colour image of I and the normalised CDT images of J by averaging the pixel-wise distances:

$$d_{r_k}(I,J) = \frac{1}{mn} \sum_{(x,y)\in r_k} \psi_{CDT}(I,J,x,y) \ , \tag{5.2}$$

where for each pixel (x, y) contained in the region, distance $\psi_{CDT}(I, J, x, y)$ is picked from the appropriate normalised CDT image of J, depending on the colour of this pixel in I. Let us also denote by $\widehat{d}_{\mathbf{S}}(I, J)$ and $\widehat{d}_{\mathbf{S},\mathbf{W}}(I, J)$ a normal and weighted average local dissimilarities between the images I and J computed over regions $r_k \in \mathbf{S}$ (weighted by $w_k \in \mathbf{W}$):

$$\widehat{d}_{\mathbf{S}}(I,J) = \frac{1}{|\mathbf{S}|} \sum_{k=1}^{|\mathbf{S}|} d_{r_k}(I,J) \ , \tag{5.3}$$

$$\widehat{d}_{\mathbf{S},\mathbf{W}}(I,J) = \frac{\sum_{k=1}^{|\mathbf{S}|} w_k d_{r_k}(I,J)}{\sum_{k=1}^{|\mathbf{S}|} w_k} \ . \tag{5.4}$$

CDT images for the model signs are pre-computed. Also, recall (Section 5.2.1.1) that the images of the observed signs can be instantly discretised using the colour look-up tables. Therefore, any on-line local-region comparison between the observed and the template images can be made extremely fast.

5.2.2.2 Region Selection Algorithm

Assuming pre-determined category of signs $C = \{T_i : i = 1, \ldots, N\}$ and a candidate image $\mathbf{x}_j$, our goal is to determine the class of $\mathbf{x}_j$ by maximising posterior:

$$p(T_i|\mathbf{x}_j, \theta_i) = \frac{p(\mathbf{x}_j|T_i, \theta_i)p(T_i)}{\sum_{k=1}^{N} p(\mathbf{x}_j|T_k, \theta_k)p(T_k)} \ , \tag{5.5}$$

where $p(T_k)$, $k = 1, \ldots, N$ are equal class priors [2]. We think that uniform feature sets are inadequate for traffic sign recognition. Some signs can be told apart by just a single distinctive pictogram element, while other need to be analysed in more detail

[2] It would be more practical to diversify class priors to reflect the fact that certain traffic signs can be found more frequently in the traffic environment than the other. However, we do not possess a reliable quantitative analysis of such occurrence frequencies that could be used to set the priors.

to be distinguished from other similar signs. Our objection to using a uniform feature space for classification makes us envisage different model parameters $\theta_i = (\mathbf{I}_i, \mathbf{W}_i)$ for each template T_i. $\mathbf{I}_i$ denotes an indexing variable determining the set $\mathbf{S}_i$ of p_i most discriminative local regions to be used and $\mathbf{W}_i$ is a vector of relevance corresponding to the regions $r_k \in \mathbf{S}_i$ selected by $\mathbf{I}_i$. p_i is not fixed and therefore may vary between classes. In order to learn the best model parameters θ_i^*, the following objective function is maximised:

$$O(\theta_i) = \sum_{j \neq i} \widehat{d}_{\mathbf{S}_i}(T_j, T_i) \ . \tag{5.6}$$

In other words, the regions best characterising a given sign are obtained through maximisation of the sum of local dissimilarities between this sign's template and all the remaining signs' templates.

In presence of model images only, each average set dissimilarity term $\widehat{d}_{\mathbf{S}_i}(T_j, T_i)$ as a function of the number of discriminative regions in $\mathbf{S}_i$ is necessarily monotonically decreasing. This is because the subsequent regions added are increasingly less dissimilar and hence give smaller contribution to this average. As a result, typically there would be just a single best region or at most a few equally good regions maximising equation (5.6). In practice, such sign descriptors are unlikely to work well for the noisy video where more support in terms of the number of image patches to match in each frame is required to make a reliable discrimination. Therefore, to balance the discriminative power and the reliability, our objective function is iteratively degraded up to the specified breakpoint, yielding a representation which is more dense and thus more useful in a real data context. The details of this procedure are given below.

Similarly to Paclík et al. [144], in the model training stage we have adopted elements of a sequential forward search strategy, a greedy technique from the family of floating search methods [38]. However, both approaches differ significantly in two main aspects. First, we think that learning the signs from the real-life images might not be worth the effort required as the publicly available templates seem to sufficiently characterise the appearance of the respective classes. Second, we believe that the possible within-class appearance variability may well be accounted for by a CDT-based image representation and a robust local pictogram distance metric, such as the one introduced in equations (5.2-5.4), instead of being learned. Our implementation outlined in Algorithm 7 and discussed below.

A given template sign is compared to each of the remaining templates. In each such comparison the algorithm loops until the appropriate number of local regions are selected. It should be noted that at a given step of the loop the most dissimilar region is fixed and removed from the pool of available regions. Moreover, at the k-th step the distance between the considered image and the image being compared to is measured with respect to the joint set comprised of the new k-th region and all previously selected regions. At the end of the loop a list of regions is produced, together with the list of corresponding weights reflecting the discriminative power of these regions. Each pairwise region set build-up is controlled by a global threshold, t_D, specifying the maximum allowed cumulative dissimilarity between any pair of templates being compared. Such a definition of STOP criterion ensures that roughly the same amount of dissimilarity between any pair of templates is incorporated in the model. This in turn allows us to treat different sign classes as directly comparable, irrespective of the actual number of local regions used to characterise them. A single comparison between the interest template T_i and template T_j, $j \neq i$, is schematically depicted in Figure 5.5a. The final region set for each class is constructed by merging the pair-specific subsets, as shown in Figure 5.5b. It is reflected in the region weights carrying the information on how often and with what importance each particular region was selected.

Algorithm 7 Discriminative local region selection algorithm.

```
input: sign category $C = \{T_j : j = 1, \ldots, N\}$, target template index $i$, region pool $R = \{r_k : k = 1, \ldots, M\}$, dissimilarity threshold $t_D$
output: ordered set $S_i$ of regions to discriminate between template $T_i$ and all other templates, ordered set $W_i$ of weights corresponding to the regions from $S_i$
initialise an array of region weights $W = \{w_k : w_k = 0, k = 1, \ldots, M\}$
for each template $T_j \in C$, $j \neq i$ do
    sort $R$ by decreasing dissimilarity $d_{r_k}(T_i, T_j)$
    initialise ordered region set $S_{i,j}$, and the corresponding weight set $W_{i,j}$, characterising the dissimilarity between templates $T_i$ and $T_j$, with the first region from $R$ and its weight respectively: $S_{i,j} = [r_j^{(1)}]$, $W_{i,j} = [w_j^{(1)}]$, where $w_j^{(1)} = d_{r_j^{(1)}}(T_i, T_j)^2$
    initialise region counter $l = 1$
    initialise the total dissimilarity, $D_{i,j}$, between templates $T_i$ and $T_j$: $D_{i,j} = d_{r_j^{(1)}}(T_i, T_j)$
    while $D_{i,j} < t_D$ and $l < M$ do
        increment region counter $l = l + 1$
        set weight of the new region to: $w_j^l = d_{r_j^{(l)}}(T_i, T_j)^2$
        add region $r_j^{(l)}$ to $S_{i,j}$ and weight $w_j^{(l)}$ to $W_{i,j}$
        update $D_{i,j}$: $D_{i,j} = D_{i,j} + d_{r_j^{(l)}}(T_i, T_j)$
    end while
    for each region $r_k \in S_{i,j}$ do
        update region weight: $w_k = w_k + w_j^{(t)}$, where $t$, $w_j^{(t)}$ are the index and weight of region $r_k$ in $W_{i,j}$
    end for
end for
build the target region set $S_i$ and the target weight set $W_i$: $S_i = \{r_k : w_k > 0\}$, $W_i = \{w_k : w_k > 0\}$
```

For each sign class the above procedure yields a set of its most unique regions. It should be noted that in the final step, depending on the actual dissimilarity threshold specified, certain number of regions will be found completely unused, and hence discarded. An example output of the proposed region selection algorithm is depicted in Figure 5.6. Obtained discriminative region maps clearly show that different signs are best distinguishable in different fragments of the contained pictogram. It can also be seen that although the same value of category-global parameter t_D was used, different numbers of relevant regions were found.

In absence of realistic images of traffic signs, it is generally hard to choose the optimal value of t_D. One possible way of tuning this parameter is via measuring the recognition rate of a nearest neighbour classifier employed to match artificially

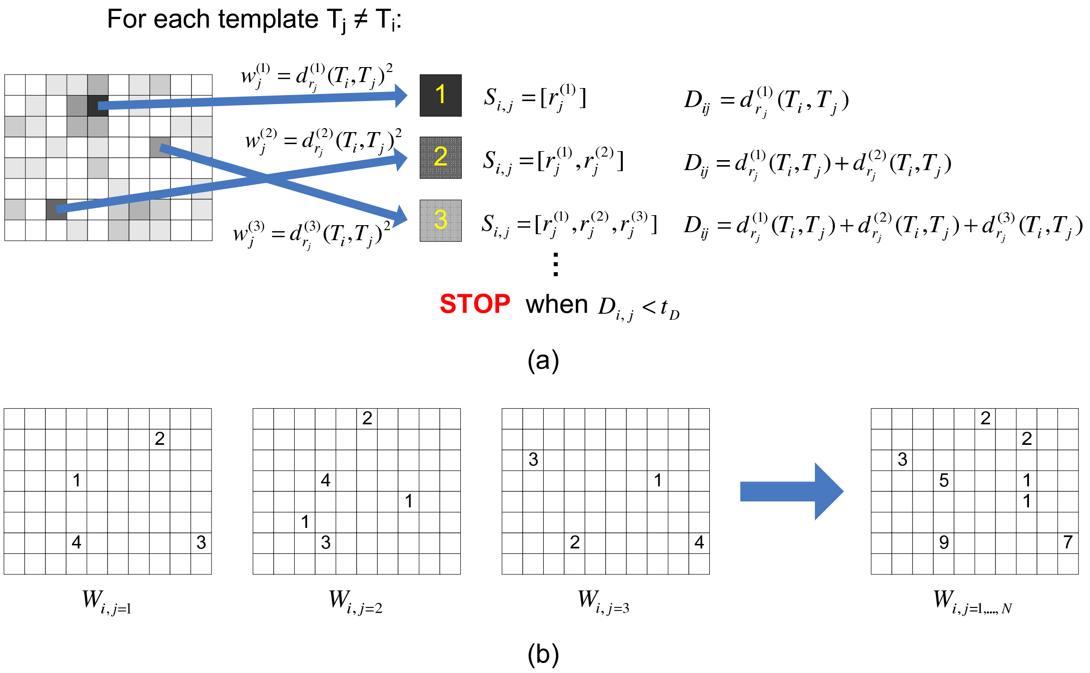

For each template $T_j \neq T_i$:
$w_j^{(1)} = d_{r_j}^{(1)}(T_i,T_j)^2$
$w_j^{(2)} = d_{r_j}^{(2)}(T_i,T_j)^2$
$w_j^{(3)} = d_{r_j}^{(3)}(T_i,T_j)^2$
1
2
3
$S_{i,j} = [r_j^{(1)}]$
$S_{i,j} = [r_j^{(1)}, r_j^{(2)}]$
$S_{i,j} = [r_j^{(1)}, r_j^{(2)}, r_j^{(3)}]$
$D_{ij} = d_{r_j}^{(1)}(T_i,T_j)$
$D_{ij} = d_{r_j}^{(1)}(T_i,T_j) + d_{r_j}^{(2)}(T_i,T_j)$
$D_{ij} = d_{r_j}^{(1)}(T_i,T_j) + d_{r_j}^{(2)}(T_i,T_j) + d_{r_j}^{(3)}(T_i,T_j)$
STOP when $D_{i,j} < t_D$
(a)
$W_{i,j=1}$
$W_{i,j=2}$
$W_{i,j=3}$
$W_{i,j=1,\dots,N}$
(b)

Figure 5.5: Discriminative local region selection algorithm: (a) construction of the local region partial ranking characterising differences between two templates, T_i and T_j, (b) merging three example partial rankings. To facilitate visualisation, non-overlapping regions are shown and the partial weights of the relevant regions are given as whole numbers.

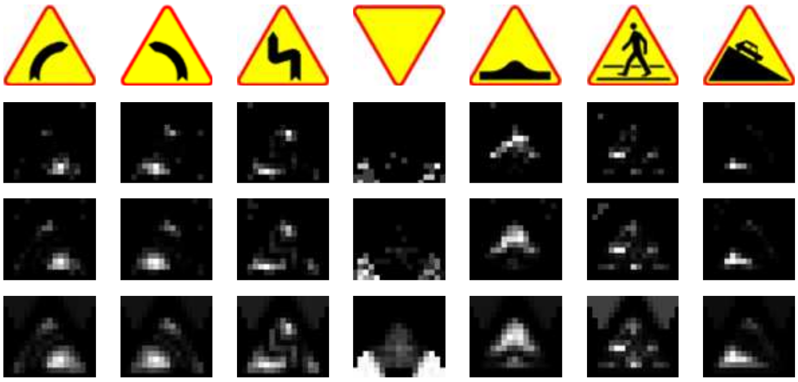


Figure 5.6: Prototype images of sample Polish cautionary signs (1^{st} row) and 4×4-pixels discriminative regions obtained for the parameter $t_D = 2.0$ (2^{nd} row), $t_D = 5.0$ (3^{rd} row), and $t_D = 50.0$ (4^{th} row). Brighter regions correspond to higher dissimilarity.

transformed (geometrically distorted, blurred, noised etc.) template images with the clean unchanged templates. However, it is generally impossible to simulate the joint effect of different types of image transformations that may simultaneously affect traffic signs in reality. Therefore, we opt for tuning the dissimilarity threshold based on a small available number of real traffic image sequences captured from a moving vehicle. More details on this tuning is given in Section 5.2.4.1.

5.2.3 Classifier Design

The proposed road sign classifier distinguishes between multiple classes contained in a category pre-determined at the detection stage. Recognition is done based on the discriminative local region representations, unique for each particular class. For simplicity, two assumptions are made: 1) the dissimilarity between each sign and all other same-category signs is Gaussian-distributed in each local region and independent of the dissimilarities in all other regions characterising this sign, and 2) class priors $p(T_i)$ are equal. In such a case Maximum Likelihood theory allows us to relate the maximisation of likelihood $p(\mathbf{x}_j|T_i, \theta_i)$ to the minimisation of weighted distance $\widehat{d}_{\mathbf{S}_i,\mathbf{W}_i}(\mathbf{x}_j, T_i)$. Therefore, for a known category $C = \{T_i : i = 1, \ldots, N\}$ and observed candidate $\mathbf{x}_t$ at time t, the winning class $L(\mathbf{x}_t)$ is determined from:

$$L(\mathbf{x}_t) = \arg\max_i p(\mathbf{x}_t|T_i, \theta_i) = \arg\min_i \widehat{d}_{\mathbf{S}_i,\mathbf{W}_i}(\mathbf{x}_t, T_i) \ , \tag{5.7}$$

where the elements of the region set $\mathbf{S}_i$ and the corresponding weights in $\mathbf{W}_i$ denote the ones learned at the training stage for the template T_i.

When a series of observations from a video sequence is available, it is reasonable to integrate the classification results through the whole sequence over time, instead of performing individual classifications. Hence, at a given time point t our temporal integration scheme attempts to incorporate all the observations made since the sign was for the first time detected until t. Denoting observation relevance at time t by $q(t)$ and assuming independence of the observations from the consecutive frames, the classifier's decision is determined by:

$$L(\mathbf{X}_t) = \arg\min_i \sum_{k=1}^{t} q(k)\widehat{d}_{\mathbf{S}_i,\mathbf{W}_i}(\mathbf{x}_k, T_i) \ . \tag{5.8}$$

Usually, the number of frames where the sign is being tracked and recognised is in between 20 and 60, depending on the size of the sign, its location in the scene, and the velocity of the vehicle. We have observed that the signs detected in the early frames are often inaccurately delimited and contain blurred pictograms due to the low image resolution. Also, as colours tend to look paler when seen at a considerable distance, the colour discretisation discussed in Section 5.2.1.1 exposes severe limitations, unless performed when the candidate sign has already grown in size in the image plane. To address this problem, we adopt the exponential observation weighting scheme from [127] in which relevance $q(t)$ of the observation at time t depends on the candidate's age (and thus size):

$$q(t) = b^{t_{last}-t} \quad , \tag{5.9}$$

where $b \in (0, 1]$ and t_{last} is the time point when the sign is for the last time seen. The optimal value of parameter b is typically in between 0.6–0.8 which effectively makes the ultimate decision of the classifier mostly dependent upon the last 5–10 observations. This strategy may appear to be losing a large amount of information gathered early in the observation process, but has been experimentally shown to provide the best recognition accuracy.

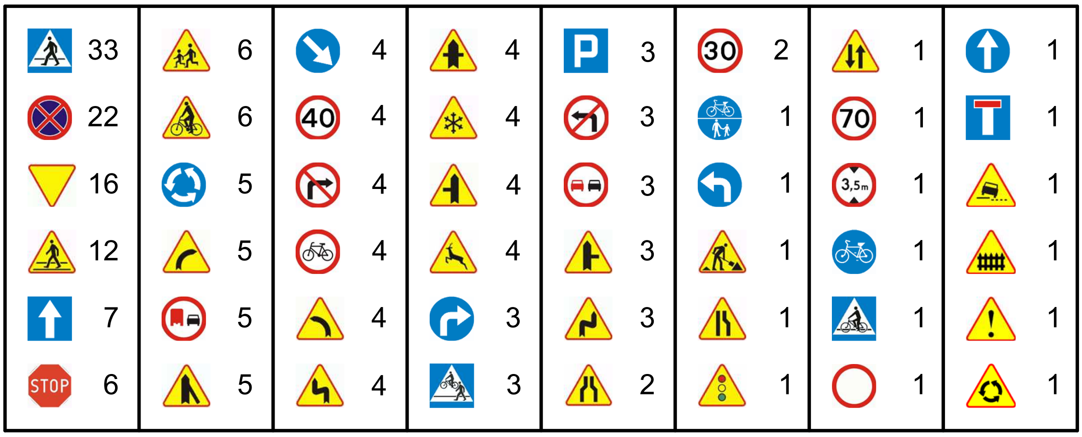
5.2.4 Experiments

In this section an experimental evaluation of our discriminative traffic sign recognition approach described in Sections 5.2.1-5.2.3 is presented. In addition, a justification of certain design patterns chosen is provided. We start with a short description of the test setup and the contents of our template database (Section 5.2.4.1). Experimental results and their discussion is given in Section 5.2.4.2.

