



# Dynamic Texture Classification Using Deterministic Partially Self-avoiding Walks on Networks

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

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

lucasribas@usp.br

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

bruno@ifsc.usp.br

**Abstract.** This paper presents a new approach to dynamic texture classification based on deterministic partially self-avoiding (DPS) walks on complex networks (or graphs). In this approach, for each pixel is assigned a vertex and two vertices are connected according to a given distance. In order to analyze appearance and motion, we propose two graph modeling: a spatial graph and a temporal graph. The DPS walks are agents that can obtain rich characteristics of the environment in which they were performed. Thus, the DPS walks are performed in the two modeled graphs (spatial and temporal) and the feature vector is obtained by calculating the statistical measures from the trajectories of the DPS walks. The results in two well-known databases have demonstrated the effectiveness of the proposed approach using a small feature vector. The proposed approach also improved the performance when compared to the previous DPS walks based method and the graph-based method.

**Keywords:** Dynamic texture · Deterministic walks · Network sciences

## 1 Introduction

Dynamic textures can be defined as a sequence of images (or video) that exhibit certain stationary in time [6]. Examples of dynamic textures in the real world include sea waves, smoke, swaying trees, moving flag, fire, a crowd of people, among others. The approaches for dynamic texture representation are applied in different problems such as traffic condition recognition [9], human activity recognition [14], surveillance [29], among others.

In the literature, many approaches have been proposed based on different strategies to analyze the spatial and temporal characteristics of the dynamic textures. These approaches can be separated into five categories: motion-based methods [17], model-based methods [11, 13, 18], filter-based methods [7],

statistical-based methods [24,29] and, agent-based methods [9,10]. The agent-based methods use the deterministic partially self-avoiding (DPS) walks to describe the dynamic textures. These methods achieved promising results in classification, clustering, and segmentation of dynamic textures.

In this paper, we propose a new method for dynamic textures analysis and classification based on deterministic partially self-avoiding walks on complex networks. The DPS walks was introduced initially to investigate the effects of simple walks in random media [15]. After that, the DPS walks methodology was applied for texture and dynamic texture analysis [3,9,10]. Basically, a DPS walk can be understood as an agent who visits points (e.g. pixels, vertices) distributed in a map (e.g. image, video, graph) based on the neighborhood, a rule of movement and memory. Starting from a given point, the next step follows the rule: go to the nearest point on the neighborhood that has not been visited in the last  $\mu$  steps (memory) [15]. Statistical features of the trajectories of the DPS walks are used to study the map.

In the proposed approach, we model the dynamic texture in two graphs (i.e. networks): spatial graph and temporal graph. The spatial graph models the appearance characteristics, while the temporal graph contains the motion properties of the dynamic texture. In this way, we apply the DPS walks on these two graphs and use statistical characteristics of the trajectories to represent the appearance and motion of the dynamic texture in a feature vector. The proposed approach is different of the previous works [9–11] because it combines graph modeling and DPS walks characteristics, while in [9,10] only the DPS walks is applied in the videos and in [11] only the complex network theory is used to modeling and characterization. In Sect. 2 our proposed method to dynamic texture analysis is detailed. The experimental setup is described in Sect. 3. In Sect. 4 the experimental results are presented and discussed. Finally, the paper is concluded in Sect. 5.

## 2 Proposed Approach

### 2.1 Network Modeling

The network sciences (also called complex network) field uses the formalism of graph theory with the incorporation of statistical mechanics and complex systems. It has attracted increased attention because of its ability to represent and study a wide range of systems and data. In computer vision, the networks have been used to model and analyze images and video analysis [2,21–23]. In this paper, we use the graph to represent the dynamic texture video. In dynamic texture analysis, it is important to obtain appearance and motion features in order to accurately represent the video. To achieve this, in the proposed approach, we model the dynamic texture video in two graphs (networks): the spatial graph  $G_S = (V_S, E_S)$  that characterizes the appearance properties and the temporal graph  $G_T = (V_T, E_T)$  that contains the motion properties.

