

# DETECTION OF NORWEGIAN SPEED LIMIT SIGNS

Lukas Sekanina  
Faculty of Information Technology  
Brno University of Technology  
Bozotechnova 2  
612 66 Brno, Czech Republic  
E-mail: sekanina@fit.vutbr.cz

Jim Torresen  
Department of Informatics  
University of Oslo  
P.O. Box 1080 Blindern  
N-0316 Oslo, Norway  
E-mail: jimtoer@ifi.uio.no

## KEYWORDS

Electronics, Image Processing, Pattern recognition, Performance analysis, Transportation.

## ABSTRACT

In this paper an approach for detecting Norwegian speed limit signs is proposed. It consists of three major steps: Color-based filtering, locating sign(s) in an image and detection of numbers on the sign. Very good results are obtained for the two first steps and for the third one work is still in progress.

## INTRODUCTION

A system for recognizing traffic signs would be beneficial in several ways. It could assist drivers about signs they did not recognize before passing them. Specifically, speed limit sign recognition – studied in this paper, could inform drivers about the present speed limit as well as giving an alert if a car is driven faster than the speed limit. In the future, autonomous vehicles would have to be controlled by automatic road sign recognition.

Recognizing road images have been studied for a long time. The first known attempt for making a real-time system was by Akatsuka and Imai (Akatsuka and Imai, 1987). Many techniques have been proposed since then.

The standard technique for detecting and recognizing road signs consists of three steps (Lalonde and Li, 1995). First, color segmentation or color thresholding is applied to emphasize possible signs in an image. Thus, restricting the search area in the image. Second, template matching is applied for shape detection. Third, specific signs are detected using template matching or neural networks.

A color image is often transformed from the RGB (Red, Green, Blue) color space into the HSV (Hue, Saturation, Value) color space. Color segmentation then becomes easier (by applying it on the hue value only rather than the three RGB values). The hue value is invariant for the illumination as well. However, the hue is not suited for greylevel segmentation since it has a constant value along the “greylevel” axis. The value will be unstable near this axis too. Small perturbations in the RGB signals may cause large variations in the hue.

A traffic sign detection system consisting of four stages has been proposed by Wei et al (Wei et al., 1999). First, a color-based filtering is applied to filter out image regions that have color characteristics similar to the color found in *one* of the signs known to the system. That is, the color input image is converted into a binary image using the filter. Second, the boundaries of the regions are smoothed by applying close and open morphological operations. Third, shape analysis is performed to evaluate the similarity between the region and a given shape. This includes major and minor axis as well as comparing area. Finally, the holes within the regions are analysed. The system performs well for four different signs in different environments as well as at different orientation and scales.

Another system proposed by Hsu and Huang (Hsu and Huang, 2001) is based on three steps for road sign detection followed by sign recognition based on Matching Pursuit (MP) filter. The detection locates regions with signs in the unknown input image by first using a priori knowledge about position, shape and color (HSV). Second, template matching is performed in the extracted regions to find road signs. Third, the road sign is extracted by further template matching and normalization.

This paper concerns detection of speed limit signs specifically. The next section introduces the signs. This is followed by two sections which outlines the proposed algorithm and reports results from implementation, respectively. Finally, the last section concludes the paper.

## NORWEGIAN SPEED LIMIT SIGNS

Speed limit signs have features making them feasible for automatic detection. First, there is a limited number of signs to distinguish.



Figure 1: Norwegian Speed Limit Signs (standard)

In Norway, the following speed limit signs – see Figure 1, are used: 30, 40, 50, 60, 70, 80 and 90 (100 are being tested

at the moment). The outer circle of a sign is in red color. Further, there are signs to nullify 30 and 40 (set speed limit to 50), 50, 60 and 70 (set speed limit to 80) and 80 (set speed limit to 90). The nullify signs have not yet been considered by the algorithm outlined in this paper. Second, there are rules (named a road grammar) for how signs can be placed along the road. After a “80” sign you will never find a “30” sign, but rather another “80” sign or a “60” sign. Third, the numbers on the signs are positioned vertically making the matching simpler. In curves only, the signs may be marginally tilted. These features make it promising to undertake speed limit sign detection with a very high rate of correct prediction.

Making a speed limit sign detection system commercially applicable, the safety is very important. If the detection system do not have a distinct best match, it should rather output that it “don’t know” the speed at the moment. Thus, it is important that the system refrain from indicating a speed rather than indicating a *wrong* speed. The system presented in the next section is based on a novel heuristic developed through a large range of experiments.

## SPEED LIMIT SIGN RECOGNITION ALGORITHM

The algorithm is divided into *three* parts: 1) Image filtering to emphasize the red parts of the sign(s), 2) Template matching to locate the sign(s) in an image and 3) Sign number recognition. These will be presented in the following sections below.

