Study on road sign recognition in LabVIEW

M Panoiu¹, C L Rat¹ and C Panoiu¹

¹Politehnica University of Timisoara, Department of Electrical Engineering and Industrial Informatics, 5 Revolution Street, 331128 Hunedoara, Romania

E-mail: manuela.panoiu@fih.upt.ro

Abstract. Road and traffic sign identification is a field of study that can be used to aid the development of in-car advisory systems. It uses computer vision and artificial intelligence to extract the road signs from outdoor images acquired by a camera in uncontrolled lighting conditions where they may be occluded by other objects, or may suffer from problems such as color fading, disorientation, variations in shape and size, etc. An automatic means of identifying traffic signs, in these conditions, can make a significant contribution to develop an Intelligent Transport Systems (ITS) that continuously monitors the driver, the vehicle, and the road. Road and traffic signs are characterized by a number of features which make them recognizable from the environment. Road signs are located in standard positions and have standard shapes, standard colors, and known pictograms. These characteristics make them suitable for image identification. Traffic sign identification covers two problems: traffic sign detection and traffic sign recognition. Traffic sign detection is meant for the accurate localization of traffic signs in the image space, while traffic sign recognition handles the labeling of such detections into specific traffic sign types or subcategories [1].

1. Introduction

Road and traffic signs, traffic lights and other traffic devices are used to regulate, warn, guide or inform road users. Traffic signs help achieve an acceptable level of road traffic quality and increase safety with orderly and predictable movement of traffic and pedestrian, by helping drivers anticipate the road ahead in addition to any associated problems and hazardous situations.

Looking at the task of road and traffic sign recognition, it appears to be a well-defined, seemingly simple task. Road signs are designed, manufactured and installed according to strict regulations in order to make them readily distinguishable from their surroundings (Figure 1). However, looking at road sign recognition in its full scale reveals a series of problems that affect the performance of the detection system [1], [3].

These signs are designed to be easily recognized by drivers, but considering the object recognition and interpretation abilities of humans, it is a hard task to try to develop a computer based system which should be able to do what people do in everyday life.

It can be concluded that it is extremely important for the algorithms to be developed for the detection and recognition of road and traffic signs to have high robustness of color segmentation, high insensitivity to noise and brightness variations, and should be invariant to geometrical effects such as translation, in-plane and out-plane rotations and scaling changes in the image [1], [4].

Our system is based on taking images through a camera from a moving vehicle and analyzing it using the recognition program. It uses traffic sign features as filters in a step by step algorithm designed to eliminate false positives with every step. It provides correct results on a wide variety of conditions.
image distortion factors, such as the weather and the environment in which the image was taken, its position in relation to the camera, etc. The program, due to the fact that it works with thresholds and probability, is robust and efficient.

Because the efficiency and speed of detection are important factors in this study, algorithms have been developed to reduce the search space and indicate potential only the potential traffic signs.

Such a system could help improve traffic flow and safety, could avoid hazardous driving conditions, such as collisions, but it does require real-time image processing [1], [2].

![Traffic Signs](image)

**Figure 1.** All the traffic signs recognized in this study

### 2. Identification Process

The software we have developed in LABVIEW works in two basic stages; namely detection and recognition (Figure 2). In the detection stage, the sign is detected according to the sign’s color information, and then according to the sign’s shape. Using these features the sign is extracted and sent through a classifier to decide what type of sign it is, in accordance with the pictogram [5].

The color of the road sign together its shape determines its category. Detection and recognition can be achieved even if one component, color or shape, is missing [1], [2], but if the two features are available, combining color and shape gives better results.

![Road Sign Recognition Diagram](image)

**Figure 2.** General road sign recognition diagram
In order to program a computer to recognize the signs in the scene, the main strategy is to find the right combination of colors in the scene, so that one color is located inside the contour of another color in combination with the shapes of the 2D objects defined by the individual colors. If a candidate possessing the right combination of shapes and colors is found, the system tries to classify the object according to the rim-pictogram combination and give the result of this classification.