In the two graphs, each pixel  $i = (x_i, y_i, t_i)$  is mapped into a vertex  $i \in V$ , where  $x_i$  and  $y_i$  are the spatial coordinates and  $t_i$  the temporal coordinate of

the pixel  $i$ . The main difference between the two graphs is the definition of the set of edges. In the spatial graph, the set  $E_S$  is defined by the connection of all vertices whose the Euclidean distance is smaller or equal than a given radius  $\sqrt{2}$  and the time coordinates  $t_i$  and  $t_j$  are equal,

$$e_{ij} \in E_S \iff \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq \sqrt{2} \text{ and } t_i = t_j \quad (1)$$

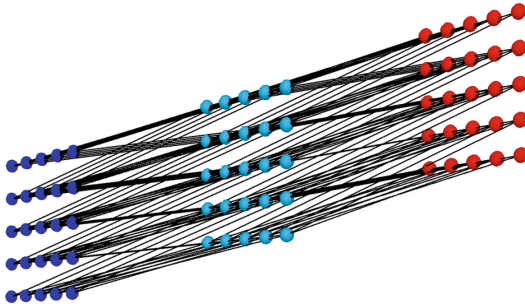
On the other hand, in the temporal graph, the set of edges is defined by connecting the vertices whose the Euclidean distance is smaller or equal than  $\sqrt{3}$  and the time coordinates are different,

$$e_{ij} \in E_T \iff \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2} \leq \sqrt{3} \text{ and } t_i \neq t_j \quad (2)$$

Figure 1 illustrates three frames modeled as a graph. The frames are represented by the spheres in blue, green and red. For each edge  $e_{ij}$  connecting two vertices  $i$  and  $j$ , a weight  $w(e_{ij})$  is defined by the difference of intensities between the two pixels that represent the vertices:

$$w(e_{ij}) = \frac{|I(i) - I(j)|}{255}, \quad (3)$$

where  $I(i) \in [0, 255]$  is the gray intensity of a pixel  $i$ .



**Fig. 1.** Three frames modeled as a temporal graph. The edges connecting only vertices of different frames. (Color figure online)

## 2.2 DPS Walks on Networks

The deterministic partially self-avoiding (DPS) walk is an agent, which was initially used to study regular and random media [15]. This deterministic walk produces a set of trajectories that are used to characterize the environment in which they were performed. The DPS walk was applied with success for feature extraction in different classification tasks, such as in texture analysis [3], dynamic texture classification [9, 10], shape analysis [20] and complex network classification [12].

In the proposed approach, the DPS walks are used for feature extraction of the graphs that model the dynamic texture videos. In this way, the DPS walks are performed on the vertices. The DPS walk is an agent that walks on the vertices of the graph based on a deterministic rule  $r$ . The agent starts the walk from a pre-defined vertex  $i$  and the movement to the next vertex  $j$  is given by: go to the vertex  $j$  in the neighborhood  $\eta(i)$  (vertices connected to the vertex  $i$ ) which minimizes the edge weight  $w(e_{ij})$  and that has not been visited in the previous  $\mu$  steps (i.e. that is not in memory  $j \notin M_\mu$ ). Here, we will call this rule of movement as  $r = \min$ . We also consider another rule of movement that moves the agent in the direction of the maximum edge weight  $w(e_{ij})$  ( $r = \max$ ). The two rules of movement are used because each one produces different trajectories and, consequently, obtain different characteristics of the graph. The walk will end when the agent to find a set of vertices in which it cannot escape, called attractor.

The memory  $M_\mu$  of size  $\mu$  is the last  $\mu$  vertices visited by the agent and that cannot be visited. This memory is updated in each step of the agent to save the last  $\mu$  vertices visited. The trajectory of the agent can be divided into two parts: an initial part of size  $\tau$  called transient, and, a final part named attractor, which is composed of vertices that form a cycle of period  $\rho \geq \mu + 1$  where the agent gets stuck. In the cases in which the agent cannot find an attractor, the trajectory is represented only by the transient part. For each vertex of the graph, a DPS walk is started with a given memory size  $\mu$  and a rule of movement  $r$ . Therefore, for a graph with  $N$  vertices, we have  $N$  different trajectories. In order to measure this set of trajectories, the transient time and attractor period joint distribution  $S_{\mu,r}(\tau, \rho)$  is considered. In this distribution, the frequency of trajectories with transient  $\tau$  and attractor  $\rho$  is stored in each position [3],

$$S_{\mu,r}(\tau, \rho) = \frac{1}{N} \sum_{i \in V} \begin{cases} 1, & \text{if } \tau_i = \tau \text{ and } \rho_i = \rho \\ 0, & \text{otherwise} \end{cases}, \quad (4)$$

where  $\mu$  is the memory size and  $r$  the rule of movement used.

