

Uma Metodologia para Detectar e Reconhecer Placas de Sinalização de Trânsito

Francisco Assis da Silva^{1,2}, Almir Olivette Artero³,
Maria Stela Veludo de Paiva², Ricardo Luís Barbosa⁴

¹*Faculdade de Informática de Presidente Prudente, Universidade do Oeste Paulista (Unoeste),
Presidente Prudente/SP*

²*Escola de Engenharia de São Carlos – Universidade de São Paulo (USP), São Carlos/SP*

³*Faculdade de ciências e Tecnologia – Universidade Estadual Paulista (Unesp), Presidente
Prudente/SP*

⁴*Campus de Sorocaba – Universidade Estadual Paulista (Unesp), Sorocaba/SP
chico@unoeste.br, almir@fct.unesp.br, mstela@sc.usp.br, ricardo@cartovias.com.br*

Abstract

This paper proposes a new methodology for the traffic sign recognition using images captured by a mobile mapping system. The proposed approach applies the SIFT algorithm to extract keypoints features, used to evaluate the correspondence between a road image containing one or more plates, and the images of some traffic signals (templates). BBF algorithm was used to evaluate the correspondence among the SIFT features. Finally, to filter only the pairs of keypoints (image-template) that are compatible as well as the orientation and positioning, we propose a new algorithm that uses this information.

1. Introdução

Os sinais de trânsito são muito importantes porque estabelecem as regras de navegação pelas vias. Suas propriedades visuais são muito fortes, pois foram projetados para serem objetos facilmente identificáveis pelos seres humanos [1]. O reconhecimento dos sinais de trânsito permite que um motorista possa ser avisado sobre ações impróprias e situações potencialmente perigosas [2]. O reconhecimento de sinais de trânsito por máquinas vem sendo estudado para diversos propósitos, tais como direção autônoma e assistida [11].

Em geral, a identificação de sinais de trânsito é feita em duas fases, que são a detecção e o reconhecimento. Primeiramente, é aplicado um algoritmo para a detecção do sinal de trânsito, sendo localizada a região do sinal. Posteriormente, o algoritmo de reconhecimento utiliza esta região para identificar o tipo do sinal de trânsito.

Na literatura, podem ser encontradas diversas abordagens na tentativa de resolver o problema do reconhecimento de sinais de trânsito, como por exemplo: utilizando propriedades de cor dos sinais [8]; a partir da geometria das placas em imagens em tons de cinza [18]; baseada nas propriedades de forma dos sinais [6]; baseada

em propriedades de cor e forma [1]; utilizando classificadores baseados em redes neurais [19]; analisando a curvatura da representação do sinal com contornos ativos (Snakes) [22]; usando *Support Vector Machine* (SVM) [5]; segmentação de sinais de trânsito baseado em conjuntos fuzzy [10]; utilizando um algoritmo de aprendizagem de máquina *Adaboost* [2].

Recentemente, descritores locais têm sido usados no contexto do TSR (*Traffic Sign Recognition*), sendo aplicados tanto para a detecção de sinais de trânsito, quanto para o seu reconhecimento, devido a seus pontos fortes e discriminativos [12].

Técnicas de descritores locais detectam as características de uma determinada imagem e, então, um conjunto de descritores é computado para estas características [20]. Os descritores representam as partes características de uma imagem, existindo na literatura muitos algoritmos para a extração de descritores locais. Um algoritmo que tem se destacado nesta etapa é o algoritmo SIFT (*Scale Invariant Feature Transform*) [17], por ser invariante a transformações de escala, rotação e pontos de vista. Höferlin e Heidemann [12] apresentam um estudo avaliando os resultados obtidos com os descritores locais obtidos com os seguintes algoritmos: SIFT [17], GLOH (*Gradient Location and Orientation Histogram*) e algumas variantes de SURF (*Speeded Up Robust Features*) e, concluem que o SIFT é o mais adequado para tratar o problema do TSR, baseado na avaliação realizada por eles e também na avaliação de descritores de características [21], que também apontam o SIFT como sendo o mais forte descritor local em termos de sensibilidade e precisão em relação às abordagens testadas.