5.2.4.1 Test Setup and Template Database

To evaluate the our traffic sign classifier, experiments were performed on the real data collected in Poland at different times of the year: February, April, June, November and December. Sample videos were acquired from a moving car with a DV camcorder mounted in front of the windscreen, and subsequently divided into short clips for off-line testing. Video content depicted the total of 210 signs and included urban, countryside, and motorway scenes in natural daytime lightning, with some signs appearing in shade and in the cluttered background. Due to extreme rarity of certain road signs, the ones in the test data represented only a part of the whole gamut recognised by our system. The entire template database is shown in Appendix B. The detailed breakdown of how particular signs were represented in the test data is provided in Table 5.1.

The basic facts about the traffic signs we focused on are given in Appendix B. All basic cautionary signs defined in the highway code were considered. Regarding the prohibition signs, we counted in speed limit variations, from 30 kph to 120 kph, but excluded the cancellation signs from the analysis. The latter are achromatic and hence difficult to detect. Our template database contained all signs giving orders, including several variants of the "minimum speed" sign, and only these information signs that are square. The rectangular information signs, such as "petrol station" or "camping site" are mostly rare and relatively insignificant. Other, more important information signs, such as those symbolically illustrating the layout of road lanes ahead, were omitted because they are not standarised, i.e. they may vary between traffic situations. Other not unambiguously defined signs, such as those giving direction or distance, or custom diversion signs, were not considered for the same



	33		6		4		4	P	3	30	2		1		1
	22		6	40	4		4		3		1	70	1		1
	16		5		4		4		3		1	3,5m	1		1
	12		5		4		4		3		1		1		1
	7		5		4		3		3		1		1		1
STOP	6		5		4		3		2		1		1		1

Table 5.1: Breakdown of the road sign class occurrence in the test data. The signs not listed were not present at all. Compare this figure with Figure B.1 in Appendix B.

reason as above. Overall, the classifier was aware of 135 different symbolic sign classes.

The physical size of a typical sign does not normally exceed 1.5 metre. With our wide-angle camera used, the image-plane diameter of such signs equal to approximately 30-40 pixels was considered an absolute minimum to start recognising a pictogram. It corresponded to roughly 30-40 metres of physical distance from the vehicle. During our test drives the lens was adjusted at the lowest available focal length of $f = 3.2\ mm$. The vehicle's velocity varied depending on the traffic situation. It was usually in between 40 kph and 70 kph, and never exceed 100 kph. Test video resolution was 640 × 480 pixels in raw format.

For optimal setting of the unknown model parameters, we considered two auxiliary datasets. The first set consisted of 200 images of road signs (50 per category), cropped from various traffic scenes. We used this dataset to 1) adjust the minimum response thresholds of the regular shape detectors (see Section 3.3.1), and 2) adjust the dissimilarity thresholds t_D, independently for each category of signs. The second dataset consisted of additional 20 video sequences used for determining the best setting of the temporal weight base b in equation (5.9). The optimal value of this parameter was found after fixing all category specific t_D-s by maximising the mean ratio of the cumulative distance between the tracked sign candidate and the best-matching template to the cumulative distance between this candidate and the second best-matching template, calculated in the last frame where the tracked sign was entirely seen in the scene.

5.2.4.2 Dynamic Recognition Results

Table 5.2 illustrates detection and classification rates obtained for all available test sequences with the model parameters tuned as described in the previous section

[3]. To illustrate the influence of the dissimilarity thresholds on the classification rates, the experiments were repeated for varying values of these thresholds. Results were compared with the performance of the classifier generated based on: 1) the exhaustive image comparison (as though no region selection was performed), and 2) image comparison with respect to the sign representations obtained for the optimal setting of t_D-s, determined from an independent dataset, as mentioned in Section 5.2.4.1. To visualise the error distribution across the classes, the individual-sign recognition results have been shown in Table 5.3.

t_D	RC (55)	BC (25)	YT (42)	BS (13)	overall (135)
1.0	82.8%	92.1%	84.9%	90.7%	87.8%
2.0	93.1%	94.7%	88.7%	90.7%	90.3%
5.0	93.1%	97.3%	78.8%	81.4%	85.6%
20.0	89.7%	97.3%	77.6%	79.1%	84.6%
all regions	82.6%	91.2%	66.1%	74.3%	76.2%
best	93.1%	97.3%	88.7%	90.7%	91.2%

Table 5.2: Recognition rates obtained for different values of dissimilarity thresholds t_D. The next to last row shows the recognition rates obtained when all possible, uniformly weighted local regions were used for image comparison, as though no feature selection was performed. The last row shows the recognition rate achieved for the best (trained) setting of t_D-s. The numbers of classes in each category: red circles (RC), blue circles (BC), yellow triangles (YT), and blue squares (BS) are given in parentheses in the column headers. The classification rates are determined from only these signs that were correctly detected.

As seen in Table 5.2, obtained classification error rate does not exceed 9%, making our method comparable to the best state-of-the-art approaches. When this classification rate is multiplied by the detection rate obtained for the same video material in the experiment described in Section 3.3.3, nearly 7 out of 8 signs are correctly detected and recognised. However, it should be noted that our template database contains significantly more signs than in most previous studies. Therefore, direct comparison with the alternative methods is not possible. The TSR system incorporating the classifier presented in Section 5.2.3 is able to run at approximately 25-30 fps on a standard PC, but can be boosted to 40 fps when the video resolution is decreased to 400 × 300 pixels. Several video sequences demonstrating these capabilities are attached to this thesis as a supplementary material.

Repetitions of the experiment for different values of dissimilarity threshold revealed that for each category of signs the optimal classifier's performance is achieved for a close to minimum value of this threshold. The following observation is vital at this point. The optimal threshold for each category must strike a balance between

[3] It is assumed that the correct classification takes place when the correct template is assigned the smallest cumulative distance in the last video frame where the candidate sign is entirely contained in the camera's field of view.

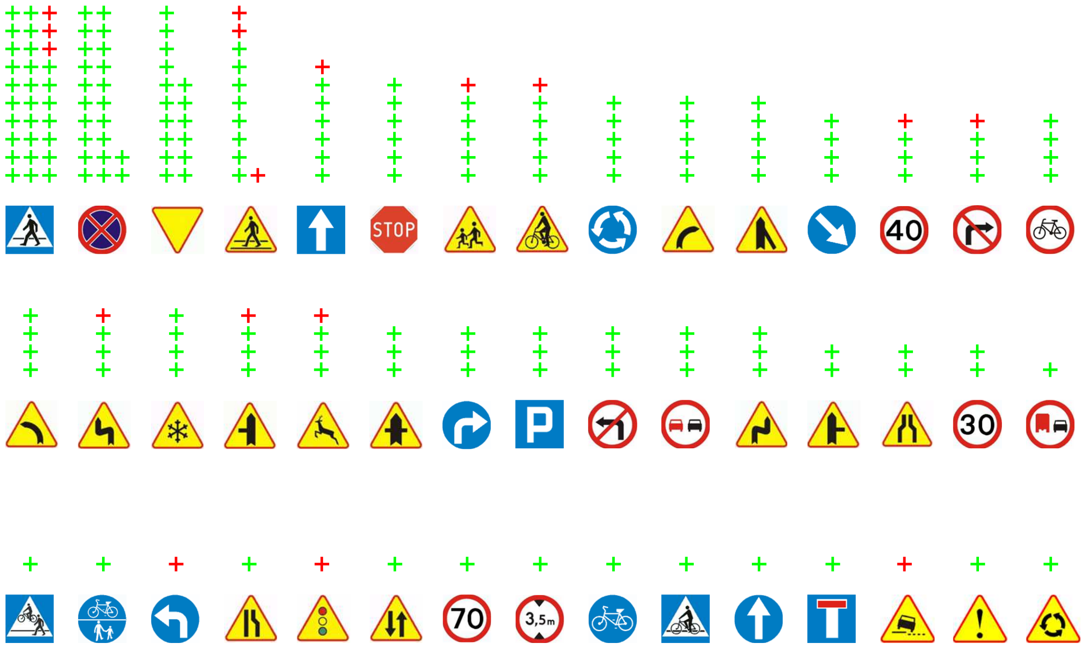


Table 5.3: Breakdown of the classification results for each class. Green crosses mean correctly classified instances, red crosses mean misclassified instances. This table is best viewed in colour.

the two: maximising template signs' separability and the reliability of the obtained dissimilarity information in the real-data context. Very low threshold values lead to the selection of very few good regions for a particular model sign. However, such sparse information may not be sufficiently stable to correctly classify a possibly distorted, blurred, or occluded object in a video frame. Very high threshold values on the other hand introduce information redundancy by allowing image regions that contribute little to the uniqueness of a given sign. In a resulting feature space signs simply look more similar to one another and are hence more difficult to tell apart, at an additional cost of the more intense computation.

After closer investigation we observed that most classification errors resulted from the confusion between the nearly identical classes, e.g. these shown in Figure 5.7. Differences between such signs were found difficult to capture, even in the discriminative feature space, resulting sometimes in the correct template receiving the second best score. However, note in Table 5.3 that the percentages of correctly classified instances of these "hard" classes are still surprisingly high. Certain number of misclassifications were caused by the motion blur or the inaccurate sign detection, which were in turn a result of car vibration affecting the stability of the camera mount. It is common when a vehicle moves on an uneven road surface. Gaussian mixture-based colour binning appeared to be relatively resilient to variations of illumination, leading directly to failure in only several cases when the signs were located in a very shady area or were themselves of poor quality. This can be a proof of usefulness of the GMM colour modelling. In a few cases the vehicle was moving directly towards the bright sun. The destroyed colour appearance information made it difficult not only for the GMM classifier to make a correct colour segmentation of the observed sign, but also for the driver to recognise the pictogram. The corresponding sequences were treated as too challenging and were therefore not used for testing.

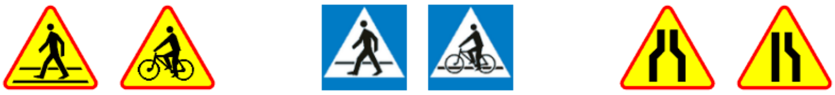


Figure 5.7: Example pairs of nearly identical signs that are sometimes confused by the classifier.

As indicated in Section 5.2.3, delimitation of sign's contour and subsequent colour discretisation in the early video frames are usually less accurate. Extensive experiments have shown that frequently the correct decision is developed by the classifier from just a few most recent frames where the sign's shape and colours are the most unambiguously determined. This fact provides justification for our exponential observation weighting which is used to promote the most recent measurements. Apparently, the classification accuracy with the temporal weighting enabled is by 10-20% higher, depending on the weight base b used. Figure 5.8 gives an illustrative example of how the temporal weights influence classification. In the sample charts a ratio of the cumulative distance from the best-matching template to the cumulative distance from the second best-matching template is shown. It is clear that the weighting shifts this ratio towards the desired lower bound of the interval $(0, 1)$.

5.3 Pairwise Similarity-based Recognition

Classifiers are often designed to distinguish between just two object classes, the second class being understood as not representing the interest class. An example of such a binary problem is a detection of humans in an image which involves discriminating between the humans and the background, i.e. "everything else". As long as a binary discriminant function is based on the estimated class density, or at least produces directly comparable numerical output, extending the binary classification to the multi-class problems, like TSR, without increasing the classifier's complexity is trivial. Such a generalisation merely requires additional responses to be computed for the extra classes, but in principle the decision-making process via selection of the class associated with the maximum response remains unchanged.

Unfortunately, the classifiers based on fitting a decision boundary or maximising the between-class margin are not subject to such a straightforward generalisation as their outputs cannot be directly compared. Such classifiers in order to learn the sophisticated discriminant function must necessarily grow in complexity. For example, a perceptron must be equipped with the additional hidden neuron layers. The support vector machines (SVM) require re-formulation of the notion of margin [42]. Similarly, a generalised AdaBoost algorithm [146] utilises a more complex multi-class exponential loss function, as opposed to the two-class version of the loss function used in the original AdaBoost formulation.

In this section the idea of learning visual similarity from image pairs labelled either "same" or "different", proposed by Nowak and Jurie [153], is developed.

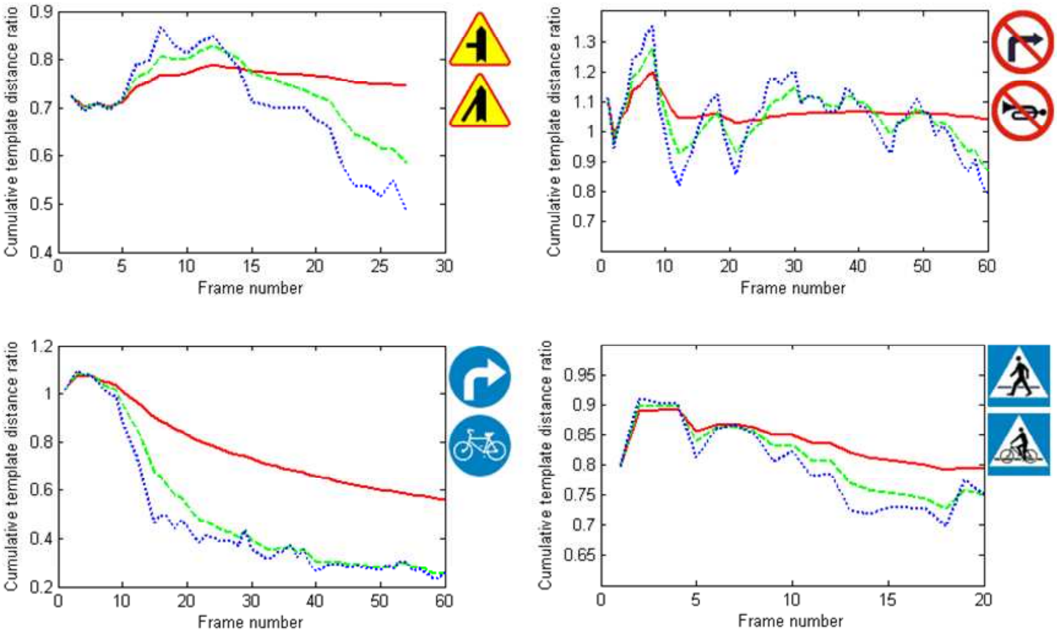


Figure 5.8: Classification of traffic signs over time. Ratio of the cumulative distance from the best-matching template (upper sign next to each chart) to the cumulative distance from the second best-matching template (lower sign next to each chart) is marked with a solid red line. The same but temporally weighted cumulative distance ratio is marked with a green dashed line for the weight base $b = 0.8$, and a blue dotted line for the weight base $b = 0.6$.

Automatic derivation of such a soft similarity measure from a number of training images makes it possible to solve the object recognition problems involving any number of classes using a simple classification scheme, such as the nearest neighbour rule. This concept is formalised in Section 5.3.1. Moreover, as in many practical problems the equivalence information is also cheaper to obtain than the exact class labels, the proposed approach becomes a good alternative to the classic multi-class classifiers, such as traditional decision trees or neural networks.

Two realisations of the pairwise similarity learning idea are introduced: *SimBoost* and the kernel regression trees. Both algorithms combine the local distances between the paired images, measured with respect to the chosen image descriptors. The first technique is an adaptation of the Schapire and Singer's extended boosting algorithm using the confidence-rated predictions [60] to the task of learning similarity from image pairs. It is discussed in Section 5.3.2. Kernel regression trees are detailed in Section 5.3.3. They encompass the concept of fuzziness in the kernel-based node splitting rules where the optimal split features and kernel function parameters are learned in the training process. For better performance, our trees are combined into bagged and boosted ensembles as well as random forests.

In Section 5.3.4 an experimental evaluation of both similarity learning techniques is presented in the context of TSR. The nearest neighbour classifiers are constructed based on the learned similarity measures and tested on both, static images and road traffic video. Prospects for integrating the similarity-based classifier in a real visual driver assistance system are outlined. Both similarity learning methods are suitable for solving other visual recognition problems where a number of potentially very similar classes are involved. In Section 5.3.5 evidence is given to support this claim. The algorithm generating more robust traffic sign classifiers is evaluated on several auxiliary datasets including cars and faces.

5.3.1 Learning and Classification from Image Pairs

Assume, we have an image of an unknown object, I_j, belonging to the one of N classes, $C_1, \ldots, C_N$, each represented by a prototype image, I_{C_i}. Let also each image be represented by some number of feature descriptors, f_k, where each f_k can be any function of the image, unrestricted in terms of the type and the dimensionality of the value returned. For example, f_k can be a scalar-valued Haar wavelet descriptor [111] dependent on the wavelet type, position and scale, or a histogram of oriented gradients (HOG) [132] parametrised by window position, scale, and the number of bins. In order to be able to classify an unknown image, we need to define two distance/similarity metrics.

The local distance between two images, I_1, I_2, with respect to a descriptor f_k is generically defined as:

$$d_k(I_1, I_2) = \phi(f_k(I_1), f_k(I_2)) \ , \tag{5.10}$$

where function ϕ takes the values returned for both input images by the chosen image descriptor and yields a scalar output. The global similarity between the images is denoted by $S(I_1, I_2)$ and can be any scalar-valued function of I_1 and I_2. In this work we are interested in the global similarity functions based on the local distances, i.e.:

$$S(I_1, I_2) = F(d_1(I_1, I_2), \ldots, d_n(I_1, I_2)) \ . \tag{5.11}$$

An example of function F is a linear combination of local features which has been successfully used within the AdaBoost framework [60]. In the next section we will show how to efficiently utilise this particular method for construction of a robust road sign similarity measure. In Section 5.3.3 we will consider the case where F is a non-linear function modelled by a fuzzy regression tree.

Assume that a road sign image I_j has to be classified into one of N categories, $C_1, \ldots, C_N$. Employing our global similarity measure S, this multi-class problem can be solved by pairing I_j with the prototype image of each class I_{C_i}, $i = 1, \ldots, N$ and picking the label maximising the similarity between them:

$$L(I_j) = \arg\max_i S(I_j, I_{C_i}) \ . \tag{5.12}$$

The choice of prototype images may be arbitrary or random. In the case of traffic sign recognition the prototypes might be taken from a highway code. This ensures that the template images are perfectly clean. However, such prototypes usually depart from the natural appearance of the traffic signs, which are frequently poorly illuminated and blurred, especially in low-resolution video captured from a moving vehicle. Alternatively, class prototypes may be chosen randomly from a set of natural road sign images that were not used for learning of the similarity function. The influence of the prototype choice strategy on the robustness of the learned class similarity will be demonstrated in Section 5.3.4.

The above classification scheme can be easily extended to the video-based recognition where the individual frame observations need to be fused over time. A simple strategy adopted here is similar to the one from Section 5.2.3 where the consecutive frame observations were assumed independent. Namely, label of a tracked candidate road sign at time t is determined from:

$$L(\mathbf{I}_{1,\ldots,t}) = \arg\max_i \sum_{k=1}^{t} q(k) S(I_k, I_{C_i}) \ , \tag{5.13}$$

where $q(k)$ are the temporal weights defined in the same way as in equation (5.9).

5.3.2 SimBoost

Let $\mathbf{x} = (I_1, I_2)$ denote a pair of images which might belong to only two classes: "same" or "different". The pairs representing the same pictograms are labelled $y = 1$ (positive), and the pairs representing two different pictograms are labelled $y = -1$ (negative). Assume that $F(\mathbf{x})$ is a real-valued discriminant function separating these two classes. We will now show that F can be learned using a modified AdaBoost algorithm introduced in [104], which we call *SimBoost*.

