17th Symposium IMEKO TC 4, 3rd Symposium IMEKO TC 19 and 15th IWADC Workshop
Instrumentation for the ICT Era
Sept. 8-10, 2010, Kosice, Slovakia

Traffic Signs Recognition Experiments with Transform based Traffic Sign Recognition System

Martin FIFIK[1], Ján TURÁN[1], Ľuboš OVSENÍK[1]

[1] Department of Electronics and Multimedia Communications *(Technical University of Košice Park Komenského 13, 041 20 Košice, Slovakia,fax:* +421556323989 *,email: martin.fifik@tuke.sk)*

Abstract- This paper describes tests on transform based traffic sign recognition system. Traffics sign recognition subsystems: preprocessing, projection transform stage, and classification are outlined. Results of traffic sign detection and recognition from real image taken from Slovak roads are shown and discussed.

I. Introduction

Video-based object recognition has many applications in different fields such as image retrieval, surveillance systems, driver assistance systems etc. [1]. Every designed system in this field has some specific requirements for input data (image versus video), requirement on computational time, cost and expectations on reliability. In general there is some basic idea: the input data are processed, and then from these data are extracted some relevant effective invariant features.

Driver assistance systems are systems to help the driver in its driver process. In driver assistance systems the most important thing is car safety and more generally road safety. The goal for developers is to create an ultimate system in car that can warm us or safe us before potential danger. The rules for safety traffic are displayed on traffic signs. Traffic signs are designed to show us some rule or warn us before something [1-4].

Goal of this paper is present tests on transform based traffic sign recognition system. Projecting transforms are used for invariant feature extraction.

II. System for Traffic Sign Recognition

The designed transform based traffic sign detection system can be divided in two subsystems[3]. One, which is called fast, composed from CSR block and Sign Class Classification block, and second slower, composed from Image Segmentation, Invariant Feature Extraction System, Feature Memory, Feature Modification and Sign Type Classification blocks (Fig.1).

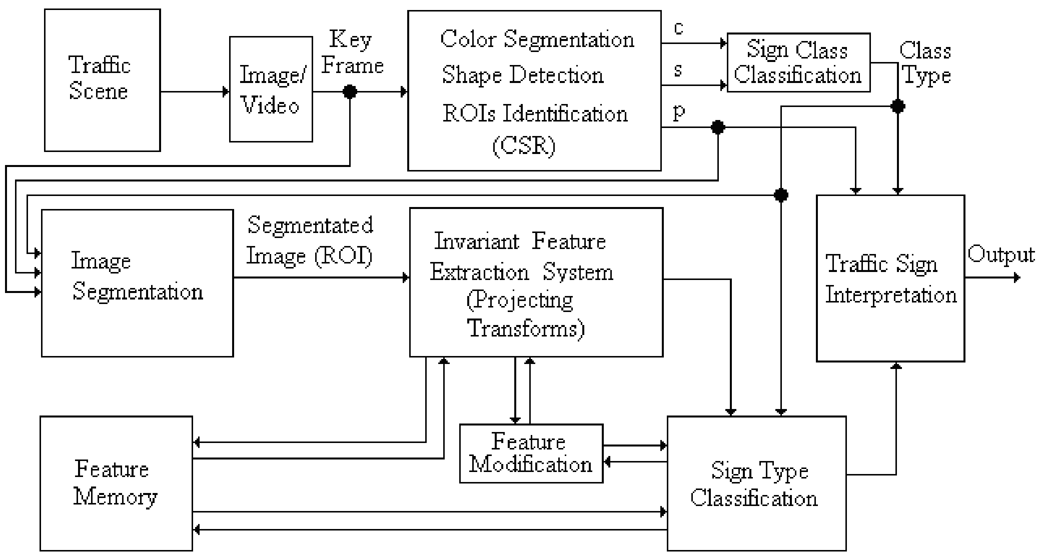

Figure 1. Traffic sign detection system architecture.

