

# Deterministic Partially Self-Avoiding Walks on Networks for Natural Shapes Classification

Lucas C. Ribas<sup>1,2</sup> and Odemir M. Bruno<sup>1</sup>

<sup>1</sup> Institute of Mathematics and Computer Science, University of São Paulo - USP,  
Avenida Trabalhador são-carlense, 400, 13566-590, São Carlos, SP, Brazil

lucascorreiaribas@gmail.com

<sup>2</sup> São Carlos Institute of Physics, University of São Paulo - USP, PO Box 369,  
13560-970, São Carlos, SP, Brazil

bruno@ifsc.usp.br

**Abstract.** Shape is an important characteristic used by different classification tasks in computer vision. In particular, shape is useful in many biological problems (e.g. plant species recognition and fish otolith classification), which are challenging due to the diversity found in nature. This paper proposes a novel method for shape analysis and classification based on deterministic partially self-avoiding walks (DPSWs) on networks. First, a shape contour is modeled as a network by mapping each contour pixel as a vertex. Then, deterministic partially self-avoiding walks are performed on the network and a robust shape signature is obtained using statistics of the trajectories of the DPSWs. We evaluate this feature vector in a classification experiment using two different natural shape databases: USPLeaves and Otolith. The experimental results demonstrate a high classification accuracy of the method when compared to the other methods. This suggests that our method is a promising option for the classification task in biological problems.

**Keywords:** Deterministic walk · Networks · Shape analysis.

## 1 Introduction

Shape is a classical attribute used in different tasks of computer vision. It is one of the most important features for object recognition with first studies dated from the 60's [1]. Furthermore, one of the interests in this feature is inspired by biological problems where shape characteristics are necessary [6].

Many methods for shape classification have been proposed in the literature. These methods can be divided into three main categories [14]: skeleton-based, region-based and contour-based techniques. The category of a method is related to the process used to extract the features from the shape. In this sense, skeleton-based methods use the medial axes of the shape for characterization and have as advantage the robustness for shapes with occlusion (e.g. [2, 5]). On the other hand, region-based techniques use the whole image to characterize the shapes. Examples of this category are Zernike moments [23] and Hu moments-based

methods [12]. The last category, uses contour information of the shape for characterization (e.g. Fourier descriptors [21], Curvature Scale Space [15] and multi-scale fractal dimension [20]). However, many methods of this category suffer when the silhouette is degraded. Recently, methods based on complex networks have been proposed to analyze shapes [3, 19, 18] and to deal with non-perfect contours that are often found in biological problems.

In this paper, we proposed a novel method for shape analysis based on the deterministic partially self-avoiding walk (DPSW) on networks. To accomplish this task, the shape contour is modeled as a network, including the information about the distribution of the contour pixels in the image. Then, we apply the DPSW on the network and use statistical measurements about the trajectories to obtain a shape signature, which can be used to distinguish different classes of shapes. The proposed method was applied on biological shape classification, which is a challenging problem, due to the diversity found in nature. Experiments results on the two natural databases showed the effectiveness of the proposed method compared to other shape methods.

This paper is organized as follows. Section 2 describes the proposed method to obtain a shape signature. Then, Section 3 presents the results obtained by the proposed method and others. Finally, in Section 4, the conclusions of the work are drawn.

