Dynamic texture analysis using networks generated by deterministic partially self-avoiding walks

Lucas C. Ribas[a,b], Odemir M. Bruno[b,a,*]

[a] Institute of Mathematics and Computer Science, University of São Paulo, USP, Avenida Trabalhador são-carlense, 400, 13566-590 São Carlos, SP, Brazil
[b] Scientific Computing Group, São Carlos Institute of Physics, University of São Paulo, USP, PO Box 369, 13560-970, São Carlos, SP, Brazil

Abstract

Dynamic textures are sequences of images (video) with the concept of texture patterns extended to the spatiotemporal domain. This research field has attracted attention due to the range of applications in different areas of science and the emergence of a large number of multimedia datasets. Unlike the static textures, the methods for dynamic texture analysis also need to deal with the time domain, which increases the challenge for representation. Thus, it is important to obtain features that properly describe the appearance and motion properties of the dynamic texture. In this paper, we propose a new method for dynamic texture analysis based on Deterministic Partially Self-avoiding Walks (DPSWs) and network science theory. Here, each pixel of the video is considered a vertex of the network and the edges are given by the movements of the deterministic walk between the pixels. The feature vector is obtained by calculating network measures from the networks generated by the DPSWs. The modeled networks incorporate important characteristics of the DPSWs transitivity and their spatial arrangement in the video. In this paper, two different strategies are tested to apply the DPSWs in the video and capture appearance and motion characteristics. One strategy applies the DPSWs in three orthogonal planes and the other is based on spatial and temporal descriptors. We validate our proposed method in Dyntex++ and UCLA (and their variants) databases, which are two well-known dynamic texture databases. The results have demonstrated the effectiveness of the proposed approach using a small feature vector for both strategies. Also, the proposed method improved the performance when compared to the previous DPSW-based method and the network-based method.

Keywords: Pattern recognition, Dynamic texture analysis, Network Science, Deterministic walkers

[*] Corresponding author
Email address: obruno@usp.br (Odemir M. Bruno)

1. Introduction

1.1. Contextualization and Motivation

Dynamic textures are sequences of images with texture patterns in motion that exhibit certain stationary in time [1, 2]. Polana et al [3] defined that dynamic textures have statistical regularity, but with a space-time extension characterized by non-rigid complex motion. This occurs because complex motions are driven by non-linear dynamical systems with certain chaos [4]. Examples of dynamic textures in the real world include sea waves, smoke, swaying trees, moving flag, fire, crowd of people, among others.

The motivation for the characterization of dynamic texture video is a great potential for application in different fields of science and problems. In addition, the increase in the number of multimedia databases has motivated the investigation and development of approaches for video characterization. Concerning the applications, the methods for dynamic textures characterization are important tools. For example, in the Ref. [5] dynamic textures approaches were used to forest fire detection. On the other hand, in [6] the authors proposed an approach for automatic segmentation of liver in ultrasound based on a dynamic textures approach. In facial recognition, the research using dynamic texture descriptors has obtained promising results for problems such as biometric systems and human-computer interaction [7]. Besides that, other important applications may be cited such as traffic condition recognition [8], human activity recognition [9], surveillance [10], video retrieval system [11], among others.

1.2. Related works

The challenges in dynamic textures analysis are related to simultaneous combination of spatial and temporal characteristics. In the last years, many approaches have been proposed based on different ways to analyze the properties of the dynamic textures. These approaches can be divided into six categories: motion-based methods, model-based methods, filter-based methods, statistical-based methods and, agent-based methods. Motion-based methods are the most popular due to the low computational cost and its efficiency in the motion analysis, such as optical flow [3, 12]. The methods from the model-based category represent the dynamic textures through mathematical models, such as dynamic linear system [13], neural networks [14] and hidden markov model [15]. Gonçalves et al [16, 17] also proposed a complex network model for modeling and characterizing of appearance and temporal features from dynamic textures. The filter-based methods characterize the dynamic texture by means of decomposition at different scales using spatio-temporal filtering. In this category, Wildes and Bergen [18] analyzed local space-time patterns, their orientation and energy, through energy-oriented filters. Other authors used wavelet transform [19, 20, 21] and Gabor filters [22]. In the statistical-based methods category, the approaches use statistical properties of the gray level distribution and calculate local characteristics of each pixel [23]. Most of the methods of this category are adapted from static textures approaches, such as the Gray Level Co-occurrence matrix (GLCM) [24] and Local Binary Pattern (LBP) [25].

Among the several methods of this category, the most known are LBP-TOP [26], VLBP [10], CVLBP [27], HLBP [28], WLBPC [29] and MEWLSP [30]. In the last category, the methods use agents to extract spatial and motion information from the dynamic textures [31, 8]. These methods use Deterministic Partially Self-avoiding Walks (DPSWs) applied in three orthogonal planes to describe the dynamic textures. The obtained results showed the potential of this approach in classification, clustering, and segmentation of dynamic textures.

1.3. Contributions of this paper

In this paper, we propose a new method for dynamic textures analysis based on networks generated by deterministic partially self-avoiding walks. In the previous work [16] based on networks, the dynamic texture is modeled as a network and degree measurements are used to characterize the dynamic textures. However, in network modeling, which is a challenging problem in network sciences theory, the method uses four parameters to represent the dynamic texture as a network. Therefore, a new and more effective network modeling for dynamic textures is needed.

On the other hand, the DPSWs was introduced initially in physics to investigate and analyze the effect of simple walks in regular and random media[32]. Then, the DPSW was applied on image and video analysis obtaining promising results [33, 34, 35, 31, 8]. The DPSW can be interpreted as a tourist or agent who wants to visit pixels distributed on an image or video. The agent starts the walk from a pre-defined pixel and follows the rule of movement: go to the nearest pixel (the distance is the difference of gray levels) that has not been visited in the last μ steps (memory) [32]. A walk is started from each pixel of the video and the trajectory produced by the DPSW is composed of two parts: the transient and the attractor. The transient is the initial part and the attractor is following pixels that form a cycle where the agent gets stuck. Despite the promising results achieved by the previous works in dynamic textures [31, 8], the authors consider only a joint probability distribution of transient size and attractor size to describe the dynamic textures.

This paper introduces a new way based on networks to describe the DPSW trajectories, which certainly obtain more information than only the sizes of transient and attractor used in the previous works. To achieve this, we consider two different strategies (three orthogonal planes and, spatial and temporal descriptors) to perform the DPSWs in the videos and thus obtain appearance and motion characteristics. From a given strategy, networks are built considering each pixel as a vertex and the edges are given by the movements of the agent between the pixels. A movement of the agent between two pixels i and j represents an edge connecting the corresponding vertices v_i and v_j. Once obtained the networks, its topological characteristics are used to compose the feature vector. In this approach, instead of using only the size and frequency of trajectories, we consider their distribution and spatial arrangement in the video. Therefore, our approach has as advantages: the ability to describe the trajectories with more discriminative information and, a new and simple network modeling for dynamic textures that uses only two parameters (rule of movement and memory size).

This paper starts presenting an overview about the deterministic partially self-avoiding walk and network sciences in Section 2. Section 3 details our proposed method to dynamic texture analysis. The experimental setup is described in Section 4. In Section 5 the experimental results Section 6.

2. Background

2.1. Deterministic Partially Self-avoiding Walks

Deterministic partially self-avoiding walks emerged as an approach to study regular and random media, obtaining promising results [32]. These deterministic walkers are agents capable of obtaining rich information about the environment in which they are performed. Posteriorly, the DPSWs was introduced in different computer vision tasks, such as static texture analysis [36, 33, 34, 35] and, dynamic texture classification and segmentation [31, 8]. Also, DPSWs have been generalized to complex network classification [37].