Traffic scene is recorded with color camera. From camera created data stream, key frame is extracted. Key frame is used in next processing. This key frame is processed in CSR block, where the base features from image are extracted. Features extracted from key frame are color, shape and ROIs position. This preprocessing is very fast, because no special transformation is there needed. From many color spaces we choose RGB and HSV colour space. By extracting color, CSR block generates 3 binary maps (red, blue and yellow). In this bitmaps we are looking for shapes of traffic signs. If traffic sign was detected, then from color and shape features traffic sign class type is identified. Traffic class type goes together with position of traffic sign to block for Image Segmentation. From key frame the ROI is segmented, which is input to block called Invariant Feature Extraction System.

In Invariant Feature Extraction System projecting transform (Trace transform) are used [6,8]. Invariant features are features that have equal values in cases when the image is translated, rotated, resized or have some other modification. The block of Invariant Feature Extraction System with cooperation with blocks Feature Memory and Feature Modification block, gives information for Sign Type Classification block. In this block are brought information about class type of detected traffic sign. An output of this block gives us completely recognized traffic sign for block called Traffic Sign Interpretation.

A. CSR Block

This block is preprocessing block. In experiments two colour spaces are used: HSV and RGB.

HSV color space

Input is converted in to the HSV color space. Every traffic sign has his dominant color. On Slovak roads most often yellow, red and blue color are used. This means we need to create three binary maps, one for each of these colors.

By analyzing hue component (H), we can identify blue, yellow and red regions in our detected image. For each image pixel, hue-based detection of blue, red and yellow colors is done. For each color one passes one of following equation [1]

$$Y = e^{\frac{-(x-42)^2}{30^2}} \qquad (1)$$

$$R = e^{\frac{-x^2}{20^2}} + e^{\frac{-(x-255)^2}{20^2}} \qquad (2)$$

$$B = e^{\frac{-(x-170)^2}{30^2}} \qquad (3)$$

Equations Y gives values close to 1 for yellow regions, R gives values close to 1 for red regions and B gives values close to 1 for blue regions. In this equations we can see, that H can be from range 0-255. Yellow can be detected near value 42, red near values 0 and 255 and blue value is 170. These equations can be tuned for every color.

Now we need saturation detection value, by exploring the S component. This is described by following equation

$$S = e^{\frac{-(x-255)^2}{115^2}} \qquad (4)$$

From equations (1), (2) and (3) we got 3 values. Every value is multiplied with S value. This value will be called D. From D we create D_n, which means D normalized. Values close to zero will be discarded. Other will follow next equation (5).

$$D_n = \begin{cases} 0, if \ \det < 0,3 \\ \left(\frac{\det - 0,3}{0,7 - 0,3} \right)^2 \\ 1, if \ \det \geq 0,7 \end{cases} \qquad (5)$$

Now the threshold can be created. For this we use Otsu's algorithm on D_n [7]. Then we can create three binary maps.
Now every binary map must be cleared. Too small regions and too big regions are discarded. In first step we must find first white point in image. Searching is done by rows. After finding first white point, method called seed-fill is used. With this method we are finding regions, and these regions are valued. If valued region has value lower than 400, then is discarded. If region has value more than 16000 then is also discarded. The regions that are left are potential candidates for next processing.
Method seed-fill we can apply only, if all white areas in image have their boundaries. Next condition is that there must be at least one white point. After successful finding white point A (seed), its neighbors are tested (Fig. 2).

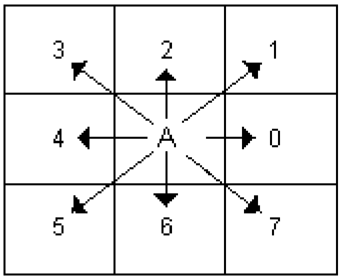

Figure 2. Values 0-7 for neighbors of seed A

This white point is filled and its neighbors' relations are checked. These points are now stored in seed vector. In next step we continue with last filled point. If this point has no unfilled neighbor, then point from seed vector is choosing for next checking. After using all points from seed vector, we have our region that has value. On the end we have selected **ROI**(s).