### 2.3 Proposed Signature

The joint distribution contains relevant information about the trajectories of the DPS walks performed in a given graph. Thus, previous works [3] have used features obtained from this joint distribution for characterization. In this way, the best results were obtained using the histogram  $h_{\mu,r}^t(l)$ , which calculates the number of trajectories with size  $l = \tau + \rho$  in a joint distribution computed with memory size  $\mu$  and rule of movement  $r$ ,

$$h_{\mu,r}(l) = \sum_{b=0}^{l-1} S_{\mu,r}^k(b, l-b). \quad (5)$$

In order to characterize the dynamic texture videos, the DPS walks are performed in the two graphs:  $G_S$  and  $G_T$ . Thus, for each graph a histogram  $h_{\mu,r}(l)$  can be obtained. Several previous works have shown that the most discriminative information of the histogram  $h_{\mu,r}(l)$  are concentrated on its first elements [3, 9]. In this work, we use the  $n$  first descriptors of the histogram  $h_{\mu,r}(l)$ , with the first position defined as  $(\mu + 1)$ , since there is no attractor smaller than  $(\mu + 1)$ . Thus, given a memory size  $\mu$  and a rule of movement  $r$ , a feature vector  $\nu_u^\Theta$  is obtained:

$$\nu_{\mu,r}^\Theta = [h_{\mu,r}^\Theta(\mu + 1), h_{\mu,r}^\Theta(\mu + 2), \dots, h_{\mu,r}^\Theta(\mu + n)] \quad (6)$$

where  $\Theta$  is the type of graph: spatial  $S$  or temporal  $T$ .

The size of the memory directly influences the complexity of the trajectories and, consequently, in the information extracted by the method. For example, DPS walks with low memory values perform better local analysis [10]. In this sense, histograms obtained with different memory sizes are used for a more robust characterization of the different patterns present in the graphs (i.e. dynamic texture videos), according to:

$$\vartheta_r^\Theta = [\nu_{\mu_1,r}^\Theta, \nu_{\mu_2,r}^\Theta, \dots, \nu_{\mu_m,r}^\Theta]. \quad (7)$$

To characterize patterns of appearance and movement of dynamic textures, a feature vector that consists of the concatenation of the spatial and temporal descriptors is considered. Thus, this feature vector is composed of the characteristics extracted from the spatial  $\vartheta_r^S$  and temporal  $\vartheta_r^T$  graphs using different memory values, as described:

$$\lambda_r = [\vartheta_r^S, \vartheta_r^T]. \quad (8)$$

The feature vector obtained above refers to a single rule of movement. Although this vector may be able to properly characterize dynamic textures, another possibility is to combine the two rules of movement. The rule of movement is another parameter that influences the trajectory of the agent. In this sense, it is considered a final feature vector that consists of the concatenation of vectors obtained with the two rules of movement  $r = min$  and  $r = max$ , as follows:

$$\lambda = [\lambda_{min}, \lambda_{max}] \quad (9)$$

## 2.4 Computational Complexity

Basically, the proposed approach models a dynamic texture with  $N = w \times h \times T$  pixels in two graphs. For modeling, each pixel is mapped into a vertex, which is linked with 8 and 18 neighbors for the spatial and temporal graphs, respectively. As the number of neighbors is a multiplicative constant and much smaller than the number of pixels in the video, it can be disregarded. Thus, the computational complexity of the modeling is given then by  $O(N)$  for each type of graph. Next, a DPS walk is started from each pixel, producing a trajectory of size  $l = \tau + \rho$ , where  $\tau$  is the transient time and  $\rho$  is the attractor period. For cases where

an attractor is not found, we finish the walk after  $M$  steps. In this work, we define  $M = 20$  because we use only the first elements of the joint distribution for characterization. Therefore, the computational complexity to compute the features from each graph (temporal and spatial) is given by the number of pixels and the size of the trajectories,  $O(N \times (\tau + 2\rho))$ . We have  $2\rho$  because it is necessary to go through twice the same cycle of pixels to identify an attractor.