Alguns trabalhos podem ser encontrados na literatura utilizando o algoritmo SIFT para aplicações de TSR, tais como: o trabalho de Farag e Abdel-Hakim [9] que utiliza um classificador de Bayes a partir do conjunto de

descritores obtidos com o SIFT; no trabalho de Kus et al. [14] são adicionados aos descritores SIFT, informações de cor; Höferlin e Zimmermann [13] realiza a detecção usando o SIFT e para a fase de reconhecimento usa uma rede neural Perceptron de múltiplas camadas; o trabalho de Reiterer et al. [23] é baseado no algoritmo SIFT e no algoritmo de agrupamento *k-means*; o trabalho proposto por Cai et al. [7] apresenta um método de reconhecimento para sinais de trânsito triangulares baseado nos descritores SIFT, onde é adotada uma árvore de decisão para classificar os sinais.

Objetivando tratar imagens com uma grande quantidade de detalhes (cenários complexos) e também contendo diversos fatores que tem dificultado o reconhecimento das placas, como: condições climáticas, variações de iluminação, sombras, placas danificadas, presença de árvores, entre outros. Neste trabalho é apresentada uma nova abordagem para o processo de TSR usando o algoritmo SIFT [17]. As demais seções deste trabalho estão organizadas da seguinte maneira: A Seção 2 descreve o algoritmo SIFT, usado para identificar e descrever pontos chave nas imagens; A Seção 3 apresenta uma nova abordagem que usa os descritores obtidos com o SIFT para identificar a correspondência entre os *templates* e as imagens de entrada; A Seção 4 apresenta alguns experimentos com a implementação da abordagem proposta; Finalmente, a Seção 5 apresenta algumas conclusões e sugestões para trabalhos futuros.

2. Algoritmo SIFT

A identificação de pontos homólogos em duas imagens não é uma tarefa simples, existindo muita pesquisa na área, para a sua execução automática [3]. A primeira dificuldade está em encontrar pontos de interesse (*keypoints*) em uma das imagens e, em seguida, localizá-los na outra. Assim como em Höferlin e Zimmermann [13] Cai et al. [7], neste trabalho, o algoritmo SIFT é usado para encontrar pontos de interesse na imagem de entrada e nos *templates* (contendo imagens das placas de sinalização). O algoritmo SIFT tem se mostrado muito eficiente para identificar e descrever pontos chave em uma imagem, o que é feito através de um mapeamento com diferentes vistas de um objeto ou cena, que resulta para cada ponto, um conjunto de informações que incluem: um vetor com 128 valores que descrevem cada ponto chave da imagem; a posição x,y ; a escala s ; a orientação θ [17]. O algoritmo consiste nas seguintes etapas:

Detecção do Espaço-Escala: Os pontos chave são detectados aplicando uma filtragem em cascata que identifica candidatos, que são invariantes à escala, usando uma função que procura por descritores estáveis ao longo das diferentes escalas. O espaço-escala é definido como

uma função $L(x,y,\sigma)$ na Equação 1, com uma imagem $I(x,y)$ [15].

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

onde $*$ indica a convolução em x e y com a Gaussiana $G(x,y,\sigma)$ (Equação 2).

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

Para detectar a localização dos pontos chave estáveis no espaço-escala, Lowe [16] propôs o uso da função de diferença de Gaussianas (DoG) no espaço-escala convoluída com a imagem $I(x,y)$, resultando em $D(x,y,\sigma)$, a qual pode ser calculada a partir de duas escalas próximas separadas por um fator multiplicativo constante k , como na Equação 3.

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

O DoG é uma aproximação da escala normalizada do Laplaciano da Gaussiana $\sigma^2 \nabla^2 G$ [15]. O máximo e o mínimo de $\sigma^2 \nabla^2 G$ produz os descritores de imagem mais estáveis, quando comparado a diversas funções, tais como gradiente, Hessian ou Harris [20].

