Journal of Urban and Environmental Engineering, v.13, n.1, p.59-68

ISSN 1982-3932
doi: 10.4090/juee.2019.v13n1.059068

Journal of Urban and Environmental Engineering

www.journal-uee.org

TSRS - A NEW APPROACH FOR TRAFFIC SIGN RECOGNITION USING THE SIFT ALGORITHM

Francisco A. Silva[1*], Danillo R. Pereira[1], João F. C. Silva[2], Almir O. Artero[3] and Marco A. Piteri[3]

[1]Department of Computer Science, University of Western São Paulo (Unoeste), Presidente Prudente, São Paulo, Brazil

[2] Department of Cartography, Faculty of Science and Technology, São Paulo State University (Unesp), Presidente Prudente, São Paulo, Brazil

[3] Department of Mathematics and Computer Science, Faculty of Science and Technology, São Paulo State University (Unesp), Presidente Prudente, São Paulo, Brazil

Received 12 December 2018; received in revised form 27 February 2019; accepted 27 February 2019

Abstract: This paper proposes a new approach for traffic sign recognition using images captured by a low-cost mapping system. The proposed approach applies the SIFT algorithm to extract keypoint features that are used to evaluate the correspondences between a road image containing one or more plates and the images of traffic signs (templates). The BBF algorithm was used to efficiently evaluate the correspondence between the SIFT features. Finally, we propose a new algorithm to filter only the pairs of keypoints (image-template) that are compatible as well as the orientation and positioning.

Keywords: Traffic Sign Recognition; Character Recognition; SIFT; RANSAC.

[*] Correspondence to: Francisco A. Silva, Tel.: +55 18 3229 1060.
E-mail: chico@unoeste.br

1 INTRODUCTION

Traffic signs are very important because they establish the navigational rules for the roads. Their visual properties are very strong because they are designed for the easy identification by human beings (Arlicot *et al.*, 2009). The recognition of traffic signs (Timofte *et al.*, 2011) (Hazelho *et al.*, 2012) (Larsson & Felsberg, 2011) (Mathias *et al.*, 2013) allows a driver to be aware of appropriate or inappropriate actions and potentially dangerous situations (Bar *et al.,* 2009). The recognition of traffic signs by machines has been studied for several purposes, such as autonomous and assisted direction (Handmann *et al.*, 1998).

In general, the identification of traffic signs is made in two stages, the detection and the recognition. Firstly, an algorithm is applied to detect the traffic sign and then the region of the sign is located. Afterwards, a recognition algorithm uses this region to identify the type of the traffic sign.

In literature several approaches can be found as an attempt to solve the problem of traffic sign recognition, such as: using color properties of the signs (Creusen *et al.*, 2010); from the plates geometry in grayscale images (Loy & Zelinsky, 2003); based on the signs shape properties (Belaroussi & Tarel, 2009); based on color properties and shape (Arlicot *et al.*, 2009) (Çağlayan *et al.*, 2015); using classifiers based on Neural Networks (Martinovi *et al.*, 2010) (Jung *et al.*, 2016); analyzing the curvature of the sign representation with Active Contours (Snakes) (Paulo & Correia, 2008); using Support Vector Machine (SVM) (Barnes & Zelinsky, 2004); traffic signs segmentation based on Fuzzy Sets (Fleyeh, 2005); using a Machine Learning algorithm Adaboost (Bar *et al.,* 2009); to overcome the difficulties of recognizing traffic signs, Ciresan *et al.* (2011) suggests to use a committee of neural network.

Local features have been used in the context of Traffic Sign Recognition (TSR); they have been applied both for the traffic signs detection and for its recognition, due to their strengths and discriminative qualities (Höferlinm & Zimmermann, 2009).

Local features techniques detect the characteristics of a particular image, and then a feature set is computed for these characteristics (Mikolajczyk, 2002). The features represent the typical parts of an image, and there are many algorithms in literature for the extraction of local features. An algorithm that has been highlighted in this step is the SIFT algorithm (Scale Invariant Feature Transform) (Lowe, 2004), that it is invariant to scale, rotation and viewpoints. Höferlin & Heidemann (2010) presented a study evaluating the results obtained with local features with the following algorithms: SIFT, GLOH (Gradient Location and Orientation Histogram) and some variants of SURF (Speeded Up Robust Features) and they conclude that SIFT is the most appropriate to treat the TSR problem. This work was based on the evaluation accomplished by them, and also on the feature evaluation proposed by Mikolajczyk & Schmid (2005), which showed SIFT as the strongest local feature in terms of the sensitivity and accuracy with regard to tested approaches.

Other studies can be found in literature using the SIFT algorithm for TSR applications: Farag & Abdel-Hakim (2004) use a Bayesian classifier from the set of features obtained by SIFT; in Kus *et al.* (2008) color information are added to SIFT features; Höferlin & Zimmermann (2009) perform the detection using SIFT and, for the recognition step, they use a neural network (multilayer perceptron); the work of Reiterer *et al.* (2009) is based on the SIFT algorithm and clustering k-means algorithm; the work proposed by Cai *et al.* (2010) presents a method for triangular traffic signs recognition based on SIFT features, where a decision tree is used to classify the signs.

