

RESEARCH

Open Access

# Local spatiotemporal features for dynamic texture synthesis

Rocio A Lizarraga-Morales<sup>1\*</sup>, Yimo Guo<sup>2</sup>, Guoying Zhao<sup>2</sup>, Matti Pietikäinen<sup>2</sup> and Raul E Sanchez-Yanez<sup>1</sup>

## Abstract

In this paper, we study the use of local spatiotemporal patterns in a non-parametric dynamic texture synthesis method. Given a finite sample video of a texture in motion, dynamic texture synthesis may create a new video sequence, perceptually similar to the input, with an enlarged frame size and longer duration. In general, non-parametric techniques select and copy regions from the input sample to serve as building blocks by pasting them together one at a time onto the outcome. In order to minimize possible discontinuities between adjacent blocks, the proper representation and selection of such pieces become key issues. In previous synthesis methods, the block description has been based only on the intensities of pixels, ignoring the texture structure and dynamics. Furthermore, a seam optimization between neighboring blocks has been a fundamental step in order to avoid discontinuities. In our synthesis approach, we propose to use local spatiotemporal cues extracted with the local binary pattern from three orthogonal plane (LBP-TOP) operator, which allows us to include in the video characterization the appearance and motion of the dynamic texture. This improved representation leads us to a better fitting and matching between adjacent blocks, and therefore, the spatial similarity, temporal behavior, and continuity of the input can be successfully preserved. Moreover, the proposed method simplifies other approximations since no additional seam optimization is needed to get smooth transitions between video blocks. The experiments show that the use of the LBP-TOP representation outperforms other methods, without generating visible discontinuities or annoying artifacts. The results are evaluated using a double-stimulus continuous quality scale methodology, which is reproducible and objective. We also introduce results for the use of our method in video completion tasks. Additionally, we hereby present that the proposed technique is easily extendable to achieve the synthesis in both spatial and temporal domains.

**Keywords:** Dynamic texture synthesis; Spatiotemporal descriptor; Non-parametric synthesis

## Introduction

Texture synthesis is an active research area with wide applications in fields like computer graphics, image processing, and computer vision. The texture synthesis problem can be stated as follows: given a finite sample texture, a system must automatically create an outcome with similar visual attributes of the input and a predefined size. Texture synthesis is a useful alternative way to create arbitrarily large textures [1]. Furthermore, since it is only necessary to store a small sample of the desired texture, the synthesis can bring great benefits in memory storage. Most texture synthesis research has been focused on the enlargement of static textures. However, dynamic texture

synthesis is receiving a growing attention during recent years.

Dynamic textures are essentially textures in motion and have been defined as video sequences that show some kind of repetitiveness in time or space [2,3]. Examples of these textures include recordings of smoke, foliage and water in motion. Comparatively to the static texture synthesis, given a finite video sample of a dynamic texture, a synthesis method must create a new video sequence which looks perceptually similar to the input in appearance and motion. The temporal domain synthesis comprises the duration of the video, while the spatial domain synthesis consists of enlarging the frame size. The synthesis in both domains must keep a natural appearance, avoiding discontinuities, jumps and annoying artifacts.

The number of methods for dynamic texture synthesis that have been proposed can be separated into

\*Correspondence: rocio\_lizarraga@laviria.org

<sup>1</sup>Universidad de Guanajuato DICIS, Salamanca, Guanajuato 36885, Mexico  
Full list of author information is available at the end of the article

two groups: parametric and non-parametric. Parametric methods address the problem as a modeling of a stationary process, where the resulting representation allows to generate a new video with similar characteristics as the input sample [2-7]. A clear disadvantage of these methods is that the estimation of the parameters may not be straightforward, being time-consuming and computationally demanding. Besides, the synthetic outputs may not be realistic enough, showing some blurred results. In contrast, we have the non-parametric methods, also known as exemplar-based techniques, which actually have been the most popular techniques up to now. The success of these methods comes from their outcomes, which look more natural and realistic than those of the parametric methods. Considerable work in non-parametric methods has been developed for dynamic texture synthesis along the time domain. Nevertheless, the synthesis in the spatial domain has not received the same attention.