Function F is defined as a sum of local image features ψ_k:

$$F(I_1, I_2) = \sum_{k=1}^{N} \psi_k(I_1, I_2) \ . \tag{5.14}$$

Each local feature evaluates to:

$$\psi_k(I_1, I_2) = \begin{cases} \alpha & \text{if } \phi(f_k(I_1), f_k(I_2)) < t_k \\ \beta & \text{otherwise} \end{cases} \ , \tag{5.15}$$

where f_k is a filter defined over a chosen class of image descriptors, ϕ is a generic distance metric that makes sense for such descriptors, and t_k is a feature value threshold. In other words, each feature ψ_k quantifies a local dissimilarity between the input images and responds to this dissimilarity depending on whether or not it is sufficiently small to consider the images as representing the same class.

Let $d_k(I_1, I_2) = \phi(f_k(I_1), f_k(I_2))$. Let us also denote by W_+^+ the total weight of these positive training examples that are labelled positive by a given weak classifier ψ_k (true positives), and by W_+^- the total weight of those positive training examples that are labelled negative (false negatives) by this weak classifier. By analogy, let W_-^- and W_-^+ be the total weight of true negatives and false positives respectively. In other words:

$$\begin{array}{llll} W_+^+ = & \displaystyle\sum_{\substack{j:\, y_j = 1 \\ \wedge\, d_k(\mathbf{x}_j) < t_k}} w_j & W_+^- = & \displaystyle\sum_{\substack{j:\, y_j = 1 \\ \wedge\, d_k(\mathbf{x}_j) \geq t_k}} w_j \\ W_-^+ = & \displaystyle\sum_{\substack{j:\, y_j = -1 \\ \wedge\, d_k(\mathbf{x}_j) < t_k}} w_j & W_-^- = & \displaystyle\sum_{\substack{j:\, y_j = -1 \\ \wedge\, d_k(\mathbf{x}_j) \geq t_k}} w_j \end{array} \ . \tag{5.16}$$

In each boosting round the image descriptor f_k and the threshold t_k are selected so as to minimise the weighted error of the training examples:

$$e_k = \sum_{\substack{j:\, y_j = 1 \\ \wedge\, d_k(\mathbf{x}_j) \geq t_k}} w_j + \sum_{\substack{j:\, y_j = -1 \\ \wedge\, d_k(\mathbf{x}_j) < t_k}} w_j = W_+^- + W_-^+ \ . \tag{5.17}$$

Secondly, the optimal values of α and β are found based on the Schapire and Singer's criterion [61] of minimising:

$$Z_k = \sum_{j=1}^{M} w_j e^{-y_j \psi_k(\mathbf{x}_j)} \ , \tag{5.18}$$

where M is the total number of training examples. First, the sum is split as follows:

$$\begin{aligned} Z_k = & \sum_{j:\, y_j=1} w_j e^{-\psi_k(x_j)} + \sum_{j:\, y_j=-1} w_j e^{\psi_k(x_j)} = \\ = & \sum_{\substack{j:\, y_j=1 \\ \wedge\, d_k(\mathbf{x}_j)<t_k}} w_j e^{-\alpha} + \sum_{\substack{j:\, y_j=1 \\ \wedge\, d_k(\mathbf{x}_j)\geq t_k}} w_j e^{-\beta} + \\ + & \sum_{\substack{j:\, y_j=-1 \\ \wedge\, d_k(\mathbf{x}_j)<t_k}} w_j e^{\alpha} + \sum_{\substack{j:\, y_j=-1 \\ \wedge\, d_k(\mathbf{x}_j)\geq t_k}} w_j e^{\beta} = \\ = & W_+^+ e^{-\alpha} + W_+^- e^{-\beta} + W_-^+ e^{\alpha} + W_-^- e^{\beta} \end{aligned} \quad . \tag{5.19}$$

Taking partial derivatives of Z_k with respect to α and β and setting each to zero determines the optimal values of each parameter to be set in a given boosting round:

$$\alpha = \tfrac{1}{2} \log\left(\frac{W_+^+}{W_-^+}\right) \qquad \beta = \tfrac{1}{2} \log\left(\frac{W_+^-}{W_-^-}\right) \quad . \tag{5.20}$$

The remaining steps of the *SimBoost* procedure resemble those known from the classical AdaBoost algorithm. Specifically, weights of the input image pairs are updated after each boosting round using the well-known exponential loss function:

$$w_j^{(t+1)} = w_j^{(t)} e^{-\psi_k(\mathbf{x}_j) y_j} \quad , \tag{5.21}$$

where ψ_k denotes the feature selected in the current round. Weights of all training examples are then normalised. One point requires special attention. Note that the total number of possible input pairs for K classes, each containing N images, is $\frac{1}{2}(NK-1)NK$, while the total number of “same” pairs is only $\frac{1}{2}K(N-1)N$. First, the number of all possible input pairs is too large to be used for training, which calls for sampling. Second, the large quantitative imbalance between the number of positive and negative pairs implies that random sampling is inappropriate because it would carry a risk of very few positive pairs being selected. Enforcing more “same” than “different” pairs could on the other hand bias the classifier. The solution to this problem proposed in [104] is as follows. We first define the cumulative example weight:

$$c_k = \sum_{j=1}^{k} w_j \quad . \tag{5.22}$$

Given the total of M examples and the target sample size S, cumulative weights $c_1, \ldots, c_M$ are computed. Then, S times a random number, r_S, is generated on the interval $[0, c_M]$. The example $\mathbf{x}_k$ with index k satisfying $c_k < r_S < c_{k+1}$ is assigned weight equal to r_S and put in the target sample. This is equivalent to choosing it multiple times, each with weight 1.

5.3.3 Similarity-Learning Kernel Regression Trees

We maintain the notion of data examples $\mathbf{x} = (I_1, I_2)$ being pairs of images, but change the possible labels to $y(\mathbf{x}) = 1$ if the images represent the same class of signs, and $y(\mathbf{x}) = 0$ if they represent different classes. Recall that our goal is to construct a real-valued discriminant function $F(\mathbf{x})$ separating the two classes. This function can be realised by a fuzzy regression tree similar to those proposed by Olaru and Wehenkel [90]. Below we develop the regression trees that are tailored to learning from image pairs and where the concept of fuzziness is generalised by the usage of kernel-based node splits. We call such trees *Similarity-Learning Kernel Regression Trees* (S-KRT). Their semantics is described in Section 5.3.3.1. Section 5.3.3.2 is devoted to the training procedure which is used to grow and prune the S-KRT trees. In Section 5.3.3.3 we combine the trees into robust bagging and boosting ensembles and random forests with the aim of improving their predictive capability.

5.3.3.1 Tree Semantics

An example of a simple hypothetical regression tree is shown in Figure 5.9. The tree has 4 test nodes and 5 terminal nodes. Each node N_j is assigned a label $L_{N_j} \in [0, 1]$ expressing its local estimation of the output. This estimation, at any level of the tree, represents the likelihood of the input pairs of images representing the same class. Each test node is additionally assigned a single scalar-valued feature, d_k, which quantifies the local dissimilarity between the paired images with respect to the underlying image descriptor f_k:

$$d_k(I_1, I_2) = \phi(f_k(I_1), f_k(I_2)) \ . \tag{5.23}$$

ϕ means here a distance metric which is adequate to the type and the dimensionality of the selected image descriptor's responses. This feature is used to determine to what extent the tested example should be directed to the left and right subtrees.

A fuzzy split of the local input space at the node is done using a univariate kernel function $\kappa_{N,k}$, where k denotes the splitting dimension, i.e. index of the feature in a large feature vector assembled from an overcomplete space of local image distances defined with respect to all possible descriptors of a chosen type. In Table 5.4 we give three examples of suitable kernel functions. Their shapes are illustrated in Figure 5.10.

As the local input space in each test node may be split into two partly or entirely overlapping subspaces, an input example is potentially propagated to both successor nodes in parallel and hence does not strictly belong to any of them. Instead, each example is assigned a degree of membership to each node N, $\mu_N(\mathbf{x}) \in [0, 1]$. The degree of membership to the root node, $\mu_R(\mathbf{x})$, is by default set to unity, which reflects the fact that all examples are equally likely to belong to it. The degree of membership to the child nodes N_L and N_R of a given node N is defined recursively:

$$\begin{array}{l} \mu_{N_L}(\mathbf{x}) = \mu_N(\mathbf{x})\kappa_{N,k}(\mathbf{x}) \\ \mu_{N_R}(\mathbf{x}) = \mu_N(\mathbf{x})(1 - \kappa_{N,k}(\mathbf{x})) \end{array} \ . \tag{5.24}$$

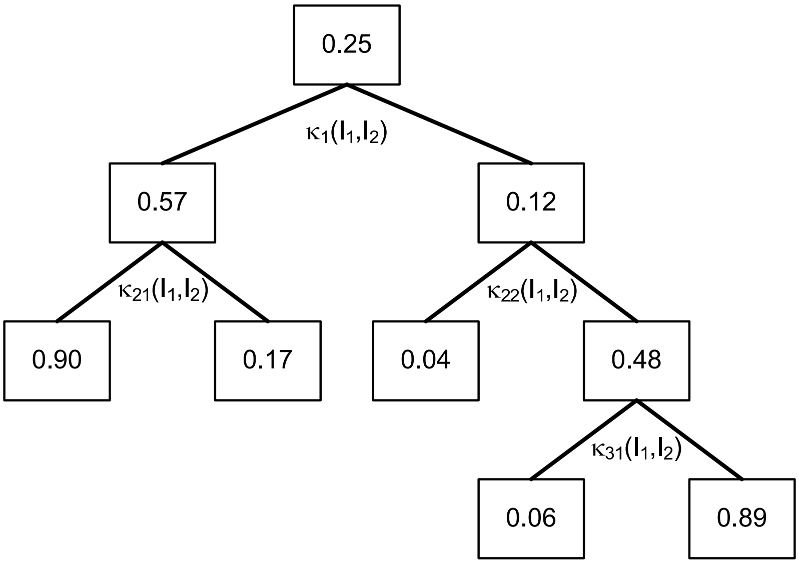


Figure 5.9: Example of a kernel regression tree.

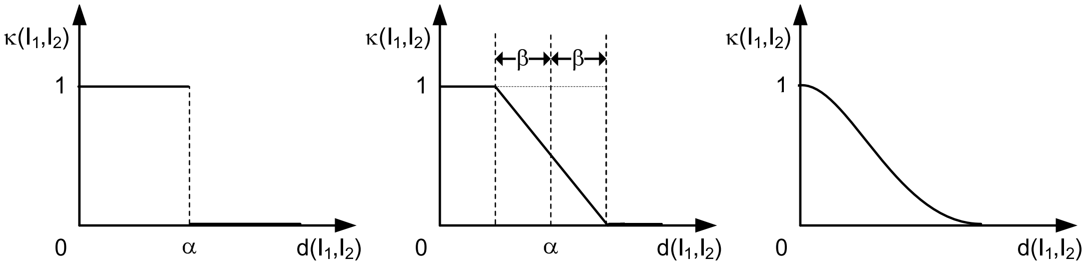


Figure 5.10: Step (left), piecewise linear (centre), and Gaussian RBF (right) kernel functions used to split the local input space in a single node of the S-KRT tree.

discriminant type	formula
step	$\kappa_{N,k}(I_1,I_2)=\begin{cases}1 & \text{if } d_k(I_1,I_2)<\alpha\\ 0 & \text{if } d_k(I_1,I_2)>=\alpha\end{cases}$
piecewise linear	$\kappa_{N,k}(I_1,I_2)=\begin{cases}1 & \text{if } d_k(I_1,I_2)<\alpha-\beta\\ \frac{\alpha+\beta-d_k(I_1,I_2)}{2\beta} & \text{if } d_k(I_1,I_2)\in[\alpha-\beta,\alpha+\beta]\\ 0 & \text{if } d_k(I_1,I_2)>\alpha+\beta\end{cases}$
Gaussian RBF	$\kappa_{N,k}(I_1,I_2)=e^{-\alpha d_k(I_1,I_2)^2}$

Table 5.4: Three possible kernel functions used to split the examples passing through each test node of a S-KRT tree.

Finally, each example passed on input of the tree may follow multiple paths at once. Therefore, in order to estimate the ultimate tree's response for a given input example, the products of the terminal node labels and their membership degrees for this example have to be summed:

$$\hat{y}(\mathbf{x})=\frac{\sum_{j\in\text{leaves}}\mu_{N_j}(\mathbf{x})L_{N_j}}{\sum_{j\in\text{leaves}}\mu_{N_j}(\mathbf{x})} \quad . \tag{5.25}$$

5.3.3.2 Training

Learning the S-KRT trees requires two independent set of examples: growing set and pruning set. In a growing phase it is necessary to define: 1) a rule for automatic determination of the split feature d_k, kernel parameters at each node, and the labels of the successors of this node, and 2) a stopping criterion for the growing process. Let us now briefly discuss both issues.

Given a node N, our objective is to find the split feature and the kernel function's parameters that minimise the squared error function:

$$E_N=\sum_{\mathbf{x}\in N}w(\mathbf{x})\mu_N(\mathbf{x})(y(\mathbf{x})-y'(\mathbf{x}))^2 \ , \tag{5.26}$$

where

$$y'(\mathbf{x})=\kappa_{N,k}(\mathbf{x})L_{N_L}+(1-\kappa_{N,k}(\mathbf{x}))L_{N_R} \ , \tag{5.27}$$

and $w(\mathbf{x})$ is a normalised weight of the training example (by default all examples' weights are equal).

The actual strategy for searching of the minimum of the above error functional depends on the type of the kernel function used to split the examples passing through the considered node. Generally, if it depends on a single parameter and introduces fuzzification, which is the case with the Gaussian RBF function, minimisation of (5.26) is straightforward. To do this, all possible local distances d_k and the values of α parameter are considered. For a fixed image descriptor f_k used to calculate d_k and fixed α, we compute the derivatives of E_N with respect to L_{N_L} and L_{N_R}, and set them to zero:

$$\frac{\partial E_N}{L_{N_L}} = 0 \quad \frac{\partial E_N}{L_{N_R}} = 0 \quad , \tag{5.28}$$

It yields:

$$\begin{array}{l} -2\sum_{\mathbf{x}\in N} w(\mathbf{x})\mu_N(\mathbf{x})\kappa_{N,k}(\mathbf{x})\left[y(\mathbf{x}) - y'(\mathbf{x})\right] = 0 \\ -2\sum_{\mathbf{x}\in N} w(\mathbf{x})\mu_N(\mathbf{x})\kappa'_{N,k}(\mathbf{x})\left[y(\mathbf{x}) - y'(\mathbf{x})\right] = 0 \end{array} \quad , \tag{5.29}$$

where $\kappa'_{N,k}(\mathbf{x})$ stands for $1 - \kappa_{N,k}(\mathbf{x})$. Solving this linear system in L_{N_L} and L_{N_R}, we obtain the formulas for the optimal successor node labels.

$$L_{N_L} = \frac{CD-EB}{B^2-AC} \quad L_{N_R} = \frac{AE-BD}{B^2-AC} \quad , \tag{5.30}$$

where:

$$\begin{array}{l} A = \sum_{\mathbf{x}\in N} w(\mathbf{x})\mu_N(\mathbf{x})\kappa_{N,k}(\mathbf{x})^2 \\ B = \sum_{\mathbf{x}\in N} w(\mathbf{x})\mu_N(\mathbf{x})\kappa_{N,k}(\mathbf{x})\kappa'_{N,k}(\mathbf{x}) \\ C = \sum_{\mathbf{x}\in N} w(\mathbf{x})\mu_N(\mathbf{x})\kappa'_{N,k}(\mathbf{x})^2 \\ D = -\sum_{\mathbf{x}\in N} w(\mathbf{x})\mu_N(\mathbf{x})y(\mathbf{x})\kappa_{N,k}(\mathbf{x}) \\ E = -\sum_{\mathbf{x}\in N} w(\mathbf{x})\mu_N(\mathbf{x})y(\mathbf{x})\kappa'_{N,k}(\mathbf{x}) \end{array} \quad . \tag{5.31}$$

Introducing (5.30) in (5.27) gives a recipe for split error computation. Ultimately, the split feature d_k and the associated kernel parameter α minimising this error are saved.

In the case of a piecewise linear kernel function, which is dependent on two parameters, the above error minimisation strategy would have to consider all possible triples (f_k, α, β), which is computationally expensive. However, it can be noticed that for fixed f_k the pursued global minimum of (5.26) as a function of four parameters, α, β, L_{N_L}, L_{N_R}, is generally very close to its minimum for any fixed β. Therefore, minimisation of E_N can be performed sequentially. First, the optimal threshold α is found for a zero fuzzy interval radius β, as though the split was completely crisp. Then, for the found value of α the best value of β is searched for.

With $\beta = 0$ our error function can be written as:

$$\begin{array}{rl} E_N = & E_{N_L} + E_{N_R} = \\ & \sum_{\mathbf{x}\in N_L} w(\mathbf{x})\mu_N(\mathbf{x})(y(\mathbf{x}) - L_{N_L})^2 + \\ & \sum_{\mathbf{x}\in N_R} w(\mathbf{x})\mu_N(\mathbf{x})(y(\mathbf{x}) - L_{N_R})^2 \end{array} \quad . \tag{5.32}$$

To find minima of (5.32), we compute the derivatives of E_{N_L} and E_{N_R} with respect to L_{N_L} and L_{N_R} respectively and set them to zero, which yields the formulas for the temporarily optimal successor node labels:

$$\begin{array}{l} L_{N_L} = \frac{\sum_{\mathbf{x}\in N_L} w(\mathbf{x})\mu_N(\mathbf{x})y(\mathbf{x})}{\sum_{\mathbf{x}\in N_L} w(\mathbf{x})\mu_N(\mathbf{x})} \\ L_{N_R} = \frac{\sum_{\mathbf{x}\in N_R} w(\mathbf{x})\mu_N(\mathbf{x})y(\mathbf{x})}{\sum_{\mathbf{x}\in N_R} w(\mathbf{x})\mu_N(\mathbf{x})} \end{array} \quad . \tag{5.33}$$

Subsequently, by incorporating (5.33) in (5.32), the formula for computing the node error is derived. With a fixed feature d_k and threshold α, minimisation of the error function (5.26) for variable β proceeds in exactly the same way as was discussed above for the Gaussian RBF kernel. Note that in the case of a step kernel, which is a special case of a piecewise linear one, the same derivations are valid, but with the β optimisation step omitted.