In this paper, we present a new approach to the TSR process using SIFT algorithm including a new step to filter only the pairs of keypoints (image-template) that are compatible as well as the orientation and positioning. The proposed approach has been tested with images with a lot of details (complex scenes) and also containing several factors that cause difficulty to the recognition of the plates, such as: weather conditions, variations in lighting, shadows, damaged plates, presence of trees, etc. The other sections of this paper are organized as follows: Section 2 describes the SIFT algorithm, used to identify and describe keypoints in the images; Section 3 presents a new approach that uses the features obtained with SIFT to accomplish the matching between the template and input images; Section 4 presents some experiments with the proposed approach; Finally, Section 5 presents some conclusions and suggestions for future work.

2 SIFT ALGORITHM

The identification of homologous points in two images is not a very simple task; there are a lot of researches in the area for its automatic execution (Basri & Jacobs, 1997) (Friedman *et al.*, 1977). In fact, the first difficulty is to find the interesting points (keypoints) in the first image, so that they can be searched in the second image. As in the works presented in Brown and Lowe (Brown & Lowe, 2007), Liu *et al.* (2008), Cai *et al.* (2010) and Takimoto *et al.* (2010), in this paper we use the SIFT algorithm to find the interesting points in the image and in the templates. The SIFT algorithm is a very efficient method to identify and to describe image keypoints, which is done by performing a mapping with different views of an object or scene, resulting in a vector with 128 values, describing each image keypoint. Additionally, the SIFT algorithm provides the position x, y, scale s and orientation θ for all keypoints (Lowe, 2004). The SIFT algorithm consists in the following steps:

Scale-space extrema detection: The keypoints are detected by applying a cascade filtering that identifies candidates that are invariant to scale. The scale-space is defined as a function $L(x, y, \sigma)$ in **Eq. (1)**, with an input image $I(x, y)$ (Koenderink, 1984) (Lindeberg, 1994).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

where * is a convolution in x and y with the Gaussian $G(x, y, \sigma)$ in **Eq. (2)**.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2)$$

To detect stable keypoint locations in space-scale, Lowe (1999) proposes the use of space-scale extrema in the difference-of-Gaussian (DoG) function convolved with the image $I(x, y)$, resulting in $D(x, y, \sigma)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k, as in **Eq. (3)**. The DoG is an approximation of the scale-normalized Laplacian of Gaussian $\sigma^2\nabla^2G^2$. The maxima and minima of $\sigma^2\nabla^2G^2$ produce the most stable image features (Mikolajczyk, 2002).

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (3)$$

Local extrema detection: From $D(x, y, \sigma)$, in Lowe (2004) it is suggested that the local maxima and minima must be detected by comparing each pixel with its eight neighbors in the current image and nine neighbors in the scale above and below (26 neighbors). SIFT guarantees that the key points are located at regions and scales of high variations, which make these locations stable for characterizing the image.

Orientation assignment: The scale of the keypoint is used to select the Gaussian smoothed image L, with the closest scale, so that all computations are performed in a scale-invariant manner. The gradient magnitude $m(x, y)$ is computed by:

$$m(x, y) = \sqrt{\Delta x^2 + \Delta y^2} \quad (4)$$

where $\Delta x = L(x + 1, y) - L(x - 1, y)$ and $\Delta y = L(x, y + 1) - L(x, y - 1)$. The orientation $\theta(x, y)$ is calculated by **Eq. (5)**.

$$\theta(x, y) = tan^{-1}(\Delta y / \Delta x) \quad (5)$$

Keypoint description: The next step is to compute a descriptor for the local image region that is distinctive and invariant to additional variations, such as the changes in illumination or 3D viewpoint. Lowe (2004) suggested that the best approach is to determine the magnitudes and directions of the gradients around the keypoint location. In this approach the Gaussian image on the keypoint scale is used.

2.1 Matching between images

To find the match between images it is possible to use the keypoints detected with the SIFT algorithm. Lowe (2004) proved that the best match for each keypoint is found by identifying its nearest neighbor, which is defined by minimizing the Euclidean distance to the features vectors. In order to avoid an exhaustive search, he suggested the use of a data structure k-d tree (Beis & Lowe, 1997) that supports a balanced binary search to find the closest neighbor of the features and the heuristic algorithm *Best-Bin-First* (BBF) is used for the search.

Several studies have shown the effectiveness of the SIFT algorithm to identify and describe keypoints in images, which is done by a vector containing 128 descriptor values, however, not always a high correlation between the corresponding values of two vectors means there is a correlation between the respective points of interest. When matching the vectors it reveals a real correspondence between points in the image, so it is said that these points are inliers. On the other hand, when a strong correlation between the values of the vector results in an erroneous correlation between the image points, these points are called outliers. The best known algorithm to delete outliers is the RANSAC (Fischler & Bolles, 1981).

2.1.1 RANSAC

RANSAC (RANdom SAmple Consensus) is the algorithm name proposed by Fischler & Bolles (1981); it is a robust estimation method designed to identify the inliers and outliers from the set of keypoints detected by the SIFT algorithm. RANSAC is widely used for object recognition (Okabe & Sato, 2003), besides it makes it possible to find the geometrically consistent correspondences to solve the problem of joining pairs of images. RANSAC is a robust estimator that gives fine results, even in extreme conditions or with some kind of outliers in the data set.

Unlike the conventional techniques that use a lot of data to obtain an initial solution, and then eliminate the outliers, RANSAC uses only a set with a minimum number of required and sufficient points for a first estimate and it continues the process by increasing the set of data points consistent. A minimum sample depends of the model to be generated.