In this section, we describe the theory of the DPSWs for image and in Section 3, we present the proposed method. Thus, consider a two-dimensional image composed of a finite set of pixels P, where each pixel has a mapping $i = (x_i, y_i)$, and an gray intensity $I(i) \in [0255]$. The neighborhood $\eta(i)$ of a pixel i is composed of pixels j whose the Euclidean distance of i is smaller or equal than $\sqrt{2}$, according to

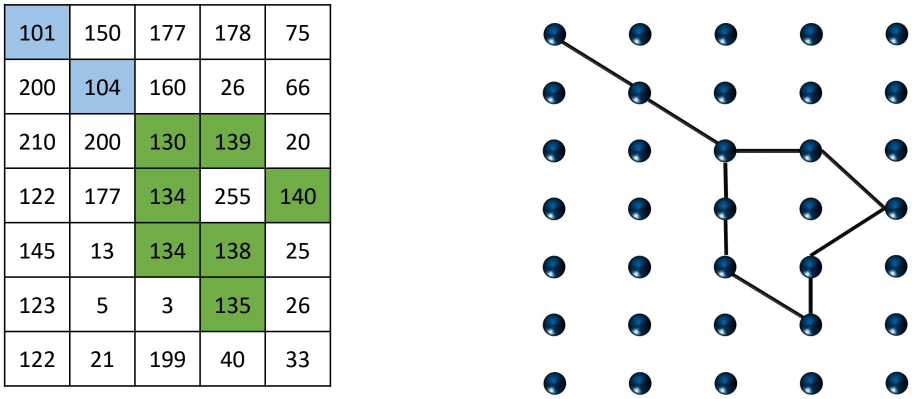
$$\eta(i) = \{j \mid \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq \sqrt{2}\}. \tag{1}$$

The distance between two neighboring pixels i and j is given by the modulus of the difference between its gray intensities:

$$\omega(i,j) =\mid I(i) - I(j) \mid . \tag{2}$$

From the above definitions, the DPSWs are performed on the pixels P. The DPSW is a walker or agent that walks on the pixels according to a deterministic rule. The agent starts the walk from a pre-defined pixel i and the rule of movement r to move to the next pixel j is: go to the pixel j in neighborhood $\eta(i)$ which minimizes the distance ω and that has not been visited in the previous μ steps (i.e. that is not in memory $j \notin M_\mu$). This rule of movement will be named as $r = min$. In addition, another rule of movement that moves the agent in the direction of the maximum distance ω is considered ($r = max$). Each rule of movement guide the agent to different trajectories, thus different characteristics of the image are analyzed. The walk ends when the agent finds an attractor, that is, a set of pixels in which it can not escape.

The memory M_μ of size μ is a set of pixels that the agent cannot visit. This memory is updated in each step of the agent in order to save the last μ pixels visited. The agent produces a trajectory that can be divided into two parts: transient and attractor. The transient is an initial part of size τ and, the attractor is the final part, which consists of pixels that form a cycle of period $\rho \geq \mu + 1$ where the agent gets stuck. In some cases, due to the layout of neighboring pixels and the memory size μ, the agent cannot find an attractor. In this case, the trajectory is considered only by the transient part. For each pixel of the image, a DPSW is started with a given memory size μ and a rule of movement r, producing different trajectories. Thus, on an image with N pixels, we obtain N trajectories. Figure 1(a) illustrates a DPSW trajectory on an image using memory size $\mu = 5$ and rule of movement $r = min$. The blue squares represent the transient and the green squares the attractor.



101	150	177	178	75
200	104	160	26	66
210	200	130	139	20
122	177	134	255	140
145	13	134	138	25
123	5	3	135	26
122	21	199	40	33

(a) Image

(b) Network generated by a walk

Figure 1: Example of a DPSW with memory $\mu = 5$ and rule of movement $r = min$ performed in a gray-scale image and its network generated.

2.2. Networks

Network sciences or complex networks is a research field that uses the formalism of graph theory, but with the incorporation of methods derived from statistical mechanics and complex systems. In the last decades, the studies of network sciences have become useful due to its ability to represent and analyze a wide variety of systems in different areas, such as social networks [38], biological [39], physics [40], technological [41], among others. Besides that, the network science theory has also been successfully applied in different computer vision tasks, such as texture analysis [42] and shape classification [43].

Formally, a network is represented by a pair $G = (V, E)$, where $V = (v_1, v_2, ..., v_N)$ is a set of N vertices and $E = (e_1, e_2, ..., e_M)$ is a set of M edges. The network may be weighted, in this case, the edges have weights $w(e_{v,v'})$, which are associated with some measure of the network, such as the distance of roads connecting cities.

For network topological analysis, measures based on the degree of the vertices are the most used. The degree k of a vertex v is the number of connections incident on it, according to

$$k_v = \sum_{v' \in V} \begin{cases} 1, & e_{v,v'} \in E \\ 0, & e_{v,v'} \notin E. \end{cases} \tag{3}$$

On the weighted network, we also have the weighted degree, which is the sum of the edge weights and is given by

$$k_v^w = \sum_{v' \in V} \begin{cases} w(e_{v,v'}), & e_{v,v'} \in E \\ 0, & e_{v,v'} \notin E. \end{cases} \tag{4}$$

Using the degree of each network vertex, the degree histogram h is obtained as

$$h(k) = \sum_{v \in V} \begin{cases} 1, & k_v = k \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The histogram is a simple and powerful representation of the statistical properties of the network topology. Using the degree histogram, a probability density function can be computed. This one indicates the probability of occurrence $P(k)$ of a given vertex v has degree k and is given by

$$P(k) = \frac{h(k)}{\sum_{k=0}^{K} h(k)}, k = 0, 1, ..., K \tag{6}$$

where K is the maximum degree of the network. By means of the probability function $P(k)$, various first-order statistics can be calculated. In order to characterize the network topology, we compute the following measures from the probability function: the energy E, entropy H and contrast C. These measures are described as

$$E = \sum_{k}^{K} P(k)^2 \tag{7}$$

$$H = \sum_{k}^{K} P(k) \log P(k) \tag{8}$$

$$C = \sum_{k}^{K} P(k) k^2. \tag{9}$$

Another vertex measure considered is the joint degree. This is a correlation measure between the degree of two connected vertices. Thus, the joint degree $\widehat{k}_v$ of a vertex v is the probability of the vertex v and a neighboring vertex v' have the same degree:

$$\widehat{k}_v = \frac{c(v)}{k_v} \tag{10}$$

where $c(v) = |\{v' \in V | e_{v,v'} \in E \land k_v = k_{v'}\}|$ is the number of connected vertices to the vertex v with the same degree k_v and $|.|$ is the cardinality of the set.

3. Proposed Method

In this section, we present the proposed method for dynamic textures analysis combining DPSWs and complex networks. The proposed method may be described in three main steps: (i) based on a strategy, DPSWs are performed in the video; (ii) the trajectories of the DPSWs are modeled as networks; (iii) a feature vector is built using the topological characteristics of the modeled networks. These three steps are illustrated in Figure 2. The main differences of the proposed approach for previous works based on DPSWs [31] and complex networks [16] are: (i) in this work, the set of trajectories is modeled as a network while previous work [31] only consider the joint probability distribution of the size of attractor and transient; (ii) the networks are generated from the DPSWs trajectories while the previous work [16] is based on four parameters for modeling. Next, we describe the proposed approach.