It should be noted that the above tree growing process is recursive. We start with the root node, R, by setting its label to the value equal to the fraction of the positive examples in the growing dataset, i.e. the prior probability of the input example representing the target class. Also, all $\mu_R(\mathbf{x})$ values are set to unity. Then we search for the optimal split feature and split kernel parameters of the root node which results in the two child nodes, N_{1L}, N_{1R}, labelled according to (5.30) or (5.33). Having computed the degree of membership of the examples to these two nodes using (5.24), we split them in turn. The process is continued until the stop condition is met. Among many possible stop rules, we choose the one based on the minimum acceptable reduction of the squared node error in the meaning of (5.26). In practice, this error reduction threshold is chosen such that the tree reaches up to 9-10 levels of depth.

As it is a standard practice in the decision/regression tree learning, a separate set of examples, $\mathbf{x}_i$, $i = 1, \ldots, M$, is used to prune the tree. Pruning is carried out in the following way. The grown tree is fed with all the pruning examples and the products $\mu_{N_j}(\mathbf{x}_i)L_{N_j}$ are stored for all nodes, as well as the tree classification rate defined as:

$$c_T = \sum_{i=1}^{M} w(\mathbf{x}_i)(1 - |y(\mathbf{x}_i) - \hat{y}(\mathbf{x}_i)|) \ . \tag{5.34}$$

Subsequently, we simulate collapsing a subtree rooted at each non-terminal node by replacing it with a terminal node with the same label. Each smaller tree obtained is evaluated using (5.34) and the stored membership degree-label products. The tree leading to the maximum increase in the classification rate over the pruning examples is saved and the process is continued until no further improvement can be made.

5.3.3.3 Combining Trees

Preliminary experiments with various datasets showed that the performance of the classifiers based on the proposed kernel regression trees can be improved by combining multiple independently learned trees. We consider here three types of combination: bagging [44], boosting [60], and random forests [59].

In a tree ensemble trained via bagging multiple trees are grown and pruned using the independent subsets of the entire training set. The response of the ensemble for a given image pair is computed as a mean of class membership degrees for this pais defined in (5.25) over all contained trees. In the case of boosting, we utilise the well-known weighting-normalisation scheme that assigns more importance to the examples previously misclassified by underweighting those classified correctly.

Specifically, in the $t+1$-th boosting round the example's weight, used in equations (5.26), (5.29), and (5.31-5.34), is updated according to:

$$w_{t+1}(\mathbf{x}_i) = w_t(\mathbf{x}_i)\lambda_t^{1-e_i} \ , \tag{5.35}$$

where $e_i = 0$ if the pair has been correctly classified in a crisp sense, i.e. $\hat{y}(\mathbf{x}_i) > 0.5$ for $y(\mathbf{x}_i) = 1$ and $\hat{y}(\mathbf{x}_i) \leq 0.5$ for $y(\mathbf{x}_i) = 0$, $e_i = 1$ otherwise, $\lambda_t = \frac{1-c_T}{c_T}$, and c_T is the classification rate of the tree defined in (5.34).

The idea of random forest combines the Breiman's bagging with the random subspace method. Specifically, like in bagging, an ensemble of trees is constructed where each tree is learned using an independent, usually random subset of all available training examples. Moreover, in each non-terminal node of each tree not all possible image descriptors are considered as potential predictors. Instead, the best split feature is found among the local image distances defined with respect to only some number of randomly selected descriptors. Finally, the trees in the forest have to be fully grown, but need not be pruned, which facilitates learning.

5.3.4 Experimental Results in Traffic Sign Recognition

In this section an experimental evaluation of the similarity learning algorithms, *SimBoost* and the kernel regression trees, is presented in the context of traffic sign recognition. In visual driver assistance the ultimate goal is not to quantify similarity between any signs, but to assign correct meanings to the pictograms of signs captured with an in-vehicle camera. Therefore, the similarity learning process itself is treated as a prerequisite and is hence not subject to experimentation. Instead, the algorithms introduced in Sections 5.3.2 and 5.3.3 are put into a classification framework. Specifically, a nearest neighbour classifier is built upon the similarity measure learned using each algorithm and is tested on the unseen static road sign images as well as live traffic video. Section 5.3.4.1 is entirely devoted to the discussion of the test datasets and the testing strategy. Classification results obtained in the experiments presented below are described in Section 5.3.4.2.

5.3.4.1 Datasets and Experimental Setup

In order to evaluate the presented traffic sign recognition approach, a sign similarity function was learned from the pairs of images cropped from the individual frames of real-life video sequences. These sequences were captured from a moving vehicle in cluttered Japanese street scenes. The quality of the input images varied from very good (short distance from the camera, good illumination, bright colours, good contrast) to very poor (adverse illumination, pale colours, motion blur). The size of the original data was 8473 training images and 9036 test images, all representing 17 types of road signs shown in Figure 5.11. It is important to note that our dataset was unbalanced, i.e. the numbers of images representing different classes varied. This reflects the frequent occurrence of certain signs in the traffic environment and the rarity of others.

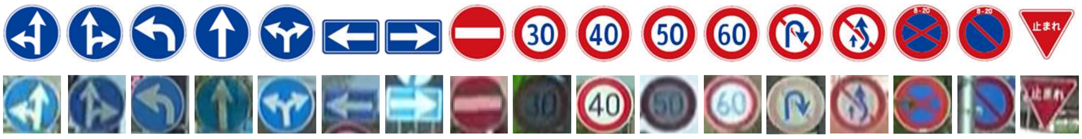


Figure 5.11: Highway code templates and example natural images representing different Japanese road sign classes in our dataset.

The sign similarity measure was independently learned from the training images using both proposed methods: *SimBoost* and the *Similarity-Learning Kernel Regression Trees* framework. For boosted similarity we set the number of local feature distances to be incorporated to 100. To make the training process tractable, the number of training image pairs was reduced by resampling described in section 5.3.2. This resampling was triggered every 5 boosting rounds on both positive and negative pairs such that only 10 randomly selected images of each class C_i were used to assemble the "same" pairs, and only 5 random images from each of the remaining classes C_j, $j \neq i$ were used to construct the "different" pairs for a given C_i. Separate classifiers were trained using: 1) natural image pairs and 2) pairs comprising one natural image and one highway code template image.

An ensemble of kernel regression trees was trained via bagging/boosting combined with cross-validation [4] on the training set in the following way. First 40% of the images of each sign were chosen to build a growing set, and all other images were put in a pool used to construct a pruning set. Further sampling was done randomly five times on such partitioned data, respecting the proportions of the images representing the classes in the original data, in order to limit the number of image pairs passed on input of each tree to approximately 2000. In each final input set the fraction of "same" pairs was kept to ≈ 0.15. Having grown and pruned the first five trees, another 40% of images in each class, half-overlapping the last growing portion, were used for the new growing set construction, against the other 60% used for the new pruning set construction. Using such a training scheme, an ensemble of 4×5 trees were produced and their responses were averaged when evaluating the classifier on the test set. Again, the classifier was trained separately from natural image pairs and the pairs containing one realistic image and one idealised class prototype.

Evaluation of both classifiers was done using two strategies, depending on the training scheme adopted. To test the classifiers trained from natural image pairs, a large pool of test examples were constructed by picking each image from each class C_i, $i = 1, \ldots, N$ and pairing it with a randomly selected image from each other class $C_j, j \neq i$, as well as with a randomly selected image from the same class C_i, but ensuring no same two images were selected. A single test example was classified correctly if among N pairs, one "same" and $N-1$ "different", the "same" pair was assigned the highest degree of similarity. To avoid bias resulting from random selection of the second image in each test pair, three tests runs per classifier were performed and the recognition rates were averaged. In the second, deterministic

[4] Random forest was not included in the analysis. As the trees in the forest are not pruned, the forest is computationally expensive and hence unsuitable for real-time driver assistance.

method, every single test image from the test set was coupled with all 17 template sign images from a Japanese highway code. This testing scheme was used to evaluate the classifiers trained using mixed, natural-synthetic image pairs.

Test results were additionally split according to the image representation, i.e. the type of the low-level image descriptors used. Three types were considered: Haar wavelets [111], Histograms of Oriented Gradients (HOG) [132], and region covariances [145], as well as a mixture of Haar and HOG features [5]. In the first case the input feature space was populated by 5 types of Haar wavelet filters, the same as those previously discussed in Section 3.3.2 and illustrated in Figure 3.8. They capture certain horizontal, vertical, and diagonal structures of the underlying image and are additionally parametrised by colour, as proposed by Bahlmann et al. [127]. The size of each rectangular part of the filters satisfied $w, h = \{4\,\text{px}, 8\,\text{px}\}$ and the filters were shifted by half of that size along each dimension.

In the case of HOGs, we used 6-bin histograms computed within regions of scale $w, h = \{10\,\text{px}, 20\,\text{px}\}$ shifted by 10 pixels along each dimension. In the latter case, a pool of all 4-feature covariance matrices were constructed in the same regions as HOGs. The matrices combined x and y positional coordinates and the first-order image derivatives along horizontal and vertical axis. The local distance metrics, ϕ used in (5.10) for the aforementioned descriptors are listed in Table 5.5. It is important to note that with the recent algorithmic developments made in visual feature extraction based on integral images: [111] (Haar wavelets), [135] (histograms), and [145] (region covariances), all three types of image descriptors can be evaluated in a very efficient way.

feature type	output value	distance formula
Haar	scalar v	$\lvert v_2 - v_1 \rvert$
HOG	vector $\mathbf{v} \in \mathbb{R}^n$	$\sqrt{\sum_{i=1}^{n}(v_{2,i} - v_{1,i})^2}$
Covariance	matrix $C_{n \times n}$	$\sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(C_1, C_2)}$, where $\{\lambda_i(C_1, C_2)\}_{i=1,\ldots,n}$ are the generalised eigenvalues of $\lambda C_1 \mathbf{x}_i - C_2 \mathbf{x_i} = 0$ (see [145] for details)

Table 5.5: Local distance metrics associated with different image descriptors that were used within the *SimBoost* and S-KRT frameworks to learn road sign similarity function.

[5] Sign similarity from a joint pool of Haar and HOG features was learned only via *SimBoost*. Learning a fuzzy regression tree ensemble using both descriptor types was omitted due to the very long training process.

5.3.4.2 Test Results

Correct classification rates obtained for the 17-class Japanese road signs problem are shown in Table 5.6 [6]. These rates are additionally compared to the results obtained using two alternative techniques: 1) PCA-TM, template matching in a PCA-reduced feature space built from concatenated regularly-spaced HOGs and 2) class-specific template matching approach (CSTM) discussed in Section 5.2, but modified to use HOGs and the Euclidean metric to measure the dissimilarities between the discriminative local image regions. In Table 5.7 we have compared the performance of the classifiers based on the trainable similarity learned using the natural image pairs and the mixed natural-template image pairs. The comparison with respect to the region covariance features has been omitted. The reason is that the evaluation of the associated distance metric based on the generalised eigenvalues fails in presence of singular covariance matrices which are often generated when clean model images are used. Figure 5.12 illustrates the influence of boosting on the classifiers' accuracy. Additionally, in Figure 5.13 the confusion matrices have been shown that correspond to the best-performing combinations of the proposed learning algorithms and the image descriptors used in this experimental evaluation.

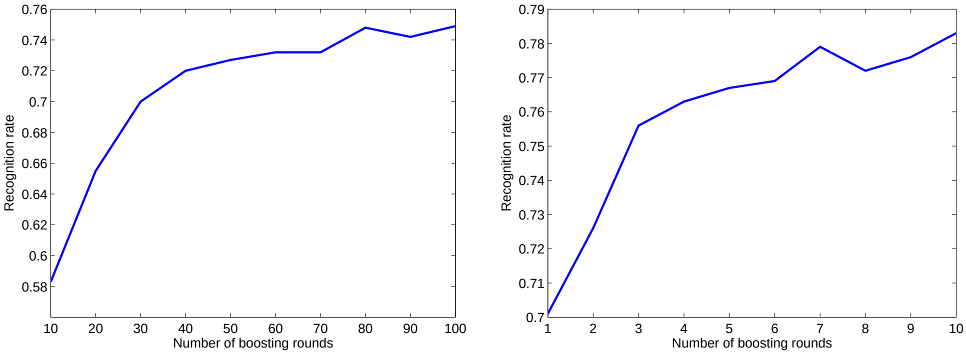
feature type	PCA-TM	CSTM	SimBoost	4 × 5 bagged S-KRT	4 × 5 boosted S-FRT
Haar	X	X	62.4%	51.9%	57.7%
HOG	22.3%	74.3%	74.7%	74.5%	76.7%
Covariance	X	X	54.4%	47.0%	54.1%
Haar & HOG	X	X	74.9%	X	X

Table 5.6: Classification rates obtained for a 17-class Japanese traffic signs problem using different methods: PCA-TM, CSTM, *SimBoost*, and the S-KRT framework.

feature type	SimBoost		4 × 5 bagged S-KRT		4 × 5 boosted S-KRT	
	natural	mixed	natural	mixed	natural	mixed
Haar	62.4%	42.5%	51.9%	49.1%	57.7%	52.6%
HOG	74.7%	61.7%	74.5%	74.0%	76.7%	71.9
Haar & HOG	74.9%	64.0%	X	X	X	X

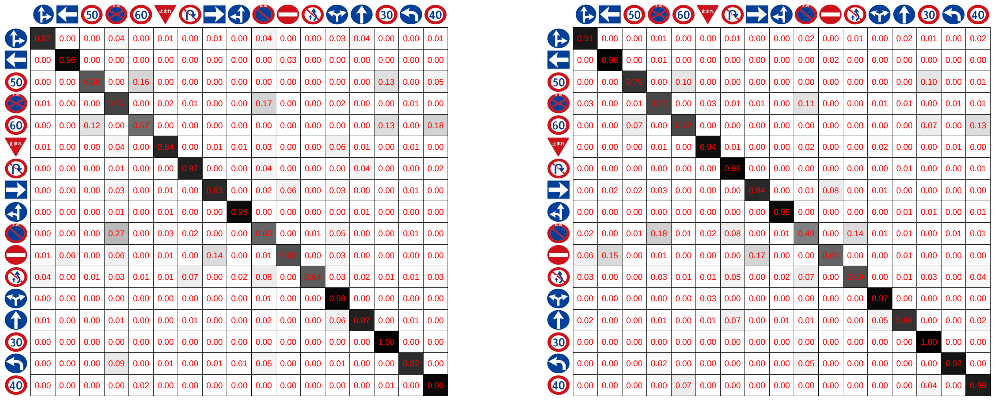
Table 5.7: Comparison of the recognition rates of the road sign classifiers based on a trainable similarity measure learned from 1) natural image pairs and 2) mixed pairs consisting of one natural image and one clean template image taken from the highway code.

[6] Piecewise linear kernel functions were used for node splitting in the S-KRT trees. Gaussian RBF kernels, although lead to the similar recognition accuracy, are more expensive to evaluate and generate much larger trees. Therefore, they are less suitable for integration in a real-time TSR system.



(a) (b)

Figure 5.12: Influence of the increasing number of boosting rounds on the performance of the road sign classifier trained using: a) *SimBoost* with mixed Haar and HOG features, b) S-KRT ensemble with piecewise linear splitting kernels and HOG features.



	↱	⬅	50	⊗	60	止まれ	⮌	➡	↰	⊘	⛔	🚫	⑂	⬆	30	↶	40
↱	0.82	0.00	0.00	0.04	0.00	0.01	0.00	0.01	0.00	0.04	0.00	0.00	0.03	0.04	0.00	0.00	0.01
⬅	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00
50	0.00	0.00	0.66	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.05
⊗	0.01	0.00	0.00	0.74	0.00	0.02	0.01	0.00	0.00	0.17	0.00	0.00	0.02	0.00	0.00	0.01	0.00
60	0.00	0.00	0.12	0.00	0.57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.18
止まれ	0.01	0.00	0.00	0.04	0.00	0.84	0.00	0.01	0.01	0.03	0.00	0.00	0.06	0.01	0.00	0.01	0.00
⮌	0.00	0.00	0.00	0.01	0.00	0.00	0.87	0.00	0.00	0.04	0.00	0.00	0.00	0.04	0.00	0.00	0.02
➡	0.00	0.00	0.00	0.03	0.00	0.01	0.00	0.83	0.00	0.02	0.06	0.00	0.03	0.00	0.00	0.01	0.00
↰	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.95	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
⊘	0.00	0.00	0.00	0.27	0.00	0.03	0.02	0.00	0.00	0.60	0.00	0.01	0.05	0.00	0.00	0.01	0.00
⛔	0.01	0.06	0.00	0.06	0.00	0.01	0.00	0.14	0.00	0.01	0.68	0.00	0.03	0.00	0.00	0.00	0.00
🚫	0.04	0.00	0.01	0.03	0.01	0.01	0.07	0.00	0.02	0.08	0.00	0.64	0.03	0.02	0.01	0.01	0.03
⑂	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.98	0.00	0.00	0.00	0.00
⬆	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.06	0.87	0.00	0.01	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
↶	0.00	0.00	0.00	0.09	0.00	0.01	0.00	0.01	0.01	0.05	0.00	0.00	0.01	0.00	0.00	0.82	0.00
40	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.96

(a)

	↱	⬅	50	⊗	60	止まれ	⮌	➡	↰	⊘	⛔	🚫	⑂	⬆	30	↶	40
↱	0.91	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.01	0.00	0.02	0.01	0.00	0.02
⬅	0.00	0.98	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
50	0.00	0.00	0.79	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.01
⊗	0.03	0.00	0.01	0.72	0.00	0.03	0.01	0.01	0.00	0.11	0.00	0.00	0.01	0.01	0.00	0.01	0.01
60	0.00	0.00	0.07	0.00	0.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.13
止まれ	0.00	0.00	0.00	0.01	0.00	0.94	0.01	0.00	0.00	0.02	0.00	0.00	0.02	0.00	0.00	0.01	0.00
⮌	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01
➡	0.00	0.02	0.02	0.03	0.00	0.00	0.00	0.84	0.00	0.01	0.08	0.00	0.01	0.00	0.00	0.00	0.00
↰	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.01
⊘	0.02	0.00	0.01	0.18	0.01	0.02	0.08	0.00	0.01	0.49	0.00	0.14	0.01	0.01	0.00	0.01	0.01
⛔	0.06	0.15	0.00	0.01	0.00	0.00	0.00	0.17	0.00	0.00	0.60	0.00	0.00	0.01	0.00	0.00	0.00
🚫	0.03	0.00	0.00	0.03	0.01	0.01	0.05	0.00	0.02	0.07	0.00	0.69	0.00	0.01	0.03	0.00	0.04
⑂	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.00	0.00	0.00	0.00
⬆	0.02	0.00	0.00	0.01	0.00	0.01	0.07	0.00	0.01	0.01	0.00	0.00	0.05	0.80	0.00	0.00	0.02
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
↶	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.92	0.00
40	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.89

(b)

Figure 5.13: Confusion matrices obtained for a 17-class Japanese traffic signs problem using: (a) a 100-feature classifier trained via *SimBoost* with mixed Haar and HOG features, and (b) a boosted ensemble of 4 × 10 piecewise linear kernel regression trees with HOG features. This figure is best viewed in colour.