The system performs a parallel search through color threshold for 2 main colors: blue and red. The program looks for the rims of potential signs. The output is a segmented binary image containing potential regions which could be recognized as possible road signs. Each of the candidates is tested against a certain set of features (a pattern) to decide whether it is in the group of road signs or not, and then according to these features they are classified into different groups. These characteristics are chosen to highlight the differences between classes of traffic signs. The shape of the sign plays a central role at this stage. It may be a triangle, a circle, an octagon, etc. Pictogram analysis is another classification step. By analyzing the pictogram, the number of patterns to test can be reduced (testing a pattern is, in itself, a time and resource-consuming task) and only the most likely models will be chosen to be tested on the sign in question [5].

2.1. Color Detection
The system performs a parallel search for 2 of the 4 main colors, namely blue and red. Regarding the other 2 colors, there are relatively few signs (2-3) that contain them and are governed by the same rules, therefore these are considered negligible. Including these colors can be a further development.

The programming environment we chose to use is LabVIEW, due to the fact that it contains a large number of color processing functions, known as IMAQ functions. Colors are represented in images through color spaces. A color space uses a three-dimensional coordinate system, each color being represented by a point. IMAQ uses the RGB color space to display images, and the HSL color space to process them. The RGB color space is an additive color system. The three primary color components can combine to reproduce almost all possible colors. The RGB space reproduces an image as the driver would expect to see it.

The HSL (hue, saturation and luminance) color space provides more accurate color information than the RGB space. The HSL space de-couples the intensity component from the color information and the close relationship between chromacity (hue and saturation) and human perception of color. This feature leads to a more robust color representation independent of light intensity variation. It is used in image processing where the lighting conditions can vary considerably [6], [4].

At this stage, a color threshold is applied to the image, transforming it into a binary image. Binary images pixel have only values of 0 or 1 and, therefore require less computing time with greater precision in processing. The threshold is defined so as to separate a specific color range found in the rims of traffic signs from other colors in the image. This process sets (logic 1) all the pixels that belong to that range (interval), and resets (logic 0) the pixels outside the range. This is used to isolate objects of interest (the rims of road signs) in an image: pixels inside the threshold interval that form particles (contiguous regions of nonzero pixels). Transforming the image to binary does present some other advantages, among which the possibility of working with IMAQ functions specific to object identification.

The particles obtained are passed through morphological functions to remove the small shape defects they may have, in order to enhance their precision for the recognition stage. Morphological functions have also been used to retouch the shapes of the particles and, correct the faulty selections that may have occurred by applying the threshold. These functions change the shapes of pixel regions by expanding or reducing particles, smoothing the borders of the preparation of particles of quantitative analysis, etc. These operations change overall size and shape of particles in the image. The operation used in this case is an expansion function: dilation removes small holes within particles and extends the particle contour according to the model defined by the structural element.

The resulting binary image is then passed through a particle filter. This has the purpose of finding particles whose spatial characteristics satisfy certain criteria and eliminating the particles that are of no
interest. In real-time applications, particle filtering is extremely important in directing the processing tasks and focusing the analysis only on the relevant particles [6], [7]. In this application, the length, width and diameter of the template particle are used as criteria for removing all particles that do not match them within some tolerance. They are also filtered by the number of holes they contain, this being the only sign that a particle might represent the rim of a road sign. A more refined search is then performed on the remaining particles.

The object or objects derived from morphological operations are modified to form masks for the inside the road sign. These operations are used for expanding or reducing particles, smoothing the borders of objects, filling the holes they contain, preparing particles for quantitative analysis, finding the external and internal boundaries of particles, observing the geometry of regions, locating particular configurations of pixels and extracting them for modeling and identification purposes [6], [8].

The resulting image is used as a mask on the original image in order to extract regions of the image that represent potential signs for the shape processing stage. A mask is a binary 8-bit image that isolates the parts of an image for processing. The pixels of the mask determine if the inspected corresponding pixels in the image are processed. If a pixel of the mask has a non-zero value, the corresponding pixel in the image is selected. If a pixel of the mask is 0, the corresponding pixel in the image is converted to a background pixel (a black pixel). Therefore, the pixels in the source image pixels are processed if the mask contains values other than zero [6], [7]. If at the end of this algorithm there is at least particle in the mask, then the analysis module is activated.