Non-parametric techniques for texture synthesis were born only for static textures and are categorized into two types: pixel-based and patch-based. Pixel-based methods are the pioneers in non-parametric sampling for static texture synthesis, starting with the innovating work by Efros and Leung [8]. As their name says, the pixel-based methods grow a new image outward from an initial seed one pixel at a time, where each pixel to be transferred to the output is selected by comparing its spatial neighborhood with all neighborhoods in the input texture. This method is time-consuming, and an extension to dynamic texture synthesis might be impractical. In order to attend this issue, Wei and Levoy [9] proposed an acceleration method, and the synthesis of dynamic texture can be achieved. However, the pixel-based methods are susceptible to deficient results because one pixel might not be sufficient to capture the large-scale structure of a given texture [10]. On the other hand, we have the patch-based methods. These techniques select and copy whole neighborhoods from the input, to be pasted one at a time onto the output, increasing the speed and quality of the synthesis results. In order to avoid discontinuities between patches, their representation, selection, and seaming processes become key issues. The main research trend in these methods is to minimize the mismatches on the boundaries between patches, after their placement in the corresponding output. The new patch can be blended [11] with the already synthesized portion, or an optimal cut can be found for seaming the two patches [12,13]. The patch-based methods have shown the best synthesis results up to now. However, some artifacts are still detected in their outputs. The reason is that in these methods, there has been more attention on the patch seaming than in the patch representation for a better selection and matching. We must select the patch that matches the best with the preceding one, depending on a

given visual feature. Usually, only the color features have been considered. This assumption may result in structural mismatches along the synthesized video. Recent approaches for static texture synthesis have proposed the use of structural features [10,14], thereby preserving the texture appearance by improving the representation and selection of patches. Nonetheless, an extension of these methods to dynamic texture synthesis has not been considered. To our knowledge, there are no synthesis methods of dynamic texture that explore the use of features that consider structure, appearance, and motion for the patch representation and selection.

In this paper, we propose the use of local spatiotemporal patterns [15], as features in a non-parametric patch-based method for dynamic texture synthesis. The use of such features allows us to capture the structure of local brightness variations in both spatial and temporal domains and, therefore, describe appearance and motion of dynamic textures. In our method, we take advantage of these patterns in the representation and selection of patches. With this improvement, we capture more structural information for a better patch matching, preserving properly the structure and dynamics of the given input sample. In this way, we can simplify the synthesis method with a very competitive performance in comparison with other patch-based methods. The main contributions of this paper are the following: (1) the extension of a patch-based approach, previously applied only for static texture, for its use in dynamic texture synthesis; (2) the dynamic texture description through local spatiotemporal features, instead of using only the color of the pixels. With this improvement, we capture the local structural information for a better patch matching, preserving properly the appearance and dynamics of a given input texture. (3) A simplified method, where the computation of an optimal seam between patches can be omitted. This can be achieved because of the fitting and matching of patches. (4) A robust and flexible method that can lead to different kinds of dynamic texture videos, ranging from videos that show spatial and temporal regularity, those conformed by constrained objects and videos that contain both static and dynamic elements, showing irregularity in both appearance and motion. (5) A combination with a temporal domain synthesis method in such a way that we can perform the synthesis in both the spatial and temporal domains. (6) The use of such method for video completion tasks.