Results of the above experiments show that the proposed similarity learning algorithms generate efficient road sign classifiers. Taking into account the number of classes and the low quality of many test images, nearly 80% recognition rate achieved is a promising result. Note (Figure 5.13) that the most confusions occurred between semantically similar classes, e.g. between speed limits, or between “no stopping” and “no parking” signs. Comparing the two algorithms, *SimBoost* and the KRT-based learning offer similar performance. However, augmenting the regression trees with boosting reduces the error rate of the classifier by extra 3%.

Regarding the choice of feature representation of the images, histograms of oriented gradients provide the most discriminative information about the traffic signs. Other image features could potentially further increase the recognition accuracy. In the case of *SimBoost*, considering overcomplete spaces of colour-parametrised Haar wavelets and HOGs jointly on input of the training algorithm brought a minimal improvement to the accuracy of the classifier. We could expect a similar gain in performance for the S-KRT method if the colour information was incorporated. Note in the right confusion matrix in Figure 5.13 that the classifier based on the sign similarity function learned by a boosted S-KRT ensemble with HOGs confused certain pairs of signs which are similar in terms of gradient orientations, but completely different in terms of colours, e.g. “turn left”, “turn right” and “no entry” signs. Finally, it is apparently better to randomly select the class prototypes from the available real-life traffic sign images than to use clean model images. In the latter case a significant drop in the correct classification rate is observed, 15-30% for *SimBoost*, and up to 10% when the S-KRT framework is used.

An additional experiment has been conducted in order to test the best-performing static image classifier in recognising signs from a real traffic video. The test sequences were captured in Japan and depict challenging, frequently highly cluttered urban street scenes. The classifier was integrated in a TSR system featuring Hough-based detector and the affine regression tracker, as detailed at the beginning of Section 4.4.3. The minimum radius of the circles captured by the detector was set to 10 pixels. Figure 5.14 illustrates the classification results obtained. As seen, an overall error rate of the classifier did not exceed 15%. Misclassifications were mainly caused by the motion blur erasing relevant image gradients, and by the cumulated reconstruction errors of the tracker. The system was able to operate at approximately 30 fps on a modern computer [7].

It is important to consider the potential of the presented traffic sign recognition method in the in-vehicle visual driver assistance (VDA). Regarding the recognition rate of the classifiers evaluated above, it might appear unacceptably low when the extreme-case consequences of every seventh misinterpreted sign are envisaged. However, if the VDA system is only to play a supportive role, without taking responsibility for vehicle manoeuvring, the decision of the classifier can be presented to the driver in a form of confidence rather than a definite class label. In that case that is the human who will make the ultimate decision. In addition, if the ambiguous classifier's decision indicates multiple, but semantically similar likely pictograms,

[7] Sample videos illustrating operation of the system are attached to this dissertation as a supplementary material.

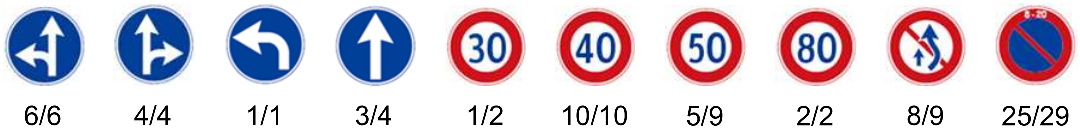


Figure 5.14: Classification accuracy obtained in the video-based recognition experiment. The road sign classifier was based on the sign similarity measure learned with a boosted ensemble of kernel regression trees. The numbers of correctly classified signs of each class are given against the total numbers of such signs detected in the input sequences.

which is usually the case when the signs are confused, a partial classification can be made, e.g. “reduce speed” rather than “50 kph speed limit” or “60 kph speed limit”. Note that although such an incomplete information is obtained, the general message or suggestion made to the driver remains essentially correct. Regarding the execution speed, the current near frame-rate processing has been achieved without any sophisticated software- and hardware-level optimisations. Therefore, real-time implementation is possible.

5.3.5 Experimental Results in Recognition of Other Object

Similarity-learning algorithms presented in Sections 5.3.2 and 5.3.3 are formulated in a sufficiently generic way to allow applying them to a broad class of visual object recognition problems, not necessarily related to traffic signs. In order to estimate the discriminative potential of our approach in such a general-purpose object recognition, the kernel regression trees were used to learn visual similarity from pairs of images representing two types of very similar objects: cars and human faces. As the classifiers based on the similarity learned within the S-KRT framework performed consistently better than the ones utilising the *SimBoost* algorithm, we decided to evaluate here only the kernel regression trees. In the below sections the test datasets and the testing methodology are described in detail and the discussion of the results obtained is given.

5.3.5.1 Description of Datasets

Evaluation of the *Similarity-Learning Kernel Regression Trees* has been done using four image datasets. Two of them: “Yale faces A” and “AT&T faces” are public and more information about them is available in the web [8]. Here only a brief characteristics of these datasets is provided. The “Yale faces A” dataset is comprised of 165 images representing 15 different individuals, 11 images per each individual. For our purposes they are scaled to the size of 150 × 114 pixels. Images representing the same person differ in terms of the facial expression shown and the illumination (intensity and angle). On some images the individuals not normally

[8] “Yale faces A” dataset is available at: `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`, “AT&T faces” dataset is available at: `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`

wearing glasses are shown with glasses to further diversify the range of views of the same person. The “AT&T faces” dataset consists of 400 90 × 110 pixels images of 40 individuals (10 images per subject). In the images representing the same subject the facial expression and the illumination vary, but these changes are not severe. However, the faces of the same person are also shown from slightly varying view angles. Also, some persons are depicted with and without glasses.

Two other datasets were collected by us and depict fronts and rears of 12 models of European and Japanese cars: Alfa Romeo 156, Audi A4, BMW series 3, Ford Mondeo, Honda Accord, Mazda 6, Mercedes C220, Opel Vectra, Renault Laguna, Toyota Avensis, Volkswagen Passat, and Volvo V50. Each model is represented by 30 front and 30 rear images, giving a total of 360 front and 360 rear images, all scaled to the size of 200 × 100 pixels. Images are well aligned and only small camera’s pan and tilt variations were allowed when the pictures were taken. The cars representing the same model differ in terms of body colour and general scene illumination as they were captured at different times of day and in different locations across Europe. For example, most cars were captured outdoors, e.g. on parking lots, and their bodies did not reflect much daylight as it was mostly an ambient light. However, on some images, where the cars are shown in poor illumination or indoors (e.g. in a showroom), many light reflections are observed. This is one of the challenging aspects of our car datasets.

Example images from all datasets above presented are shown in Figure 5.15.

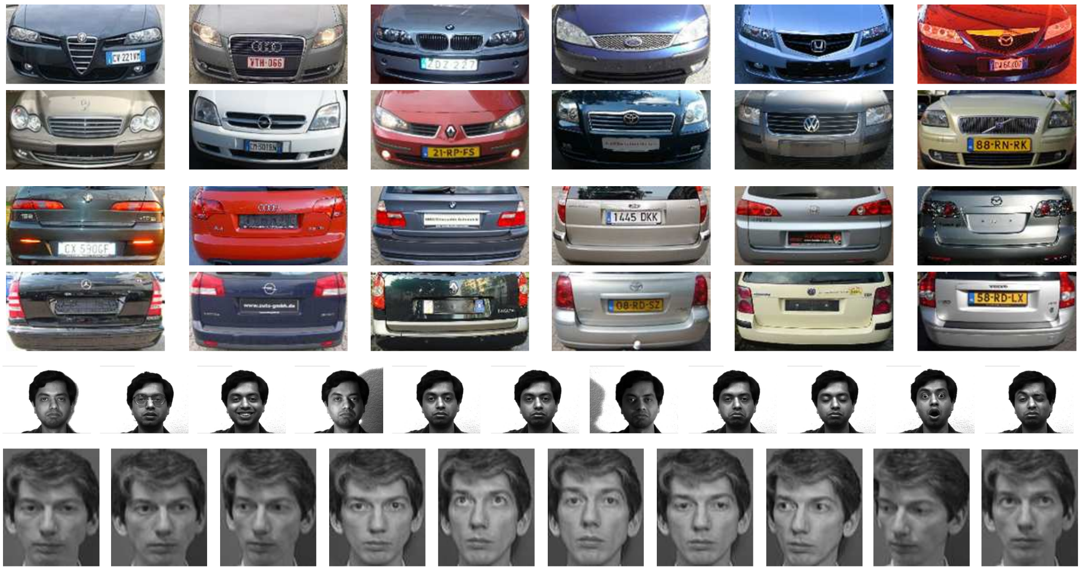


Figure 5.15: Example images from the test non-sign datasets. From top to bottom: car fronts, car rears, Yale faces and AT&T faces. For the last two datasets different views of the same subject are shown.

5.3.5.2 Test Results

In the first experiment we tested a similarity-based nearest neighbour classifier incorporating the similarity measure learned from the car datasets using the kernel regression trees. The effect of combining multiple trees on the overall performance of the classifier was investigated. We were particularly interested in how much gain in performance can be achieved using different kernels and different combination methods, and at what computational cost. In this experiment several tree ensembles were trained separately from the front and the rear car images, using HOGs as the underlying image representation, in order to predict the model of new, unseen cars. Both datasets were divided into training and test parts. 10 images per car model were used for tree training and the other 20 images per model were used for evaluation. Each bagged and boosted ensemble was trained via cross-validation in the same way as described in Section 5.3.4.1, but the number of trees built using the same portion of data was treated as a variable. The random forests were constructed based on the same growing sets as those used to grow the bagged and boosted trees, but pruning was omitted. The testing scheme was also borrowed from the traffic sign recognition experiment.

Results of the experiment are shown in Table 5.8. In addition, in Table 5.9 the average numbers of visited tree nodes per example are listed, depending on the combination scheme and the split kernel type used.

setting	4 × 1 trees		4 × 3 trees		4 × 10 trees	
	front	rear	front	rear	front	rear
bagg./step	55.2%	53.7%	62.5%	63.0%	67.7%	76.2%
bagg./p-l	59.1%	66.2%	65.9%	71.7%	67.1%	78.7%
bagg./RBF	64.0%	80.2%	67.5%	83.9%	68.8%	82.6%
boost./step	X	X	64.9%	73.4%	73.4%	80.8%
boost./p-l	X	X	69.9%	69.6%	76.1%	81.2%
boost./RBF	X	X	72.5%	83.7%	$\boxed{81.6\%}$	$\boxed{87.8\%}$
forest/step	46.6%	38.7%	60.3%	63.3%	69.4%	79.3%
forest/p-l	45.9%	53.3%	66.4%	72.4%	74.0%	82.5%
forest/RBF	75.4%	82.7%	79.5%	85.1%	$\boxed{84.3\%}$	$\boxed{87.4\%}$

Table 5.8: Classification rates obtained for the car front and car rear datasets using combined kernel regression trees. Results associated with the two best-performing combinations for each dataset are marked.

In the second experiment the capabilities of our recognition system were demonstrated on two face datasets: "Yale faces A" and "AT&T faces". In the first case the face similarity was learned in the same way as for the car datasets, but 5 images per class were used for training and 6 images per class were used for testing. Table 5.10 illustrates the obtained results for the two best-performing ensemble type-kernel type combinations. Using the "AT&T faces" dataset, our goal was to

dataset	bagg. step	bagg. p-l	bagg. RBF	boost. step	boost. p-l	boost. RBF	forest step	forest p-l	forest RBF
car front	2.48	2.73	54.05	3.87	5.25	37.85	12.95	16.53	89.00
car rear	2.35	3.20	64.20	4.22	5.14	51.55	11.98	17.88	65.85

Table 5.9: Average numbers of visited tree nodes per example depending on the combination scheme and the split kernel function.

demonstrate that the learned similarity function can also be used to recognise previously never seen objects, as in [153]. Specifically, we learned the similarity from the pairs of same/different images representing 20 individuals. Then, we tested the nearest neighbour classifier based on this similarity on the same/different pairs generated from the images of 20 completely different individuals. To avoid bias in the results, upon completion of the first test run the datasets were swapped, i.e. the training was done using the pairs generated from the images of those 20 persons that were previously used for testing, and vice versa. The training/test pair generation scheme described earlier in this section was adopted. The averaged recognition rates of both classifiers are shown in Table 5.11.

setting	4 × 1 trees	4 × 3 trees	4 × 10 trees
boost./p-l	71.3%	80.7%	87.4%
boost./RBF	69.6%	83.3%	81.3%
forest/p-l	81.3%	86.2%	86.0%

Table 5.10: Classification rates obtained for the "Yale faces A" dataset using the two best-performing kernel regression tree ensembles.

setting	4 × 1 trees	4 × 3 trees	4 × 10 trees
boost./p-l	64.8%	70.4%	73.6%
forest/p-l	67.4%	68.1%	74.3%

Table 5.11: Classification rates obtained for the "AT&T faces" dataset using the two best-performing kernel regression tree ensembles. In this experiment the previously never seen faces were recognised.

In the above experiments it has been demonstrated that the proposed kernel regression trees can be used for learning visual similarity between various object classes. This similarity measure can further be employed to efficiently recognise objects of the previously seen or unseen classes. The similarity-learning trees prove to be flexible in terms of the image representation. Note that any kind of features can be used for node splitting, irrespective of the type and the dimensionality of the descriptor's output, as long as a sensible distance metric for the chosen class of descriptors is available. This is a desired property as, depending on the problem, the actual image description and the local distance formulation can be tuned to the particular application without changing the core of the similarity learning and the classification frameworks.

As expected, all tree combination methods improved the stability of the prediction, but the additional gain in performance appeared to be significantly higher when using boosting ensembles and random forests. As to the choice of split kernel functions, interestingly the Gaussian RBF kernel does not always lead to the best results. While the RBF kernel trees outperform the trees adopting other kernels when classification of cars is involved, the simpler piecewise linear kernels surprisingly offer comparable performance in the task of face discrimination. In addition, as they are simpler to compute and reduce the range of fuzzification, i.e. introduce only limited overlap between the local subspaces obtained via splitting a given node, the time of propagation of an input image pair through the tree is dramatically reduced (see Table 5.9 for comparison). This makes the piecewise linear kernel trees suitable for video-based face recognition applications.

5.4 Summary

Classification is the final step in the TSR processing pipeline. There are a large number of different signs around the world. They differ with respect to the shape, colour, size, and the pictogram symbols contained. Moreover, different instances of the same semantic class of signs may differ both across the countries and within the same country, but this variability is minor in the latter case, compared to the cross-country differences. Construction of a classifier that would be able to recognise the entire gamut of pictograms is probably not possible. In some cases it is even not practical as certain signs, e.g. those giving directions, are not standarised, i.e. their individual instances are non-repeatable. Besides, a vehicle equipped with a visual driver assistance system is unlikely to encounter all possible variations of all possible signs on its way. It seems more practical then to tune the road sign classifier only to the subset of the most popular, highly standarised signs that are likely to be observed in a given, currently operated administrative region, e.g. country. In the same time the classifier may be re-trained to adopt the new pictogram database, when necessary. It should be noted that despite such a commonly adopted problem simplification still more than one hundred pictograms are to be recognised, which is a very challenging task.

In this chapter, we have developed a road sign recognition strategy where the key concept is image similarity or dissimilarity. This led us to the two, conceptually different approaches to building efficient road sign classifiers. The first method proposed is based on the class-specific template matching. The critical step here is the extraction of the optimal local image features, where the term "optimal features" is associated with the local image regions where a given idealised target pictogram differs possibly the most from all the remaining idealised pictograms. In our approach this *one-vs-all* dissimilarity maximisation is carried out using a forward search algorithm and the local region distance metric is based on the so-called *Colour Distance Transform* (CDT). This distance metric is strongly related to the term "idealised template". Actually, we note that because the traffic signs are composed of only several named colours, it is impractical to compare the noisy realistic sign images with the clean sign prototypes in the full colour spectrum. Instead, the images of signs detected and tracked in the incoming video are discretised by a fast, off-line learned Gaussian Mixture-based colour classifier. Furthermore, separate distance transforms [16] are computed for each discrete colour obtained, which is what we call a CDT.

While the colour discretisation makes the proposed road sign classifier robust to significant illumination changes, CDT efficiently models the viewpoint-related appearance variability of traffic signs, which would otherwise have to be learned from a large number of geometrically transformed real-life sign images. In practice, a sufficient number of such training images is difficult to obtain and no appropriate public databases exist. Overall, our feature extraction method generates a compact set of the most discriminative local regions characterising each class separately. Also, through the colour discretisation and the CDT, it makes it possible to classify the image of an unknown sign by simply comparing it to the clean class templates and minimising the resulting distance. Good recognition accuracy and fast processing obtained in the experimental evaluation of the class-specific template matching justify further development of this promising approach.

The other TSR strategy introduced is based on a different concept – learning sign similarity from the pairs of images representing either the same or two different pictograms. This real-valued similarity function combines the local image distances understood as the differences between the values of the selected image descriptors, evaluated at the corresponding locations in both input images. With the similarity function available, an unknown sign image can be classified in a straightforward way - by pairing it with the prototype images of each possible class, one by one, and picking the label of the prototype maximising similarity of the pair. This approach is suitable for problems involving a large number of potentially very similar classes. Through combining the local image distances rather than feature values directly, it becomes flexible with respect to the actual feature description of the target objects.

Two robust approaches to learning the visual similarity from the equivalence information have been proposed: *SimBoost* and the *Kernel Regression Trees*. The first technique is essentially an adaptation of the Schapire and Singer's boosting algorithm using the confidence-rated predictions [61] to learning the pairwise similarity. Kernel regression trees are our novel contribution. At each node of such a tree the local estimation of the output is made. This output is associated with the degree of similarity between the pairs of images, which are input to the tree. The pairs passing through each non-terminal node are typically directed to both subtrees simultaneously, which makes the tree fuzzy. The actual range of this fuzzification is dependent on the choice of split kernel functions. The notion of kernel is used to emphasise the fact that each local discriminant is a function of the (distance between the) responses of the locally selected image descriptor evaluated for both images in an input pair. *Similarity-Learning Kernel Regression Trees* have been combined into robust ensembles via bagging and boosting, as well as random forests, to achieve better recognition stability and accuracy.

Extensive experimentation has been conducted to reveal the discriminative potential of the classifiers based upon the visual similarity function learned using *SimBoost* and the kernel regression tree ensembles. The recognition rates obtained on the challenging road sign dataset are promising and make the proposed approach superior to the alternative methods in the context of visual driver assistance. The regression tree ensembles have been additionally evaluated on several auxiliary datasets, including car and face images, to check whether the pairwise similarity learning can be used to build a robust, general-purpose object classifier. Good recognition accuracy reported in this experimental evaluation seems to confirm this hypothesis and has revealed a large potential of our approach.

Chapter 6

Conclusions and Future Work

Traffic sign recognition (TSR) is one of the critical tasks for the intelligent vehicles which are themselves a technology of the future. Solving the TSR problem will have deep implications on our safety while driving and should therefore reduce the number of road traffic fatalities. Computer-aided recognition of traffic signs has now a long record of related academic and industrial research projects and the first industry-scale applications are now emerging on the market.