From a minimum necessary set of points randomly selected, RANSAC measures the accuracy of the model for all other points, classifying them in inliers or outliers. A threshold value t is defined from the maximum distance of the model that a data can be considered an inlier. The set of inliers associated to a model is the consensus set of the model. Thus, with several iterations, the algorithm is able to find an accurate model for the problem at hand. According to

Fischler & Bolles (1981) and Hartley & Zisserman (2003), the steps of the RANSAC execution are:

1. Randomly select a sample of s data points from S and instantiate the model from this subset;
2. Determine the set of data points S_i which is within a distance threshold t of the model. The set S_i is the consensus set of the sample and defines the inliers of S;
3. If the size of S_i (the number of inliers) is greater than some threshold T, re-estimate the model using all the points in S_i and terminate;
4. If the size of S_i is less than T, select a new subset and repeat the above;
5. After N trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i.

The number of the iterations N is chosen sufficiently high to ensure with a probability p (normally chosen at 0.99), that at least one of the random samples of s points is free from outliers. Suppose w is the probability that any selected data point is an inlier and $e = 1 - w$ is the probability that it is an outlier. Hartley & Zisserman (2003) define the amount of selections N using the **Eq. (6)**.

$$(1-w^s)^N = 1-p \tag{6}$$

and thus the number of iterations required is given by **Eq. (7)**.

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)} \tag{7}$$

RANSAC calculates the best fit of matching SIFT features previously extracted from two images. Although the RANSAC works well in many situations and it returns the homographic matrix between the corresponding points in the two images, it presents problems when the SIFT algorithm detects few keypoints, aligned and very close together. This occurs frequently in the problem of recognition of traffic signs, because of the simplicity of the templates. In this work a new proposal is presented for the removal of outliers, and it is able to operate in conditions where the RANSAC fails.

3 TSRS ALGORITHM

Traffic Sign Recognition using SIFT (TSRs) is the algorithm proposed in this paper; it is displayed in **Fig. 1**. Initially, the SIFT algorithm is applied to all templates containing the traffic signs used; then a database containing the keypoints features from these templates is built. This is done to avoid that the SIFT has to be applied to all templates for each input image, because its processing time is not negligible. A very important feature of this approach is that a minimal training is necessary because the algorithm can work with only one model of each traffic sign template.

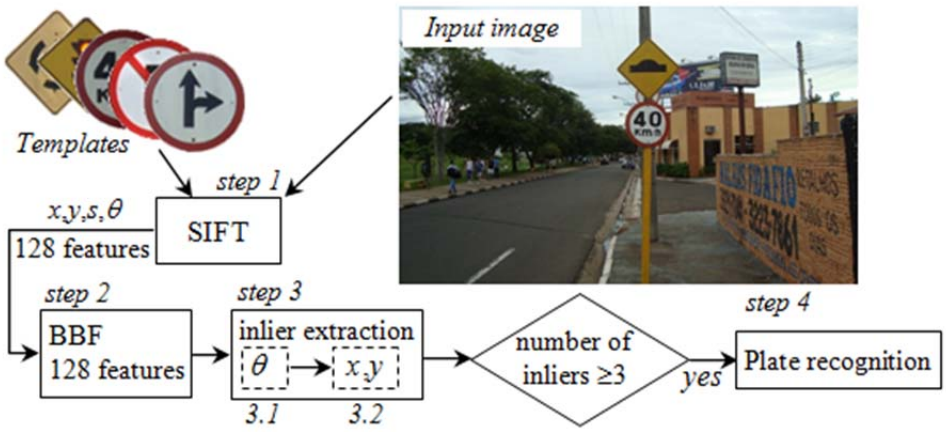


Fig. 1 The TSRs algorithm proposed for the traffic sign recognition.

In Step 1 the SIFT algorithm is applied only to the input images, generating the information that describes each keypoint. In Step 2, the 128 features obtained from each keypoint of the input image are compared with the 128 features obtained from each keypoint of the templates.

In **Fig. 2**, the straight lines connecting the points in the template and the points in the input image indicate their features that are matched. At this time, most of the input image keypoints which do not match a template keypoint were eliminated. However, there are still some points that do not correspond with the template. Thus, Step 3 needs to identify only the inliers (data points that fit a particular model within an error tolerance), that is done in two stages: 1) comparing the orientation angles of local gradients, 2) searching for a region that concentrates the most of the inliers and calculating the correlation between spatial coordinates of keypoints in the input image and the template.

The templates and the input images may present differences in orientation for several reasons: misalignment of roads, poorly fixed plates, and others.

Thus, the algorithm used for finding correspondence between images must be rotation invariant, as it occurs with the SIFT.

Fig. 2 Matching between a) template and b) input image, after the keypoints after evaluation of the 128 SIFT features.