### 3 Experimental Setup

To classify the feature vectors, we adopted the 1-Nearest Neighbor (1NN) classifier and a specific scheme for each database to separate the training and test set. The Dyntex++ [8] database is a compiled version of the Dyntex database [16]. The samples are preprocessed in order to eliminate static or dynamic non-representative backgrounds, zoom, and textures without movement. The database has 3600 samples divided into 36 classes (e.g. boiling water, river water, colony of ants and smoke). In the experiments, a 10-fold cross-validation scheme with 10 trials was used [11]. The accuracy is reported as the average performance of all experimental trials.

The UCLA [4] database is composed of 200 dynamic texture videos separated into 50 classes with 4 samples per class (named UCLA-50 version). Each sample has  $48 \times 48 \times 75$  pixels. This database also has two variations of the original database proposed in [19]. On the UCLA-9 version, the samples are reorganized into 9 classes: boiling water (8 samples), fire (8), flower (12), fountains (20), plants (108), sea (12), smoke (4), water (12) and waterfall (16). In the UCLA-8 version, the plant class is eliminated due to the large number of samples. The experimental setup adopted in these databases is similar to [19]. For the UCLA-50 is used a 4-fold cross-validation scheme with 10 repetitions. On the other versions, it is used for the testing set, half of the sequences (randomly selected from each class), and the remaining half is used for training. For these databases, the correct classification rate (CCR) or accuracy is reported as the average performance of all experimental trials.

### 4 Results and Discussion

First, we investigate the effects of the parameters of our proposed approach in the task of dynamic texture classification. The parameters analyzed were: (i) memory sizes  $\mu$  and (ii) rules of movement  $r$ . In the experiments, it was used the first  $n = 3$  elements of the histogram  $h_{\mu,r}^{\mathcal{O}}(l)$  for the UCLA databases and the first  $n = 5$  positions for the Dyntex++ database. These values were defined based on the idea that the main information are in the first elements and from the experimental tests.

Figure 2 shows the results of our proposed method on the two databases for different combinations of memory sizes and rules of movement. On both databases, the rule of movement  $r = max$  obtained higher accuracies than the

rule of movement  $r = min$ . The rule of movement  $r = max$  is related to heterogeneous regions of the video (i.e. graph), that is, the agent walks on regions where the difference of the gray level between the pixels (i.e. high edge weight) is higher. On the other hand, in the rule  $r = min$ , the agent walks on homogeneous regions, that is, where the edge weight is smaller. This indicates that heterogeneous regions have more discriminative information of the dynamic texture. However, the best results are obtained when both rules of movement are combined.

Concerning the memory sizes, we note that low memory sizes provide inferior accuracies. Thus, as we increase the memory sizes, the accuracy also is increased. However, when using a combination of memory size higher than  $[0, 1, 2, 3]$ , the accuracy obtained starts to stabilize, suggesting that the proposed descriptors have reached their limits in terms of discrimination ability. Such behavior is expected: the larger the memory sizes  $\mu$ , the harder find an attractor. From the results, we set up as default parameters of the proposed method  $\mu = [0, 1, 2, 3, 4, 5, 6, 7, 8]$  and  $r = [min, max]$ . On both databases, the highest accuracy was using this configuration (94.5% and 96.0% for the Dyntex++ and UCLA-50, respectively). These results are interesting because they indicate that the method is not strongly parameter dependent.

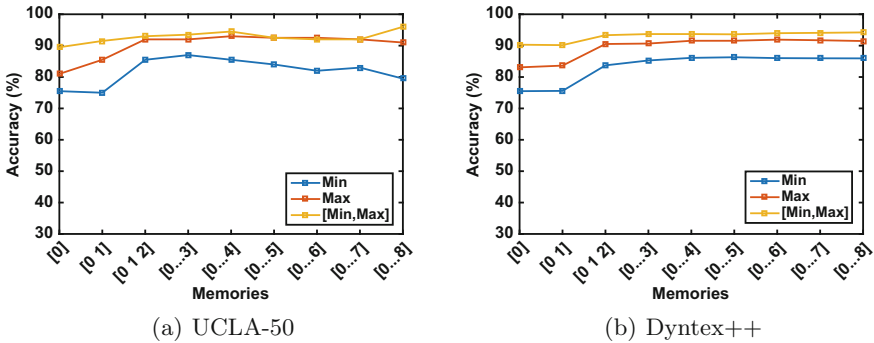


Fig. 2. Accuracies using different combination of memory sizes and rules of movement.