## 2 Proposed Method

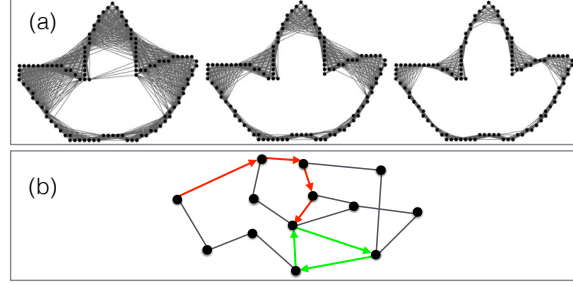
### 2.1 Modeling contour as network

Different approaches have been proposed to model contour as network [19, 3]. Based on this, we model the contour as follows. Let a network represented by a set of vertices  $V = \{v_1, v_2, \dots, v_n\}$  and a set of edges  $E$  where each edge  $e_{v_i, v_j}$  connects two vertices  $v_i$  and  $v_j$ . The neighbors of a vertex are given by  $\nu(v)$ . Now, consider a set of pixels  $S$  of size  $N$  that belong to the contour  $C$  of a shape. Each pixel  $s_i \in S$  is represented by discrete numerical values  $s_i = (x_i, y_i)$  representing its Cartesian coordinates. To build a network, each point of the contour  $s_i \in S$  is mapped into a vertex  $v_i \in V$  of the network. Then, we connected each pair of vertices with a non-directed edge. For each edge  $e_{v_i, v_j}$  is attributed a weight  $w(e_{v_i, v_j})$  according to the Euclidean distance considering the coordinates of the pixels  $s_i$  and  $s_j$  represented by the vertices  $v_i$  and  $v_j$ ,  $w(e_{v_i, v_j}) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ . The edges weights are normalized by the largest distance between all pixels  $w_{max}$  of the contour,  $w(e_{v_i, v_j}) = \frac{w(e_{v_i, v_j})}{w_{max}}$ . This normalization aims to keep the edges weights in the interval  $[0, 1]$  and invariant to scale of the shape.

At the moment, the network presents a regular behavior, i.e., all vertices have the same degree. Thus, it is necessary to apply a procedure to transform the network and to reveal important characteristics [17]. To accomplish this task, a simple and widely used approach is to apply a threshold  $t$  to remove edges. In this approach, a new network  $G_t = (V_t, E_t)$  is obtained for each value of

threshold  $t$  (Figure 1(a)). Therefore, the transformed network  $G_t$  is obtained removing all edges of  $E$  whose the weight is equal or smaller than  $t$ ,  $E_t = \{e_{v_i, v_j} \in E | w(e_{v_i, v_j}) \leq t\}$ .

Each value of  $t$  highlights different patterns of the network topology. In this way, we use different threshold values to achieve a richer set of measurements that describe the network dynamics [9, 7]. Thus, the network characterization is performed using a set of thresholds  $T$ , which is defined by an initial threshold  $t_0$ , an incremental threshold  $t_i$  and a final threshold  $t_f$ .



**Fig. 1.** (a) Example of the application of different values of threshold in a contour network. (b) Illustration of a DPSW on a network using the  $r = \max$ ,  $\mu = 1$  and the vertex degree as criterion of distance. The edges in red represent the movements of the tourist until find the attractor (edges in green).

## 2.2 Deterministic Partially Self-Avoiding Walk on Network

Recently, DPSW has emerged as a promising approach for pattern recognition, obtaining great results in different problems such as texture analysis [4], dynamic texture analysis [8] and synthetic complex networks classification [9]. A DPSW on the network can be interpreted as a tourist that visit  $N$  vertices of the network. The tourist starts the walk from a pre-defined vertex  $v_i$  and the rule of movement  $r$  to visit the next vertex  $v_j$  is: go to the neighboring vertex that minimizes the distance  $d(v_i, v_j)$  and has not been visited in the previous  $\mu$  steps. In this rule, named as  $r = \min$ , the tourist moves by minimizing the distances between vertices. On the other hand, in the rule  $r = \max$ , the tourist maximize the distances between vertices. On networks, it is necessary to define the distance  $d(v_i, v_j)$  between two vertices  $v_i$  and  $v_j$ . In this way, we consider two different criterion of distance  $c$ : the degree of the neighboring vertex  $d(v_i, v_j) = k_{v_j}$  and the edge weight  $d(v_i, v_j) = w(e_{v_i, v_j})$ . It can occur of two neighboring vertices have the same distance, in this case, we consider as tiebreaker: the edge weight for the criterion of degree and the vertex degree for the criterion of the edge weight.