It must be mentioned that this is a formal review and an extension of our previous work on synthesis only in the spatial domain [16]. In this study, we have carried out new tests that show our contribution more objectively. New results were obtained to test the boundaries of our proposal, using different types of dynamic textures for synthesis. For the evaluation of the results, the

previous work only includes personal comments of the quality achieved, while in the current manuscript, the results are evaluated using a double-stimulus continuous quality scale (DSCQS) methodology for the video quality assessment, which is reproducible and objective. A comparison with other state-of-the-art parametric and non-parametric approaches, following the same assessment methodology, is also presented in this manuscript. Furthermore, we hereby present results for the use of our method in constrained synthesis tasks, where missing parts in a given video can be considered as holes that must be filled. Moreover, we also propose results for the combination of our technique with a temporal synthesis method, in order to achieve the synthesis in both spatial and temporal domains. This last change is important since previous methods are mainly focused only on synthesis in temporal domain.

This paper is organized as follows: in the section ‘Dynamic texture synthesis using local spatiotemporal features’, the local spatiotemporal features used in this work and the proposed approach are defined. In the ‘Experiments and results’ section, we include a set of tests using a standard database of dynamic textures, a comparison with other parametric and non-parametric approaches in order to validate our method, and the application of our algorithm in constrained synthesis tasks. Finally, ‘Conclusions’ section presents a summary of this work and our concluding remarks.

### Dynamic texture synthesis using local spatiotemporal features

In this section, the spatiotemporal features used for the representation of dynamic texture patches are defined. Also, the proposed method for the texture synthesis in the spatial domain is described. After that, with the combination of a method in spatial domain and a temporal domain approach, we can achieve a full synthesis in both spatial and temporal domains.

#### The spatiotemporal descriptor

The local binary pattern from three orthogonal planes (LBP-TOP) [15] is a spatiotemporal descriptor for dynamic textures. The LBP-TOP considers the co-occurrences in three planes  $XY$ ,  $XT$  and  $YT$ , capturing information about space-time transitions. The LBP-TOP is an extension of the local binary patterns (LBP) presented by Ojala et al. [17]. As it is known, the LBP is a theoretically simple yet an efficient approach to characterize the spatial structure of a local texture. Basically, the operator labels a given pixel of an image by thresholding its neighbors in function of the pixel intensity and summing the thresholded values weighted by powers of two. According to Ojala et al., a monochrome texture image  $T$  in a local neighborhood is defined as the joint

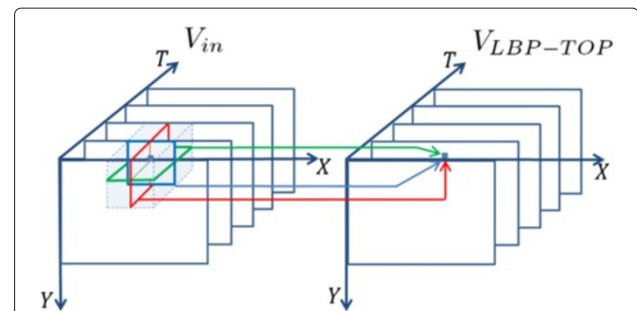
distribution of the gray levels of  $P(P > 1)$  image pixels  $T = t(g_c, g_0, \dots, g_{P-1})$ , where  $g_c$  is the gray value of the center pixel and  $g_p (p = 0, 1, \dots, P - 1)$  are the gray values of  $P$  equally spaced pixels on a circle radius  $R (R > 0)$  that form a circularly symmetric neighbor set. If the coordinates of  $g_c$  are  $(x_c, y_c)$ , then the coordinates of  $g_p$  are  $(x_c - R \sin(2\pi p/P), y_c + R \cos(2\pi p/P))$ . The LBP value for the pixel  $g_c$  is defined as in Equation 1:

$$\text{LBP}_{P,R}(g_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \quad s(t) = \begin{cases} 1, & t \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

More details can be further consulted in [17].