Detecção de Extremos Locais: A partir de $D(x,y,\sigma)$, Lowe [17] sugere que os máximos e mínimos locais devem ser detectados pela comparação de cada pixel com os seus 8 vizinhos na imagem corrente e 9 vizinhos nas escalas superior e inferior.

O candidato é preservado se ele for maior (para o máximo) ou menor (para o mínimo) que todos os seus 26 vizinhos. O próximo passo executa um ajuste detalhado nos dados da imagem local, determinando a posição e a escala. Pontos chave candidatos que possuem baixo contraste são rejeitados, por serem sensíveis ao ruído ou localizados ao longo das bordas. Para corrigir este problema, o algoritmo elimina os pontos chave que possuem uma localização fracamente determinada, mas possuem uma alta resposta na borda [17].

Atribuição de Orientação: A escala do ponto chave é usada para selecionar a imagem suavizada pela Gaussiana L , com a escala mais próxima, de modo que toda a computação seja realizada de modo invariante à escala. O gradiente de magnitude $m(x,y)$ é calculado com a Equação 4.

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

onde: $\Delta x = L(x+1,y) - L(x-1,y)$ e $\Delta y = L(x,y+1) - L(x,y-1)$. A orientação $\theta(x,y)$ é calculada pela Equação 5.

$$\theta(x,y) = \arctan(\Delta y / \Delta x) \quad (5)$$

Em seguida, é construído um histograma de orientação (36 *bins*) a partir das orientações dos gradientes dos pontos amostrados, com uma região ao redor do ponto chave e, os picos neste histograma correspondem à direção dominante do gradiente local. Orientações com frequência acima de 80% do valor do pico no histograma são usadas para criar pontos chave com essa orientação. Desta forma, para múltiplas orientações de magnitude similar, existem múltiplos pontos chave criados no mesmo local e escala, mas com diferentes orientações.

Descrição dos pontos chave: Após a localização, escala e orientação da imagem ter sido atribuída a cada ponto chave, é possível adotar um sistema de coordenadas bidimensional para descrever a região local da imagem e prover invariância com respeito a esses parâmetros. O próximo passo é calcular um descritor para a região da imagem local que é distinta e invariante a variações adicionais, tais como mudanças na iluminação ou ponto de vista 3D. Para alcançar a invariância de orientação, as coordenadas do descritor e as orientações do gradiente são rotacionadas em relação à orientação do ponto chave. Este procedimento é mostrado na Figura 1, onde o descritor de um ponto chave é criado pela determinação da magnitude e orientação do gradiente em cada pixel da imagem amostrada em uma região ao redor do ponto chave, como mostrado em (a). Os descritores são ponderados em uma janela Gaussiana indicada pelo círculo. Estas amostras são acumuladas em histogramas de orientação, como exibido em (b).

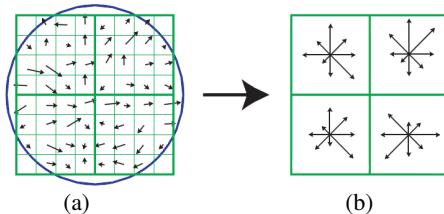


Figura 1. Descrição dos Pontos Chave [17].

2.2 Correspondência entre duas imagens

Para se encontrar a correspondência (*matching*) entre duas imagens, é possível usar os pontos chave detectados com o algoritmo SIFT. Cada ponto chave está associado a um vetor de descritores com 128 dimensões, a comparação desses descritores torna possível encontrar a correspondência de uma imagem com outra. Lowe [17] provou que a melhor correspondência para cada ponto chave é encontrada pela identificação de seus vizinhos mais próximos, que é definida minimizando a distância Euclidiana para os vetores de descritores.