### Image Filtering

In this part, a specialized robust color filter is applied on the image to mark the red circle surrounding the speed limit numbers. The algorithm has to use a broad definition of red and white colors since various intensities and saturations can occur in real images (depending on weather, day/nightlight, etc.). On the other hand, only red color *on* traffic signs should be detected to minimize the number of starting points for the following template matching algorithm. Thus, detection of red color *outside* a traffic sign (like on buildings, cars, etc.) should be minimized.

The RGB color model was found useful for detection. RGB is also suitable according to planned hardware implementation where calculation of e.g. HSV model could be computational expensive. The colors are defined (by experiments) as follows:

- A pixel is RED if:  $(R > 80) \text{ AND } (R - G > 20) \text{ AND } (R - B > 20)$
- A pixel is WHITE if:  $(R > 150) \text{ AND } (G > 150) \text{ AND } (B > 150)$
- A pixel is BLACK if:  $(R < 180) \text{ AND } (G < 180) \text{ AND } (B < 180)$

RED, WHITE and BLACK are symbolic names that will be used below. As one can see, the definition of the basic colors is broad to assure robustness. WHITE and BLACK are used mainly in different parts of the algorithm. When detecting WHITE, we would like every light pixel. When detecting BLACK, every pixel not being BLACK should *definitely* be of white color. Thus, there is a value overlap (150 to 180) between BLACK and WHITE. Good selectivity is achieved

by combining these colors during pixel classification. The following three images are used by the filtering algorithm:

- OrigImg – the original RGB image
- Image1 – temporal image1 (256 gray levels)
- Image2 – temporal image2 (256 gray levels)

The filtering algorithm is as follows:

1. Set all pixels in Image1 to grey.
2. Set all pixels in Image2 to white.
3. Find all 2x2 WHITE squares in the OrigImg and mark them in Image1.
4. Find (recursively) all RED neighbors in OrigImg of WHITE squares in Image1 and mark them in Image1.
5. Mark RED pixels in Image2.

### Locating Sign by Template Matching

Locating a sign in the image is the most computational demanding part of sign recognition. We have chosen to use template matching for this operation. One reason for this is that template matching can be very effectively implemented in hardware. Ten templates (of size 160x160, 130x130, 100x100, 80x80, 60x60, 50x50, 40x40, 32x32, 26x26 and 20x20 pixels) are used to detect the *position* of a speed limit sign. The templates are shown in Figure 3. The 20x20 template is used only in this testing version to find the limits of the system. We need a very high reliability of the system and small templates tend to produce incorrect classification of the numbers. The templates are stored in an NMP format (Numbers Map – a format defined by ourselves). The template matching algorithm is shown in Figure 2.

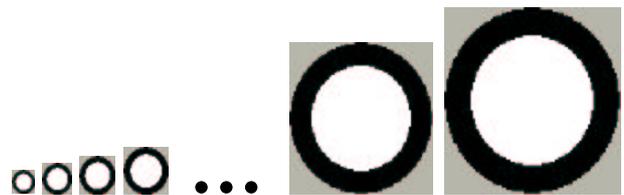


Figure 3: Templates Used for Locating Signs in an Image

### Sign Number Recognition

The last part of the algorithm is to detect the speed limit *number* on a sign. This is conducted as follows:

1. Clean the space defined by the best template (remove RED and surrounding colors), but keep the numbers.
2. Find boundaries of numbers (width and height).
3. Work only with the first number (the second is always zero).

```

Besttotalmatch = 0
1) FOR (every template) DO
  BestP = 0
  2) FOR (every RED pixel at even line and even column in the Image2) DO
    a) P = Percentage of pixels that match with a template.
    b) IF (there are some RED pixels in the center of the sign) THEN P = 0 (avoid crossing lines)
    c) IF (RED pixels are not distributed symmetrically) THEN P = 0 (avoid triangles)
    d) IF (there is more than 95% of WHITE pixels) THEN P = 0
    e) IF (there is less than 10% of WHITE pixels) THEN P = 0
    f) IF (P > BestP) THEN update BestP
  3) IF (BestP > Besttotalmatch) THEN update record of the Besttotalmatch
IF (Besttotalmatch < 50%) THEN Print: No speed limit
ELSE continue by recognition of number on the sign

```

Figure 2: The Template Matching Algorithm

4. Create 6 (rows) x 4 (columns) bit array for width and height of the given number (downscaling): Set each bit of the array to 1 if there is more BLACK than WHITE pixels, else set the bit to 0.
5. Use the array for number classification. It is supposed that special features belonging to different numbers are reflected in the array.



Figure 4: Image2 After Recognition of “50” in the Image in Figure 5.