The memory  $(\mu - 1)$  is the window in which the tourist performs a deterministic walk partially self-repulsive. In other words, the tourist will consider a visited

vertex attractive again after having traveled  $\mu$  vertices. The rule of movement is based on the neighborhood of the vertex  $\nu(v)$  and a short memory  $\mu$ , which although simple, can generate DPSW of great complexity [4, 7]. The tourist walk can be divided into two parts: transient and attractor. The transient is the initial part of the walk that has size  $\tau$ , and the attractor is the final part of the walk, consisting of a cycle of period  $\rho \geq (\mu + 1)$  in which the attractor gets trapped. The attractor is a cycle of vertices of period  $\rho$  that will always be visited by the tourist, i.e. when entering an attractor the tourist cannot escape. Therefore, the transient is the route of the tourist until finding an attractor, where its size  $\tau$  is the number of vertices traveled by the tourist from the initial vertex to the first vertex of the attractor. Figure 1(b) illustrates a trajectory performed by a DPSW. In some cases, depending on the topology of neighboring vertices and the size of memory  $\mu$ , it can occur that the tourist does not find an attractor. In this case, the trajectory is considered only by the transient time  $\tau$ .

Each vertex of the network is considered as an initial condition to perform a walk. Thus, given a network composed of  $N$  vertices, we obtain a set of  $N$  trajectories. To measure this set of trajectories performed on the network, a transient time and attractor period joint distribution  $S_{\mu,r}(\tau, \rho)$  is used. This distribution combines the transient time  $\tau$  and the attractor period  $\rho$ , storing in each position of the distribution the frequency of trajectories with transient  $\tau$  and attractor  $\rho$  [4, 7],

$$S_{\mu,r}^t(\tau, \rho) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1, & \text{if } \tau_i = \tau \text{ and } \rho_i = \rho \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where  $i$  is the vertex where the walk was started and  $t$  is the value of the threshold applied in the network.

In the last years, several approaches have used features obtained from the joint distribution for classification [4, 7]. In this sense, the histogram  $h_{\mu,r}^t(l)$  proved to achieve better results in classification tasks. This histogram computes the number of trajectories with size  $l = \tau + \rho$  in a joint distribution calculated with memory size  $\mu$ , rule of movement  $r$ , value of threshold  $t$  and criterion of distance  $c$  according to  $h_{\mu,r}^{t,c}(l) = \sum_{b=0}^{l-1} S_{\mu,r}^k(b, l-b)$ .

In order to construct a feature vector to characterize a shape, we use the  $n$  first descriptors of the histogram  $h_{\mu,r}^{t,c}(l)$ . In this work, we define empirically  $n = 4$ . The first position of the histogram is  $(\mu + 1)$  since there is no attractor smaller than  $(\mu + 1)$ . Thus, the feature vector  $\Psi(\mu, r)_t^c$  is given by  $\Psi(\mu, r)_t^c = [h_{\mu,r}^t(\mu + 1), h_{\mu,r}^t(\mu + 2), \dots, h_{\mu,r}^t(\mu + n)]$ . Each rule of movement extracts different properties from the network [9]. Therefore, we combine the feature vector  $\Psi(\mu, r)_t^c$  for the two rule of movement. To obtain a robust feature vector, we also concatenate the feature vector  $\Psi(\mu, r)_t^c$  for the two criterion of distance (degree and edge weight). Thus, a combined feature vector  $\Upsilon_\mu^t$  is given by  $\Upsilon_\mu^t = [\Psi(\mu, min)_t^k, \Psi(\mu, max)_t^k, \Psi(\mu, min)_t^w, \Psi(\mu, max)_t^w]$ , where  $k$  is the criterion of distance based on degree and  $w$  is the criterion of distance based on edges weights.