Para evitar uma busca exaustiva, Lowe [17] sugere o uso de uma estrutura de dados *k-d tree* [4], que suporta uma busca binária balanceada para encontrar o vizinho mais próximo dos descritores. O algoritmo heurístico

Best-Bin-First (BBF) é usado nesta busca, pois retorna o vizinho mais próximo, com alta probabilidade.

3. Metodologia proposta

A metodologia proposta neste trabalho pode ser entendida pela Figura 2, que aplica o algoritmo SIFT na imagem de entrada e, em seguida, em cada *template*, até alcançar a melhor correspondência, considerando os 128 descritores e também as informações de orientação e posicionamento.

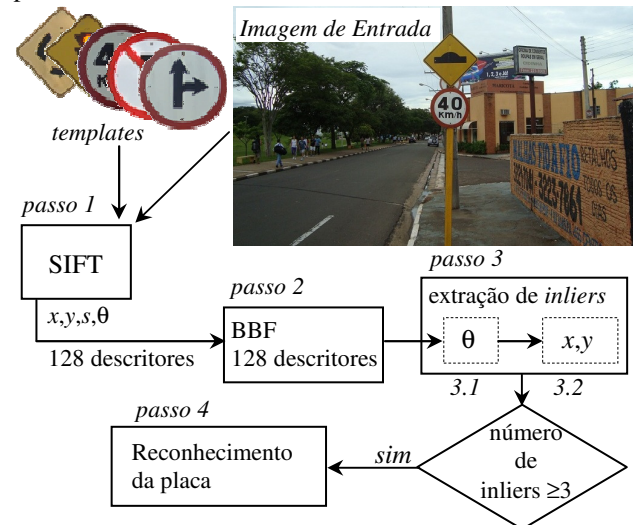


Figura 2. Metodologia proposta para o reconhecimento de placas de sinalização de trânsito.

Na metodologia proposta, o algoritmo SIFT é aplicado a todos os *templates* contendo os sinais de trânsito usados e, então, um banco de dados contendo os descritores de pontos chave desses *templates* é construído, de modo a evitar que o SIFT tenha que ser aplicado a todos os *templates* para cada imagem de entrada, pois o seu tempo de processamento é relativamente alto.

No passo 1 da metodologia proposta, o algoritmo SIFT é aplicado apenas na imagem de entrada, gerando as informações que descrevem cada ponto chave. No passo 2, os 128 descritores obtidos em cada ponto da imagem de entrada são comparados com os 128 descritores obtidos em cada ponto dos *templates*. Na Figura 3, as linhas retas ligando os pontos no *template* e os pontos na imagem de entrada indicam que seus descritores são correspondentes. Neste momento, a maioria dos pontos chave da imagem que não combinam com algum ponto chave dos *templates* foram eliminados. Entretanto, ainda persistem alguns pontos que, visualmente, não correspondem com o *template*.

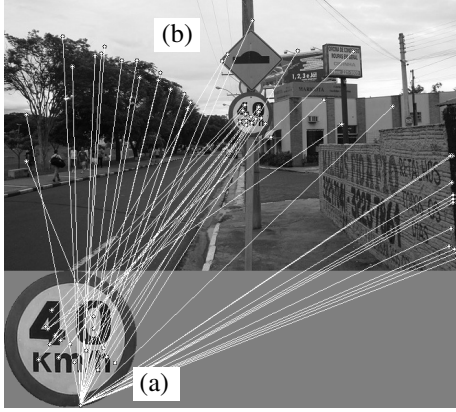


Figura 3. Correspondência entre os pontos chave após avaliação dos 128 descritores do SIFT.

Assim, o passo 3, deve identificar apenas os *inliers*¹, o que é feito em dois estágios: 1) comparando os ângulos de orientação dos gradientes locais; 2) buscando uma região onde se concentra a maior parte dos *inliers* e calculando a correlação entre as coordenadas espaciais dos pontos chave na imagem de entrada e no *template*.