A final Image2 is given in Figure 4. The array used for number recognition is shown in the top left corner of the image.

## RESULTS

This section reports about the experimental work undertaken. As a part of the experiments, a Graphical user interface (GUI) was developed and is the first to be described below.

### Graphical User Interface

Figure 5 shows a picture of the main part of the GUI. The purpose of the control elements and indicators are as follows:



Figure 5: The Graphical User Interface

- RGB – informs about RGB pixel values at the mouse position (which is indicated in the leftmost panels).
- HSV – informs about HSV pixel values at the mouse position (which is indicated in the leftmost panels).
- Save result to: denotes a file where the report will be stored.
- Template folder: denotes a folder where templates used for recognition are stored.
- Acceptable match in %: if a match is less than this % value, then there is not any speed limit sign in the image.
- Color detection – allows the user to define sensitivity to red, white and black colors.

- Set defaults – sets default sensitivity to colors.
- Image 1 – shows the image after application of red/white filter. Position of the best match is indicated.
- Image 2 – shows the image after thresholding and cleaning inside a template. Only the number should be visible.
- Find Speed Limit – activates speed limit detection. Image1 and Image2 are created (available through buttons with the same names), a report (in the message box) is created and the final decision is presented.
- Possible final decisions (50 in the given picture):
  - NOT – no speed limit found.
  - ??? – probably no speed limit sign since some template matched well, but no number was recognized.
  - x? – unknown (not supported) number recognized.
  - The speed limit number detected – only 5 and 6 are supported at the moment.

### Performance results

A database of 84 traffic situations were used in the experiments. Some contained a speed limit sign and some contained other signs or no sign at all. The results were as follows:

- Is there a speed limit sign? 82 correct answers out of 84 tested.
- Speed Limit Number Recognition: ca. 50% is correct (the algorithm is only partially implemented).

There is a large potential of improvement in the speed limit sign recognition algorithm. Thus, this part of the algorithm should be able to give a very high rate of correct recognition as well. Speed of recognition in the current solution is mainly determined by:

- The number of templates used for locating signs in an image.
- The number of RED pixels found during the filtering operation (only RED pixels at *even* rows and columns are taken into account to reduce the computation).
- Image size.

Future work consists of both improving the algorithm as well as studying the real-time performance and possible implementation in special hardware. Biology-inspired algorithms (e.g. artificial neural networks) could be an alternative as well for improving the performance.

### CONCLUSIONS

The paper has presented a novel approach to locate speed limit signs and detect the numbers on them. Promising results are achieved for detecting the signs. However, further improvement of the number recognition is required and this is a part of the ongoing and future research.

### ACKNOWLEDGMENTS

Lukas Sekanina would like to acknowledge The Research Council of Norway for funding his stay at Department of Informatics at University of Oslo.

### REFERENCES

- Akatsuka, H. and Imai, S. (1987). Road signpost recognition system. In *Proc. of SAE vehicle highway infrastructure: safety comptatibility*, pages 189–196.
- Hsu, S. and Huang, C. (2001). Road sign detection and recognition using matching pursuit method. *Image and Vision Computing*, 19:119–129.
- Lalonde, M. and Li, Y. (1995). Road sign recognition, survey of the state of the art. In *CRIM/IIT (Centre de recherche informatique de Montreal)*.
- Wei, G. et al. (1999). Traffic sign detection and recognition for safe driving. In Dagli et al., editors, *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems, Proc. of ANNIE'99*. ASME Press.

### BIOGRAPHY

**LUKAS SEKANINA** received MSc degree in Computer Science and Engineering from Brno University of Technology, Czech Rep. in 1999. He was a visiting lecturer at Pennsylvania State University, USA and a visiting researcher with Department of Informatics, University of Oslo in 2001. Currently he is a PhD candidate at the Brno University of Technology. His research interests focus on evolvable hardware and evolutionary design.

E-mail: sekanina@fit.vutbr.cz

Web: <http://www.fit.vutbr.cz/~sekanina>

**JIM TORRESEN** was born in Mandal, Norway. He received his M.Sc. and Ph.D. degrees in computer architecture and design from the Norwegian University of Science and Technology, University of Trondheim in 1991 and 1996, respectively. He was then employed as a senior embedded system designer in several different companies. Since 1999, he has been an associate professor at the Department of Informatics at the University of Oslo. Torresen has been a visiting researcher at Kyoto University, Japan for one year (1993-1994) and four months at AIST (formerly Electrotechnical laboratory), Tsukuba, Japan (1997 and 2000). His main research interests are evolvable hardware and real-world applications.

E-mail: jimtoer@ifi.uio.no

Web: <http://www.ifi.uio.no/~jimtoer>