Since there can be more than one road sign in an image, there can be more than one particle in the mask. Hence, the mask must be divided in single particle masks. To this purpose, particle analysis is used. This consists of a series of processing operations and analysis functions that produce information about particles in an image, including statistical information, such as the particle size or the number, location, and the presence of particles. A typical particle analysis process scans through an entire image, detects all the particles in the image, and builds a detailed report on each particle. This can be used to uniquely identify an object (a particle) in an image [6], [8]. The positions and dimensions of the mask are inserted into a vector, further processing being applied for the generation of each individual mask.

The resulting binary images (single particle masks) are passed through a shape detection function (Figure 3).

2.2. Shape Detection

Two major shape detection algorithms have been used in this project depending on the characteristics of the sign category, namely Shape Matching and Shape Detection. Shape Detection (Figure 4.a) is more accurate, but it can only detect rectangles and circles. Shape Matching (Figure 4.b) may seek a shape after a given model, but it cannot find regular shapes with accuracy with which Shape Detection can. Therefore circles and rectangles are sought for with Shape Detection and triangles are sought for with Shape Matching. Both functions are able to detect shapes regardless of size and orientation. The output of these functions is a report detailing the location of each part, the centroid of the part, and a score that indicates the degree of match between the template and the part. This helps focus the search on the regions of the image that are likely to be traffic signs, rather than scanning the entire image for possible results. If the following stage is based on a different characteristic of a road sign than the stage that produced the mask, it also becomes less likely to yield false positives [9].

Depending on the results of the shape function, the single-particle mask then is used to extract the sign candidate from the original image. Before the original image is masked, color equalization is applied in order to enhance it and to improve contrast and brightness. The extracted symbol is rotated in accordance with the detection results of the shape matching step, and then resized to 105x105 pixels in size. Since IMAQ functions use template-based recognition, they only work with objects (signs) the same size as the template. Hence resizing the image is a key point in the algorithm. A black 10-pixel border is also applied to the sign, all to ensure recognition.
2.3. Recognition

The obtained image is then passed through three different recognition techniques, grouped in a recognition module. These consist of pattern matching, color matching and color-pattern matching.

**Pattern matching** locates regions of a grayscale image that match a predetermined template, a reference pattern which constitutes an idealized representation of a feature. Pattern matching works reliably under various conditions including poor lighting (only under uniform changes in the lighting across the image), blur, and noise, shifting of the template or rotation of the template. The machine vision application searches for the model in each acquired image, calculating a score for each match,
that relates how closely the model matches the pattern found. If the measurement falls within a tolerance range, the part is considered good. If it falls outside the tolerance range, the component is rejected. The result is a fast and accurate pattern matching [6], [8].

**Figure 4.** Shape functions

**Color matching** is used to select regions within an image that contain the color information used as a reference. The color information in the image may consist of one or more colors. The machine vision software then learns the three-dimensional color information in the image and represents it as a one-dimensional color spectrum, comparing the color information in the entire image or regions in the image to the learned color spectrum, calculating a score for each region. The match score, ranging from 0 to 1000, relates how closely the color information in the image region matches the information represented by the color spectrum, defining the similarity between the color spectrums. A score of zero represents no similarity between the color spectrums, whereas a score of 1000 represents a perfect match.

**Color pattern matching** is a unique approach that combines color and spatial information to quickly find color patterns in an image. It uses a combination of color location and grayscale pattern matching to search for the template. Color pattern matching uses color matching first to locate the objects, and then pattern matching to refine the locations, providing more accurate results. The addition of color helps to improve the accuracy and reliability of the pattern matching tool, making it less sensitive to factors such as orientation of the part, scale changes, and lighting changes and maintaining its ability to locate the reference patterns and gives accurate results despite these changes [6], [7].

The machine vision application searches for a model or template that represents the object, in each acquired image, calculating a score for each match. The score indicates how closely the model matches the color pattern found. It also provides the application with information about the number of instances and location of the template within an image.

