24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-1193
PERFORMANCE EVALUATION OF A MULTISCALE DESCRIPTOR TO DETECT FEATURE POINTS ON OBJECT BORDERS IN RANGE IMAGES

Gerardo Antonio Idrobo Pizo
Departamento de Engenheira Eletrônica, Universidade de Brasília.
gerardo.idrobo@gmail.com

José Mauricio Santos Torres Motta
Departamento de Engenheira Mecânica e Mecatrônica, Universidade de Brasília.
jmmotta@unb.br

Abstract. *In this article, it is presented a new method to detect object borders in a multiscale level using descriptors that encode the neighborhood of points belonging to the borders of the object with numeric values named signature. The process generates as output feature points that are used for the registration of the range images. The proposal is to contribute to the state of the art of 3D reconstruction of workspaces that are mapped for navigation of service robots. For evaluating the performance of the proposed descriptor to detect borders in noisy images, the results were compared to the results obtained by other authors and techniques by using test images. The results are presented statistically as the number of feature points identified as true and false by each technique. It can be observed that the proposed technique provides more true points and less false points, approaching the results obtained with a manual identification.*

Keywords: *Object Borders, Multiscale Descriptor, Range Image Registration*

1. INTRODUCTION

The registration process is a fundamental problem in computer vision and pattern recognition (Yang et al., 2013). The process can transform different sets of data in a single coordinate system. This topic plays an important role in the field of biometrics (Bhowmick and Bhattacharya, 2004), remote sensing image registration (Song and Li, 2010), bioinformatics (Hoffmann et al., 1999), industrial applications, among others. Because of the large number of applications, it is impossible that there is a single standard method for range image registration. Several algorithms have been proposed for this purpose, but they still suffer from several limitations, primarily related to ensuring the existence of corresponding points.

According to Pedrosa (2011) correspondent points usually are associated to high values of curvature along the object contour and they are quite useful for the characterization of shapes, due to its compact representation and invariance to rotation and translation. In practice, these feature points facilitate the identification of possible areas of overlapping during the registration process.

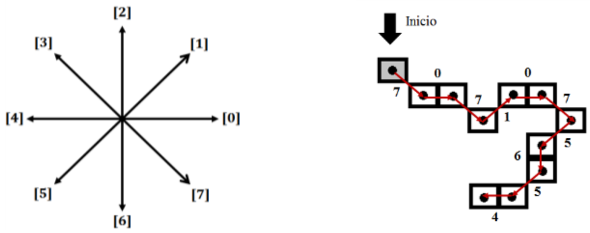
According to Arrebola and Sandoval (2005), the characterization of curvatures is accounted in two large groups: curve approximation (Sato, 1992) and curvature-based algorithms (Rosenfeld and Johnston, 1973, Freeman and Davis, 1977, Beus and Tiu, 1987). The literature provides different methods for curve approximations, which are characterized by fitting a polygon on a curve that is composed of a minimum number of sides. On the other hand, curvature based algorithms classify curve segments point by point, identifying feature points by discontinuous variation of the curvature. Null curvature intervals correspond to straight segments and intervals with continuous curvature variation correspond to circles or arcs. The target contour in this work is made up of segments of open and/or closed curves. The best option to be chosen so that the descriptor is used in multiple cases is a descriptor that characterizes and classifies curves point by point. According to this criterion, the selection can be obtained from curvature-based algorithms. Several authors have worked on the analysis at the neighborhood of a point. Rosenfeld and Johnston (1973) detected feature points using k-cosine as estimate of the curvature at points in the borders of an object. Rosenfeld and Weszka (1975) proposed a modification of Rosenfeld and Johnston (1973), using average k-cosines. Asada and Brady (1984) used Gaussian smoothing and cubic B-Splines to calculate the feature points of a curve. Freeman and Davis (1977) detected feature points analyzing curvature variation by moving a straight line segment along the curve, so that the angular differences between successive segments were used to measure the local curvature. The technique of Beus and Tiu (1987) is similar to the technique of Freeman and Davis (1977) except for one parameter that limits the size of the line segment. Chetverikov and Szabo (2003) proposed a fast and efficient algorithm, known as IPAN, consisting of classifying as feature points those points that have a specific opening angle and with a minimum spacing between adjacent feature points. Most previous methods make use of the chain code algorithm. There are other works that use the Hough transform (Kiryati and Bruckstein, 1992), wavelets (Antoine et al. .1997) and neural networks as tools to detect feature points.

This work presents a new descriptor to circumvent the problem of the existence of correspondence, representing better the geometry of the borders of an object image and relieving the combinatorial complexity in the search of these correspondences between the images. The proposed descriptor is invariant to similarity geometric transformations, it has low sensitivity to noise and generates vectors, or digital signatures, that describe the object borders. The vectors require little storage space and increase the efficiency of the feature detection.