RGB color space
Input is in RGB format. First step is brightness correction of input image. Brightness correction was done manually.
Next step is color segmentation. This has been done by using following equations

$$\begin{aligned} &\alpha_{\min} * G < R < \alpha_{\max} * G \\ &\beta_{\min} * B < R < \beta_{\max} * B \qquad (6) \\ &\gamma_{\min} * B < G < \gamma_{\max} * B \end{aligned}$$

Three binary maps have been generated; bitmaps are cleared with the same way as it was at HSV. Those bitmaps are used for ROI detection. ROI detection is done by same method at HSV detection[1].

Shape detection
Now with pattern matching method, on that size of ROI are created perfect shapes, like it is show on Fig.3. Here every **ROI** is tested pixel by pixel with every perfect shape, with circle, rhomb, triangle, reverse triangle, filled circle-STOP sign, square /rectangle [1,5].

Figure 3. Four perfect shapes

B. Invariant Feature Extraction System Block

This block is used for image detection inside traffic signs. The major problem of this image is that it is not perfect. For example: it is resized, rotated or moved.
For invariant feature extraction projecting transform (Trace transform) is used here. Trace transform can be understood as generalization of the well-know Radon transformation. Trace Transform based invariant feature extraction is detaily described in [8].

C. Sign Type Classification Block

TABLE I. TRAFFIC SIGN TYPE CLASSIFICATION

Shape \ Color	Red	Blue	Yellow
Square/Rectangle	-	Information	-
Circle	Obligation	Prohibition	-
Rhomb	-	-	Highway
Triangle	Danger	-	-
Reverse triangle	Yield sign	-	-
Cut square	Stop sign	-	-

Now when we got shape and colour we can classify traffic signs to classes (Type of traffic sign). This is shown in Tab. I.
All possibilities are not obtained in Tab. I. When we know type and image as output from feature extraction system, then we can classify sign in to the exact classe. For example, it's red triangle with image symbolizing kids, so this sign is classified and recognized as beware of children.

III. Experiments

Experiments were done for HSV and RGB color space. Input images were in resolution 640x480 pixels. Tests were done on Pentium PC with dual core processor 2 x 2 GHz.
In HSV color space, the number of tested images was 326; number of false positive tested was 32, which means 90,18 % successful rate. Details are shown in Tab. II.
In RGB color space, the number of tested images was 326; number of false positive tested was 26, which means 92,02 successful rate. Details are shown in Tab. III**.**

TABLE II. RESULTS FOR HSV COLOR SPACE

Brightness condition	Number of sings	Detected signs	Rate[%]	Average rate[%]
low	93	80	86,022	
normal	220	211	95,909	
extreme	13	3	23,077	90,18

TABLE III. RESULTS FOR RGB COLOR SPACE

Brightness condition	Number of sings	Detected signs	Rate[%]	Average rate[%]
low	93	81	87,097	
normal	220	214	97,273	
extreme	13	5	38,462	92,02

Preprocessing time in both color spaces was average from 1 second. Average processing time in invariant feature extraction system was 2,2 seconds. Traffic sign classification average time was 0,2 seconds.

IV. Conclusion

Experiments show us average system recognition rate 90% at HSV color space and 92% at RGB color space. Factors which affect images in preprocessing stage was darkness and sun lightning. Trace transform based invariant image recognition system effecivity depends on chosed functional combination. For best result were used three best combinations of these functionals. All reached results was for standard PC and non optimised software solution in 2-3 seconds per image, which means that this system can be used in real time.

Acknowledgement

This work was partially supported from the grants VEGA, project COST IC0802 and by Agency of the Ministry of Education of the Slovak Republic for the Structural Funds of the EU under the project Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120020).