Neste trabalho, não se espera encontrar rotações acima de 10 graus entre o *template* e a imagem de entrada, assim, propõe-se que o processo de extração de *inliers* compare as direções dos gradientes locais θ (também obtida com o algoritmo SIFT – Equação 5 [17]). Isto é feito no passo 3.1 (Figura 2) que toma n pontos chave K_i^j ($j=1,...,n$) na imagem de entrada e seus correspondentes K_i no *template*, e os considera candidatos a *inliers* se suas orientações $K_i\theta$ e $K_i^j\theta$, respectivamente, satisfazem $|K_i\theta - K_i^j\theta| \leq 2\pi/36$. Caso contrário, eles são classificados imediatamente como *outliers*² e são removidos do conjunto de candidatos a *inliers*. A Figura 4 mostra os pontos chave que ainda permanecem após este passo.

Neste momento, boa parte dos pontos chave do *template* estão ligados pelas linhas retas aos pontos chave da imagem de entrada na região do sinal de trânsito correspondente ao *template*, porém, ainda persistem alguns pontos chave dispersos pela imagem de entrada, que devem ser considerados *outliers*.



Figura 4. Correspondência entre os pontos chave após avaliação da orientação θ .

Para remover estes pontos, no passo 3.2, é rastreada a região com a mais alta concentração de pontos chave na imagem de entrada, levando em conta também as coordenadas espaciais dos pontos chave do *template*. Todo este processo é demonstrado no Algoritmo 1.

Algoritmo 1

- 1: Ordene as coordenadas x e y dos pontos chave candidatos a *inliers* separadamente.
 - 2: Encontre os pontos mais próximos na horizontal (x) e na vertical (y) buscando a menor soma de um conjunto de coordenadas.
 - 3: Calcular uma região aproximada da placa (*template*) na imagem da sequência, a partir da região encontrada na fase anterior.
 - 4: Eliminar os pontos chave candidatos a *inliers* que estiverem fora da região encontrada.
-

Após o passo 3.2, apenas os pontos chave que estão dentro da região aproximada da placa permanecem, como mostrado na Figura 5.

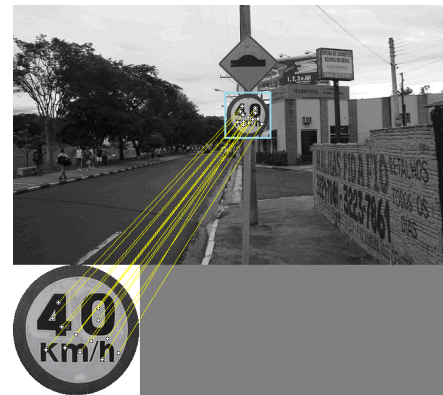


Figura 5. Conjunto de pontos chave classificados como *inliers*.

¹ *inliers*: pontos de dados que se ajustam com um modelo específico dentro de uma certa tolerância de erro.

² *outliers*: pontos de dados que não se ajustam ao modelo correspondente ao objeto desejado, estão fora de uma certa tolerância de erro.

É importante observar que, para ser caracterizada uma correspondência razoável entre um dos *templates* e uma imagem de entrada, é necessário e suficiente que seja encontrado uma quantidade mínima de *inliers*, que empiricamente foi definido como três *inliers*.

No caso de placas que sinalizam limites de velocidade, no passo 4, é aplicado um algoritmo para o reconhecimento dos números na placa, o que é feito usando o algoritmo baseado em transições entre pixels, proposto por Silva et al. [24]. Para as demais placas de sinalização, o processo termina no passo 3.2.

4. Experimento

Esta seção apresenta os resultados de um experimento com 60 imagens com resolução de 2144 x 1424 pixels, obtidas a partir de uma câmara Nikon D300S instalada sobre um veículo (Sistema de Mapeamento Móvel). As imagens foram obtidas com o veículo em movimento em condições normais de tráfego. A Figura 6 apresenta os *templates* usados neste experimento.



Figura 6. Placas de sinais de trânsito usadas.