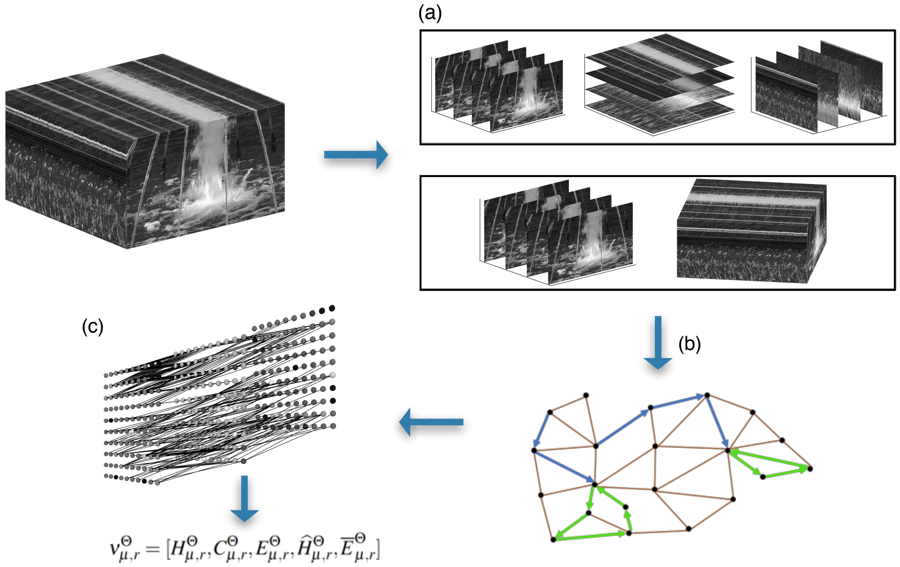

(a)
(b)
(c)
$v_{\mu,r}^{\Theta} = [H_{\mu,r}^{\Theta}, C_{\mu,r}^{\Theta}, E_{\mu,r}^{\Theta}, \widehat{H}_{\mu,r}^{\Theta}, \overline{E}_{\mu,r}^{\Theta}]$

Figure 2: Illustration of the proposed method. (a-b) DPSWs are performed in the video using a strategy (orthogonal planes or temporal and spatial descriptors), (c) the trajectories are modeled as networks and its topological characteristics are used to compose the feature vector.

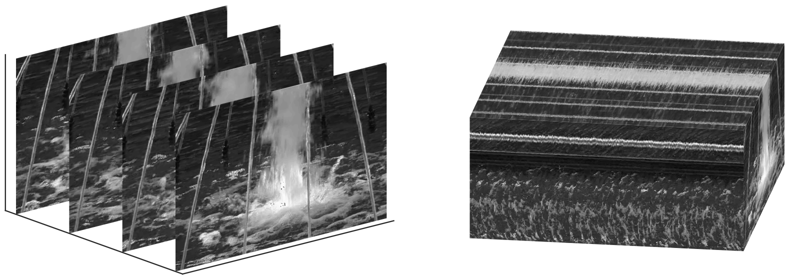
3.1. Strategies for DPSW in videos

In this section, we present two different strategies to apply the DPSWs in videos. Basically, these strategies define the neighborhood function $\eta(i)$ used to perform the DPSWs in the videos of dynamic texture. Therefore, the difference of the DPSWs in videos for images is the definition of the neighborhood function.

3.1.1. Spatial and Temporal Descriptors - STD

A video of dynamic texture consists of a finite set of pixels P and a mapping I that assigns to each pixel $i = (x_i, y_i, t_i) \in P$ a gray level $I(i) \in [0, 255]$. The coordinates of the video are defined as: $x_i \in [0, W]$ and $y_i \in [0, H]$ are the spatial coordinates and $t_i \in [0, T]$ is the temporal coordinate of the video. Each pixel has a neighborhood $\eta(i)$ containing the neighboring pixels.

The proposed strategy is to separate the video in spatial and temporal descriptors to apply the DPSWs, as illustrated in Figure 3. Basically, a neighborhood function $\eta(i)$ is defined for each descriptor, changing the trajectory of the DPSWs and allowing to explore different properties of the dynamic texture present in the video.



(a) Spatial Descriptor

(b) Temporal Descriptor

Figure 3: Illustration of the spatial and temporal descriptors of a dynamic texture video.

The spatial descriptor S aims to describe the appearance characteristics present in dynamic textures. Thus, the agent will walk only on pixels that belong to the same frame of the video, as illustrated in Figure 3(a). That is, the spatial variance of the dynamic textures are described for each frame, independently [31]. Therefore, the neighborhood $\eta(i)^S$ is defined as the pixels j whose the Euclidean distance between i and j is less than $\sqrt{2}$ and the time coordinates t_i and t_j are equals, according to:

$$\eta(i)^S = \{j \mid \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq \sqrt{2} \text{ and } t_i = t_j\}. \tag{11}$$

The temporal descriptor T highlights the motion characteristics of the dynamic texture. In this descriptor, the agent walks between different frames hence it characterizes the temporal variance of the dynamic textures (Figure 3(b)). Thus, the neighborhood $\eta(i)^T$ is given by pixels j whose Euclidean distance is less than $\sqrt{3}$ and the coordinate t_i and t_j are different, according to

$$\eta(i)^T = \{j \mid \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2} \leq \sqrt{3} \text{ and } t_i \neq t_j\}. \tag{12}$$

3.1.2. Three Orthogonal Planes - TOP

The second strategy considered to perform DPSWs in the dynamic texture videos is the slicing of the video into Three Orthogonal Planes (TOP) [26, 31]. The three planes named XY, XT and YT are shown in Figure 4. Basically, the planes define the neighborhood function $\eta(i)$ of a pixel, changing the trajectories of the agent. The idea is each plan highlight different characteristics of the video: the XY plane enhances appearance characteristics, while the XT and YT planes correspond to the motion characteristics.

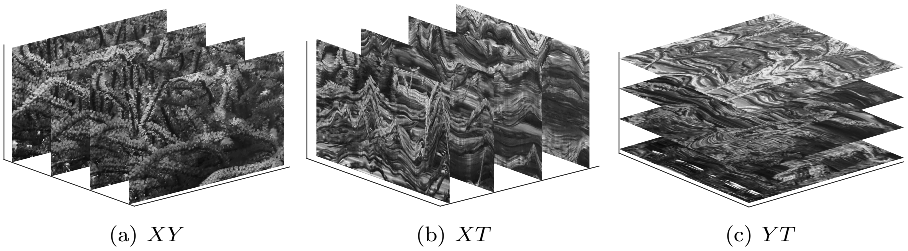

(a) XY (b) XT (c) YT

Figure 4: Illustration of the three orthogonal planes of a dynamic texture video.

In the XY plan, the agent walks only on pixels belonging to the same frame, characterizing the spatial variance of the gray level distribution of the dynamic texture. Therefore, the neighborhood $\eta^{XY}(i)$ of a pixel i in the XY plan is defined as all pixels j that are at a Euclidean distance lesser or equal to $\sqrt{2}$ and the temporal indexes t_i and t_j are equals, according to:

$$\eta(i)^{XY} = \{j \mid \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq \sqrt{2} \text{ and } t_i = t_j\}. \tag{13}$$

For the XT plan, the neighborhood $\eta^{XT}(i)$ of a pixel i is defined as the pixels j whose Euclidean distance is lesser or equal to $\sqrt{2}$ and if the coordinates y_i and y_j are equal, as given by:

$$\eta(i)^{XT} = \{j \mid \sqrt{(x_i - x_j)^2 + (t_i - t_j)^2} \leq \sqrt{2} \text{ and } y_i = y_j\}. \tag{14}$$

Finally, in the YT plan, a pixel j belongs to the neighborhood $\eta^{YT}(i)$ if the coordinates x_i and x_j are equals, according to:

$$\eta(i)^{YT} = \{j \mid \sqrt{(t_i - t_j)^2 + (y_i - y_j)^2} \leq \sqrt{2} \text{ and } x_i = x_j\}. \tag{15}$$

3.2. Modeling DPSWs Trajectories in Networks

Basically, each trajectory consists of a sequence of pixels visited by the deterministic walker based on a given memory μ and a rule of movement r. To analyze such trajectories, each movement performed by the walker between two pixels is considered an edge of a network [34]. In this way, a network can be constructed from the set of trajectories performed by DPSWs in a video.