In order to improve the analysis of our proposed approach, we performed a comparison experiment using literature methods of dynamic textures. To achieve this, we considered the accuracy, standard deviation and number of features of the methods, when described in the original papers. In all comparison, we use the same experimental setup described in Sect. 3.

Table 1 presents the classification results of the proposed method and others on UCLA-50 database. Note that the proposed method obtained the best accuracy when compared to the others. Concerning the complex network based methods, the proposed method improves the accuracy compared to the CNDT [11] method by 1.0%. This method uses traditional complex networks measures

while our method uses the DPS walk for complex network characterization. Thus, the results suggest that the DPS walk is more effective to describe the graph and, consequently, the dynamic texture.

**Table 1.** Comparison of the classification results of the proposed method and others on UCLA-50 database.

Methods	Number of features	Accuracy (%)
KDT-MD [5]	-	89.50
DFS [27]	-	89.50
3D-OTF [28]	290	87.10
CVLBP [24]	-	93.00
RI-VLBP [30]	16384	77.50 ( $\pm$ 8.98)
LBP-TOP [29]	768	95.00 ( $\pm$ 4.44)
DPSW-TOP [9]	75	95.00 ( $\pm$ 4.78)
CNDT [11]	420	95.00 ( $\pm$ 5.19)
Proposed method	180	96.00 ( $\pm$ 3.16)

Table 2 summarizes the results on the UCLA-9 database. On this database, our approach yields the second best result (96.80%). This result is slightly inferior to the one obtained by CVLBP method (96.90%). On the other hand, on the UCLA-8 database, the proposed method achieved the best accuracy, as can be seen in Table 3. Here, the proposed method gives an accuracy of 96.59% against 95.65% of the CVLBP method. The proposed method also outperformed the method DPSW-TOP, which is a DPS walk based method. This method applies the DPS walk on three orthogonal planes to analyze the appearance and motion properties of the dynamic textures. In this way, the results indicate that our approach based on DPS walk applied on the graph is more effective for dynamic texture characterization.

Table 4 presents the results on the Dyntax++ database for different methods. The proposed method shows an improvement of 10.74% and 3.21% compared to the CNDT and DPSW-TOP methods, respectively. However, on this database, the proposed method obtained a performance lower than the RI-VLBP and LBP-TOP methods. Nevertheless, it is important to emphasize that the feature vector size of these methods is significantly higher than the feature vector of our method. Therefore, our method is still competitive due to the small feature vector, for example, the RI-VLBP extracts a long feature vector of dimension 16384, whereas our method produces only 180 characteristics.

Besides these compared methods, called hand-craft methods, we also compare our proposed signature with a method based on learned features. This method proposed in [1] uses a convolutional neural network (GoogleNet) to learn the characteristics of the dynamic textures in three orthogonal planes and obtain a signature. On the UCLA-50, UCLA-9 and UCLA-8 databases, the CNN-based



**Table 2.** Classification results for all methods on the UCLA-9 database.

Methods	Number of features	Accuracy (%)
3D-OTF [28]	290	96.32
CVLBP [24]	-	96.90
High level feature [25]	-	92.60
Chaotic vector [26]	300	85.10
RI-VLBP [30]	16384	96.30
LBP-TOP [29]	768	96.00
DPSW-TOP [9]	75	96.33 ( $\pm 2.46$ )
CNDT [11]	336	95.61 ( $\pm 2.72$ )
Proposed method	180	96.80 ( $\pm 2.36$ )