2. CHAIN CODE DESCRIPTOR

Any continuous open or closed curve can be modeled as a discrete curve C(u) that can be represented by an ordered sequence of straight line segments connecting two points, such that $C_u = u_1 u_2 \ldots u_n$, where $\{u_i: 1 \leq i \leq n\}$ and n indicates the number of points of the curve and u_i can be labeled by eight states. States associated with each u_i element represent the relative direction between two feature points along the curve, approximated to a relative direction stepped in 45 degrees (Fig. 1a).

A chain code [Freeman and Davis, 1977] is generated at the same time as the curve points are tracked. Figure 1b provides an example of the assignment of the states. The process starts at the shaded element.



a. 8-state directional code. b. Example of chain code (c.8) 7071075654.

Figure 1. Chain code

In the case of closed curves, encoding finishes when the curve tracing returns to the starting point. The encoding described is absolute, so that each state represents a constant direction with reference to the initial point. According to Gonzales (2008), the method has many limitations: (1) the resulting code is too long; (2) any small disturbance along the curve due to noise or imperfect segmentation causes code variations, which may not necessarily be related to the curve shape; (3) the chain-code varies according to the chosen starting point (in the case of open curves there is ambiguity between the initial and end points).

In this section, it is described a new algorithm to reduce the limitations of the chain-code algorithm. The initial steps of the proposed optimization algorithm filter out noise and select feature points. Then, each selected feature point is assigned to a signature according to its neighbors. The signature is invariant to similarity transformations. The descriptor is defined by operations with integer numbers, which produces benefits for being simple for programming in any type of hardware (FPGAs, microprocessors, microcontrollers, among others) and it is computationally efficient.

The proposed algorithm can label feature points on a contour curve and it is described following:

Step [1]. *Noise Attenuation of the curve C(u) with the Normalized Least Mean Squares (NLMS) Adaptive Filter.*

A curve ($\boldsymbol{C}_u$) in a point cloud can be described as an ordered sequence of straight line segments, $\boldsymbol{u}_n$, connecting two points and can be modeled as $\boldsymbol{C}_u = \boldsymbol{H}_u \boldsymbol{u}_n + \boldsymbol{v}_u$, in which $\boldsymbol{H}_u$ is the observation model that transforms the real state space in the observation space and $\boldsymbol{v}_u$ is the observation noise modeled as a Gaussian with zero mean white noise.

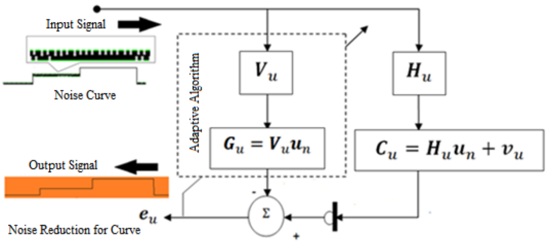
Figure 2a shows the noise attenuation algorithm in a block diagram. The process consists in comparing the desired output signal $\boldsymbol{G}_u = \boldsymbol{V}_u \boldsymbol{u}_n$ from the filter and the input signal $\boldsymbol{C}_u$, calculating an estimation of the error $\boldsymbol{e_u} = \boldsymbol{C}_u - \boldsymbol{G}_u = (\boldsymbol{H}_u - \boldsymbol{V}_u)\boldsymbol{u}_n + \boldsymbol{v}_u$. An ideal case of noise filtering would be $\boldsymbol{H}_u = \boldsymbol{V}_u$, producing therefore $\boldsymbol{e_u} = \boldsymbol{v}_u$, so the noise would be completely identified and eliminated. However, the approximation of $\boldsymbol{V}_u$ to $\boldsymbol{H}_u$ by a finite length filter with limited accuracy prevents noise from being perfectly modeled.

The $\boldsymbol{V}_u$ operator is obtained iteratively by the NLMS algorithm, and can be expressed by Eq. (1) [Haykin, 2002]:

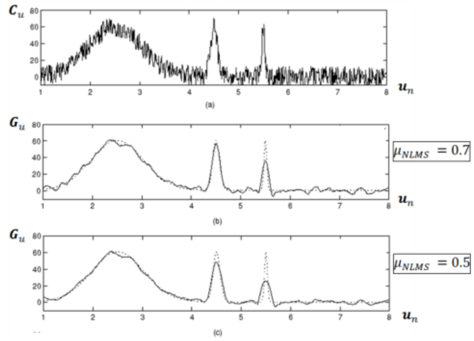
$$\boldsymbol{V}_u = \boldsymbol{V}_{u-1} + \frac{\mu_{NLMS}\, \boldsymbol{e_u} \boldsymbol{u}_n}{\boldsymbol{u}_n^T \boldsymbol{u}_n} \quad , \qquad (1)$$

where μ (NLMS) is the performance parameter value, with values in the range [0 - 1].

Figure 2b shows the input signal to the NLMS filter and its output signal for different values of the performance parameter. As the value of μ_{NLMS} is decreased the $\boldsymbol{G_u}$ curve gets smoother compared to the curve $\boldsymbol{C_u}$.