Consider a network $G_{\mu,r}$ generated by DPSWs with memory size μ and rule of movement r. This network consists of a pair $G_{\mu,r} = (V, E)$, where V is a set of vertices and E the set of edges. Each vertex v_i corresponds to a pixel i of the video, and as the agent moves from pixel i to pixel j in the video, an non-directed edge e_{v_i,v_j} is added to the set E connecting the vertices v_i and v_j. The agent will move according to a deterministic rule r (minimum min or maximum max) and a memory M_μ of size μ until it gets stuck in an attractor. It is important to note that the agent can move between two pixels many times, however, only one edge connecting the corresponding vertices is added to the set E. That is, an edge e_{v_i,v_j} is only added to the set E if $e_{v_i,v_j} \notin E$. In addition, each edge has a weight corresponding to the modulus of the intensity difference between the two pixels representing the vertices $\omega(e_{v_i,v_j}) =\mid I(i) - I(j) \mid$. Therefore, given a trajectory of size $(l = \tau + \rho)$ started from a pixel, its conversion into edges can be defined as

$$\bigcup_{j=1}^{l-1} E = \{e_{v_j,v_{j+1}} \mid e_{v_j,v_{j+1}} \notin E\}. \tag{16}$$

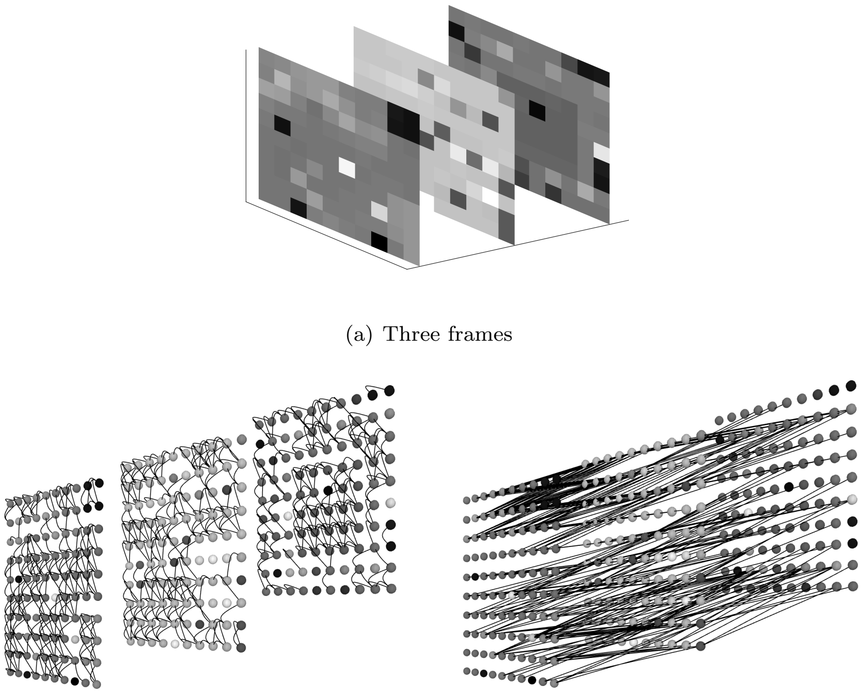
Figure 1 shows a trajectory of a DPSW performed in an image and the network generated by it. In the image (Figure 1(a)), each pixel is represented by a square with its gray level. The transient of the trajectory are the blue squares and the attractor is represented by the green squares. Considering each pixel as a vertex and each step of the DPSW as an edge, a network is generated (Figure 1(b)).

Each pixel is taken as an initial condition of a DPSW. Thus, N trajectories are produced for a video of $N = W \times H \times T$ pixels. Given these N trajectories, a network without disconnected vertices can be constructed. Since the DPSWs are based on the gray level variation of the video, the network edges directly reflect the texture patterns. In this way, the network topology has information about the transients and attractors and, consequently, important properties of the dynamic texture. In order to characterize the movement and appearance patterns present in the dynamic textures, we present two different strategies to apply the DPSWs in the videos and obtain networks. Next, these two strategies are described in detail.

3.2.1. Spatial and Temporal Descriptors - STD

In this section, we build the network using the spatial and temporal descriptors strategy (Section 3.1.1). Thus, given a video and considering the spatial descriptor S, DPSWs with a memory size μ and rule r are performed to generate a network $G^S_{\mu,r}$. This network represents the transitive and attractive regions of the spatial descriptor and, consequently, characterizes the appearance properties of the dynamic texture.

On the other hand, for the temporal descriptor T, another network $G^T_{\mu,r}$ is generated with DPSWs using a memory size μ and rule r. The generated network contains information of the trajectories performed by the agent in the time. Thus, movement characteristics of the dynamic texture are modeled. Figure 5 illustrates the networks generated by the DPSWs in three frames (Figure 5 (a)) for the two descriptors.



(a) Three frames

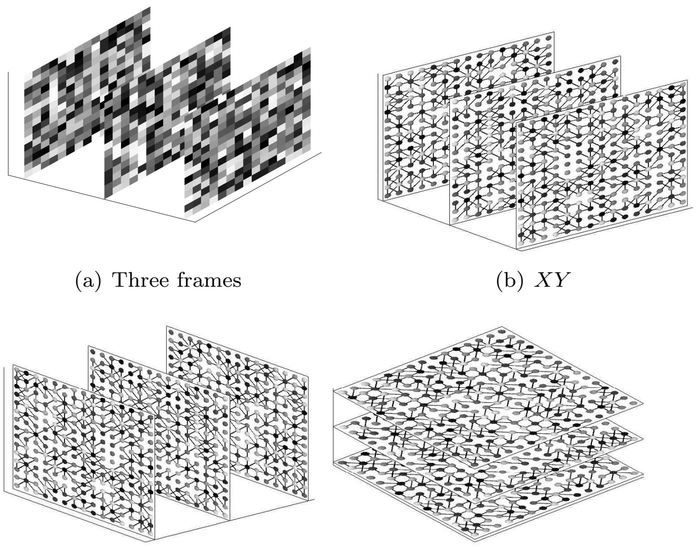
(b) Spatial Descriptor

(c) Temporal Descriptor

Figure 5: Networks generated by DPSWs for the spatial and temporal descriptors of a video with three frames (a). In the spatial descriptor, the edges connect only vertices that correspond to pixels of the same frame, while in the temporal descriptor the edges connect vertices that represent pixels of different frames.

3.2.2. Three Orthogonal Planes - TOP

In order to generate networks with characteristics of movement and appearance, DPSWs are performed independently in each orthogonal plane. Thus, given a video, DPSWs with memory μ and rule r are performed independently in the planes XY, XT and YT to generate the corresponding networks $G^{XY}_{\mu,r}$, $G^{XT}_{\mu,r}$ and $G^{YT}_{\mu,r}$. Such networks are responsible for describing the characteristics of appearance ($G^{XY}_{\mu,r}$) and movement ($G^{XT}_{\mu,r}$ and $G^{YT}_{\mu,r}$) of the dynamic texture. Figure 6 illustrates the networks generated by the DPSWs for each orthogonal plane of three frames (Figure 6(a)).