**Table 3.** Comparison results on the UCLA-8 database.

Methods	Number of features	Accuracy (%)
3D-OTF [28]	290	95.80
CVLBP [24]	-	95.65
High level feature [25]	-	85.65
Chaotic vector [26]	300	85.00
RI-VLBP [30]	16384	91.96
LBP-TOP [29]	768	93.67
DPSW-TOP [9]	75	93.41 ( $\pm 6.01$ )
CNDT [11]	336	94.32 ( $\pm 4.18$ )
Proposed method	180	96.59 ( $\pm 7.12$ )

**Table 4.** Comparison results for different dynamic texture methods on the Dyntex++ database.

Methods	Number of features	Accuracy (%)
RI-VLBP [30]	16384	96.14 ( $\pm 0.77$ )
LBP-TOP [29]	768	97.72 ( $\pm 0.43$ )
DPSW-TOP [9]	75	91.39 ( $\pm 1.29$ )
CNDT [11]	336	83.86 ( $\pm 1.40$ )
Proposed method	180	94.60 ( $\pm 1.20$ )

method obtained 99.50%, 98.35% and 99.02% of accuracy, respectively. These results are higher than the obtained by our method. However, it is important to highlight that even with inferior results, our method is still competitive due to its computational simplicity.

## 5 Conclusion

This paper presents a new method for characterization and classification of dynamic textures using deterministic partially self-avoiding walks on complex networks. In this method, we have shown a graph modeling from dynamic texture videos, which allows us to analyze appearance (spatial graph) and motion (temporal graph) properties. Thus, the DPS walks are performed on these two graphs and the statistical information of its trajectories are used to compose a feature vector. Experimental results obtained on the UCLA and Dyntex++ databases showed that our method is very competitive when compared to other methods. Our method also outperformed the other previous DPSW-based and complex network based methods. In addition, the proposed approach is competitive in terms of dimensionality, producing feature vectors significantly smaller than other literature methods. In this way, the tradeoff between performance and feature vector size demonstrates the great potential of the proposed method for dynamic texture classification.

**Acknowledgments.** Lucas C. Ribas gratefully acknowledges the financial support grant #2016/23763-8 and #2019/03277-0, São Paulo Research Foundation (FAPESP). Odemir M. Bruno thanks the financial support of CNPq (Grant #307797/2014-7) and FAPESP (Grant #14/08026-1).

## References

1. Andrearczyk, V., Whelan, P.F.: Convolutional neural network on three orthogonal planes for dynamic texture classification. *Pattern Recogn.* **76**, 36–49 (2018)
2. Backes, A.R., Casanova, D., Bruno, O.M.: Texture analysis and classification: a complex network-based approach. *Inf. Sci.* **219**, 168–180 (2013)
3. Backes, A.R., Gonçalves, W.N., Martinez, A.S., Bruno, O.M.: Texture analysis and classification using deterministic tourist walk. *Pattern Recogn.* **43**(3), 685–694 (2010)
4. Chan, A.B., Vasconcelos, N.: Probabilistics for the classification of auto-regressive visual processes. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 1, pp. 846–851. IEEE (2005)
5. Chan, A.B., Vasconcelos, N.: Classifying video with kernel dynamic textures. In: *CVPR*, pp. 1–6 (2007)
6. Doretto, G., Jones, E., Soatto, S.: Spatially homogeneous dynamic textures. In: Pajdla, T., Matas, J. (eds.) *ECCV 2004*. LNCS, vol. 3022, pp. 591–602. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-24671-8\\_47](https://doi.org/10.1007/978-3-540-24671-8_47)
7. Dubois, S., Péteri, R., Ménard, M.: A comparison of wavelet based spatio-temporal decomposition methods for dynamic texture recognition. In: Araujo, H., Mendonça, A.M., Pinho, A.J., Torres, M.I. (eds.) *IbPRIA 2009*. LNCS, vol. 5524, pp. 314–321. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-02172-5\\_41](https://doi.org/10.1007/978-3-642-02172-5_41)
8. Ghanem, B., Ahuja, N.: Maximum margin distance learning for dynamic texture recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010*. LNCS, vol. 6312, pp. 223–236. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-15552-9\\_17](https://doi.org/10.1007/978-3-642-15552-9_17)