Earlier works [2, 19, 3] have demonstrated that characteristics of the network dynamic evolution using different values of threshold significantly improves the network topology analysis. In this way, a combined feature vector  $\Phi_\mu$  that considers information of networks transformed by different values of  $t$  and uses memory size  $\mu$  is given by:  $\Phi_\mu = [\mathcal{Y}_\mu^{t_0}, \mathcal{Y}_\mu^{t_1}, \dots, \mathcal{Y}_\mu^{t_f}]$ . Finally, the feature vector  $\Omega$  is given by concatenating feature vectors  $\Phi_\mu$  for different values of memory size  $\mu$ , starting from 0 to the maximum value of memory size  $M_\mu$ . This final feature vector is defined as  $\Omega = [\Phi_0, \Phi_1, \dots, \Phi_{M_\mu}]$ .

### 3 Results and Discussion

#### 3.1 Experimental setup

The proposed method and others were applied on two databases: USPLeaves and Otolith. The USPLeaves [3] database contains images of different species of plants. The classification of this database is a challenging task since that different species of plants have similar shapes and in the digitalization step of the images, overlaps can occur in the adjacency of the objects. This database is composed of 30 leaf species with 20 samples each. The Otolith database is composed of the fish natural database for fish recognition. The database was provided by AFORO [13], which is an open online catalog of otolith images. This database contains 180 different otolith images with 20 different species each.

The literature methods considered for this experiment are: CN Degree [3], Fourier descriptors [16], Curvature descriptors [22], Zernike Moments [23], Multi-Scale Fractal Dimension [20], Segment Analysis [11] and Angular Descriptors of Complex Networks (ADCN) [19]. In order to evaluate the proposed method and other methods, we perform a classification task over its feature vectors using a 10-fold cross-validation scheme and ten repetitions. We use the SVM classifier [10] with a linear kernel, a well-known supervised method.

#### 3.2 Parameter Evaluation

The evaluated parameters of the proposed method are the set of threshold  $T$  and the values of memory size  $\mu$ . The set of threshold  $T$  is defined by an initial threshold  $t_0$ , an incremental threshold  $t_i$  and a final threshold  $t_f$ . Table 1 presents the classification results for different configurations of  $t_0$ ,  $t_i$  and  $t_f$ . Firstly, we consider  $t_0 = 0.025$  and  $t_i = 0.025$  for different values of final threshold  $t_f$ . Note that the best accuracy was obtained for  $t_f = 0.95$ . We also note that low values of  $t_i$  provided better accuracies. Thus, we define the values  $t_0 = 0.025$ ,  $t_i = 0.025$  and  $t_f = 0.950$  as the default parameters of the proposed method, since these values obtained good results on the two databases.

The values of memory size are evaluated in function of the maximum value of memory size  $M_u$  in Table 2. Note that the accuracies tend to decrease or oscillate very little as we increase the value of  $M_\mu$ . Also, as we increase the values of  $M_\mu$  the number of features increases as well. In this way, the value  $M_\mu = 2$  presents a good tradeoff between performance and number of features on the two databases. Therefore, the final feature vector is given by  $\Omega = [\Phi_0, \Phi_1, \Phi_2]$ .

**Table 1.** Accuracies obtained for different set of thresholds.

Thresholds			Accuracy (%)	
$t_0$	$t_i$	$t_f$	USPLeaves	Otolith
0.025	0.025	0.650	87.25 (0.11)	63.33 (0.78)
0.025	0.025	0.800	87.42 (0.35)	65.83 (0.39)
<b>0.025</b>	<b>0.025</b>	<b>0.950</b>	<b>88.00 (0.11)</b>	<b>68.88 (0.11)</b>
0.025	0.050	0.625	85.16 (0.94)	61.94 (1.17)
0.025	0.050	0.775	85.66 (0.47)	64.44 (1.57)
0.025	0.050	0.925	85.00 (0.23)	67.50 (0.39)
0.050	0.025	0.950	88.25 (0.35)	68.33 (0.33)
0.075	0.025	0.950	88.00 (0.47)	68.33 (0.78)
0.100	0.025	0.950	87.66 (0.47)	68.61 (0.39)