(a) Three frames

(b) XY

(c) XT

(d) YT

Figure 6: Modeling DPSWs trajectories in networks for the three orthogonal planes of the video.

3.3. Feature Vector

The proposal of this approach is to use the topological characteristics of the networks generated by the DPSWs to characterize the dynamic textures. For this, we consider measures of complex networks based on the degree k, weighted degree $\overline{k}$ and joint degree $\widehat{k}$ of the vertices. These measures may quantify the agent behavior in a certain region of the network and as a consequence in the dynamic texture [34]. For example, vertices with the same degree can indicate that the agent has the same behavior in that region.

From these vertices properties, it is possible to calculate the degree distribution $P(k)$. Then, the following heterogeneity measures are obtained from the distribution $P(k)$ (described in Section 2.2): entropy H, contrast C and energy E. In addition, we also use the entropy of the joint degree

$$\widehat{H} = \sum_{v_i}^{N} \widehat{k}_{v_i} \log_2[\widehat{k}_{v_i}], \tag{17}$$

and the energy of the weighted degree,

$$\overline{E} = \sum_{v_i}^{N} \overline{k}_{v_i}^{\ 2}. \tag{18}$$

Although simple, these measures are effective for network characterization and it can be calculated quickly.

To characterize a dynamic texture video, a network $G^{\Theta}_{\mu,r}$ for a descriptor or orthogonal plane Θ is generated by DPSWs with a memory size μ and rule r. Then, the five statistical measures described above are extracted from the network to compose the feature vector,

$$V^{\Theta}_{\mu,r} = [H^{\Theta}_{\mu,r}, C^{\Theta}_{\mu,r}, E^{\Theta}_{\mu,r}, \widehat{H}^{\Theta}_{\mu,r}, \overline{E}^{\Theta}_{\mu,r}] \tag{19}$$

From this feature vector, it is possible to measure the dispersion of vertex pairs with the same degree and the network heterogeneity, characterizing the agent behavior in the dynamic texture.

To combine appearance and movement characteristics, we concatenate the feature vector $V^{\Theta}_{\mu,r}$ according to the strategy to perform the DPSWs in the video. In the strategy of spatial and temporal descriptors, measurements obtained of networks generated from each descriptor are combined to create a vector $\upsilon_{\mu,r}$, which contains appearance and movement characteristics, according to

$$\upsilon_{\mu,r} = [V^{S}_{\mu,r}, V^{T}_{\mu,r}]. \tag{20}$$

Similarly, in the strategy of orthogonal planes, measurements extracted from the three planes are concatenated to compose a vector $\varphi_{\mu,r}$, given by:

$$\varphi_{\mu,r} = [V^{XY}_{\mu,r}, V^{XT}_{\mu,r}, V^{YT}_{\mu,r}]. \tag{21}$$

The trajectory depends on the rule of movement and the memory size used by the DPSW. In order to characterize different properties of the dynamic texture, statistical measures are extracted from networks generated by DPSWs with different memory sizes μ. Thus, a combined feature vector υ_r for the STD strategy, which considers information from networks generated by different memory sizes and using a rule r, is given by:

$$\upsilon_r = [\upsilon_{\mu_1,r}, \upsilon_{\mu_2,r}, ..., \upsilon_{\mu_n,r}]. \tag{22}$$

Similarly, a combined feature vector φ_r for the TOP strategy is used, according to:

$$\varphi_r = [\varphi_{\mu_1,r}, \varphi_{\mu_2,r}, ..., \varphi_{\mu_n,r}]. \tag{23}$$

Finally, a final feature vector combining the two rules of movement can be considered. Indeed, each rule of movement produces different trajectories and, consequently, networks with different information. Thus, the combination of the two rules may produce a vector with new information that improves the classification performance. Although this occurs in most cases, this improvement is not a rule, as seen in Section 5.1. The final vector for the strategy of spatial and temporal descriptors is given by

$$\upsilon = [\upsilon_{min}, \upsilon_{max}], \tag{24}$$

on the other hand, the feature vector for the strategy of orthogonal planes is defined as:

$$\varphi = [\varphi_{min}, \varphi_{max}]. \tag{25}$$

4. Experimental Setup

In all experiments, a feature vector is extracted from each dynamic texture video and the 1-Nearest Neighbor (1NN) classifier is used to classify the samples. A specific scheme was adopted for each database to separate the training and test set. Next, each database is described in detail.

- Dyntex++ [44]: this database is a compiled version of the Dyntex database [45]. On this database, the samples were preprocessed by [44] to evidence only their dynamic texture, eliminating samples with static or dynamic non-representative backgrounds, zoom, and textures without movement. This preprocessing is described in [44]. Thus, each sample has a unique dynamic texture class. The database consists of 3600 dynamic texture videos, divided into 36 classes with 100 samples each. Some dynamic textures present are boiling water, river water, colony of ants, smoke, among others. In the experiments, a 10-fold cross-validation scheme with 10 trials was used as well as in [16].

- UCLA [46]: is widely used by the computer vision community as a benchmark for evaluating methods of dynamic texture recognition. It consists of 200 dynamic texture videos separated into 50 classes with 4 samples per class (UCLA-50 version). Each video contains $160 \times 110 \times 75$ pixels and it was captured from a single point of view. In this way, each video was cut in a window of $48 \times 48 \times 75$ pixels in order to obtain representative statistical and dynamic characteristics. This database also has two variations of the original database proposed in [47]. In these variations, the database is reorganized in order to combine samples from different viewpoints. In the first variation, the samples are reorganized into 9 classes (UCLA-9): boiling water (8 samples), fire (8), flower (12), fountains (20), plants (108), sea (12), smoke (4), water (12) and waterfall (16). As the plants class has more samples than others, their samples are eliminated, generating a second variation with 8 classes (UCLA-8). To evaluate the proposed method, we have adopted an experimental setup similar to [47]. In UCLA-50 was used a $4-$fold cross-validation scheme to separate the training and testing sets with 10 repetitions In UCLA-8 and UCLA-9 databases, for the testing set, half of the sequences are 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.

5. Results and discussion

5.1. Parameter Analysis

In this section, we analyze the parameters of the proposed method and its impact on the task of dynamic textures recognition. The parameters of the proposed method are: (i) memory sizes μ − (ii) rule of movement r − (iii) measures of complex networks. In these experiments, we used the UCLA-50 and Dyntex++ databases.

Initially, we considered the five measures of complex networks for the method analysis: entropy H, contrast C, energy E, entropy of the joint degree $\widehat{H}$ and energy of the weighted degree $\overline{E}$. Table 1 summarizes the results for different memory sizes combined (feature vector υ_r) and rules of movement using the STD strategy to perform the DPSWs in the videos. The results show an improvement in the correct classification rate as we combine more than two memory sizes. However, when using more than five memory sizes, the accuracy obtained starts to oscillate down, suggesting that the proposed descriptors have reached their limits.

Concerning to rule of movement, in most cases the rule $r = min$ obtained better classification rates than $r = max$ on the UCLA-50 database, on the other hand, the rule $r = max$ achieved highest accuracies on the Dyntex++ database. Nevertheless, the highest accuracies were obtained using the rules $r = min$ and $r = max$ combined (feature vector υ). For the UCLA-50 database, the highest accuracy was 98.0%, while for the Dyntex ++ database it was 92.6%. These results were obtained by concatenating the memory sizes $\mu = [0, 1, 2, 3]$ and $\mu = [0, 1, 2, 3, 4, 5, 6, 7, 8]$ for the UCLA and Dyntex++ databases, respectively. However, note that the variance in accuracy for different concatenations of memory sizes is very low. This indicates that the method is not very sensitive to this parameter.