References

[1] F. P. Paulo, L. P. Correia, "Automatic detection and classification of traffic signs," Image Analysis for Multimedia Interactive Services, 2007, WIAMIS, 6-8 June 2007.
[2] A. Laika, W. Stechele, "A review of different object recognition methods for the application in driver assistance systems, " Image Analysis for Multimedia Interactive Services, 2007, WIAMIS, 6-8 June 2007.
[3] J. Turán, M. Fifik, Ľ. Ovseník and J. Turán Jr., "Transform based system for traffic sign recognition" 15th International Conference on Systems, Signals & Image Processing IWSSIP'08, Bratislava, Slovakia, 2008.
[4] A. Broggi, P. Cerri, P. Medici, P. P. Porta, "Real Time Road Signs Recognition", Intelligent Vehicles Symposium, 2007 IEEE Volume , Issue , 13-15, 981 – 986, June 2007.
[5] M. Nixon, A. Aguado "Feature extraction and image processing", Academic Press, 2008
[6] J. Turán, "Fast translation Invariant Transforms and their Application," Elfa, Košice, 1999.
[7] N. Otsu. "A threshold selection method from gray-level histogram" IEEE Transactions on Pattern Analysis and machine Intelligence, vol. 25, nº8, August 2003, pp 959- 973
[8] J. Turán, D. Šiškovičová, Ľ. Ovseník and J. Turán Jr., P.Filo, "Trace transform and KLT Based Invariant Feature and Image Recognition System", Acta Electrotechnika et Informatica, no.3, vol. 6, 2006

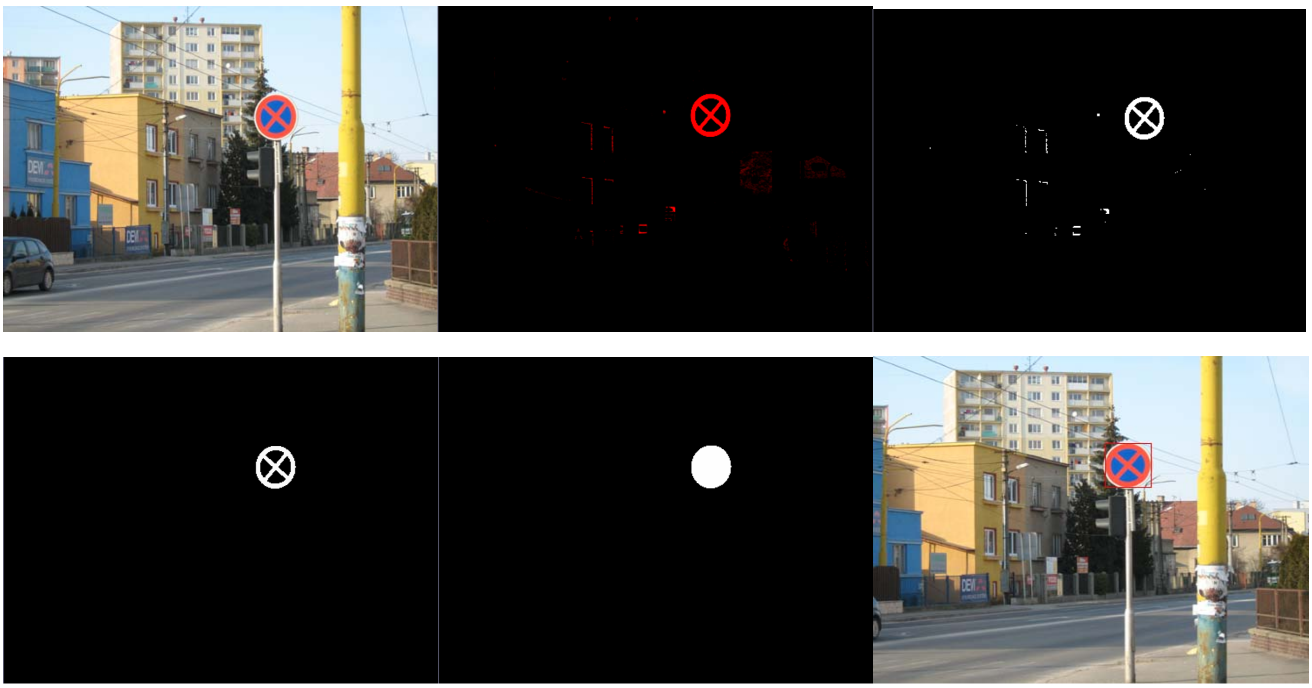

Figure 4. Samples of images for preprocessing and ROI extraction from red color in HSV color space