A sign is recognized (identified) if it obtains a score higher than a threshold level on each of these algorithms. An object is passed through the recognition module of a particular indicator if there is a chance that the object might represent the sign in question. The possibility is computed by analyzing the score with which the rim shape was detected. This is not necessarily an accurate result, and if it were left only to this, a sign would usually have to be passed through all the templates for that particular sign category. Considering the fact that the execution time for a sign category that contains only one sign (the stop sign) is 0.12 seconds, the duration for a category containing 34 signs, as is the triangular red category, would be 4.08 seconds. This is obviously too long for real-time sign recognition. Considering that the Stop sign is the only sign in its category (red hexagon), its duration is minimal possible in this configuration (Figure 5), but for other sign categories further processing is requires in order to reduce the number of templates tested and, implicitly, the execution duration.
2.4. Comparison between processing algorithms

Several attempts have been made in order to reduce the duration of this algorithm. First, we used the color and shape detection functions applying them, this time, to the background of the shape. The image is then taken through another color threshold in order to detect the background color (usually white but can also be blue, depending on the sign type) or the interior color of the sign. Therefore, the lack of the specific background color in the candidate constitutes a possible sign rejection criterion. The resulting binary image is taken through the shape matching step, this time for the shape of the background. Using Shape matching instead of Shape detection is mandatory in this case, since the shape of the background can form the pictogram of the sign and does not constitute a regular shape [6].

Shape in road signs is particularly difficult to assess; it is, generally speaking, irregular and can be easily influenced by objects that obstruct the image. From this point of view it is necessary that the shape detection be as robust as possible, making it, therefore, less accurate. This facilitates confusion between signs of similar shape. Therefore, shape detection is used to classify rather than to recognize the individual road sign. Pattern matching is used to detect the individual signs. This automatically means that the pattern matching function will have to search through several models and choose the best fit for the sign sought after. Shape analysis helps sort the number of models which should be tested and focus the search on the most likely model. This step has the purpose of activating only the recognition module based on the most likely matches for the sign in question. The search result represents the sign with the maximum found score, score that exceeds a certain threshold. If the maximal score is below the threshold, the candidate either does not represent a sign or does not represent a sign belonging to that category. One example of such an algorithm, which can be seen in Figure 6, is used on the circular blue signs. It presents an average execution time of 0.66 seconds to 11 signs.

Using this algorithm, for a number of 34 signs, the duration would increase to around 2 seconds, which is a significant improvement from the performance of the stop algorithm. Another improvement would be the use of a more complex shape matching function that searches for a correspondence between the background shape and the rim shape. The program basically looks for an irregular shape formed from the exterior and the interior margins of the contour of the sign (Figure 7). This verifies...
the existence of a correspondence between the size and orientation of the contour and the background. This operation has not shown significant improvements in duration (15 signs in 0.87 seconds), but it increases the accuracy of the algorithm.

![Diagram of subprograms that use pictogram analysis](image)

**Figure 7.** Diagrams of subprograms that use pictogram analysis

Another option is to search according to the color/colors of the pictogram. There are road signs where the pictogram is the background, but many have both a background and a pictogram, different from one another. Therefore a secondary search based on the color of the pictogram could improve processing time. This operation was used on the rectangular blue signs, these having different colored pictograms. The resulting algorithm can be seen in Figure 7.a.

A more complex version of this algorithm was used for the red circular sign category. There is significant improvement in this case (27 signs in 1.2 seconds). The diagram can be seen in Figure 7.b.
A variation of the latter is the triangular red sign recognition algorithm. It can check 34 patterns in 0.63 seconds. The improvement consists of the use of pattern matching for determining the shape of the pictograms. This seems counterintuitive, since pattern matching functions are those that give the duration of the process, being the most time consuming of all the functions used in this program. The logical motivation would be the fact that the execution time of these functions depends on size of the template and on the image to be checked. Since the pictogram of the sign is significantly smaller than the sign itself, it takes much less to check its presence in the image than of the entire pictogram. It is also much more precise than a shape matching function, leaving less pictogram patterns to check in comparison to shape matching. This algorithm can be seen in figure 8. These methods could be improved in order to reduce the execution time, and so increase the execution speed. Execution duration is influenced by the number of different types of signs simultaneously detected in the image, the results of the intermediate functions of shape and color, the number of matching modules simultaneously active, the time it takes to receive data from the module, etc. Therefore, the execution time can vary within a fairly wide range (± 100%). However, the average duration of an algorithm still represents a possibility to compare algorithms between them.