Table 1: Classification results of the STD strategy using different combinations of memory sizes and rules of movement on two databases.

$\{u_1, u_2, ..., u_m\}$	UCLA-50			Dyntex++		
	υ_{min}	υ_{max}	$\{\upsilon_{min}, \upsilon_{max}\}$	υ_{min}	υ_{max}	$\{\upsilon_{min}, \upsilon_{max}\}$
{0}	87.00	91.00	96.50	78.56	84.28	90.00
{0, 1}	87.00	91.00	96.50	78.56	84.28	90.00
{0, 1, 2}	93.50	93.50	97.50	83.22	87.17	91.53
{0, ..., 3}	94.50	93.50	98.00	84.19	87.89	91.75
{0, ..., 4}	94.00	94.00	97.50	84.86	88.31	92.00
{0, ..., 5}	95.50	94.00	97.50	85.08	88.58	92.06
{0, ..., 6}	95.50	94.00	97.50	85.11	88.64	92.19
{0, ..., 7}	96.00	93.50	97.50	85.39	88.67	92.39
{0, ..., 8}	96.50	94.00	97.50	85.69	88.67	92.60

Table 2 presents the results of the experimental analysis conducted for the strategy of three orthogonal planes. On the UCLA-50 database, the accuracy does not improve significantly as we combine more than one memory size. On this database, the highest accuracy 92.50% was obtained using $\mu = [0, 1]$ and rule of movement $r = max$. On the Dyntex++ database, an improvement in accuracy is obtained as we combine more than two memory size. For this database, the rules of movement $r = min$ and $r = max$ combined and the memory sizes $\mu = [0, 1, 2, 3, 4, 5, 6]$ achieved the highest accuracy.

Table 2: Classification results of the TOP strategy using different combinations of memory size and rules of movement on two databases.

$\{u_1, u_2, ..., u_m\}$	UCLA-50			Dyntex++		
	φ_{min}	φ_{max}	$\{\varphi_{min}, \varphi_{max}\}$	φ_{min}	φ_{max}	$\{\varphi_{min}, \varphi_{max}\}$
{0}	82.50	91.50	92.50	83.00	89.97	92.56
{0, 1}	82.50	91.50	92.50	83.00	89.97	92.56
{0, 1, 2}	82.00	91.50	90.50	87.33	91.86	93.22
{0, ..., 3}	81.50	92.00	91.00	88.11	92.22	93.39
{0, ..., 4}	82.00	92.50	90.50	88.75	92.50	93.36
{0, ..., 5}	82.50	91.50	90.00	89.17	92.31	93.42
{0, ..., 6}	82.00	91.00	89.00	89.17	92.47	93.47
{0, ..., 7}	83.00	91.00	89.00	89.19	92.64	93.42
{0, ..., 8}	83.00	91.50	89.00	89.39	92.67	93.42

Finally, we analyze the performance of the method for different combinations of complex networks measures. Table 3 presents the correct classification rates for the different combinations of statistical measures using the STD strategy. Note that the combination of the five statistical measures obtained the best performance on two databases. The results also show that when considering a single measurement for characterization, the best performance was obtained with the entropy H.

Table 4 presents the results with different combinations of statistical measures for the TOP strategy. In this strategy, the combination of the five measures also presented the best accuracy on the two databases. In addition, the contrast of the degree distribution obtained a higher accuracy than other measures. In both strategies, the combination of the five measures provided the best accuracies. Such behavior corroborates that the five statistical measures are robust for characterization of the dynamic textures. However, it is important to note that in some cases other combinations have provided interesting results, such as the energy and the weighted degree ($[E, \overline{E}]$), and entropy and the energy of the degree weighted ($[H, \overline{E}]$).

5.2. Comparison with other methods

In this section, the proposed method is compared to other methods of the literature. For this, the correct classification rates, standard deviation, and the number of features (when described in the original articles) of the methods are considered. We used the parameters of the proposed method obtained in Section 5.1.

Table 3: Results for different combinations of statistical measures of networks generated using the STD strategy.

Measures					Databases	
H	E	C	$\widehat{H}$	$\overline{E}$	UCLA-50	Dyntex++
x					93.50	88.30
	x				92.00	87.41
		x			93.00	88.30
			x		89.00	87.69
				x	91.50	71.77
x				x	98.00	89.66
	x			x	97.50	89.25
x	x				93.50	89.47
x	x	x			95.00	90.27
x	x	x	x		94.50	91.69
x	x	x	x	x	98.00	92.60

Table 4: Results for different measures of complex networks generated by the DPSWs using the TOP strategy

Measures					Database	
H	E	C	$\widehat{H}$	$\overline{E}$	UCLA-50	Dyntex++
x					84.00	89.91
	x				76.00	88.67
		x			86.50	90.63
			x		67.50	90.61
				x	79.00	80.41
x				x	90.00	91.22
	x			x	90.00	90.58
x	x				89.00	90.00
x	x	x			90.00	91.80
x	x	x	x		87.50	92.94
x	x	x	x	x	92.50	93.50

Table 5 summarizes the classification results obtained by the methods on the UCLA-50 database. As can be seen, the proposed method using the STD strategy (DPSWNet-STD) outperformed the other methods. On the other hand, the proposed method using the TOP strategy obtained an accuracy lower than other methods. Concerning the network-based methods, the proposed method improves the classification rate compared to the CNDT[16] method by 3.5%. This method models the network using four parameters (radius and thresholds), while the proposed approach uses the DPSW for modeling. Therefore, the results suggest that the modeling based on DPSW is more effective for dynamic texture recognition. In addition, the proposed modeling uses only two parameters against four parameters of the CNDT method.

The second comparison was performed using the UCLA-9 database as can

Table 5: Comparison of the classification results of the proposed method and others on UCLA-50 database.

Methods	Number of features	Accuracy (%)
KDT-MD [48]	-	89,50
DFS [4]	-	89,50
3D-OTF [49]	290	87,10
WLBPC [29]	-	96,50
CVLBP [27]	-	93,00
HLBP [28]	1536	95,00
MEWLSP [30]	1536	96,50
RI-VLBP [10]	16384	77,50 (± 8.98)
LBP-TOP [26]	768	95,00 (± 4.44)
DPSW -TOP [8]	75	95,00 (± 4.78)
CNDT [16]	420	95,00 (± 5.19)
DPSWNet - STD	80	98,00 (± 3.50)
DPSWNet - TOP	60	93,00 (± 4.25)

be seen in Table 6. Note that the DPSWNet-TOP method obtained a small improvement over the HLBP, MEWLSP and DNGP methods. In contrast with the UCLA-50 database, on the UCLA-9 database, dynamic texture samples obtained at different viewpoints are considered the same class. For the UCLA-9 database, the DPSWNet-TOP method gives an accuracy of 99.10% (± 0.86) against 93.00% (± 4.25) on the UCLA-50 database, an improvement of 6.10%. This suggests that the DPSWNet-TOP approach is better to recognize dynamic textures captured at different viewpoints in the same class. In relation to complexity-based methods (i.e. High level feature, Chaotic vector, 3D-OTF, and DFS), the proposed methods also improve the classification rate.