(a). Diagram showing the noise attenuation process for a curve.



b) Noise attenuation for different performance parameters.
The dotted line is the original points before using the performance parameter on the curve $\boldsymbol{C_u}$.

Figure 2. Normalized Least Mean Squares (NLMS) Adaptive Filter.

Step [2]. *Detection of Feature Points*

The developed method for detection of feature points consists in:

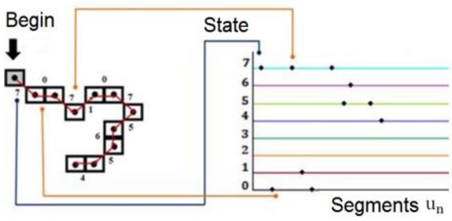


Figure 2. Data Representation of *State vs. Segments*.

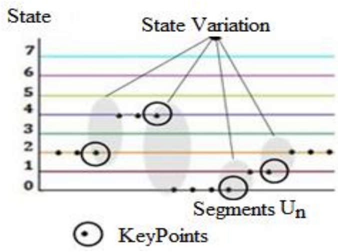


Figure 3. Detection of feature points in the graphic *State vs. Segments*.

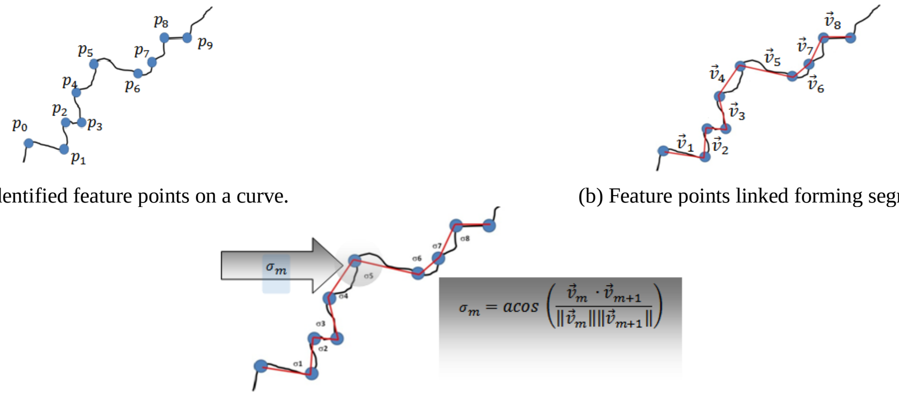
a. Relating the u_n segments of the curve in a graph State vs Segments (Fig. 2).
b. Assignment of angle values $\alpha(u_n)$ to the variation of the segment directions between successive points (Eq. (2)). The general strategy to build the descriptor is not to find local points but the values defined by the corresponding state variations between segments constructed between points. (Fig. 3).

$$\alpha(u_n) = abs\left[tan^{-1}\left(\frac{u_{n+1}}{u_n}\right)\right] \quad (3)$$

A feature point is detected if $\alpha(u_n) \neq 0$.

Step [3]. *Signatures of Feature Points*

c. Selection of feature points on the curve $C(u)$ that were detected in the graph S*tate vs. Segments*. Connect the feature points by straight lines generating new segments $\vec{v}_m$ (Figs. 4a and 4b);



(a) Identified feature points on a curve.

(b) Feature points linked forming segments.

(c) Calculation of the angle between segments connecting feature points.

Figure 4. Selected feature points on the curve and calculation of the angle between segments connecting the feature points.

d. Calculation of the change in direction of the segments (angle σ_m) between $\vec{v}_m$ and $\vec{v}_{m+1}$ from the scalar product by Eq. (4) (Fig. 4c).

e.

$$\sigma_m = acos\left[\frac{\vec{v}_m . \vec{v}_{m+1}}{\|\vec{v}_m\| . \|\vec{v}_{m+1}\|}\right] \tag{4}$$

f. Discretization of σ_m in 4 states of a directional code as in Tab. 1 and association of these states to discrete values of the angles $(\sigma_d)_{4E}$ (4E=four states).

Table 1. Discrete values associated to the 4-state directional codes.

Angle variation(σ_m)	Discrete Angle$(\sigma_d)_{4E}$
$\sigma_m < 45°$	$\sigma_d = 0$
$45° \leq \sigma_m < 90°$	$\sigma_d = 1$
$90° \leq \sigma_m < 135°$	$\sigma_d = 2$
$135° \leq \sigma_m < 180°$	$\sigma_d = 3$

g. To generate the signature, the information is collected from the neighborhood of each feature point PP by connecting this point to its six closest neighbors. The process can be seen graphically in Fig. 5.

h. The variation of the directions of the segments connecting the feature point PP and its six neighbors is calculated from the scalar product between the direction vectors of these segments, as shown in Tab. 2.

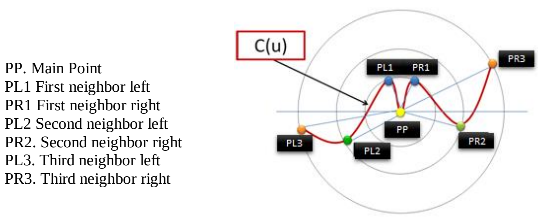


Figure 5. Process to generate the signature associated to a feature point.