**Table 2.** Accuracies for different values of maximum memory  $M_\mu$ .

$M_\mu$	Accuracy (%)	
	USPLeaves	Otolith
<b>2</b>	<b>88.58 (<math>\pm 0.82</math>)</b>	<b>70.55 (<math>\pm 0.39</math>)</b>
3	88.66 ( $\pm 0.23$ )	68.88 ( $\pm 0.16$ )
4	87.92 ( $\pm 0.35$ )	66.94 ( $\pm 1.18$ )
5	88.25 ( $\pm 0.58$ )	66.11 ( $\pm 1.57$ )
6	88.50 ( $\pm 0.47$ )	65.83 ( $\pm 0.39$ )

### 3.3 Comparison with other methods

This section presents a comparison experiment between the proposed method and others (cited in Section 3.1). Table 3 shows the accuracies achieved for each shape method evaluated on the two shape databases. For the proposed method, we used the best parameter setting evaluated in Section 3.2. For the literature methods, we considered the default parameters defined by the authors. The results showed that the proposed method presents the highest accuracy for both USPLeaves and Otolith databases. On the USPLeaves database, the performance of the proposed method is 3.42% superior to the second best method (C.N. Degree). On the other hand, the proposed method significantly improves the accuracy on the Otolith database when compared to the second best method (Segment Analysis), e.g. from 63.47% to 70.55%. The Otolith database is a much harder classification database due to the presence of little inter-class variability.

Concerning the method based on networks, the proposed method also achieved the highest accuracy on the two databases. On the USPLeaves database, the proposed method improves the accuracy compared to the C.N. Degree and ADCN methods in 3.42% and 9.75%, respectively. On the Otolith database, the proposed method significantly improves correct classification rate compared to the network-based methods, e.g. from 48.08% to 70.55% (C.N. Degree) and from 36.32% to 70.55% (ADCN). These results indicate that the DPSW obtains more significant characteristics of the network for the classification task than other measures.

**Table 3.** Classification performance of various shape methods on the two databases.

Methods	Accuracy (%)	
	USPLeaves	Otolith
C.N. degree	85.16 ( $\pm$ 3.63)	48.08 ( $\pm$ 1.07)
Fourier	83.16 ( $\pm$ 4.61)	50.35 ( $\pm$ 0.72)
Curvature	81.66 ( $\pm$ 3.33)	49.93 ( $\pm$ 1.31)
Zernike	76.33 ( $\pm$ 5.07)	20.04 ( $\pm$ 1.44)
M.S. Fractal	73.66 ( $\pm$ 5.81)	46.89 ( $\pm$ 0.91)
Segment Analysis	83.50 ( $\pm$ 4.26)	63.47 ( $\pm$ 0.77)
ADCN	78.83 ( $\pm$ 3.68)	36.32 ( $\pm$ 0.71)
Proposed method	<b>88.58 (<math>\pm</math> 0.82)</b>	<b>70.55 (<math>\pm</math> 0.39)</b>

## 4 Conclusion

In this paper, we proposed a novel method for shape classification based on DPSW on networks. After network modeling, DPSWs are performed on networks and statistical measures of the trajectories are used as a signature. These statistical measurements proved to be powerful for shapes classification. In the classification experiments on biological shapes, the proposed method provided better results when compared to other methods. Furthermore, the proposed method outperformed the other network-based methods.

## Acknowledgments

Lucas Correia Ribas gratefully acknowledges the financial support grant #2016/23763-8, São Paulo Research Foundation (FAPESP). Odemir M. Bruno thanks the financial support of CNPq (Grant # 307797/2014-7) and FAPESP (Grant #s 14/08026-1 and 16/18809-9).