The results on the UCLA-8 database are presented in Table 7. Note that the proposed method obtained 96.55 (± 7.13) and 96.15 (± 7.02) using the STD and TOP strategies, respectively. This performance is slightly lower than some literature methods, such as HLBP, MEWLSP and MBSIF methods. However, it is important to emphasize that the feature vector size of these methods (i.e., 1536 for HLBP and MEWLSP, and 6144 for MBSIF) is significantly higher than the feature vector of our method.

Finally, Table 8 shows the results on the Dyntex++ database for different methods. The proposed method obtained 92.60% (± 1.86) and 93.50% (± 1.27) using the STD and TOP strategies, respectively. In general, the DPSWNet-TOP method shows an improvement of 2.11% compared to the DPSW-TOP method and 9.64% over the CNDT method. The RI-VLBP and LBP-TOP methods achieved the best accuracies, with 96.14% and 97.72%, respectively. Nevertheless, the proposed method is still competitive due to the feature vector size, for example, the RI-VLBP produces a long feature vector of size 16384, whereas the DPSWNet-TOP extracts only 180 features.

In general, the proposed method obtained competitive results compared to the other methods. A key point for the dynamic texture methods is the di-

Table 6: Classification results for all methods on the UCLA-9 database.

Methods	Number of features	Accuracy (%)
3D-OTF [49]	290	96,32
CVLBP [27]	-	96,90
HLBP [28]	1536	98,30
MEWLSP [30]	1536	98,50
MBSIF [50]	6144	98,75
High level feature [51]	-	92,60
DNGP [52]	-	98,10
WMFS [53]	702	96,95
WLBPC [29]	-	97,17
Chaotic vector [54]	300	85,10
RI-VLBP [10]	16384	96,30
LBP-TOP [26]	768	96,00
DPSW -TOP [8]	75	96,33 (± 2.46)
CNDT [16]	336	95,61 (± 2.72)
DPSWNet-STD	80	98,40 (± 2.01)
DPSWNet-TOP	60	99,10 (± 0.86)

Table 7: Comparison results on the UCLA-8 database.

Methods	Number of features	Accuracy (%)
3D-OTF [49]	290	95.80
CVLBP [27]	-	95.65
HLBP [28]	1536	97.50
MEWLSP [30]	1536	98.04
MBSIF [50]	6144	97.80
High level feature [51]	-	85.65
DNGP [52]	-	97.00
WMFS [53]	702	97.18
WLBPC [29]	-	97.60
Chaotic vector [54]	300	85.00
RI-VLBP [10]	16384	91.96
LBP-TOP [26]	768	93.67
DPSW -TOP [8]	75	93.41 (± 6.01)
CNDT [16]	336	94.32 (± 4.18)
DPSWNet-STD	80	96.55 (± 7.13)
DPSWNet-TOP	60	96.15 (± 7.02)

mension of the feature vectors. Nowadays, dynamic texture methods have been used in different real-time applications in which computational time is important. Therefore, a high dimensional feature vector increases the computational time for dynamic textures classification. Several studies over the last few years indicate that a low dimensional feature vector is a key point for the success of the dynamic texture recognition methods [26, 51, 25, 28]. This fact is interesting

Table 8: Comparison results for different dynamic texture methods on the Dyntex++ database.

Methods	Number of features	Accuracy (%)
RI-VLBP [10]	16384	96.14 (± 0.77)
LBP-TOP [26]	768	97.72 (± 0.43)
DPSW -TOP [8]	75	91.39 (± 1.29)
CNDT [16]	336	83.86 (± 1.40)
DPSWNet-STD	180	92.60 (± 1.86)
DPSWNet-TOP	210	93.50 (± 1.27)

for the proposed method since their feature vectors are small. For example, for all UCLA databases, the proposed methods DPSWNet-STD and DPSWNet-TOP achieve competitive correct classification rates with feature vectors of size 80 and 60, respectively. On the other hand, methods such as HLBP, MBSIF, and RI-VLBP have feature vectors of size 1536, 6144 and 16384, respectively.

6. Conclusion

In this paper, we have proposed a new method for characterization and classification of dynamic textures using networks generated by deterministic partially self-avoiding walks. Using the DPSWs, we have shown two different strategies to build networks from dynamic texture videos, which allows us to compose a feature vector based on network topological measures for dynamic texture classification.

We showed promising results obtained on the UCLA and Dyntex++ databases. The obtained results are very competitive when compared to other methods, demonstrating that our approach is highly discriminative in its two strategies used to generate the networks by DPSWs. Our method also outperformed the other previous DPSW-based and network-based methods. Moreover, the proposed method showed that is competitive in terms of feature vector size, producing feature vectors significantly smaller than other literature methods. Thus, this tradeoff between performance and dimensionality demonstrates the great potential of the proposed method in dynamic texture analysis. The future related works would be to investigate the use of directed networks and different network measures.

Acknowledgments

Lucas C. Ribas gratefully acknowledges CAPES (Grant Nos. 9254772/m) and São Paulo Research Foundation (FAPESP) (Grant No. 2016/23763-8) for financial support. Odemir M. Bruno thanks the financial support of CNPq (Grant # 307897/2018-4) and FAPESP (Grant #s 14/08026-1 and 16/18809-9).

References

[1] D. Chetverikov, R. Péteri, A brief survey of dynamic texture description and recognition, in: Computer Recognition Systems, Springer, 2005, pp. 17–26.

[2] G. Doretto, E. Jones, S. Soatto, Spatially homogeneous dynamic textures, in: Computer Vision-ECCV 2004, Springer, 2004, pp. 591–602.

[3] R. C. Nelson, R. Polana, Qualitative recognition of motion using temporal texture, CVGIP: Image Underst. 56 (1) (1992) 78–89.

[4] Y. Xu, Y. Quan, H. Ling, H. Ji, Dynamic texture classification using dynamic fractal analysis, in: 2011 International Conference on Computer Vision, 2011, pp. 1219–1226. doi:10.1109/ICCV.2011.6126372.

[5] Y. Zhao, J. Zhao, E. Dong, B. Chen, J. Chen, Z. Yuan, D. Zhang, Dynamic texture modeling applied on computer vision based fire recognition, in: International Conference on Artificial Intelligence and Computational Intelligence, Springer, 2011, pp. 545–553.

[6] S. Milko, E. Samset, T. Kadir, Segmentation of the liver in ultrasound: a dynamic texture approach, International Journal of Computer Assisted Radiology and Surgery 3 (1-2) (2008) 143.

[7] S.-J. Wang, W.-J. Yan, X. Li, G. Zhao, X. Fu, Micro-expression recognition using dynamic textures on tensor independent color space, in: 2014 22nd International Conference on Pattern Recognition, IEEE, 2014, pp. 4678–4683.

[8] W. N. Gonçalves, O. M. Bruno, Dynamic texture analysis and segmentation using deterministic partially self-avoiding walks, Expert Systems with Applications 40 (11) (2013) 4283 – 4300.

[9] V. Kellokumpu, G. Zhao, M. Pietikäinen, Human activity recognition using a dynamic texture based method., in: BMVC, Vol. 1, 2008, p. 2.

[10] G. Zhao, M. Pietikäinen, Dynamic texture recognition using volume local binary patterns, in: R. Vidal, A. Heyden, Y. Ma (Eds.), Dynamical Vision, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 165–177.

[11] A. Ravichandran, R. Vidal, Video registration using dynamic textures, Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (1) (2011) 158–171. doi:10.1109/TPAMI.2010.61.

[12] R. Polana, R. Nelson, Motion-Based Recognition, Springer Netherlands, Dordrecht, 1997, Ch. Temporal Texture and Activity Recognition, pp. 87–124.