Table 2. Angles between segments connecting feature points for the generation of a signature associated to a point – curve $\boldsymbol{C(u)}$.

	Angles between points				
Points	PL1, PP PR1	PL1 PP PL2	PL2 PP PL3	PR1 PP PR2	PR2 PP PR3
(σ_m)	Main Angle ~33°	~98°	~18°	~87°	~52°

i. Angle values are sorted in ascending order. The ordering of the angles starts from the main angle and thus the proposed descriptor becomes invariant to rotations. (Tab. 3);

j. Discretization of the angles σ_m in σ_d discrete angles making use of the values of the 4-state directional code, as shown in Tab. 1. With this angle encoding, a complete signature is generated in the form of a numeric code, ordering the discretized angles in digits (Tab. 4).

During the tracking process along the $C(u)$ curve the optimization algorithm does not prevent that there are closed curves. To avoid this, a further step can be included.

Table 3. Data organized from the angles between feature points for the generation a signature – curve $\boldsymbol{C(u)}$.

	Angles between points				
Points	PL1 PP PR1	PL2 PP PL3	PR2 PP PR3	PR1 PP PR2	PL1 PP PL2
(σ_m)	Main Angle	~18°	~52°	~87°	~98°

Table 4. Discrete angles making use of the 4-state directional code – full signature.

	Angle between point segments				
Points	PL1 PP PR1	PL2 PP PL3	PR2 PP PR3	PR1 PP PR2	PL1 PP PL2
(σ_m)	Main Angle ~33°	~18°	~52°	~87°	~98°
Full Signature	0	0	1	1	2
$(\sigma_d)_{4E}$	1° Digit	2° Digit	3° Digit	4° Digit	5° Digit

k. When the start and end points are equal, the algorithm ends.

The proposed algorithm extracts feature points from a curve that are represented numerically. The proposed descriptor has fewer limitations than the original chain-code descriptor. Tab. 5 presents a comparison between the original chain-code and this proposed descriptor.

Table 5. Comparison between the original Chain-Code Algorithm and the proposed Algorithm.

Curve Representation		
	Limitations of the Chain-Code Algorithm [Gonzales and Woods, 2008]	Proposed Descriptor
1	The resulting code can be too long.	Only feature points receive a signature chain code.
2	Any small disturbance along the border, due to noise or imperfect segmentation causes variations in the code that may not be necessarily related to the shape of the contour.	Noise is attenuated through adaptive filters.
3	The chain-code depends on the starting point (there is ambiguity in the case of open curves).	The code generated for each feature point does not depend on the starting point.

3. DETECTION OF FEATURE POINTS – EXPERIMENTAL RESULTS

The task of comparing results is not easy, since classifying a point depends largely on the application or from the user's point of view. The parameter used by the proposed descriptor is the performance parameter (Eq. 1), that can be used to minimize the influence of noise and to remove irregularities. Thus, the performance parameter to be selected depends on the amount of noise in the contour to be minimized. In a search of comparative results with other techniques to detect feature points the work of Dmitry and Zsolt (1999) was selected, which brings together results from other authors such as: Ronself and Jonhston (1973), Ronself and Weszka (1975), Freeman and Davis (1977) and Beus and Tiu (1987). To test the proposed descriptor, images were taken from the same article of Dmitry and Zsolt (1999), printed and subsequently scanned (Fig. 6). It is worth mentioning that during the scanning process, involuntarily, noise was added to the images. These images could be used as input parameters to test the robustness of the algorithm to noise.