Video-based traffic sign recognition typically involves three separate tasks: detection of the likely sign candidates, tracking the already detected candidates over time, and recognition of the detected and tracked signs' pictograms. All these tasks are related to separate types of problems which have been addressed in this thesis:

1. **Rapid localisation of the interest regions in the scene**
   Real-time performance requirements for the TSR systems necessitate quick determination of the local image regions where the traffic signs are most likely present in a potentially large, complex and cluttered scene.

2. **Discriminating road signs from the background**
   This is a challenging problem even when the target is roughly localised and its prototype appearance is well defined. The factors hampering sign detection include: adverse illumination, low image resolution, motion blur, background clutter, occlusions, viewpoint changes, and signs' aging (physical degradation).

3. **Invariant target tracking over time**
   Frame-rate video processing requires the traffic sign recognition system to estimate the state of the target object over time in a robust, invariant way, such that the current state estimate can be instantly inferred based on the current and the previous observations. This dramatically reduces the computation involved.

4. **Discriminative representation of traffic signs**
   A road sign pattern is difficult to recognise because there are numerous possible, frequently very similar classes in the pictogram database, and because the observation is usually noisy. Therefore, to enable robust classification, discriminative feature representation of traffic signs must be constructed.

5. **Recognition and observation fusion**
   It is a non-trivial problem which approach to adopt to possibly most correctly identify the road sign's pictogram based on a sequence of noisy observations, in particular, how to fuse these observations over time.

6.1 Outcomes

In this thesis a comprehensive approach to detection, tracking and recognition of traffic signs from video input has been presented. Different functional modules have been combined and implemented as a complete prototype TSR system that demonstrated a good detection and recognition accuracy as well as near frame-rate processing speed in a number of experiments. This work adds new values to the current state of the research in the area of machine vision and intelligent vehicles and has several contributions characterised below:

6.1.1 Detection

At the detection stage the fundamental problem is to discriminate the traffic signs from the possibly complex background. Because signs do not always clearly stand out, are often distorted and poorly illuminated, or because the observation process is imperfect, i.e. the input video is affected by the motion blur or vehicle's vibrations, accurate and fast capturing of road signs in the subsequent images of an input sequence is challenging. To address these problems, we have proposed a detection framework that is both, satisfactorily accurate with respect to the balance between the true and false positives rates, and enables frame-rate processing. It is outlined below.

To quickly reduce the analysis region, a quad-tree focus of attention operator has been introduced. It measures local density of sign-specific colour gradients within an input image region, starting from the entire image, and iteratively analyses the quarters of this region in smaller scales, depending on whether or not this density exceeds a predefined threshold. Computationally inexpensive implementation of this algorithm is possible thanks to the integral image representation [111] which enables determination of the amount of feature contained in any rectangular image region in constant time. This method is also versatile as it allows interest region extraction based on thresholding the concentration of other pixel-based image features, e.g. temporal differences for moving object localisation.

While the state-of-the-art object detectors (regular polygon detector of Barnes et al. [128] and a boosted classifier cascade [111]) are used to capture traffic signs in the scene, appropriate colour image preprocessing is done and a novel clustering algorithm is proposed to refine the output of these detectors. It is called *Confidence-Weighted Mean Shift* and has been developed as an extension of the original mean shift algorithm [82] for accurate estimation of the signs' locations and scales in the scene and to eliminate the redundant positive hypotheses produced around the true sign instances. This algorithm treats the detector's response space as a probability distribution and iteratively finds its maxima. Positive detector's hypotheses, which are input to the algorithm, are assigned confidence weights directly associated with the real-valued detector's responses. As they are indicative of the likelihood of a sign's presence at a given location and scale, plugging these weights into the mean shift formula makes the algorithm converge at the more accurate sign locations and scales.

Overall, the presented detector demonstrated good performance in the experiments involving both static images and image sequences, capturing majority of the true signs in complex scenes at close to frame-rate and yielding small number of false alarms. Taking into consideration the fact that these false alarms can be rejected at a later stage of the processing, when the consecutive frame observations are fused over time, the presented approach, after necessary optimisations, has good prospects for implementation in a real in-vehicle driver assistance system.

6.1.2 Tracking

Road sign tracking in typical traffic scenario seems easy as the apparent motion of the target in the image plane is slow when the vehicle approaches it along a straight line. However, in certain situations, when the car undergoes non-linear motion, pose changes of traffic signs may be more substantial. Also, due to occlusions or car vibrations, which are common when the vehicle moves on an uneven road surface, signs may be temporarily impossible to detect. These observations call for appropriate tracker's design. Besides, the appearance of traffic signs, e.g. their colours and contrast, may change significantly over time as a result of varying illumination. The above problems mark out two major directions for development of a road sign tracker: geometry-based and appearance-based. In this thesis a general strategy has been chosen in which the tracker explicitly models only the apparent geometrical transformations of signs (position and scale in the first instance). The appearance variability, potentially hindering the recognition process, is either compensated by on-line re-training of the tracker, or is addressed at the classifier level.

Three different road sign trackers have been proposed in this work. A combined Kalman Filter-Particle Filter (KF-PF) tracker has been developed as a baseline algorithm for handling most common traffic situations. In this technique the Kalman Filter component [1] is responsible for the current-frame sign's centroid and scale estimation based on the immediately preceding estimates. It also reduces the search region for the detector. The location and the size of this region is determined from the current state's mean and process error variance estimates. The Particle Filter corrector [21] is used to refine the apparent width and height estimates of the detected sign. They might be inaccurate when the offset between the sign's symmetry axis perpendicular to its plate, and the camera's optical axis is large relative to the distance between the camera and the target. This is because the detection process is driven by the radial symmetry property of signs which is no longer valid in the abovementioned situations.

A contour tracking algorithm has been presented as an extension to the above KF-PF technique. The main idea behind this method is to maintain a map of pixel relevances within the local interest region determined by the Kalman Filter.

The pixel relevance is understood as a likelihood of it being part of the sign's contour. This likelihood is propagated from frame to frame via spatio-temporal voting and modulated by a distance transform calculated around the hypothesised motion-compensated contour of a sign. As a result, the confusing edges inside and around the signs are attenuated and the risk of inaccurate sign delimitation is minimised. The resulting tracker has been shown particularly useful in cluttered scenes.

An affine motion tracker of Tuzel et al. [156] has been implemented to show that the full structure of the apparent geometrical deformation of traffic signs can be modelled in a robust way. Tracking is carried out using a matrix-valued function of the observation vector. When multiplied on the left by the previous-frame motion matrix, this function gives an accurate 3D pose estimate of the target in the current frame. This function is learned and periodically updated based on the Lie group theory from an image of a sign in an *a priori* known pose, either arbitrarily assumed frontal (initial detection) or obtained as a result of previous estimations. The learning process proceeds by applying random affine distortions to this image and minimising the sum of the squared geodesic distances between the estimated and the known motion matrices. The resulting tracker, if updated appropriately, is accurate and very fast. Moreover, ability of reconstructing a full-face view of a distorted sign makes it viewpoint-invariant. This property enables efficient recognition of signs that are already close to the vehicle and hence their pictograms are the most unambiguous. The regression tracker has been tested on a number of road traffic sequences captured from a real vehicle. It demonstrated good accuracy and could run in real time on a standard computer.

6.1.3 Recognition

The main challenge in road sign recognition is related to the necessity of discriminating between numerous, frequently very similar classes. In this respect this problem is radically different from generative object categorisation in which also a large number of object classes are involved, but they are much more diverse. In the existing approaches to TSR the above problem is often avoided by explicitly focusing on a narrow specific category of signs, such as prohibition signs. As in that case the input problem is much easier than when the entire gamut of signs are considered, such classifiers often demonstrate impressive performance. To facilitate recognition, the road signs family is often also decomposed in an arbitrary tree structure reflecting its semantic hierarchy.

In this thesis we have proposed two different strategies for multi-class symbolic sign recognition. In the first approach, similarly to certain previous studies, a preliminary decomposition of traffic signs is done based on their shapes and dominant colours, which is intended to facilitate both, detection and classification. Then, within each subcategory of signs, for each individual pictogram belonging to this subcategory a discriminative local feature representation is built based essentially on the clean template images available in a highway code. As a prerequisite to this learning process, noisy sign images cropped from the incoming video are subject to colour binning based on an off-line learned Gaussian Mixture colour classifier. In the same time, for each discrete colour present in each clean template image a separate distance transform (DT) is computed as though the pixels of this particular colour were feature pixels and the pixels of all remaining colours were non-feature pixels. The resulting set of DTs, which we call a *Colour Distance Transform*(CDT), enables comparisons between the discrete-colour images of the observed and the idealised signs, which are robust to small geometrical distortions and misalignments introduced in the detection process.

In the learning process a compact number of local image regions are selected for each template road sign where this template differs possibly the most from the remaining templates with respect to the distance metric based on CDT. A category-global threshold is introduced to ensure that roughly the same amount of dissimilarity between one sign and any other sign is incorporated in the model and to control the dimensionality of the feature space. When a likely road sign is captured in the input video, its image is discretised into the category-specific colours as mentioned above. Further, it is compared to each template pictogram with respect to the class-specific set of discriminative local regions found in the training process and using the CDT-based distance metric. Individual-frame comparisons are fused over time and the class of an unknown sign is determined from the template image to which the cumulative distance is minimised at the time when the tracked sign disappears from the scene. Experimental evaluation has shown that the above method is suitable for discriminating between a large number of traffic signs. Moreover, its advantage over the alternative methods is that it requires relatively few real-life training images. In particular, it does not necessarily require such images to represent all possible types of signs, which is convenient as certain signs are very rare in reality and hence their images are difficult to acquire.

In the second proposed approach we have developed a sign similarity learning framework based on two different techniques: *SimBoost* and the *Similarity-Learning Kernel Regression Trees*. The main feature of both algorithms is that they aim at estimating the real-valued similarity function of the two input images. Similarly, they are learned from image pairs, instead of individual images. These pairs are labelled either “same” or “different”, depending on whether the paired images represent the same or two different pictograms. The abovementioned similarity function combines the local image distances quantified by chosen image descriptors evaluated at the corresponding locations in both input images. As only the descriptor value differences are important, our approach becomes independent of the actual type and the dimensionality of the image descriptors used, as long as adequate distance metrics for these descriptors exist.

The first proposed technique is a modification of the improved boosting algorithm using confidence-rated predictions [61], adapted to pairwise similarity learning. In the second method the fuzzy regression trees have been used for this task. The fuzziness is realised at the level of kernel-based node splitting rules where multiple kernel functions have been considered. Kernelised trees are combined into robust bagged and boosted ensembles, as well as random forests, to achieve additional gain in accuracy of the similarity estimation. In a series of experiments involving low-resolution road sign images it has been shown that a simple nearest neighbour classifier based on the learned similarity function successfully discriminates between multiple similar pictograms. Video-based recognition experiments confirmed these promising results. In addition, similar experiments repeated on several non-sign datasets proved that the proposed visual similarity-learning algorithms can be used to generate efficient general-purpose object classifiers.

6.2 Future Work

In the field of visual driver assistance significant advances were made in the recent years. This resulted in the extensive research work done all over the world and the first commercial industry-scale applications released on the market, e.g. [162]. In this thesis several interesting ideas have been presented on how the contemporary in-vehicle traffic sign recognition systems could be improved. However, despite their promise shown in the previous chapters, this work still has limitations. Below, these limitations are discussed and the possible directions for further research are outlined.

6.2.1 Problem Decomposition

Traffic sign recognition problem calls for appropriate decomposition as groups of signs exhibiting similar shape and colour characteristics can be easily identified. Currently, such decomposition is done manually and is driven by the general shape and dominant colours of a traffic sign. This facilitates both detection and recognition because instead of one monolithic detector and classifier, several more specialised (and hence more efficient) detectors and classifiers can be constructed from each distinct group. Although such an arbitrary problem decomposition is generally well-justified, it is not necessarily optimal. Possibly, other partitionings, orthogonal to the semantic road sign decomposition would generate better detectors/classifiers.

We have tried automatic partitioning of traffic signs based on the k-means clustering of the training images in various low-dimensionality feature spaces. It was repeated for the increasing numbers of randomly initialised clusters so as to record possibly the best partitioning in the meaning of large mean between-cluster to mean within-cluster distance ratio. However, this clustering invariably led to assigning the images representing different pictograms to separate clusters in the optimal case. It implied the necessity of training the individual pictogram detectors and classifiers, which would have caused various decision-making problems and prohibitively slow processing in system runtime.

A more complex approach for the automatic TSR problem decomposition has to be proposed. In addition, a robust mechanism of probabilistic assignment of the detected and tracked signs to the discovered subcategories has to be devised to avoid misclassifications at an early stage of the processing. Note that from this perspective the currently used manual decomposition is very good as the risk of confusing different regular polygons with different rim colours is minimal. However, it requires extensive image preprocessing to be done, i.e. separate colour enhancement, colour-specific edge and gradient computation, colour- and shape-specific detection. Employing an appropriately pruned decision tree as a clusterer of the training data, in place of k-means mentioned above, could prove useful in this case. Note that such a tree would enable construction of the cluster proximity measure, e.g. based on the length of the shortest path between the terminal nodes. Such a measure could be helpful in the design of a probabilistic cross-category classification model.

6.2.2 Illumination Invariance

One of the major limitations of our approach is that the sign detectors considered in this work are not illumination-invariant and depend on various thresholds. As a result, they tend to be oversensitive when the signs clearly stand out from the background or insufficiently discriminative when the illumination is adverse. In general, we preferred to set the operating points of the detectors [1] such that they captured even the hard instances of traffic signs, e.g. those with poor figure-background contrast, at the cost of yielding more false positives in more favourable scene fragments. However, simply increasing the sensitivity of the detector does not fundamentally solve the illumination dependence problem. Moreover, it causes other problems. In particular, in cluttered scenes containing objects similarly coloured and shaped to traffic signs, it not only increases the risk of generating false alarms, but also significantly increases the computational overhead needed to cluster the numerous redundant positive hypotheses of the detector (Section 3.4). This, in turn, makes the real-time implementation of the TSR system difficult.

Dealing with the lightning variations in the scene requires explicit modelling of these variations at the tracking stage, which is currently not done. One possible way such modelling could be performed is similar to learning the dependency between the 3D pose of the target with its feature descriptor by the affine motion tracker (Section 4.4). For instance, a single contrast-stretching factor or independent colour channel amplification/attenuation factors could be learned on-the-fly via regression from a sufficiently stable view of a sign. Then, these regularisation factors would be used to normalise the appearance of a tracked sign in each frame of the input video such that its colours/contrast would be roughly constant over the entire observation sequence. This would in turn make the detection and tracking process illumination-invariant.

To solve the illumination dependence problem at the time of initial candidate sign discovery, the only promising direction seems to be towards improving the discriminative power of the image features on which the detector relies. The fundamental problem here is that these features must be extremely simple to compute to enable frame-rate operation of the detector. Currently used features, although combining both primary cues, colour and shape, are computationally sufficiently fast, but are not sufficiently discriminative. A worthwhile idea that could bring the desired improvement would be to encode the spatial arrangement of colours in the low level image descriptors, e.g. via parametrising the Haar wavelets by pairs of colours, instead of single colours.

[1] Depending on the actual detector used, it means setting the minimum acceptable number of votes in the Hough parameter space or setting the layer-specific thresholds in the rejection cascade.

6.2.3 False Alarm and Ambiguity Rejection

An apparent deficiency of the classifiers presented in this work is a lack of the false alarm rejection mechanism. However, there are several steps earlier in the processing pipeline, e.g. detection and temporal integration, where a vast majority of such false alarms are effectively discarded. To further minimise the chance of unnecessary tracking and recognition of false traffic signs, a maximum dissimilarity/minimum similarity threshold can be introduced at the classifier level. The candidate objects for which (accordingly) the distance to the least similar template or similarity to the most different class prototype go above/below this threshold, would be simply discarded.

A more severe problem that the proposed sign classifiers have to face is rejection of the too ambiguously recognised signs for the classifier's decision to be considered trustworthy. It is difficult as the ambiguity accidentally developed over time, e.g. through inaccurate target tracking, and involving two physically substantially different signs, is no different from the ambiguity caused by the fact that simply two or more nearly identical pictograms resembling the observed sign exist in the template database. Having to discriminate between such nearly identical signs is unavoidable because they are common on the roads in many countries (see Figure 5.7 for examples). Unfortunately, they look similar even in the discriminative feature spaces, such as those spanned by the local regions discussed in Section 5.2.2. It is so because such discriminative spaces are obtained via *one-vs-all* dissimilarity maximisation. Therefore, even if two signs look very similar in a given space, globally this space may still be very discriminative.

Regarding the class-specific template matching (CSTM) classifier introduced in Section 5.2, a natural way to cope with the ambiguous classifications is to use an extra threshold which would specify the minimum difference or ratio between the cumulative distance of the observation sequence from the best-scored template to the cumulative distance of this sequence from the second best-scored template. However, due to the aforementioned problem, in the case of certain sign pairs this difference is likely to be almost always very small, but in most other cases it will be usually large and smoothly increasing, as the image of a tracked sign becomes larger and clearer. It is why we considered using a single uniform threshold for ambiguity detection inadequate. Perhaps, one possible way of addressing this problem would be to learn the class-specific sign representations based on the *one-vs-one* dissimilarity maximisation, together with the currently used representations learned via maximising the *one-vs-all* dissimilarity. Such extra representations could be used dynamically (when needed) for comparisons between the actual observation and each of the two most likely templates. This could resolve at least those confusions involving two very similar pictograms.

The classifiers based on the learned sign similarity measure, discussed in Section 5.3, suffer from the similar problem as the CSTM technique. Again, one possible solution involves learning auxiliary similarity functions only from the images representing two classes. As in this case large numbers of natural images are required in the training process, learning the similarity between all possible pairs would be impractical. Therefore, it could be restricted to only the most commonly confused pairs. The extra similarity functions could then be used to resolve the potential conflicts between the two prototypes best-scored in the course of regular classification.

Regardless of the possible improvements outlined above, a practical approach, applicable to any traffic sign recognition system that is to be installed in real, human-operated vehicles, is to present to the driver the confidence of the classification rather than a definite classifier's decision. Besides, simple generalisations could be made by the decision interpreter if the confused signs were semantically similar. For instance, if two speed limits were confused the human-understandable message could sound: "Speed limit ahead" instead of "60 kph speed limit ahead" or "80 kph speed limit ahead".

Bibliography

[1] R. E. Kalman: “A New Approach to Linear Filtering and Prediction Problems”, *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35-45, 1960.

[2] M.-K. Hu: “Visual Pattern Recognition by Moment Invariants”, *IEEE Transactions on Information Theory*, 8:179-187, 1962.

[3] R. O. Duda, P. E. Hart: “Use of the Hough Transformation to Detect Lines and Curves in Pictures”, *Communications of Association for Computing Machinery*, 15(1):11-15, 1972.

[4] A. Dempster, N. Laird and D. Rubin: “Maximum likelihood from incomplete data via the EM algorithm”, *Journal of the Royal Statistical Society*(B), 39(1):1-38, 1977.