For the spatiotemporal extension of the LBP, named as LBP-TOP, the local patterns are extracted from the  $XY$ ,  $XT$ , and  $YT$  planes, with  $XY$  the frame plane and  $XT$  and  $YT$  the temporal variations planes. Each code is denoted as  $XY$ -LBP for the space domain, and  $XT$ -LBP and  $YT$ -LBP for space-time transitions [15]. In the LBP-TOP approach, the three planes intersect in the center pixel, and three different patterns are extracted in function of that central pixel (see Figure 1). The local pattern of a pixel from the  $XY$  plane contains information about the appearance. In the local patterns from  $XT$  and  $YT$  planes, statistics of motion in horizontal and vertical directions are included. In this case, the radii in axes  $X$ ,  $Y$ , and  $T$  are  $R_X$ ,  $R_Y$ , and  $R_T$ , respectively, and the number of neighboring points in each plane is defined as  $P_{XY}$ ,  $P_{XT}$ , and  $P_{YT}$ . Supposing that the coordinates of the center pixel  $g_{t_c,c}$  are  $(x_c, y_c, t_c)$ , the coordinates of the neighbors  $g_{XY,p}$  in the plane  $XY$  are given by  $(x_c - R_X \sin(2\pi p/P_{XY}), y_c + R_Y \cos(2\pi p/P_{XY}), t_c)$ . Analogously, the coordinates of  $g_{XT,p}$  in the plane  $XT$  are  $(x_c - R_X \sin(2\pi p/P_{XT}), y_c, t_c - R_T \cos(2\pi p/P_{XT}))$ , and the coordinates of  $g_{YT,p}$  on the plane  $YT$  are  $(x_c, y_c - R_Y \cos(2\pi p/P_{YT}), t_c - R_T \sin(2\pi p/P_{YT}))$ .

For the implementation proposed in this paper, each pixel in the input sequence  $V_{in}$  is analyzed with the



**Figure 1** Creation of the LBP-TOP-coded sequence. Each pixel in the corresponding LBP-TOP sequence is obtained by extracting the LBPs from the three orthogonal planes in the input sequence.

LBP-TOP<sub>P<sub>XY</sub>,P<sub>XT</sub>,P<sub>YT</sub>,R<sub>XY</sub>,R<sub>XT</sub>,R<sub>YT</sub></sub> operator, in such a way that we obtain a LBP-TOP-coded sequence  $V_{\text{LBP-TOP}}$ . Each pixel in the  $V_{\text{LBP-TOP}}$  sequence is coded by three values, comprising each of the space-time patterns of the local neighborhood, as can be seen in Figure 1. As we said before, in patch-based methods, each patch must be carefully selected, depending on a given visual feature. To accomplish this task, we use  $V_{\text{LBP-TOP}}$  as a temporary sequence for the patch description in the selection process. This means that instead of comparing the similarity of patches using only the intensity, we compare them using their corresponding LBP values in  $V_{\text{LBP-TOP}}$ .

### Dynamic texture synthesis in spatial domain

In this paper, as we said before, we propose the synthesis of dynamic textures using local spatiotemporal features [15] in a patch-based method for dynamic texture synthesis. As mentioned, patch-based algorithms basically select regions from the input as elements to build an output. Since our method synthesizes textures in motion, we take video volumes as such building blocks to obtain the desired sequence. The selection of these volumes is crucial, in order to obtain a high quality synthesis and smooth transitions. In our approach, we achieve this by including local structural information for a better video volume matching. The use of this information allows us to consider the local spatial and temporal relations between pixels and, therefore, get more insight about the structure of a given dynamic texture.

In general, our method can be described in an algorithmic manner: the synthesized output video is built by sequentially pasting video volumes or blocks in raster scan order. In each step, we select a video block  $B_k$  from the input video  $V_{\text{in}}$  and copy it to the output  $V_{\text{out}}$ . To avoid discontinuities between adjacent volumes, we must carefully select  $B_k$  based on the similarity of its spatiotemporal cues and the features of the already pasted neighbor  $B_{k-1}$ . At the beginning, a volume  $B_0$  of  $W_x \times W_y \times W_t$  pixel size is randomly selected from the input  $V_{\text{in}}$  and copied to the upper left corner of the output  $V_{\text{out}}$ . The following blocks are positioned in such a way that they are partially overlapped with the previously pasted ones. The overlapped volume between two blocks is of size  $O_x \times O_y \times O_t$  pixels. If the input sample  $V_{\text{in}}$  is of  $V_x \times V_y \times V_t$  pixel size, we set the synthesis block size as  $W_x \times W_y \times W_t$ , and the overlapped volume of two adjacent blocks as  $O_x \times O_y \times O_t$ . In this process we consider  $V_t = W_t = O_t$ .