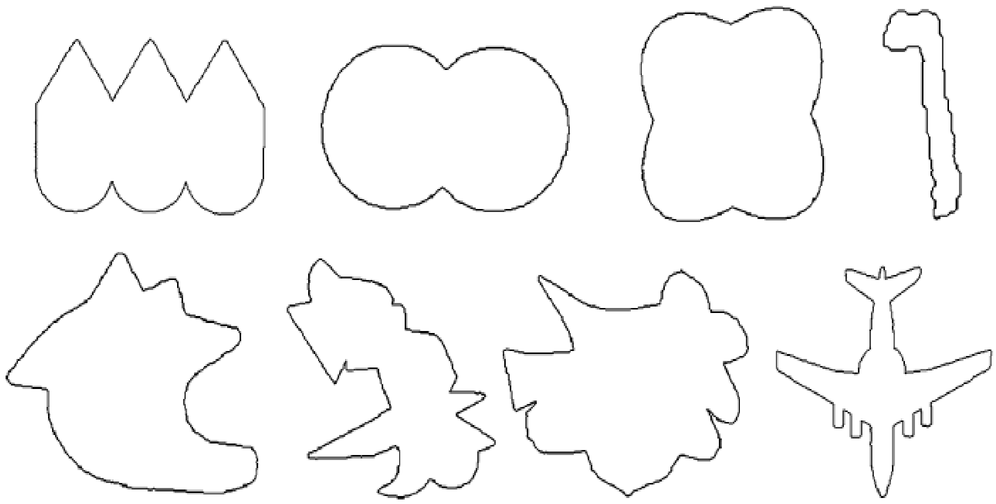
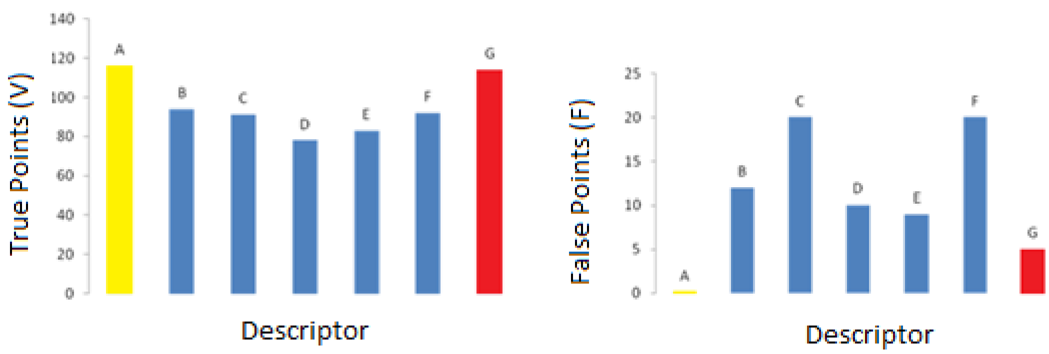


Figure 6. Examples of images used in the test.

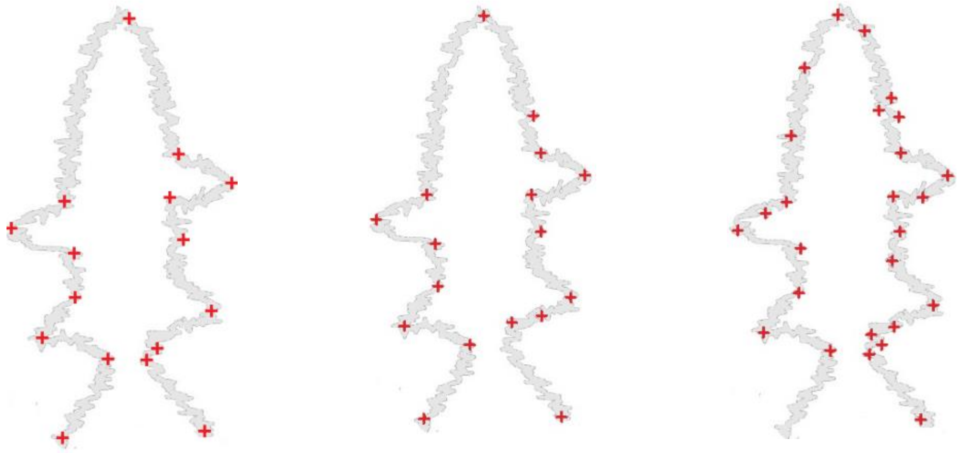
In the tests, the algorithms of each technique to detect feature points were not programmed, but just used comparatively with the results already published in the article of Dmitry and Zsolt (1999). To evaluate the performance of the proposed descriptor to detect feature points in noisy contours, results will be evaluated from values calculated with a parameter of selectivity, which sorts the number of real points (V) and number of false points (F) for each technique, when the resulting image is compared to a reference image. A reference image has features points that would be detected through manual selection, with no influence of the noise level. Next the results obtained during this research are presented.



Descriptors	Symbol	True Points	False Points
Manual Identification	A	116	0
RONSELF e JONHSTON, (1973)	B	94	12
RONSELF e WESZKA, (1975)	C	91	20
FREEMAN e DAVIS, (1977)	D	78	10
BEUS e TIU (1987)	E	83	9
CHETVERIKOV e SZABO (1999)	F	92	20
Proposed descriptor (2014)	G	114	5

Figure 7. Statistical results of the performance of several descriptors

Following, the results from images with high level of noise are shown (Fig. 8).