A Tabela 1 mostra o resultado obtido aplicando a metodologia proposta, sendo a primeira coluna referente aos índices numéricos existentes na Figura 6.

Tabela 1. Resultados no reconhecimento das placas.

<i>Templates</i>	Acertos	Erros	Não reconhecidas
1	4 (100%)	0 (0%)	0 (0%)
2	3 (100%)	0 (0%)	0 (0%)
3	14 (73,68%)	0 (0%)	5 (26,32%)
4	0 (0%)	0 (0%)	1 (100%)
5	4 (100%)	0 (0%)	0 (0%)
6	5 (100%)	0 (0%)	0 (0%)
7	7 (77,78%)	0 (0%)	2 (22,22%)
8	8 (100%)	0 (0%)	0 (0%)
9	4 (66,67%)	0 (0%)	2 (33,33%)
10	1 (100%)	0 (0%)	0 (0%)
11	2 (100%)	0 (0%)	0 (0%)
12	6 (100%)	0 (0%)	0 (0%)
13	1 (100%)	0 (0%)	0 (0%)
14	3 (75%)	0 (0%)	1 (25%)
15	2 (100%)	0 (0%)	0 (0%)
16	2 (66,67%)	1 (33,33%)	0 (0%)
17	12 (85,71%)	2 (14,29%)	0 (0%)

Nas 60 imagens usadas nestes experimento existem 92 placas de sinalização. Conforme apresentado na tabela, 78

placas foram reconhecidas corretamente, obtendo-se uma taxa de acerto de 84.78%. Ocorreram apenas 3 erros de identificação (3.26% de erros) e 11 placas não foram reconhecidas (11.96%). O tempo médio de processamento destas 60 imagens em uma máquina com processador Intel Core i3 M330 2.13 GHz (RAM de 4 Giga Bytes) é de 90 segundos.

A Figura 7 apresenta alguns exemplos de *template* com os seus respectivos recortes na imagem de entrada, em que o reconhecimento funciona, apesar de alguns fatores complicadores.



Figura 7. Exemplos de reconhecimento apesar de fatores complicadores.

5. Conclusões e Trabalhos Futuros

Este trabalho apresenta uma nova metodologia para realizar o reconhecimento de placas de sinalização de trânsito, usando imagens com uma grande quantidade de detalhes (pedestres, árvores, veículos, placas de publicidade, sombras, etc), obtidas por um veículo em movimento em condições normais de tráfego.

O uso do algoritmo SIFT, que consegue fornecer os pontos de interesse nas imagens, com uma descrição muito satisfatória é um dos diferenciais desta metodologia, contribuindo decisivamente para a qualidade dos resultados.

Este algoritmo está sendo aplicado em um sistema de georreferenciamento automático, para o cadastro de placas de sinalização de trânsito. Em trabalhos futuros deverão ser investigadas otimizações nesta metodologia, objetivando a redução do tempo de processamento.