In Figure 2, the following elements are illustrated: a video block, the boundary zone, where two video blocks should match, and an example of the overlapped volume between two blocks. In Figure 2b, the selected block  $B_k$  has a boundary zone  $E_{B_k}$ , and the previously pasted volume in  $V_{\text{out}}$  has a boundary zone  $E_{\text{out}}$ .

According to our method and in order to avoid discontinuities,  $E_{B_k}$  and  $E_{\text{out}}$  should have similar local spatiotemporal structure properties. It could be easy to think that the video block that matches the best is selected to be pasted on the output. However, as it was pointed out by Liang et al. [11], for the case of static texture synthesis, this can lead to a repeatability of patterns, and some randomness in the output is desirable.

In order to accomplish a better video block selection and preserve a certain degree of randomness in the outcome, we build a set of candidate video patches  $A_B$ . This set is built by elements which are considered to match with previously pasted volumes with some tolerance. Then, we select one block randomly from this set. Let  $B_{(x,y,t)}$  be a volume whose upper left corner is at  $(x, y, t)$  in  $V_{\text{in}}$ . We construct

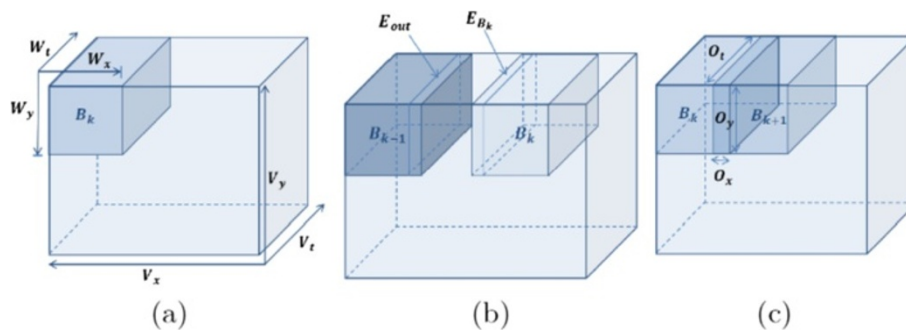
$$A_B = \{B_{(x,y,t)} | d(E_{B_{(x,y,t)}}, E_{\text{out}}) < d_{\text{max}}\}, \quad (2)$$

where  $E_{B_{(x,y,t)}}$  is the boundary zone of  $B_{(x,y,t)}$ , and  $d_{\text{max}}$  is the distance tolerance between two boundary zones. Details on how to compute  $d(\cdot)$  are given later.

When we have determined all the potential blocks, we pick one randomly from  $A_B$  to be the  $k$ th video block  $B_k$  to be pasted on  $V_{\text{out}}$ . The cardinality of  $A_B$  depends on how many video blocks from the input satisfy the similarity constraints given by  $d_{\text{max}}$ . With a low value of  $d_{\text{max}}$ , the output will have a better quality, but few blocks will be considered to be part of  $A_B$ . By contrast, with a high tolerance, a big number of blocks will be part of the set and there will be more options to select, but the quality of the output will be compromised. For a given  $d_{\text{max}}$ , the set  $A_B$  could be empty. In such case, we choose  $B_k$  to be the block  $B_{(x,y,t)}$  from  $V_{\text{in}}$ , whose boundary zone  $E_{B_{(x,y,t)}}$  has the smallest distance to the boundary zone of the output  $E_{\text{out}}$ . In our implementation, the similarity constraint  $d_{\text{max}}$  is set to be  $d_{\text{max}} = 0.01 VL$ , where  $V$  is the number of pixels in the overlapped volume and  $L$  is the maximum LBP value in the sequence.