$\mu_{NLMS} = 0.2$ $\mu_{NLMS} = 0.4$ $\mu_{NLMS} = 0.7$

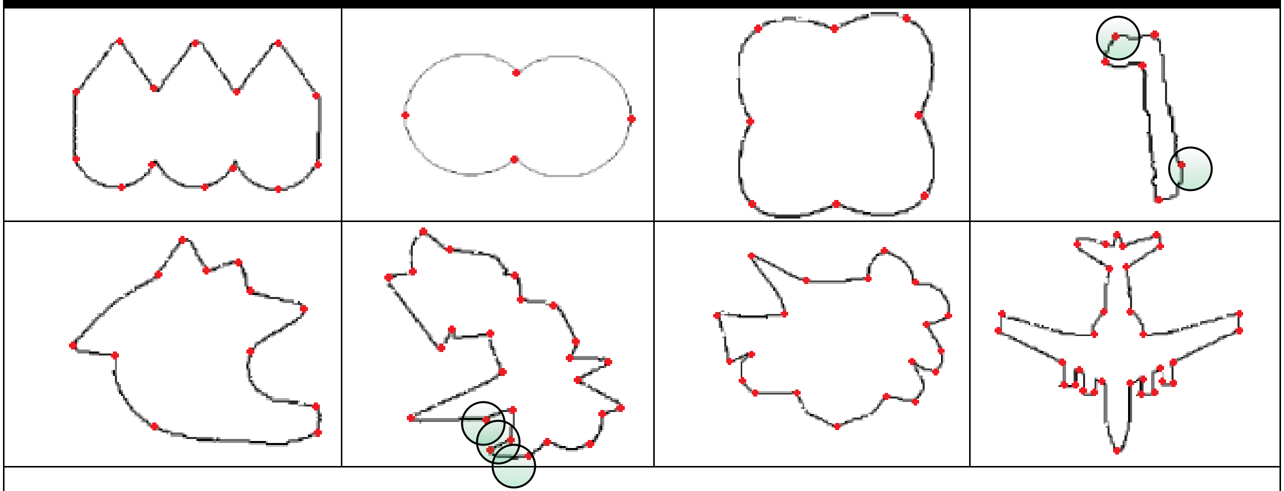
Figure 8. Results with different Performance Parameter values μ_{NLMS}.

To evaluate the performance of the descriptor in multiple scales reference images were reduced to two different scales as is presented in the Tab. 6.

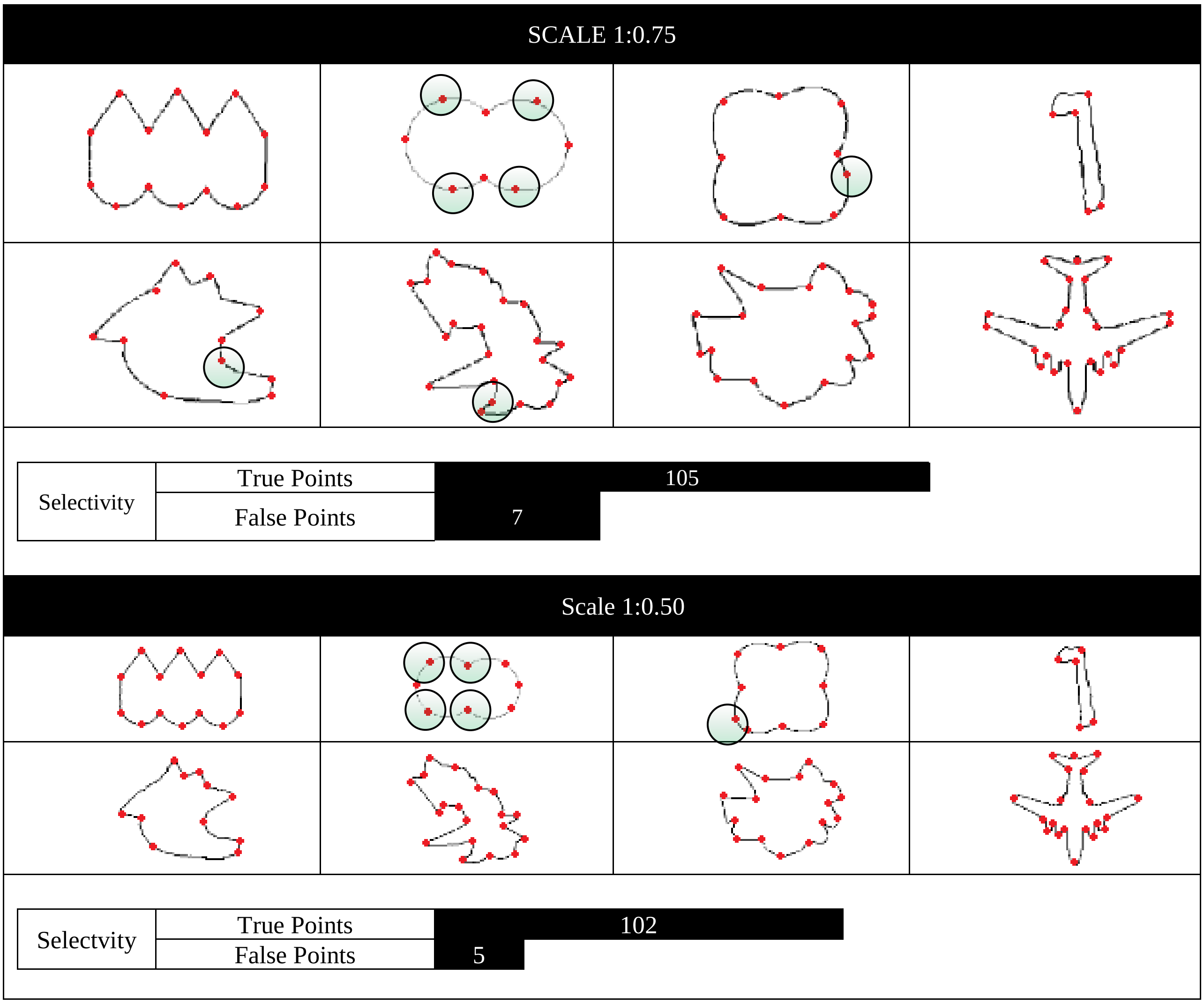
Table 6. Descriptor performance with different scales.