The proposed method is based on rotations below ten degrees between the template and the input image, thus the process of inliers extraction compares the directions of local gradients θ (also obtained with the SIFT algorithm). This is done in Step 3.1, using the n keypoints $K_j^i (j = 1,..,n)$ from the input image and its corresponding K_i in the template. The pair K_i and K_j^i are considered candidates to inliers if the $K_i\theta$ and $K_j^i\theta$, the orientation of K_i and K_j^i, respectively, satisfies $\left|K_i\theta - K_i^j\right| \le 2\pi/36$. Otherwise, they are immediately classified as outliers and removed from the set of inliers candidates. **Fig. 3** shows the keypoints that remain after this step. At this time, the template keypoints and the input image keypoints satisfy the 128 features and the θ from SIFT matching. However, there are still some keypoints scattered across the input image, which should be considered outliers. This is illustrated in the **Fig. 3** by the straight lines connecting the keypoints from the template to the input image.

To remove these keypoints, in Step 3.2, the region with the highest concentration of keypoints in the input image is searched, taking into account also the spatial coordinates of the template keypoints. All this process is presented in Algorithm 1.

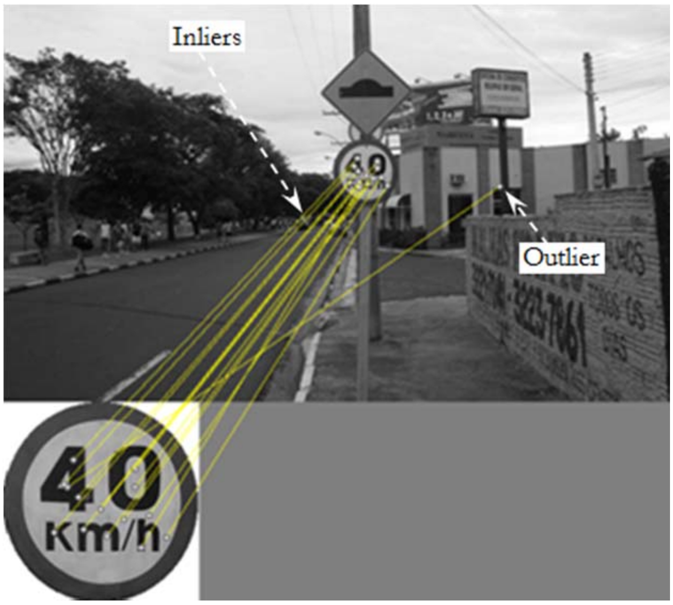


Fig. 3 Correspondence between the keypoints after evaluation of the orientation θ, with some inliers and one outlier.

Algorithm 1 – TSRs

A_1: *Sort the x and y coordinates of inliers keypoints candidates separately.*
A_2: *Find the closest points in the horizontal (x) and the vertical (y).*
A_3: *Calculate an approximate area of the plate (template) in the image sequence from the region found in the previous phase.*
A_4: *Delete the keypoints candidates to inliers that are found outside the region.*

After Step 3.2 (**Fig. 1**), there only remain the keypoints that are within the approximate plate region, as shown in **Fig. 4**.

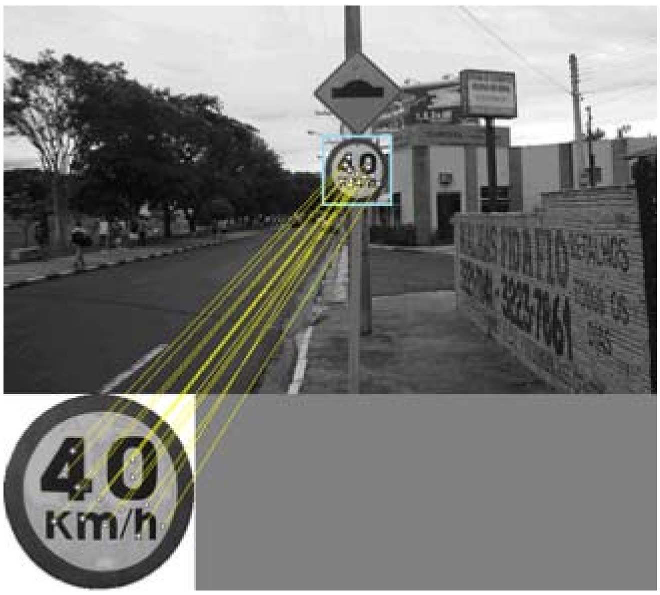


Fig. 4 The final set of keypoint classified as inliers.

It is important to observe that, for a reasonable match between one of the templates and an input image, it is necessary that a minimum amount of inliers is found. In this work, we empirically defined a minimum amount of inliers such as three inliers.

In the case of plates which indicate speed limit, in Step 4 (**Fig. 1**), an algorithm for number recognition on the plate is applied, which is done using an algorithm based on transitions between pixels, proposed by Silva *et al.* (2011). Otherwise, for the other traffic sign, the process ends in Step 3.2 (**Fig. 1**).

The TSRs algorithm (Fig. 1) is performed with each of the templates in the database. Thus, if there is more than one template containing a traffic sign in the input image, all of them are processed.

3.1 Character Recognition

The character recognition (letter or number) is an important area of pattern recognition with several applications in automation and information treatment (Theodoridis & Koutroumbas, 2009). The solution adopted in Step 4 of the TSRs algorithm (developed in Silva *et al.* (2011)) was selected because its recognition time is adequate for real time tasks.