![Figure 8. Triangular red signs recognition diagram](image)

3. Conclusions
Identifying road signs is a simple task for human beings, but it appears to be quite the challenge for computers. If a person is asked to point out the traffic sign in an image, they can do this easily. However, from the point of view of computer vision, there are a number of difficulties, such as lighting, weather condition, incorrect positioning, obstructions, etc. To deal with them, the road sign recognition system is provided with large amounts of training data to allow the system to respond correctly when a traffic sign is encountered.

The program, because it works with thresholds and probability, is robust and effective. The chance of false alarms has been reduced by filtering the results gradually through successive stages. Each potential sign is tested with a set of features to decide if it can represent a road sign or not, and then, depending on these characteristics, potential signs are classified into different groups. Different levels of detection work with different procedures and are based on different principles of seeking different attributes of road signs, eliminating false signs.

Road signs can be either pictogram-based or text-based. The road and traffic signs considered in this paper are those that use a visual/symbolic language that can be interpreted by drivers. The signs
containing text are not included, as these are subject to further developments based on optical character recognition. Supplementary signs are also not included, because their meaning is dependent on the associated road sign. In this regard, an algorithm capable of combining sign meanings based on proximity is needed. This is also an opportunity for further improvement.

Road signs may appear rotated, resulting in perspective errors. Perspective errors are caused by the incorrect positioning of the sign itself, which is not perpendicular to the axis of the road, and therefore not perpendicular to the axis of the camera. The sign should be directed towards the road to which it corresponds, but this is not always the case, making the sign difficult to identify even for experienced drivers. Our system works well regardless of lighting conditions, faded colors, obstructions, etc. but it has limited capabilities regarding perspective errors. A further improvement to the program would be, therefore, to make it invariant to perspective errors.

The algorithm presented in this paper is based on the detection of the contour color of road signs, but there are signs that do not have a continuous contour, for example the End of compulsory minimum speed sign. Generally speaking, these are known signs crossed with a red stripe. Theoretically, by using a HSI threshold composed of both colors, this sign could be detected with this system. The problem is that the signs that have a contour consisting of only one color would also be detected in this category and would complicate the identification of the sign category. This could be solved through a color matching function that would determine the category to which the sign belongs, based on its color spectrum.

References
[1] Fleyeh H 2008 Traffic and Road Sign Recognition, Napier University, UK, Doctoral Thesis
[2] Fleyeh H and Dougherty M 2005 Road and traffic sign detection and recognition, Proceedings of the 16th Mini-EURO Conference and 10th Meeting of EWGT, Poznan, Poland, September 13-16, pp 644-653
[3] Greenhalgh J and Mirmehdi M 2012 Real-time detection and recognition of road traffic signs, IEEE Transactions on Intelligent Transportation Systems 13(4) 1498-1506
[4] Mathias M, Timofte R, Benenson R and Van Gool L 2013 Traffic Sign Recognition - How far are we from the solution?, Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN 2013), Dallas, USA, August 4-9, pp 1-8
[5] Torresen J, Bakke J W and Sekanina L 2004 Efficient recognition of speed limit signs, Proceedings of IEEE Conference on Intelligent Transportation Systems (ITS), Washington DC, USA, October 3-6, pp 652-656
[6] ***National Instruments 2003 IMAQ Vision Concepts Manual, June 2003 Edition, 322916B-01
[7] ***National Instruments 2004 IMAQ Vision for LabVIEW, August 2004 Edition, 371007A-01
[8] ***National Instruments 2005 NI Vision for LabVIEW User Manual, November 2005, 371007B-01
[9] ***National Instruments 2008 NI Vision Assistant Tutorial, June 2008, 372228K-01