Proposed Descriptor

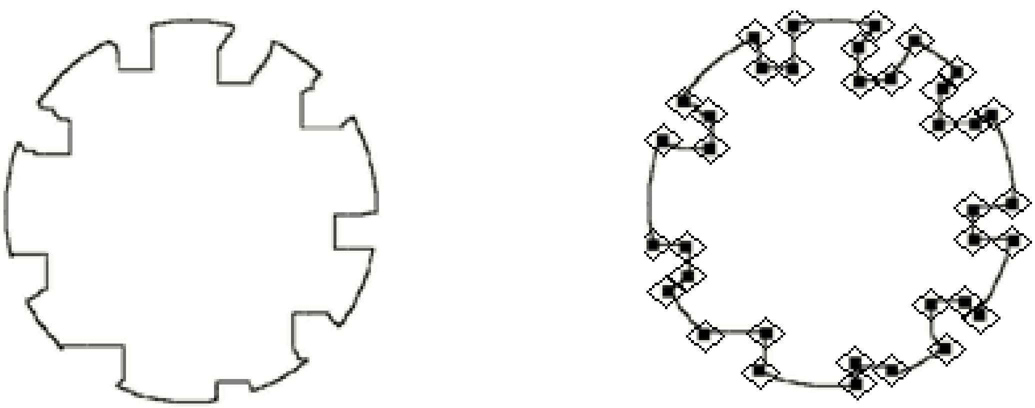
Scale 1:1



Selectivity	True Points	114
	False Points	5



Another procedure performed to verify the performance of the proposed descriptor is to use images with different degrees of perspective view. It was initially selected a reference image and, later, through the proposed descriptor, detected the feature points. These points were identified as points of reference through the symbol (♦) (Fig. 9).



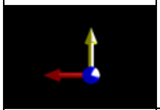
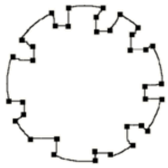
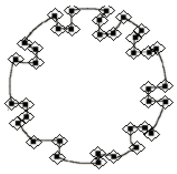
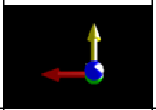
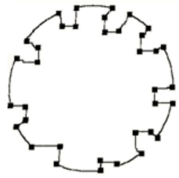
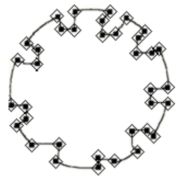
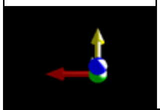
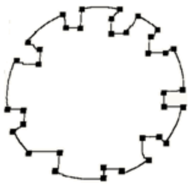
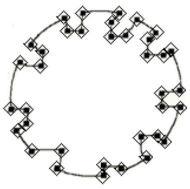
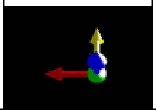
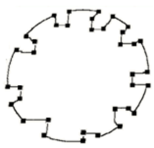
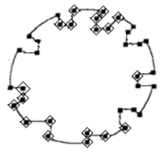
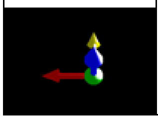
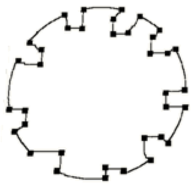
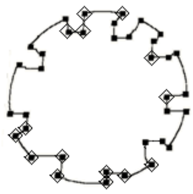
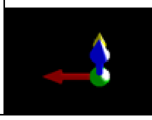
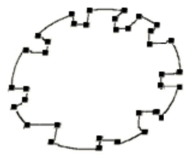
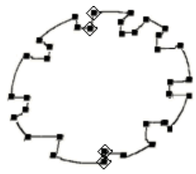
Number of Points (♦)=36

(a) Reference Image (b) Points detected with the proposed descriptor

Figure 9. Reference images used to evaluate the descriptor performance at different perspective angles. ($\mu_{NLMS} = 1.0$).

Next, the image (fig. 9b) was rotated around its geometric center at different angles (θ_i). Then, for each image the proposed algorithm was run and the feature points were detected and compared to the reference image using the symbol (♦).

Table 7. Descriptor Performance with images at different perspective angles.