3.1.1 Algorithm for Character Recognition

The algorithm for character recognition (Silva *et al.*, 2011) consists of a simple and rapid strategy to model the behavior of characters. This algorithm samples the characters images in a grid and it annotates the transitions between the pixels values (0 and 1 – binary images) of adjacent pixels. In this way, a character image sampled in a grid with dimensions *m* x *n* generates a transition vector with $m.n - 1$ values. Silva *et al.* (2011) suggest that the grid has at least the size 20 x 16. For a high precision, in this work we used a grid with size 80 x 64. The transitions are defined on the image character using the path illustrated in **Fig. 5**.

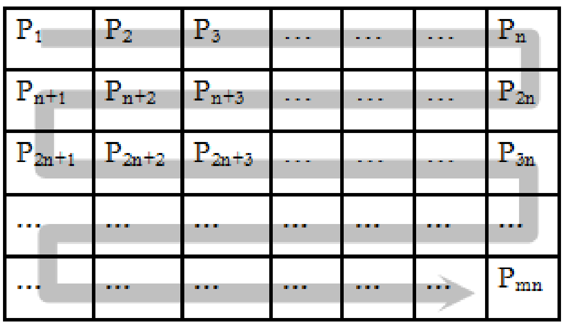


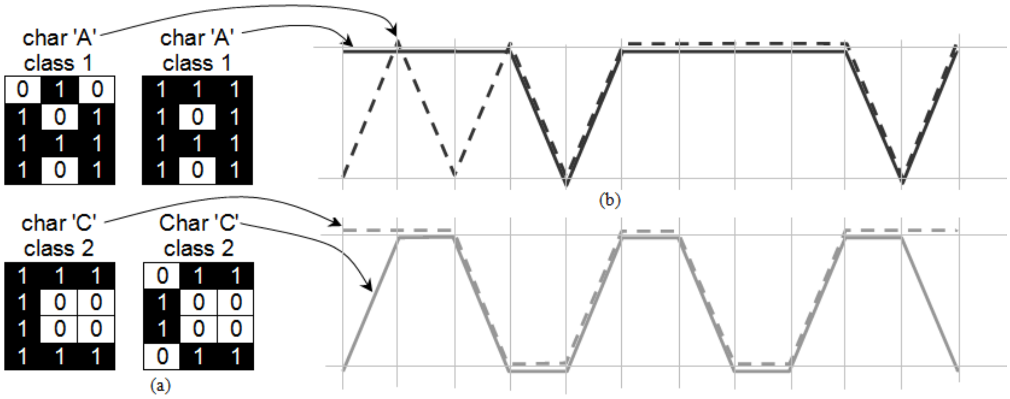
Fig. 5 Pixel sequence in the image used to model the behavior of the character.

After comparing the behavior of these transitions to a set of characters from a given class (supervised training), a set of rules is defined to be used later in the classification of the characters whose classes are not known. It is suggested to use a list structure to store the transitions that do not occur between each adjacent pair of pixels in the image of each character in their respective classes. As the images used are binary, the possible transitions between two adjacent pixels are: 00, 01, 10 and 11. **Fig. 6 (a)** shows the images of two 'A' characters (class 1) and two 'C' characters (class 2). In (b) a graph, using Parallel Coordinates (Inselberg, 1985) illustrates the transitions between adjacent pixels of the images in (a). In (c) a list with the transitions that do not occur between adjacent pixels in these images, where it is possible to identify that between the attributes a_3 and a_4, the transitions 01 and 11 occur, and the transitions 00 and 10 do not occur. While class 2 just possesses the transition 10 between the same attributes.

The list structure in **Fig. 6 (c)** has 30 restrictions for class 1 and 31 restrictions for class 2. After obtaining all the transitions that do not occur between adjacent attributes in each class, the characters can be classified in the class that presents the lowest number of inconsistencies in relation to characteristic transitions annotated in each class. The three rules for the character classification are:

R1 Recognize the character in the class whose transitions are not violated by the transitions in the class;
R2 Having more than one class that satisfies this condition, the class that has the most restrictions (most restrictive class) should be chosen;
R3 When transitions do not satisfy all the restrictions of any class, the character may be inserted in the least violated class, or to be classified as noise in extreme cases.

The last rule (R_3) requires a user-defined threshold to determine when the character should be classified as noise. For example, when it does not meet at least 30% of the restrictions of any class rules. Two strategies are also presented in Silva *et al.* (2011) to increase the quality of the classifications: 1) Accomplishing a denser sampling of the characters, the number of transitions between adjacent pixels will increase; 2) Using the transitions among three pixels instead of transitions between two adjacent pixels. Thus, the eight transitions to be checked are: 000, 001, 010, 011, 100, 101, 110 and 111.



(c)

	Transitions that do not occur between adjacent pixels										
	a_1-a_2	a_2-a_3	a_3-a_4	a_4-a_5	a_5-a_6	a_6-a_7	a_7-a_8	a_8-a_9	a_9-a_{10}	a_{10}-a_{11}	a_{11}-a_{12}
class 1	00 10	00 01	00 10	00 01 11	00 10 11	00 01 10	00 01 10	00 01 10	00 01 10	00 01 11	00 10 11
class 2	00 10	00 01 10	00 01 11	01 10 11	00 10 11	00 01 10	00 01 11	01 10 11	00 10 11	00 01 10	00 01

Fig. 6 a) Images representing two 'A' characters in the class 1 and two 'C' character in the class 2; b) Exhibition of the records of the class 1 in black and of the class 2 in gray in parallel coordinates; c) Transitions do not occur among adjacent attributes.