[5] H. P. Moravec: “Towards Automatic Visual Obstacle Avoidance”, In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, 584, 1977.

[6] M. A. Fischler and R. C. Bolles: “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, *Communications of the ACM*, 24:381-395, 1981.

[7] B. D. Lucas and T. Kanade: “An iterative image registration technique with an application to stereo vision”, In *Proceedings of the DARPA Image Understanding Workshop*, 121-130, 1981.

[8] B. K. P. Horn and B. G. Schunck: “Determining optical flow”, *Artificial Intelligence*, 17:185-203, 1981.

[9] D. H. Ballard: “Generalizing the Hough Transform to Detect Arbitrary Shapes”, *Pattern Recognition*, 13(2):111-122, 1981.

[10] D. H. Ballard and C. M. Brown: “Computer Vision”, Prentice Hall, 1982.

[11] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi: “Optimization by Simulated Annealing”, *Science*, 220(4598):671-680, 1983.

[12] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone: “Classification and Regression Trees”, Wadsworth International Group, 1984.

[13] H. W. Sorenson: “Kalman Filtering: theory and application” IEEE Press, 1985.

[14] J. Canny: “A computational approach to edge detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679-698, 1986.

[15] D. M. Titterington: “Statistical Analysis of Finite Mixture Distributions”, John Wiley & Sons, 1986.

[16] G. Borgefors: “Distance transformations in digital images”, *Computer Vision, Graphics and Image Processing*, 34(3):344-371, 1986.

[17] L. R. Rabiner and B. H. Juang: “An Introduction to Hidden Markov Models”, *IEEE ASSP Magazine*, 4-15, 1986.

[18] B. D. Ripley: “Stochastic Simulation”, Wiley & Sons, 1987.

[19] G. Borgefors: “Hierarchical chamfer matching: a parametric edge matching algorithm”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849-865, 1988.

[20] C. Harris and M. Stephens: “A Combined Corner and Edge Detector”, In *Proceedings of the 4th Alvey Vision Conference*, Manchester, 147-151, 1988.

[21] D. B. Rubin: “Using the SIR algorithm to simulate posterior distributions”, *Bayesian Statistics*, 3:395-402, 1988.

[22] L. Gupta, M. R. Sayeh and R. Tammana: “A neural network approach to robust shape classification”, *Pattern Recognition*, 23(6):563-568, 1990.

[23] F. Meyer and S. Beucher: “Morphological Segmentation”, *Journal of Visual Communication and Image Representation*, 1:21-46, 1990.

[24] L. Vincent and P. Soille: “Watersheds in digital spaces: An efficient algorithm based on immersion simulations”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583-598, 1991.

[25] C. Tomasi and T. Kanade: “Detection and Tracking of Point Features”, Technical Report CMU-CS-91-132, Carnegie-Mellon University, 1991.

[26] B. E. Boser, I. M. Guyon and V. N. Vapnik: “A training algorithm for optimal margin classifiers”, In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 144-152, 1992.

[27] I. Masaki: “Vision-based vehicle guidance”, Springer-Verlag New York, New York, USA, 1992.

[28] W. Ritter: “Traffic Sign Recogition in Color Image Sequences”, In *Proceeding of the IEEE Intelligent Vehicles Symposium*, 12-77, 1992.

[29] D. C. W. Pao, H. F. Li and R. Jayakumar: “Shapes Recognition Using the Straight Line Hough transform: Theory and Generalization”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1076-1089, 1992.

[30] F. Leymarie and M. D. Levine: “Tracking deformable objects in the plane using an active contour model”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6), 617-634, 1993.

[31] D. P. Huttenlocher, G. A. Klanderman and W. J. Rucklidge: “Comparing Images Using the Hausdorff Distance”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850-863, 1993.

[32] N. J. Gordon, D. J. Salmond and A. F. M. Smith: “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”, *IEE Proceedings F Radar and Signal Processing*, 140(2):107-113.

[33] J. Shi and C. Tomasi: “Good Features to Track”, In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 593-600, 1994.

[34] L. Priese, J. Klieber, R. Lakmann, V. Rehrmann and R. Schian: “New results on traffic sign recognition”, In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 249-254, 1994.

[35] Y. J. Zheng, W. Ritter and R. Janssen: “An adaptive system for traffic sign recognition”, In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 165-170, 1994.

[36] D. S. Kang, N. C. Griswold and N. Kehtarnavaz: “An invariant traffic sign recognition system based on sequential color processing and geometrical transformation”, In *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, 88-93, 1994.

[37] P. J. Werbos: “The roots of backpropagation: from ordered derivatives to neural networks and political forecasting”, Wiley Series On Adaptive And Learning Systems For Signal Processing, Communications, And Control, 1994.

[38] P. Pudil, J. Novovicová and J. Kittler: “Floating search methods in feature selection”, *Pattern Recognition Letters*, 15(11):1119-1125, 1994.

[39] M. Lalonde and Y. Li: “Road Sign Recognition, Survey of the State of the Art”, Technical report, Centre de recherche informatique de Montréal, 1995.

[40] H. Wang and M. Brady: “Real-time corner detection algorithm for motion estimation”, *Image and Vision Computing*, 13(6):695-703, 1995.

[41] H. Breu, J. Gil, D. Kirkpatrick and M. Werman: “Linear-time Euclidean distance transform algorithms”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):529-533, 1985.

[42] V. N. Vapnik: "The Nature of Statistical Learning Theory", Springer, New York, 1995.

[43] T. G. Dietterich and G. Bakiri: "Solving Multiclass Learning Problems via Error-Correcting Output Codes", *Journal of Artificial Intelligence Research*, 2:263-286, 1995.

[44] L. Breiman: "Bagging predictors", *Machine Learning*, 24(2):123-140, 1996.

[45] Y. Aoyagi and T. Asakura: "A study on traffic sign recognition in scene image using genetic algorithms and neural networks", In *Proceedings of the 22nd IEEE International Conference on Industrial Electronics, Control, and Instrumentation*, 3:1838-1843, 1996.

[46] S. J. Julier and J. K. Uhlmann: "A new extension of the Kalman filter to nonlinear systems", In *Proceedings of the International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 182-193, 1997.

[47] Y. Q. Shi and X. Xia: "A thresholding multiresolution block matching algorithm", *IEEE Transactions on Circuits and Systems for Video Technology*, 7(2):437-440, 1997.

[48] A. S. Wright, S. T. Acton: "Watershed pyramids for edge detection", In *Proceedings of the International Conference on Image Processing*, 2:578-581, 1997.

[49] G. Piccioli, E. De Micheli, P. Parodi and M. Campani: "A Robust Method for Road Sign Detection and Recognition", *Image and Vision Computing*, 14:209-223, 1997.

[50] A. de la Escalera, L. E. Moreno, M. A. Salichs and J. M. Armingol: "Road traffic sign detection and classification", *IEEE Transactions on Industrial Electronics*, 44(6):848-859, 1997.

[51] C. Schmid and R. Mohr: "Local Grayvalue Invariants for Image Retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530-535, 1997.

[52] M. Isard and A. Blake: "Condensation - conditional density propagation for visual tracking" *International Journal of Computer Vision*, 29(1):5-28, 1998.

[53] M. Sonka, V. Hlavac, R. Boyle: "Image Processing, Analysis, and Machine Vision", 2nd Edition, Thomson-Engineering, 1998.

[54] M. Trajkovic and M. Hedley: "Fast Corner Detection", *Image and Vision Computing*, 16(2):75-87, 1998.

[55] D. Gavrila: "Multi-feature Hierarchical Template Matching Using Distance Transforms", In *Proceedings of the IEEE International Conference on Pattern Recognition*, Brisbane, Australia, 439-444, 1998.

[56] M. C. Burl, M. Weber and P. Perona: "A probabilistic approach to object recognition using local photometry and global geometry", In *Proceedings of the 5th European Conference on Computer Vision*, 2:628-641, 1998.

[57] M. Pontil and A. Verri: "Support vector machines for 3D object recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637-646, 1998.

[58] M. Akmal Butt and P. Maragos: "Optimum design of chamfer distance transforms", *IEEE Transactions on Image Processing*, 7(10):1477-1484, 1998.

[59] T. K. Ho: "The random subspace method for constructing decision forests", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832-844, 1998.

[60] Y. Freund and R. E. Schapire: "A short introduction to boosting", *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, 1999.

[61] R. E. Schapire and Y. Singer: "Improved boosting algorithms using confidence-rated predictions", *Machine Learning*, 37(3):297-336, 1999.

[62] C. Stauffer and W. E. L. Grimson: "Adaptive background mixture models for real-time tracking", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2:246-252, 1999.

[63] N. M. Oliver, B. Rosario and A. P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831-843, 2000.

[64] A. M. Elgammal, D. Harwood and L. S. Davis: "Non-parametric Model for Background Subtraction", In *Proceedings of the 6th European Conference on Computer Vision*, 751-767, 2000.

[65] J. Miura, T. Kanda and Y. Shirai: "An active vision system for real-time traffic sign recognition", In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, Darborn, MI, USA, 52-57, 2000.

[66] P. Paclík and J. Novovicová: "Road Sign Recognition without Color Information", In *Proceedings of 6th Annual Conference of the Advances School for Computing and Imaging*, Lommel, Belgium, 84-90, 2000.

[67] P. Paclík, J. Novovicová, P. Pudil and P. Somol: "Road Sign Classification using the Laplace Kernel Classifier", *Pattern Recognition Letters*, 21(13-14):1165-1173, 2000.

[68] M. Sato, S. Lakare, M. Wan and A. Kaufman: "A Gradient Magnitude Based Region Growing Algorithm for Accurate Segmentation", In *Proceedings of the International Conference on Image Processing*, 3:448-451, 2000.

[69] M. K. Lee, S. W. Leung, T. L. Pun, H. L. Cheung and A. M. K. Lee: "Edge detection by genetic algorithm", In *Proceedings of the International Conference on Image Processing*, 1:478-480, 2000.

[70] M. Mirmehdi and M. Petrou: "Segmentation of Color Textures", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):142-159, 2000.

[71] G. Guo, S. Z. Li and K. Chan: "Face recognition by support vector machines", In *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, 196-201, 2000.

[72] P. Douville: "Real-Time Classification of Traffic Signs", *Real-Time Imaging*, 6(3):185-193(9), 2000.

[73] M. Weber, M. Welling and P. Perona: "Unsupervised learning of models for recognition", In *Proceedings of the 6th European Conference on Computer Vision*, 2:101-108, 2000.

[74] S. H. Hsu and C. L. Huang: "Road sign detection and recognition using matching pursuit method", *Image and Vision Computing*, 19(3):119-129, 2001.

[75] B. Heisele, P. Ho and T. Poggio: "Face recognition with support vector machines: global versus component-based approach", In *Proceedings of the 8th International Conference on Computer Vision*, 688-694, 2001.

[76] C.-C. Chang, C.-S. Chan, J.-Y. Hsiao: "A Color Image Retrieval Method Based on Local Histogram", *Proceedings of the 2nd IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing*, 831-836, 2001.

[77] G.-D. Guo, H.-J. Zhang and S. Z. Li: "Pairwise face recognition", In *Proceedings of the 2001 International Conference on Computer Vision*, 2:282-287, 2001.

[78] T. Kadir and M. Brady: "Saliency, Scale and Image Description", *International Journal of Computer Vision*, 45(2):83-105, 2001.

[79] J. Lafferty, A. McCallum and F. Pereira: "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", In *Proceedings of the 18th International Conference on Machine Learning*, 282-289, 2001.

[80] Y.-S. Chen, Y.-P. Hung and C.-S. Fuh: "Fast block matching algorithm based on the winner-update strategy", *IEEE Transactions on Image Processing*, 10(8):1212-1222, 2001.

[81] D. M. J. Tax and R. P. W. Duin: "Using two-class classifiers for multi-class classification", In *Proceedings of the 2002 International Conference on Pattern Recognition*, 2:124-127, 2002.

[82] D. Comaniciu and P. Meer: "Mean shift: a robust approach towards feature space analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603-619, 2002.

[83] A. M. Ferman, A. M. Tekalp, R. Mehrotra: "Robust Color Histogram Descriptors for Video Segment Retrieval and Identification", *IEEE Transactions on Image Processing*, 11(5):497-508, 2002.

[84] S.-O. Shim, T.-S. Choi: "Edge Color Histogram for Image Retrieval", *Proceedings of the International Conference on Image Processing*, 3:957-960, 2002.

[85] D. A. Forsyth, J. Ponce: "Computer Vision, A Modern Approach", Prentice Hall, 2002.

[86] C. Palm, T. M. Lehmann: "Classification of Color Textures by Gabor Filtering", *Machine Graphics & Vision International Journal*, 11(2/3):195-219, 2002.

[87] P. Gouton, I. Foucherot: "Optimal Contours Detection By Using a Directional Filtering", *Proceedings of the 2nd IASTED International Conference on Visualisation, Imaging, and Image Processing*, ACTA Press, Malaga, Spain, 459-462, 2002.

[88] K. Crammer and Y. Singer: "On the algorithmic implementation of multiclass kernel-based vector machines", *Journal of Machine Learning Research*, 2:265-292, 2002.

[89] M. Seki, T. Wada, H. Fujiwara and K. Sumi: "Background detection based on the cooccurrence of image variations", In *Proceedings the IEEE International Conference on Computer Vision and Pattern Recognition*, 2:65-72, 2003.

[90] C. Olaru and L. Wehenkel: "A complete fuzzy decision tree technique", *Fuzzy Sets and Systems*, 138(2):221-254, 2003.

[91] E. Abdelkawy and D. McGaughy: "Wavelet-based Image Target Detection Methods", *Proceedings of the Automatic Target Recognition Conference*, Orlando, USA, 5094:337-347, 2003.

[92] C.-Y. Fang, S.-W. Chen, C.-S. Fuh: "Road-Sign Detection and Tracking", *IEEE Transactions on Vehicular Technology*, 52(5):1329-1341, 2003.

[93] G. Loy and A. Zelinsky: "Fast Radial Symmetry for Detecting Points of Interest", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):959-973, 2003.

[94] W. Zhao, R. Chellappa, A. Rosenfeld, P. J. Phillips: "Face Recognition: A Literature Survey", *ACM Computing Surveys*, 399-458, 2003.

[95] S. Mahamud, M. Hebert: "The optimal distance measure for object detection", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 1:248-255, 2003.

[96] D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar: “Handbook of Fingerprint Recognition”, Springer, 2003.

[97] M. Tkalčič, J. F. Tasič: “Colour spaces: perceptual, historical and applicational background”, *The IEEE Region 8 EUROCON 2003*, 1:304-308, 2003.

[98] A. de la Escalera, J. M. Armingol and M. Mata: “Traffic sign recognition and analysis for intelligent vehicles” *Image and Vision Computing*, 21(3):247-258, 2003.

[99] P. Viola, M. Jones and D. Snow: “Detecting Pedestrians using Patterns of Motion and Appearance”, In *Proceedings of the 9th IEEE International Conference on Computer Vision*, 2:734-742, 2003.

[100] D. Comaniciu, V. Ramesh and P. Meer: “Kernel-based object tracking”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564-577, 2003.

[101] A. A. Efros, A. C. Berg, G. Mori and J. Malik: “Recognizing Action at a Distance”, In *Proceedings of the 9th International Conference on Computer Vision*, 2:726-734, 2003.

[102] R. Fergus, P. Perona and A. Zisserman: “Object class recognition by unsupervised scale-invariant learning”, In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2:264-271, 2003.

[103] L. Fei-Fei, R. Fergus and P. Perona: “Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories”, *Computer Vision and Image Understanding*, 106(1):59-70, 2003.

[104] M. Jones and P. Viola: “Face Recognition Using Boosted Local Features”, Technical report TR2003-25, Mitsubishi Electric Research Laboratories, 2003.

[105] K.-C. Lee, J. Ho, M.-H. Yang and D. Kriegman: “Video-based face recognition using probabilistic appearance manifolds”, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1:313-320, 2003.

[106] H. Schneiderman and T. Kanade: “Object Detection Using the Statistics of Parts”, *International Journal of Computer Vision*, 56(3):151-177, 2004.

[107] K. Mikolajczyk and C. Schmid: “Scale & Affine Invariant Interest Point Detectors”, *International Journal of Computer Vision*, 60(1):63-86, 2004.

[108] S. B. Park, J. W. Lee and S. K. Kim: “Content-based image classification using a neural network”, *Pattern Recognition Letters*, 25(3):287-300, 2004.

[109] A. Quattoni, M. Collins and T. Darrell: “Conditional random fields for object recognition”, *Advances in Neural Information Processing Systems*, 1097-1104, 2004.

[110] N. Barnes and A. Zelinsky: "Real-time radial symmetry for speed sign detection", In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 566-571, 2004.

[111] Viola, P. and Jones, M.: "Robust Real-time Face Detection", *International Journal of Computer Vision*, vol.57, no.2, pp.137-154, 2004.

[112] D. G. Lowe: "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, 60(2):91-110, 2004.

[113] F. Mindru, T. Tuytelaars, L. Van Gool and T. Moons: "Moment invariants for recognition under changing viewpoint and illumination", *Computer Vision and Image Understanding*, 94(1-3):3-27, 2004.

[114] Y. Ke and R. Sukthankar: "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2:506-513, 2004.

[115] R. Rifkin and A. Klautau: "In Defense of One-Vs-All Classification", *Journal of Machine Learning Research*, 5:101-141, 2004.

[116] B. Han, D. Comaniciu, L. Davis: "Sequential kernel density approximation through mode propagation: applications to background modeling", In *Proceedings of the Asian Conference on Computer Vision*, 2004.

[117] T. Kadir, A. Zisserman and M. Brady: "An affine invariant salient region detector", In *Proceedings of the 8th European Conference on Computer Vision*,345-457, 2004.

[118] J. Matas, O. Chum, M. Urbana and T. Pajdlaa: "Robust wide-baseline stereo from maximally stable extremal regions", *Image and Vision Computing*, 22(10):761-767, 2004.

[119] D. Serby, E. K. Meier and L. Van Gool: "Probabilistic object tracking using multiple features", In *Proceedings of the 17th International Conference on Pattern Recognition*, 2:23-26.

[120] A. de la Escalera, J. M. Armingol, J. M. Pastor and F. J. Rodríguez: "Visual Sign Information Extraction and Identification by Deformable Models for Intelligent Vehicles", *IEEE Transcations on Intelligent Transportation Systems*, 5(2):57-68, 2004.

[121] C. Dance, J. Willamowski, L. Fan, C. Bray and G. Csurka: "Visual categorization with bags of keypoints", In *Proceedings of the ECCV International Workshop on Statistical Learning in Computer Vision*, 1-22, 2004.

[122] L. Fei-Fei and P. Perona: "A Bayesian Hierarchical Model for Learning Natural Scene Categories", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2:524-531, 2005.

[123] J. Sivic, B. C. Russell, A. A. Efros and W. T. Freeman: "Discovering Object Categories in Image Collections", In *Proceedings of the 10th International Conference on Computer Vision*, 1:370-377, 2005.