9. Gonçalves, W.N., Bruno, O.M.: Dynamic texture analysis and segmentation using deterministic partially self-avoiding walks. *Expert Syst. Appl.* **40**(11), 4283–4300 (2013)
10. Gonçalves, W.N., Bruno, O.M.: Dynamic texture segmentation based on deterministic partially self-avoiding walks. *Comput. Vis. Image Underst.* **117**(9), 1163–1174 (2013)
11. Gonçalves, W.N., Machado, B.B., Bruno, O.M.: A complex network approach for dynamic texture recognition. *Neurocomputing* **153**, 211–220 (2015)
12. Gonçalves, W.N., Martinez, A.S., Bruno, O.M.: Complex network classification using partially self-avoiding deterministic walks. *Chaos: Interdisc. J. Nonlinear Sci.* **22**(3), 033139 (2012)
13. Junior, J.J.d.M.S., Ribas, L.C., Bruno, O.M.: Randomized neural network based signature for dynamic texture classification. *Expert Systems with Applications* (2019)
14. Kellokumpu, V., Zhao, G., Pietikäinen, M.: Human activity recognition using a dynamic texture based method. In: *BMVC*, vol. 1, p. 2 (2008)
15. Lima, G.F., Martinez, A.S., Kinouchi, O.: Deterministic walks in random media. *Phys. Rev. Lett.* **87**, 010603 (2001)
16. Péteri, R., Fazekas, S., Huiskes, M.J.: Dyntex: a comprehensive database of dynamic textures. *Pattern Recogn. Lett.* **31**(12), 1627–1632 (2010)
17. Polana, R., Nelson, R.: Temporal texture and activity recognition. In: Shah, M., Jain, R. (eds.) *Motion-Based Recognition*, pp. 87–124. Springer, Dordrecht (1997). [https://doi.org/10.1007/978-94-015-8935-2\\_5](https://doi.org/10.1007/978-94-015-8935-2_5)
18. Qiao, Y., Weng, L.: Hidden markov model based dynamic texture classification. *IEEE Sig. Process. Lett.* **22**(4), 509–512 (2015)
19. Ravichandran, A., Chaudhry, R., Vidal, R.: View-invariant dynamic texture recognition using a bag of dynamical systems. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1651–1657, June 2009
20. Ribas, L.C., Bruno, O.M.: Deterministic partially self-avoiding walks on networks for natural shapes classification (2019)
21. Ribas, L.C., Gonçalves, W.N., Bruno, O.M.: Dynamic texture analysis with diffusion in networks. *Digit. Signal Proc.* **92**, 109–126 (2019)
22. Ribas, L.C., de Sá Junior, J.J.M., Scabini, L.F.S., Bruno, O.M.: Fusion of complex networks and randomized neural networks for texture analysis. *CoRR abs/1806.09170* (2018). <http://arxiv.org/abs/1806.09170>
23. Ribas, L.C., Neiva, M.B., Bruno, O.M.: Distance transform network for shape analysis. *Inf. Sci.* **470**, 28–42 (2019)
24. Tiwari, D., Tyagi, V.: Dynamic texture recognition based on completed volume local binary pattern. *Multidimension. Syst. Signal Process.* **27**(2), 563–575 (2016)
25. Wang, Y., Hu, S.: Exploiting high level feature for dynamic textures recognition. *Neurocomputing* **154**, 217–224 (2015)
26. Wang, Y., Hu, S.: Chaotic features for dynamic textures recognition. *Soft. Comput.* **20**(5), 1977–1989 (2016)
27. Xu, Y., Quan, Y., Ling, H., Ji, H.: Dynamic texture classification using dynamic fractal analysis. In: *2011 International Conference on Computer Vision*, pp. 1219–1226, November 2011
28. Xu, Y., Huang, S., Ji, H., Fermüller, C.: Scale-space texture description on sift-like textons. *Comput. Vis. Image Underst.* **116**(9), 999–1013 (2012)

29. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(6), 915–928 (2007)
30. Zhao, G., Pietikainen, M.: Dynamic texture recognition using volume local binary patterns. In: Vidal, R., Heyden, A., Ma, Y. (eds.) *WDV 2005-2006*. LNCS, vol. 4358, pp. 165–177. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-70932-9\\_13](https://doi.org/10.1007/978-3-540-70932-9_13)