4 EXPERIMENTS

This section presents the results of an experiment using 60 images with resolution of 2,144 x 1,424 pixels, obtained with a Nikon D300S camera, mounted on a moving vehicle under normal conditions of traffic. The low-cost mobile mapping system used to capture the images was developed in LaMMov – *Laboratório de Mapeamento Móvel* of the São Paulo State University (Unesp), Presidente Prudente, São Paulo, Brazil (Silva *et al.*, 2003) (Gallis *et al.*, 2002).

In order to train the character recognition algorithm a set of images (**Fig. 7**) containing the used characters (0,...,9) was used. The characters are very similar to those adopted in the speed limit traffic sign in Brazil. To apply the recognition algorithm based on the transitions between adjacent pixels, each character was sampled into a grid with 80 x 64 pixels so that each character was represented by 5,119 transitions.

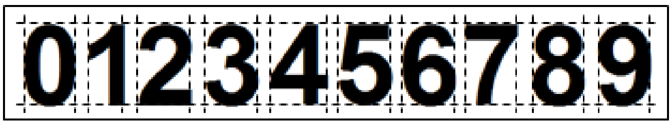


Fig. 7 Set of characters used in classifier training.

The 60 images used present complex scenes containing sign plates and also various other objects, such as: trees, people, vehicles, building façades, advertising boards, etc. In this experiment, our proposed methodology was applied, as well as the RANSAC (Fischler & Bolles, 1981) algorithm for the extraction of inliers in the detection and recognition stage of traffic signs.

Fig. 8 shows the Brazilian traffic signs (templates) used in this work (obtained from different images of those used in the experiment): (a) Speed limit 20 Km/h; (b) Speed limit 40 Km/h; (c) Regulated parking; (d) Go straight or right; (e) No parking; (f) No stopping and parking; (g) No return; (h) No right return; (i) No left return; (j) Signalled pedestrian crossing strip; (k) Roundabout; (l) Stop light ahead; (m) Maximum height allowed 5,4 m; (n) Stop; (o) Signalled Student crossing; (p) Pedestrian crossing plate; (q) Emergency parking. In this figure, it is possible to observe the different sizes of traffic sign images used as templates.

Fig. 8 Traffic signs plates used as templates.

Table 1 shows the results obtained by applying the TSRs and the RANSAC algorithms for detection and recognition of the traffic signs.

Table 1. Results in the traffic sign recognition.

	TSRs algorithm		RANSAC algorithm	
Templates	Recognized	Undetected	Recognized	Undetected
(a) Speed limit 20 Km/h	4	0	1	3
(b) Speed limit 40 Km/h	3	0	3	0
(c) Regulated parking	14	4	6	12
(d) Go straight or right	0	1	0	1
(e) No parking	4	0	0	4
(f) No stopping and parking	5	0	0	5
(g) No return	7	3	0	10
(h) No right return	8	0	1	7
(i) No left return	4	3	1	6
(j) Signalled pedestrian crossing strip	1	0	1	0
(k) Roundabout	2	0	1	1
(l) Stop light ahead	7	0	0	7
(m) Maximum height allowed 5,4 m	1	0	1	0
(n) Stop	3	1	3	1
(o) Signalled student crossing	2	0	2	0
(p) Pedestrian crossing plate	2	0	2	0
(q) Emergency parking	6	6	6	6
Total	73	18	28	63

The first column refers to existing indexes in **Fig. 8**. In the 60 images used in this experiment there were 91 traffic signs. RANSAC algorithm extracted inliers correctly recognized only in 28 plates, representing 30.77% of the hit rate, and in 63 plates they were not recognized (69.23%).

This proposal recognized and eliminated outliers correctly in 73 plates, yielding a hit rate of 80.22%. From the 60 plates, 18 plates were not recognized (19.78%).

Fig. 9 shows some examples of templates with their respective cuttings from the input image, in which the recognition was successful, in spite of some complicating factors, such as branches, twigs and leaves; rusted or damaged plates; slanted plate, etc.

Fig. 9 Recognition examples in spite of some complicating factors.

Fig. 10 shows one of the input images (2,144 x 1,424 pixels) in which the two traffic signs existing in the scene have been detected and recognized, in spite the high complexity of the urban scene, which includes the traffic of vehicles, people, trees, buildings and other objects.

Fig. 10 Traffic sign recognition in real urban scene.

The experiment was performed on a machine with an Intel Core i7 2600K processor with 3.40 GHz (6 Gigabytes of RAM), where the average processing time for each pair of images was 52 seconds.

The next figures show the result of the application of our proposal in other bases of images known by the community. **Fig. 11** shows the result of the recognition of traffic signs using the public dataset from Belgium Traffic Signs, available at http://btsd.ethz.ch/shareddata/ (Timofte et al., 2011). The **Fig. 12** shows the result of the recognition of traffic signs using the public dataset from Swedish Traffic Signs, available at http://www.cvl.isy.liu.se/research/datasets/traffic-signs-dataset/ (Larsson & Felsberg, 2011).

Fig. 11 Traffic signs recognition from Belgium Traffic Signs Dataset.