Image Perspective Angle (θ_i)	Points detected by the proposed descriptor	Points Identified	Image Perspective Angle (θ_i)	Points detected by the proposed descriptor	Points Identified
0^0 		 Number of Points (♦)=36	5^0 		 Number of Points (♦)=36
10^0 		 Number of Points (♦)=36	15^0 		 Number of Points (♦)=19
20^0 		 Number of Points (♦)=15	25^0 		 Number of Points (♦)=4

4. RESULTS AND DISCUSSION

In Fig. 7, we presented the results obtained in the registration process. In a comparison of the results it can be noted that the proposed technique offers a larger number of true corresponding points and a low number of false feature points, approaching the results obtained with manual identification. It should be mentioned that the proposed descriptor used the performance parameter value, μ (NLMS) = 0.9, (Eq. 1) for all images. This value was chosen because there is a low level of noise in the images according to the criteria in Fig. 2b.

In figure (4) it can be observed that, by increasing the value of the performance parameter, the feature points appear following the contour in the figure. This process is typical of adaptive algorithms. For the process of aligning two images (registration), it is recommended the availability of a large amount of points in both images, since the likelihood of finding corresponding signatures is increased. The performance parameter also allows the adjust the number of points of a contour.

From Tab. 7, the angles of rotation versus number of points identified are shown in Fig. 10, as a histogram. It is observed that the number of feature points decreases as the angle of rotation increases, as expected.

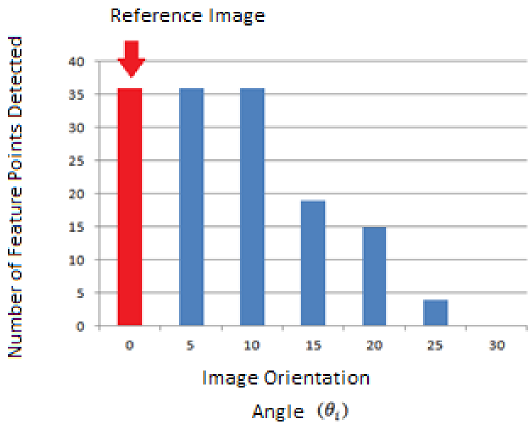


Figure 10. Chart showing perspective angle vs number of feature points detected.

The latter procedure was repeated for different types of curves and the result was the same. The proposed technique can identify points of the same image, when it is twisted up to a maximum value of 20°.

5. CONCLUSIONS

This article describes a set of experiments performed to evaluate the performance of a proposed technique to detect feature points on noisy contours and the steps of the complete algorithm. The results were acceptable in comparison to other authors' results. The proposed technique has generated a larger number of true corresponding points and a low number of false feature points, approaching the results obtained by a manual identification process. The descriptor has proved to be tolerant to affine transformations, since it considers angles in discrete values, i.e. the variations of image perspectives up to approximately 20°.

6. REFERENCES

Arrebola, F., and Sandoval, F, (2005). Corner detection and curve segmentation by multiresolution chain-code linking. *In:Elseiver*, Vol.38, pp 1596 – 1614.

Asada, I. and Brady, M., (1984). "The Curvature Primal Sketch". *Acquisition of Visual Motion Guided Behaviors.*

Bhowmick,P and BHATTACHARYA, BB.,(2004).Aproximate finger print matching using kd-tree.,*in: Proceddings of the 17th internacional conference Pattern Recognition*, Vol.1. page 544-547.

Dmitry C., and Zsolt S., 1999. A simple and efficient algorithm for detection of high curvature points in planar curves. *In : CiteSeer*. < http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.5752>. Data de acesso 10 agosto 2012.

Freeman, H., and Davis, L. S.(1977). "A Corner Finding Algorithm for Chain Coded Curves". In: *IEEE Transactions in Computing*, Vol. 26, issue 3, pp 297–303.

Haykin, Simon (2002). *Adaptive Filter Theory*, Prentice Hall, ISBN 0-13-048434-2.

Hoffmann,F., Kriegel,K., Wenk, C., (1999)., An applied point pattern matching problem: comparing 2D patterns of protein spots.In: *13th European Workshop on Computational Geometry CG '97*., Volume 93, Issue 1, Pages 75–88 <doi:10.1016/S0166-218X(99)00007-4>.

Rosenfeld. H., and Johnston. E., (1973). Angle Detection on Digital Curves. In: *IEEE Transactions on Computers*, Vol. C-22, issue 9, pages 875–878.

Rosenfeld. A., and Wezka. J. S.,(1975), An Improved Method of Angle Detection on Digital Curves. In: *IEEE Transactions on Computers*, Vol. C-24, issue 9, pages 940–941.

Sato,Y. (1992). "Piecewise linear approximation of plane curves by perimeter optimization". *In:Pattern Recognition*. Vol. 25, issue (1992), pp 1535–1543.

Song, Z., LI, T.,(2010). Remote sensing image registration approach base o a retrofitted SIFT algorithm and lissajous-curve trajectories. *In:Opt. Express* 18. Pages (513-522)< doi: 10.1364/OE.18.000513>.

Yang,Su, Erling Wei and Yuanyuan Wang (2013). Triangle chain codes for image matching. In: Elseivier. Neurocomputing.Vol.1, pp (268-276).

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.