[124] E. B. Sudderth, A. Torralba, W. T. Freeman and A. S. Willsky: "Describing Visual Scenes Using Transformed Objects and Parts", *International Journal of Computer Vision*, 77(1-3):291-330, 2005.

[125] I. Ulusoy and C. M. Bishop: "Generative versus discriminative methods for object recognition", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2:258-265, 2005.

[126] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool: "A Comparison of Affine Region Detectors", *International Journal of Computer Vision*, 65(1-2):43-72, 2005.

[127] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, T. Koehler: "A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information", In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 255-260, 2005.

[128] N. Barnes, G. Loy, D. Shaw, A. Robles-Kelly: "Regular polygon detection", In *Proceedings of the 10th IEEE International Conference on Computer Vision*, 1:778-785, 2005.

[129] M. D. Fairchild: "Color Appearance Models", 2nd Edition, Wiley-IS&T, Chichester, UK, 2005.

[130] J. Fan, G. Zeng, M. Body, M.-S. Hacid: "Seeded region growing: an extensive and comparative study", *Pattern Recognition Letters*, 26(8):1139-1156, 2005.

[131] A. L. C. Barczak, F. Dadgostar, M. J. Johnson: "Real-Time Hand Tracking using the Viola and Jones Method", In *Proceedings of the 7th IASTED International Conference on Signal and Image Processing*, ACTA Press, Honolulu, Hawaii, USA, 336-341, 2005.

[132] N. Dalal, B. Triggs: "Histograms of Oriented Gradients for Human Detection", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 1:886-893, 2005.

[133] C. Changjiang, R. Duraiswami and L. S. Davis: "Efficient mean-shift tracking via a new similarity measure", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 1:176-183, 2005.

[134] C.-J. Yang, R. Duraiswami and L. S. Davis: "Fast Multiple Object Tracking via a Hierarchical Particle Filter", In *Proceedings of the International Conference on Computer Vision*, 1:212-219, 2005.

[135] F. Porikli: “Integral histogram: a fast way to extract histograms in Cartesian spaces”, In *Proceedings of the 2005 IEEE International Confernce on Computer Vision and Pattern Recognition*, 1:829-836, 2005.

[136] A. Ferencz, E. Learned Miller and J. Malik: “Building a classification cascade for visual identification from one example”, In *Proceedings of the International Conference on Computer Vision*, 1:286-293, 2005.

[137] S. B. Wang, A. Quattoni, L. P. Morency and D. Demirdjian: “Conditional Random Fields for Gesture Recognition”, In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2:1521-1527, 2006.

[138] S. Munder, D. M. Gavrila: “An Experimental Study on Pedestrian Classification”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1863-1868, 2006.

[139] Q. Zhu, S. Avidan, M.-C. Yeh, K.-T. Cheng: “Fast Human Detection Using a Cascade of Histograms of Oriented Gradients”, In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2:1491-1498, 2006.

[140] E. Rosten, T. Drummond: “Machine learning for high-speed corner detection”, In *Proceedings of the 9th European Conference on Computer Vision*, 1:430-443, 2006.

[141] I. Kunttu, L. Lepistö, J. Rauhamaa, A. Visa: “Fourier-Based Object Description in Defect Image Retrieval”, *Machine Vision and Applications*, 17:211-218, 2006.

[142] X. W. Gao, L. Podladchikova, D. 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*, vol.17, no.4, pp.675-685, 2006.

[143] W. Zhang, H. Deng, T. G. Dietterich and E. N. Mortensen: “A Hierarchical Object Recognition System Based on Multi-scale Principal Curvature Regions”, In *Proceedings of the IEEE 18th International Conference on Pattern Recognition*, 1:778-782, 2006.

[144] P. Paclík, J. Novovicová and R. P. W. Duin: “Building Road-Sign Classifiers Using a Trainable Similarity Measure”, *IEEE Transactions on Intelligent Transportation Systems*, 7(3):309-321, 2006.

[145] O. Tuzel, F. Porikli and P. Meer: “Region covariance: A fast descriptor for detection and classification”, In *Proceedings of the 9th European Conference on Computer Vision*, 589-600, 2006.

[146] W. Hao and J. Luo: "Generalized Multiclass AdaBoost and its Applications to Multimedia Classification", In *Proceedings of the 2006 Computer Vision and Pattern Recognition Workshop*, 113, 2006.

[147] G. S. W. Klein and D. W. Murray: "Full-3D Edge Tracking with a Particle Filter", In *Proceedngs of the 17th British Machine Vision Conference*, 3:1119-1128, 2006.

[148] X.-Y. Xu and B.-X. Li: "Learning Motion Correlation for Tracking Articulated Human Body with a Rao-Blackwellised Particle Filter", In *Proceedings of the International Conference on Computer Vision*, 1-8, 2007.

[149] Y. Xia, D. Feng, T. Wang, R. Zhao and Y. Zhang: "Image segmentation by clustering of spatial patterns", *Pattern Recognition Letters*, 28(12):1548-1555, 2007.

[150] E. Bayro-Corrochano and J. Ortegón-Aguilar: "Lie algebra approach for tracking and 3D motion estimation using monocular vision", *Image and Vision Computing*, 25(6):907-921, 2007.

[151] P. Sabzmeydani and G. Mori: "Detecting Pedestrians by Learning Shapelet Features", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 1-8, 2007.

[152] O. Tuzel, F. Porikli and P. Meer: "Human Detection via Classification on Riemannian Manifolds", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 1-8, 2007.

[153] E. Nowak and F. Jurie: "Learning Visual Similarity Measures for Comparing Never Seen Objects", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 1-8, 2007.

[154] L. Oliveira, U Nunes and P. Peixoto: "Scheme of Primate's Visual Cortex Cells for Pedestrian Recognition", In *Proceedings of the 3rd European Symposium on Nature-inspired Smart Information Systems*, 2007.

[155] J. Stallkamp, H. K. Ekenel, R. Stiefelhagen: "Video-based Face Recognition on Real-World Data", In *Proceedings of the 11th IEEE International Conference on Computer Vision*, 1-8, 2007.

[156] O. Tuzel, F. Porikli and P. Meer: "Learning on Lie Groups for Invariant Detection and Tracking", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 1-8, 2008.

[157] Y-Y. Nguwi and A. Z. Kouzani: "Detection and classification of road signs in natural environments", *Neural Computing and Applications*, 17(3):265-289, 2008.

[158] Z. Lin, L. S. Davis, D. Doermann and D. DeMenthon: “Hierarchical Part-Template Matching for Human Detection and Segmentation”, In *Proceedings of the 11th IEEE International Conference on Computer Vision*, 1-8, 2008.

[159] W.-H. Yun, S. Y. Bang and D. Kim: “Real-time object recognition using relational dependency based on graphical model”, *Pattern Recognition*, 41(2):742-753, 2008.

[160] DARPA Grand Challenge, `http://www.darpa.mil/grandchallenge/index.asp`, [last accessed: February 16, 2009].

[161] Mobileye vision system for driver assistance, `http://www.mobileye.com/default.asp?PageID=202`, [last accessed February 16, 2009].

[162] “Opel Cars Can See: Opel Eye camera reads signs, improves safety”, 18 June 2008, `http://media.gm.com/servlet/GatewayServlet?target=http://image.emerald.gm.com/gmnews/viewmonthlyreleasedetail.do?domain=103&docid=46499` [last accessed: February 16, 2009]

[163] Daimler-Chrysler pedestrian classification benchmark, `http://www.science.uva.nl/research/isla/downloads/pedestrians/`, [last accessed February 16, 2009].

[164] Open Graphics Library, `www.opengl.org/`, [last accessed: February 16, 2009].

[165] Open Computer Vision Library, `http://sourceforge.net/projects/opencvlibrary/`, [last accessed: February 16, 2009].

[166] B. Mecánico: Road signs arranged by country. [www] Available from: `http://www.elve.net/rcoulst.htm` [last accessed: February 16, 2009].

Appendix A

Learning of the Affine Tracking Function

To recall, the affine tracker uses a function, $f : \mathbb{R}^m \longmapsto A(2)$, to estimate the 3D pose of a sign in each frame of an input video based on the previous-frame pose estimate and the current-frame observation. m is the dimensionality of the observation vector constructed from the previously observed image, after mapping it to the object coordinates and $A(2)$ denotes a two-dimensional affine transformation. When multiplied on the left by the previous-frame motion matrix, function f gives an accurate pose estimate of the target in the current frame. To learn the optimal parameters of f, given the initial pose of a sign, $\mathbf{M}_0$, in the image I_0 at time t_0, random affine deformation matrices $\Delta\mathbf{M}_i$, $i = 1, \ldots, n$ around identity are generated (where $n < m$), the image is transformed through each such matrix, and in the resulting object-coordinate images descriptor $\mathbf{o}_0^i$ is computed. Then, the sum of the squared geodesic distances between the pairs of motion matrices: estimated $f(\mathbf{o}_0^i)$, and known $\Delta\mathbf{M}_i$ is minimised.

The affine motion matrices can be considered points on the Lie group with a structure of a 6-dimensional differentiable manifold given by (4.25). The tangent space to the identity element $\mathbf{I}$ of the group forms the associated Lie algebra, which in our case is a set of matrices:

$$\mathbf{m} = \begin{pmatrix} \mathbf{U} & \mathbf{v} \\ 0 & 0 \end{pmatrix} , \tag{A.1}$$

where $\mathbf{U}$ is a 2×2 matrix and $\mathbf{v} \in \mathbb{R}^2$. The matrix m is often treated as a 6-dimensional vector obtained by selecting each entry of $\mathbf{U}$ and $\mathbf{v}$ as an orthonormal basis.

As previously said, the goal of the tracking is to minimise the sum of squared distances between the motion matrices estimated by function f and the known random transformations. An adequate measure of distance between two motion matrices treated as points on the manifold is the minimum length of a curve connecting these points, called geodesic. It is given by:

$$\rho(\mathbf{M}_1, \mathbf{M}_2) = \| \log(\mathbf{M}_1^{-1}\mathbf{M}_2) \| \ . \tag{A.2}$$

With (A.2), the error measure becomes:

$$J = \sum_{i=1}^{n} \rho(f(\mathbf{o}_0^i), \Delta\mathbf{M}_i)^2 \ . \tag{A.3}$$

Tuzel et al. [156] show that if two vectors $\mathbf{m}_1$, $\mathbf{m}_2$ can be expanded into motion matrices $\mathbf{M}_1$, $\mathbf{M}_2$ respectively, then the first-order approximation to the geodesic distance between them is:

$$\rho(\mathbf{M}_1, \mathbf{M}_2) = \|\mathbf{m}_2 - \mathbf{m}_1\| \ . \tag{A.4}$$

Therefore, selecting $d = 6$ orthonormal bases on the Lie algebra, the error function in (A.3) can be computed as a sum of the squared Euclidean distances between the vectors $\log(f(\mathbf{o}_0^i))$ and $\log(\Delta\mathbf{M}_i)$, i.e.:

$$J = \sum_{i=1}^{n} \| \log(f(\mathbf{o}_0^i)) - \log(\Delta\mathbf{M}_i)\|^2 \ . \tag{A.5}$$

Taking the form of equation (A.5) into account, the target regression function $f(\mathbf{o})$ is modelled as:

$$f(\mathbf{o}) = exp(g(\mathbf{o})) \tag{A.6}$$

and function $g : \mathbb{R}^m \mapsto \mathbb{R}^d$ is learned. It estimates the tangent vectors, $\log(\Delta\mathbf{M})$, on the Lie algebra, and is assumed linear:

$$g(\mathbf{o}) = \mathbf{o}^T \boldsymbol{\Omega} \ , \tag{A.7}$$

where $\boldsymbol{\Omega}$ is a $m \times d$ matrix of regression coefficients. Assuming that $\mathbf{X}$ is a $n \times m$ matrix of initial observations, $[\mathbf{o}_0^i]^T$, and $\mathbf{Y}$ is a $n \times d$ matrix of tangent vectors $[\log(\Delta\mathbf{M}_i)]^T$, $i = 1, \ldots, n$, minimisation of the error function can be reformulated as:

$$J = tr[(\mathbf{X}\boldsymbol{\Omega} - \mathbf{Y})^T(\mathbf{X}\boldsymbol{\Omega} - \mathbf{Y})] + \lambda\|\boldsymbol{\Omega}\|^2 \ . \tag{A.8}$$

Since $m > n$, the above linear system is underdetermined and hence the least squares estimate becomes inaccurate. To prevent this, the last term in (A.8), $\lambda\|\boldsymbol{\Omega}\|^2$, is introduced to avoid overfitting. It defines an additional constraint on the size of the regression coefficients, where λ is a constant that needs to be adjusted based on the training data. The minimum of J is given by:

$$\boldsymbol{\Omega} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \ , \tag{A.9}$$

where $\mathbf{I}$ is an $m \times m$ identity matrix.

Since traffic signs can undergo appearance changes in time, it is necessary to adapt to these variations by updating the tracking function. This update process is carried out in a similar way as the initial training, i.e. by minimising the sum of the squared geodesic distances between the estimated and the known, randomly generated motion matrices. However, another constraint is introduced on the difference between the current and the previous regression coefficients:

$$J_u = tr[(\mathbf{X\Omega} - \mathbf{Y})^T(\mathbf{X\Omega} - \mathbf{Y})] + \lambda\|\mathbf{\Omega}\|^2 + \gamma\|\mathbf{\Omega} - \mathbf{\Omega}'\|^2 \ , \tag{A.10}$$

where $\mathbf{\Omega}'$ denotes the matrix of regression coefficients used prior to the update, and γ is an another constant that needs to be tuned based on the training data. The minimum of (A.10) is given by:

$$\mathbf{\Omega} = (\mathbf{X}^T\mathbf{X} + (\lambda + \gamma)\mathbf{I})^{-1}(\mathbf{X}^T\mathbf{Y} + \gamma\mathbf{\Omega}') \ . \tag{A.11}$$

Appendix B

Description of Polish Symbolic Traffic Signs

There are four main categories of traffic signs defined in the Polish highway code: cautionary signs, prohibition signs, signs giving orders, and information signs. Below, these four categories are briefly outlined. Other signs, such as those giving direction or distance, or custom diversion signs, are not discussed. Such signs are not focused on in this work as they are not uniquely defined, i.e. they vary between roads or traffic situations.

Cautionary signs warn against various dangers on the road. They are equilateral triangles with yellow background and essentially black symbols in the inner part. The "give way" sign is special as it is turned upside down. The prohibition signs inform the driver that certain manoeuvres are not allowed in places where these signs are mounted. They are circular, have red rim, white interior (with the exception of "no stopping" and "no parking" signs) and usually black symbols in it. A special group of prohibition signs are the cancellation signs, e.g. cancellations of the previously introduced speed limits. Pictograms of these signs are completely achromatic, i.e. they only contain shades of gray. The signs giving orders inform the drivers about the mandatory actions to be done. For example, they may enforce a given driving direction. These signs are also circular and contain white symbols on a blue background. Occasionally, they contain a thick red strip to cancel the meaning of a sign previously introduced. The information signs conduct messages about the road where they are mounted and its surroundings, e.g. parking places, petrol stations, garages, etc. They are usually blue squares or blue rectangles containing black, white or red colour symbols in their central part. Exceptions to this rule are two signs: "road with right of way" and its cancellation, which are white-yellow squares rotated by 45 degrees. Other information signs, e.g. lane restrictions, are usually custom-made and vary between places.

Figure B.1 lists the templates of traffic signs characterised above, with the exception of rectangular information signs and all other not uniquely defined signs. The classifier defined in Section 5.2 is trained to recognise all of the pictograms shown in this figure, including all possible "speed limit" and "minimum speed" signs' variations, but excluding the "road with right of way" sign, its cancellation, and the colourless cancellation prohibition signs.

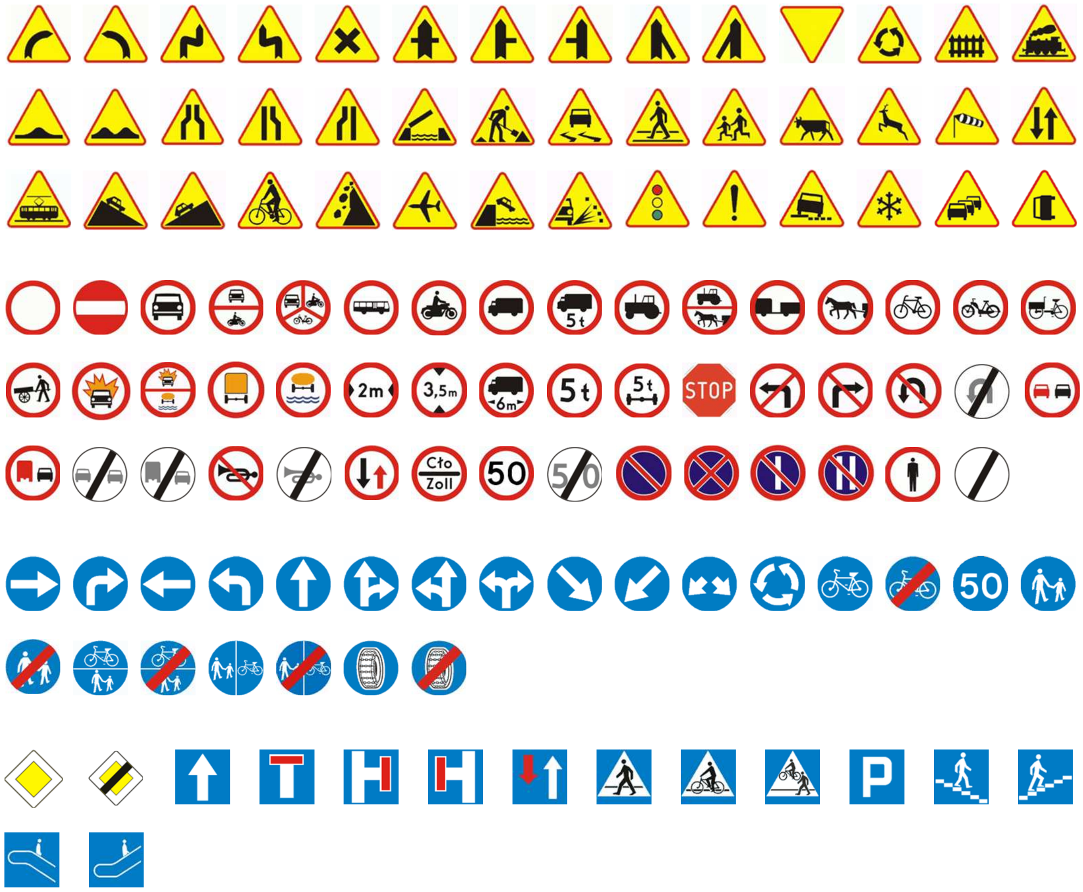


Figure B.1: The most popular traffic sign defined in the Polish highway code. From top to bottom: cautionary signs, prohibition signs, signs giving orders, and information signs. A successful in-vehicle road sign classifier should be able to discriminative between the signs hereby depicted.