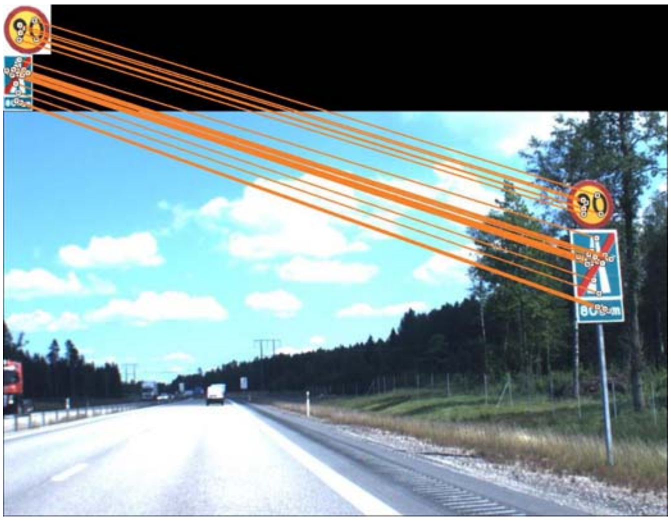


Fig. 12 Traffic signs recognition from Swedish Traffic Signs Dataset.

5 CONCLUSIONS AND FUTURE WORK

This paper presented a new approach to accomplish traffic sign recognition, using images with a great amount of details (pedestrians, trees, cars, advertising boards, shadows, etc.), obtained by a vehicle in movement under normal traffic conditions.

The use of the SIFT algorithm, which can provide points of interest in images with a very satisfactory description, is one of the differentials of this methodology, contributing significantly to the quality of the results. Furthermore, the use of the information orientation and positioning, also obtained by SIFT, as a complement in the matching process between the input image and the template, represents a strategy that besides using the full potential of SIFT it also eliminates the need for more complex filtering, proposed by other studies.

The TSRs algorithm was applied in an automatic georeferencing system for the registration of traffic signs, and it showed satisfactory results even in images with various complicating factors, common in real urban scenes, such as: many vehicles; pedestrians; trees; obstacles obstructing the plate; branches; twigs and leaves; rusted or damaged plates; slanted plate, etc. And considering additionally that the input images were captured by a moving vehicle, this means an extra complicating factor that our method dealt with very well. Future studies should investigate the optimization of this methodology, aiming to reduce the processing time.

REFERENCES

Arlicot, A.; Soheilian, B. & Paparoditis, N. (2009) Circular Road Sign Extraction from Street Level Images Using Colour, Shape and Texture Database Maps. In: *Stilla U, Rottensteiner F, Paparoditis* N (Eds) CMRT09. IAPRS, (38), France.

Bar, X.; Escalera, S.; Vritri, J.; Pujol, O. & Radeva, P. (2009) Traffic Sign Recognition using Evolutionary Adaboost Detection and Forest-ECOC classification. *IEEE Transactions on Intelligent Transportation Systems*, 2009, 113–126.

Basri, R. & Jacobs, D.W. (1997) Recognition using region correspondences. *International Journal of Computer Vision*, 25(2), 145–166.

Barnes, N. & Zelinsky, A. (2004) Real-Time Radial Symmetry for Speed Sign Detection, Proc. *Intelligent Vehicles Symposium*, 566–571.

Beis, J. & Lowe, D.G. (1997) Shape Indexing using Approximate Nearest-Neighbour Search in High Dimensional Spaces, Conference on Computer Vision and Pattern Recognition, Puerto Rico, 1000–1006.

Belaroussi, R. & Tarel, J.P. (2009) A real-time road sign detection using Bilateral Chinese Transform. *Proc. International Symposium on Visual Computing*, ISVC, 1161–1170.

Brown, M. & Lowe, D.G. (2007) Automatic Panoramic Image Stitching using Invariant Features. *International Journal of Computer Vision*, 74(1):5973.

Cai, N.; Liang, W.; Xu, S. & Li, F. (2010) Traffic Sign Recognition Based on SIFT Features. *Advanced Materials Research. Nanotechnology and Computer Engineering*, 596–599.

Çağlayan, T.; Ahmadzay, H. & Kofraz, G. (2015) Real-time traffic sign recognition based on shape and color classification. In: *23th Signal Processing and Communications Applications Conference* (SIU), Malatya, Turkey.

Ciresan, D.C.; Meier, U.; Masci, J. & Schmidhuber, J. (2011) A committee of neural networks for traffic sign classification, In: *International Joint Conference on Neural Networks*, San Jose, California, USA, 1918–1921.

Creusen, I.M.; Wijnhoven, R.G.J.; Herbschleb, E. & With, P.H.N. (2010) Color Exploitation in Hog-Based Traffic Sign Detection. *Prof. of the IEEE 17th International Conference on Image Processing*, Hong Kong, 2669–2672.

Farag, A.A. & Abdel-Hakim, A.E. (2004) Detection, Categorization and Recognition of Road Signs for Autonomous Navigation. *Proc. of Advanced Concepts for Intelligent Vision Systems*, Belgium, 125–130.

Fischler, M.A. & Bolles, R.C. (1981) Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, (24), 381–395.

Fleyeh, H. (2005) Road and Traffic Sign Color Detection and Segmentation – A Fuzzy Approach. In: *Conference on Machine Vision Applications*, MVA2005 IAPR, Tsukuba Science City, Japan, May, 124–127.

Friedman, J.H.; Bentley, J.L. & Finkel, R.A. (1977) An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3), 209–226.