## References

1. Ataer-Cansizoglu, E., Bas, E., Kalpathy-Cramer, J., Sharp, G.C., Erdogmus, D.: Contour-based shape representation using principal curves. *Pattern Recognition* **46**(4), 1140–1150 (2013)
2. Backes, A.R., Bruno, O.M.: Shape skeleton classification using graph and multi-scale fractal dimension. In: *International Conference on Image and Signal Processing*. pp. 448–455. Springer (2010)
3. Backes, A.R., Casanova, D., Bruno, O.M.: A complex network-based approach for boundary shape analysis. *Pattern Recognition* **42**(1), 54–67 (2009)
4. Backes, A.R., Gonçalves, W.N., Martinez, A.S., Bruno, O.M.: Texture analysis and classification using deterministic tourist walk. *Pattern Recognition* **43**(3), 685–694 (2010)
5. Bai, X., Latecki, L.J.: Path similarity skeleton graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(7), 1282–1292

6. Costa, L.d.F.D., Cesar Jr, R.M.: Shape analysis and classification: theory and practice. CRC Press, Inc. (2000)
7. Gonçalves, W.N., Backes, A.R., Martinez, A.S., Bruno, O.M.: Texture descriptor based on partially self-avoiding deterministic walker on networks. *Expert Systems with Applications* **39**(15), 11818–11829 (2012)
8. Gonçalves, W.N., Bruno, O.M.: Dynamic texture segmentation based on deterministic partially self-avoiding walks. *Computer Vision and Image Understanding* **117**(9), 1163–1174 (2013)
9. Gonçalves, W.N., Martinez, A.S., Bruno, O.M.: Complex network classification using partially self-avoiding deterministic walks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **22**(3), 033139 (sep 2012)
10. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. *IEEE Intelligent Systems and their Applications* **13**(4), 18–28 (1998)
11. Junior, J.J.d.M.S., Backes, A.R.: Shape classification using line segment statistics. *Information Sciences* **305**, 349–356 (2015)
12. Liao, S.X., Pawlak, M.: On image analysis by moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(3), 254–266 (1996)
13. Lombarte, A., Chic, Ò., Parisi-Baradad, V., Olivella, R., Piera, J., García-Ladona, E.: A web-based environment for shape analysis of fish otoliths. the aforo database. *Scientia marina* **70**(1), 147–152 (2006)
14. Loncaric, S.: A survey of shape analysis techniques. *Pattern recognition* **31**(8), 983–1001 (1998)
15. Mokhtarian, F., Bober, M.: Curvature scale space representation: theory, applications, and MPEG-7 standardization, vol. 25. Springer Science & Business Media (2013)
16. Osowski, S., et al.: Fourier and wavelet descriptors for shape recognition using neural networks a comparative study. *Pattern Recognition* **35**(9), 1949–1957 (2002)
17. Ribas, L.C., Junior, J.J., Scabini, L.F., Bruno, O.M.: Fusion of complex networks and randomized neural networks for texture analysis. *arXiv preprint arXiv:1806.09170* (2018)
18. Ribas, L.C., Neiva, M.B., Bruno, O.M.: Distance transform network for shape analysis. *Information Sciences* **470**, 28 – 42 (2019)
19. Scabini, L.F., Fistarol, D.O., Cantero, S.V., Gonçalves, W.N., Machado, B.B., Rodrigues Jr, J.F.: Angular descriptors of complex networks: A novel approach for boundary shape analysis. *Expert Systems with Applications* **89**, 362–373 (2017)
20. Torres, R.d.S., Falcao, A.X., Costa, L.d.F.: A graph-based approach for multiscale shape analysis. *Pattern Recognition* **37**(6), 1163–1174 (2004)
21. Wallace, T.P., Wintz, P.A.: An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors. *Computer Graphics and Image Processing* **13**(2), 99–126 (1980)
22. Wu, W.Y., Wang, M.J.J.: Detecting the dominant points by the curvature-based polygonal approximation. *CVGIP: Graphical Models and Image Processing* **55**(2), 79–88 (1993)
23. Zhenjiang, M.: Zernike moment-based image shape analysis and its application. *Pattern Recognition Letters* **21**(2), 169–177 (2000)