6. Referências

- [1] A. Arlicot, B. Soheilian, N. Paparoditis, "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, Vol. 38, France, 2009.
- [2] X. Baró, S. Escalera, J. Vritrià, O. Pujol and P. Radeva, "Traffic Sign Recognition using Evolutionary Adaboost Detection and Forest-ECOC classification", *IEEE Transactions on Intelligent Transportation Systems*, 2009, pp. 113-126.
- [3] R. Basri and D.W. Jacobs, "Recognition using region correspondences". *International Journal of Computer Vision*, vol. 25, n. 2, 1997, pp. 145-166.
- [4] J. Beis and D.G. Lowe, "Shape Indexing using Approximate Nearest-Neighbour Search in Highdimensional Spaces", *Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 1997, pp.1000-1006.
- [5] N. Barnes, A. Zelinsky, "Real-Time Radial Symmetry for Speed Sign Detection", *Proc. Intelligent Vehicles Symposium*, 2004, pp. 566-571.
- [6] R. Belaroussi, J.-P. Tarel, "A real-time road sign detection using Bilateral Chinese Transform", *Proc. International Symposium on Visual Computing*, ISVC, 2009, pp. 1161-1170.
- [7] N. Cai, W. Liang, S. Xu and F. Li, "Traffic Sign Recognition Based on SIFT Features", *Advanced Materials Research*. Nanotechnology and Computer Engineering, pp. 596-599, 2010.
- [8] I.M. Creusen, R.G.J. Wijnhoven, E. Herbschleb and P.H.N de With, "Color Exploitation in Hog-Based Traffic Sign Detection", *Proc. IEEE 17th International Conference on Image Processing*. Hong Kong, 2010, pp. 2669-2672.
- [9] A.A. Farag, A.E. Abdel-Hakim, "Detection, Categorization and Recognition of Road Signs for Autonomous Navigation". *Proc. Advanced Concepts for Intelligent Vision Systems*, Belgium, 2004, pp. 125-130.
- [10] H. Fleyeh, "Road and Traffic Sign Color Detection and Segmentation - A Fuzzy Approach", *MVA2005 IAPR Conference on Machine Vision Applications*, Tsukuba Science City, Japan, May 16-18, 2005, pp. 124-127.
- [11] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner and W. V. Seelen, "An Image Processing System for Driver Assistance", *Proc. IEEE International Conference on Intelligent Vehicles*, Stuttgart, Germany, 1998, pp. 481-486.
- [12] B. Höferlin and G. Heidemann, "Selection of an Optimal Set of Discriminative and Robust Local Features with Application to Traffic Sign Recognition", *Proc. WSCG, 18th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*. vol. 18, 2010, pp. 9-16.
- [13] B. Höferlin and K. Zimmermann, "Towards Reliable Traffic Sign Recognition", *IEEE Intelligent Vehicles Symposium*, 2009, pp 324-329.
- [14] M.C. Kus, M. Gokmen and S. Etaner-Uyar, "Traffic sign recognition using Scale Invariant Feature Transform and color classification", *23rd International Symposium on Computer and Information Sciences*, pp. 1-6, 2008.
- [15] T. Lindeberg, "Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales", *Journal of Applied Statistics*, 1994, pp. 224-270.
- [16] D.G. Lowe, "Object Recognition from Local Scale-Invariant Features", *International Conference on Computer Vision*, Greece, 1999, pp. 1150-1157.
- [17] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, vol. 60, n.2, 2004, pp 91-110.
- [18] G. Loy and A. Zelinsky, "Fast Radial Symmetry for Detecting Points of Interest". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, 2003, pp. 959-973.
- [19] A. Martinović, G. Glavaš, M. Juribašić, D. Sutić and Z. Kalafatić, "Real-Time Detection and Recognition of Traffic Signs". *Proc. 33rd International Convention MIPRO 2010*, vol. III, 2010, pp. 247-252.
- [20] Mikolajczyk, K., *Detection of local features invariant to affine transformations*, Ph.D. thesis, Institute National Polytechnique de Grenoble, France, 2002.
- [21] K. Mikolajczyk, and C. Schmid, "A Performance Evaluation of Local Descriptors". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 27, n. 10, 2005, pp. 1615-1630.
- [22] C.F. Paulo and P.L. Correia, "Traffic Sign Recognition Based on Pictogram Contours". *Ninth Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'2008)*, Klagenfurt, Austria, May 2008, pp. 67-70.
- [23] A. Reiterer, T. Hassan, N. El-Sheimy, "Robust Extraction of Traffic Signs from Georeferenced Mobile Mapping Images", *6th International Symposium on Mobile Mapping Technology*. Presidente Prudente, Brasil, July 21-24, 2009.
- [24] F.A. Silva, A.O. Artero, M.S.V. Paiva and R.L. Barbosa, "Um Algoritmo Rápido para o Reconhecimento de Caracteres", *VII Workshop de Visão Computacional*, Curitiba, 2011, pp. 149-154.