Gallis, R.B.A.; Silva, J.F.C.; Camargo, P.O. & Barbosa, R.L. (2002) Mapeamento Móvel no Brasil: Resultados Obtidos com a Utilização da Unidade Móvel de Mapeamento Digital. *Série em Ciências Geodésicas / Pesquisas em Ciências Geodésicas*, Curitiba, 1(2), 248–266.

Handmann, U.; Kalinke, T.; Tzomakas, C.; Werner, M. & Seelen, W.V. (1998) An Image Processing System for Driver Assistance. *Proc. of the IEE International Conference on Intelligent Vehicles*, Stuttgart, Germany, 481–486.

Hartley, R. & Zisserman, A. (2003) Multiple View Geometry in Computer Vision. University Press, Cambridge, second edition.

Hazelho, L.; Creusen, I.M. & With, P.H.N. (2012) Robust detection, classification and positioning of traffic signs from street-level panoramic images for inventory purposes. *Proc. of the IEEE Workshop on Applications of Computer Vision* (WACV), 313–320.

Höferlin, B. & Heidemann, G. (2010) Selection of an Optimal Set of Discriminative and Robust Local Features with Application to Traffic Sign Recognition. *Proc. WSCG, 18th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, (18), 9–16.

Höferlinm B. & Zimmermann, K. (2009) Towards Reliable Traffic Sign Recognition. In: *IEEE Intelligent Vehicles Symposium*, 324–329.

Inselberg, A. (1985) The Plane with Parallel Coordinates. *Visual Computer*, 2(1), 69–91.

Jung, S.; Lee, U.; Jung, J. & Shim, D.H. (2016) Real-time Traffic Sign Recognition system with deep convolutional neural network. In: *13th International Conference on Ubiquitous Robots and Ambient Intelligence* (URAI), Xian, China.

Koenderink, J.J. (1984) The structure of images. *Biological Cybernetics*, 50, 363–396.

Kus, M.C.; Gokmen, M. & Etaner-Uyar, S. (2008) Traffic sign recognition using Scale Invariant Feature Transform and color classification. In: *23rd International Symposium on Computer and Information Sciences*, 1–6.

Larsson, F. & Felsberg, M. (2011) Using Fourier Descriptors and Spatial Models for Traffic Sign Recognition. *Image Analysis - Lecture Notes in Computer Science*, 238–249.

Lindeberg, T. (1994) Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales. *Journal of Applied Statistics*, 224–270.

Liu, J.; Chen, Z. & Guo, R. (2008) A Mosaic Method for Aerial Image Sequence by R/C Model. In: *International Conference on Computer Science and Software Engineering*, 58–61.

Lowe, D.G. (1999) Object Recognition from Local Scale-Invariant Features. In: *International Conference on Computer Vision*, Greece, 1150–1157.

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

Loy, G. & Zelinsky, A. (2003) Fast Radial Symmetry for Detecting Points of Interest. *Transaction on Pattern Analysis and Machine Intelligence*, IEEE, 25(8), 959–973.

Martinovi, A.; Glava, G.; Juribai, M.; Suti, D. & Kalafati, Z. (2010) Real-Time Detection and Recognition of Traffic Signs". *Proc. 33rd International Convention MIPRO 2010*, 247–252.

Mathias, M.; Timofte, R.; Benenson, R. & Gool, L.V. (2013) Traffic sign recognition – how far are we from the solution? *Proc. of IEEE International Joint Conference on Neural Networks*.

Mikolajczyk, K. (2002) Detection of local features invariant to affine transformations, Ph.D. thesis, Institute National Polytechnique de Grenoble, France.

Mikolajczyk, K. & Schmid, C. (2005) A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10) 1615–1630.

Okabe, T. & Sato, Y. (2003) Object recognition based on photometric alignment using RANSAC. *Proc. of Computer Society Conference on Computer Vision and Pattern Recognition*, 221–228.

Paulo, C.F. & Correia, P.L. (2008) Traffic Sign Recognition Based on Pictogram Contours. In: *9th Workshop on Image Analysis for Multimedia Interactive Services* (WIAMIS2008), Klagenfurt, Austria, 67–70.

Reiterer, A.; Hassan, T. & El-Sheimy, N. (2009) Robust Extraction of Traffic Signs form Georeferenced Mobile Mapping Images, In: *6th International Symposium on Mobile Mapping Technology*, Presidente Prudente, Brasil, 21–24.

Silva, J.F.C.; Camargo, P.O. & Gallis, R. (2003) Development of a low-cost mobile mapping system: a South American experience. *The Photogrammetric Record*, The London Society, 18(101), 5–26.

Silva, F.A.; Artero, A.O.; Paiva, M.S.V. & Barbosa, R.L. (2011) A Fast Algorithm for Character Recognition, In: *VII Workshop on Computer Vision*, Curitiba, Brazil, 149–154.

Takimoto, R.Y.; Sato, A.K. & Tsuzuki, M.S.G. (2010) Retrieval of epipolar geometry using two images, *XVIII Brazilian Congress Automatic*, Brazil, 12–16.

Theodoridis, S. & Koutroumbas, K. (2009) *Pattern Recognition*. Elsevier Inc Fourth Edition.

Timofte, R.; Zimmermann, K. & Van Gool, L. (2011) Multi-view Traffic Sign Detection, Recognition, and 3D Localisation, *Machine Vision and Applications*, December, 1–15.