[13] G. Doretto, A. Chiuso, Y. Wu, S. Soatto, Dynamic textures, International Journal of Computer Vision 51 (2) (2003) 91–109.

[14] J. J. de Mesquita S Junior, L. C. Ribas, O. M. Bruno, Randomized neural network based signature for dynamic texture classification, Expert Systems with Applications 135 (2019) 194 – 200.

[15] Y. Qiao, L. Weng, Hidden markov model based dynamic texture classification, Signal Processing Letters, IEEE 22 (4) (2015) 509–512. doi:10.1109/LSP.2014.2362613.

[16] W. N. Gonçalves, B. B. Machado, O. M. Bruno, A complex network approach for dynamic texture recognition, Neurocomputing 153 (0) (2015) 211 – 220.

[17] L. C. Ribas, W. N. Gonalves, O. M. Bruno, Dynamic texture analysis with diffusion in networks, Digital Signal Processing 92 (2019) 109 – 126.

[18] R. P. Wildes, J. R. Bergen, Qualitative spatiotemporal analysis using an oriented energy representation, in: European Conference on Computer Vision, Springer, Berlin, Heidelberg, 2000, pp. 768–784.

[19] P. Wu, Y. M. Ro, C. S. Won, Y. Choi, Texture descriptors in mpeg-7, in: W. Skarbek (Ed.), Computer Analysis of Images and Patterns, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 21–28.

[20] J. Smith, C.-Y. Lin, M. Naphade, Video texture indexing using spatio-temporal wavelets, in: Image Processing. 2002. Proceedings. 2002 International Conference on, Vol. 2, 2002, pp. II–437–II–440 vol.2.

[21] S. Dubois, R. Péteri, M. Ménard, Pattern Recognition and Image Analysis: 4th Iberian Conference, IbPRIA 2009 Póvoa de Varzim, Portugal, June 10-12, 2009 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, Ch. A Comparison of Wavelet Based Spatio-temporal Decomposition Methods for Dynamic Texture Recognition, pp. 314–321.

[22] W. N. Gonçalves, B. B. Machado, O. M. Bruno, Spatiotemporal gabor filters: a new method for dynamic texture recognition, Workshop de Viso Computacional (2011) 184–189.

[23] D. Tiwari, V. Tyagi, Dynamic texture recognition: a review, in: Information Systems Design and Intelligent Applications, Springer, 2016, pp. 365–373.

[24] R. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, Systems, Man and Cybernetics, IEEE Transactions on SMC-3 (6) (1973) 610–621. doi:10.1109/TSMC.1973.4309314.

[25] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (7) (2002) 971–987.

[26] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (6) (2007) 915–928.

[27] D. Tiwari, V. Tyagi, Dynamic texture recognition based on completed volume local binary pattern, Multidimensional Systems and Signal Processing 27 (2) (2016) 563–575.

[28] D. Tiwari, V. Tyagi, A novel scheme based on local binary pattern for dynamic texture recognition, Computer Vision and Image Understanding 150 (2016) 58 – 65.

[29] D. Tiwari, V. Tyagi, Improved weber's law based local binary pattern for dynamic texture recognition, Multimedia Tools and Applications (2016) 1–18.

[30] D. Tiwari, V. Tyagi, Dynamic texture recognition using multiresolution edge-weighted local structure pattern, Computers & Electrical Engineering 62 (2017) 485 – 498.

[31] W. N. Gonçalves, O. M. Bruno, Dynamic texture segmentation based on deterministic partially self-avoiding walks, Computer Vision and Image Understanding 117 (9) (2013) 1163 – 1174.

[32] G. F. Lima, A. S. Martinez, O. Kinouchi, Deterministic walks in random media, Phys. Rev. Lett. 87 (2001) 010603.

[33] A. R. Backes, W. N. Gonçalves, A. S. Martinez, O. M. Bruno, Texture analysis and classification using deterministic tourist walk, Pattern Recognition 43 (3) (2010) 685–694.

[34] A. R. Backes, A. S. Martinez, O. M. Bruno, Texture analysis using graphs generated by deterministic partially self-avoiding walks, Pattern Recognition 44 (8) (2011) 1684 – 1689.

[35] W. N. Goncalves, A. R. Backes, A. S. Martinez, O. M. Bruno, Texture descriptor based on partially self-avoiding deterministic walker on networks, Expert Systems with Applications 39 (15) (2012) 11818 – 11829.

[36] M. Campiteli, A. Martinez, O. Bruno, An image analysis methodology based on deterministic tourist walks, in: Advances in Artificial Intelligence - IBERAMIA-SBIA 2006, Vol. 4140 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 159–167.

[37] W. N. Gonçalves, A. S. Martinez, O. M. Bruno, Complex network classification using partially self-avoiding deterministic walks, Chaos: An Interdisciplinary Journal of Nonlinear Science 22 (3) (2012) 033139.

[38] M. E. Newman, J. Park, Why social networks are different from other types of networks, Physical Review E 68 (3) (2003) 036122.

[39] A.-L. Barabasi, Z. N. Oltvai, Network biology: understanding the cell's functional organization, Nature reviews genetics 5 (2) (2004) 101–113.

[40] S. Bornholdt, H. G. Schuster, Handbook of graphs and networks: from the genome to the internet, John Wiley & Sons, 2006.

[41] P. Holme, Congestion and centrality in traffic flow on complex networks, Advances in Complex Systems 6 (02) (2003) 163–176.

[42] L. C. Ribas, J. J. de M. Sá Junior, L. F. S. Scabini, O. M. Bruno, Fusion of complex networks and randomized neural networks for texture analysis, CoRR abs/1806.09170. `arXiv:1806.09170`.

[43] L. C. Ribas, M. B. Neiva, O. M. Bruno, Distance transform network for shape analysis, Information Sciences 470 (2019) 28 – 42.

[44] B. Ghanem, N. Ahuja, Maximum margin distance learning for dynamic texture recognition, in: Proceedings of the 11th European Conference on Computer Vision: Part II, ECCV'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 223–236.

[45] R. Péteri, S. Fazekas, M. J. Huiskes, Dyntex: A comprehensive database of dynamic textures, Pattern Recogn. Lett. 31 (12) (2010) 1627–1632.

[46] A. B. Chan, N. Vasconcelos, Probabilistics for the classification of autoregressive visual processes, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 846–851.

[47] A. Ravichandran, R. Chaudhry, R. Vidal, View-invariant dynamic texture recognition using a bag of dynamical systems, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1651–1657.

[48] A. B. Chan, N. Vasconcelos, Classifying video with kernel dynamic textures., in: CVPR, 2007, pp. 1–6.

[49] Y. Xu, S. Huang, H. Ji, C. Fermüller, Scale-space texture description on sift-like textons, Computer Vision and Image Understanding 116 (9) (2012) 999–1013.

[50] S. R. Arashloo, J. Kittler, Dynamic Texture Recognition Using Multiscale Binarized Statistical Image Features, Multimedia, IEEE Transactions on 16 (8) (2014) 2099–2109.

[51] Y. Wang, S. Hu, Exploiting high level feature for dynamic textures recognition, Neurocomputing 154 (2015) 217–224.

[52] A. R. Rivera, O. Chae, Spatiotemporal directional number transitional graph for dynamic texture recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (10) (2015) 2146–2152.

[53] H. Ji, X. Yang, H. Ling, Y. Xu, Wavelet domain multifractal analysis for static and dynamic texture classification, IEEE Transactions on Image Processing 22 (1) (2013) 286–299.

[54] Y. Wang, S. Hu, Chaotic features for dynamic textures recognition, Soft Computing 20 (5) (2016) 1977–1989.