The final computation of the overlapping distance between a given block  $E_{B_{(x,y,t)}}$  and the output  $E_{\text{out}}$  is estimated by using the L2 norm through the corresponding LBP values for each pixel. As it was mentioned before, we use the  $V_{\text{LBP-TOP}}$  representation as a temporary sequence for the patch description. The input sequence  $V_{\text{in}}$  is analyzed with the LBP-TOP<sub>P<sub>XY</sub>,P<sub>XT</sub>,P<sub>YT</sub>,R<sub>XY</sub>,R<sub>XT</sub>,R<sub>YT</sub></sub> operator, in such a way that we obtain a LBP-TOP-coded sequence  $V_{\text{LBP-TOP}}$ . Here, we set the operator to be LBP-TOP<sub>8,8,8,1,1,1</sub>, and therefore, each pixel from the  $V_{\text{in}}$  sequence is coded in  $V_{\text{LBP-TOP}}$  by three basic LBP values, one for each orthogonal plane. The overlapping distance is defined as

$$d(E_{B_{(x,y,t)}}, E_{\text{out}}) = \left[ \frac{1}{V} \sum_{i=1}^V \sum_{j=1}^3 \left[ p_{B_{(x,y,t)}}^j(i) - p_{\text{out}}^j(i) \right]^2 \right]^{1/2}, \quad (3)$$



**Figure 2** Illustration of the building blocks. Examples of (a) a video block, (b) the boundary zone of two different video volumes, and (c) the overlapped volume between two blocks. The boundary zones must have similar LBP-TOP features.

where  $V$  is the number of pixels in the overlapped volume.  $p_{B(x,y,t)}^j(i)$  and  $p_{out}^j(i)$  represents the LBP values of the  $i$ th pixel on the  $j$ th orthogonal plane, respectively. For color textures, we compute the LBP-TOP codes for each color channel. In this paper, we use the RGB color space, and the final overlapping distance is the distance average for all the color components.

In summary, the creation of the output is shown in Figure 3, where the three possible configurations of the overlapping zones are also shown. Figure 3a presents the step where the second block is pasted, and the boundary zone is taken only on the left side. Figure 3b shows when  $B_k$  is the first block in the second or subsequent rows, and the boundary is taken on the upper side of it. The third case, illustrated in Figure 3c, is when  $B_k$  is not the first on the second or subsequent rows. Here, the total distance is the addition of the distances from the above and left boundaries.

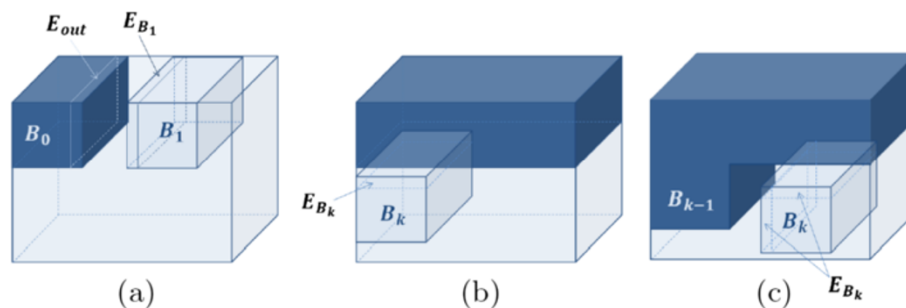
It is important to mention that the size of a given block and the overlapped volumes are dependent on the properties of a particular texture. The size of the video block must be appropriate; it must be large enough to represent the structural composition of texture, but at the same time, small enough to avoid redundancies. This

characteristic makes our algorithm flexible and controllable. In our proposal, this parameter is adjusted empirically, but still, it can be automatically approximated with methods to obtain the fundamental pattern or texel size, like the ones proposed in [18,19]. In the same way, the boundary zone should avoid mismatching features across the borders, but at the same time, be tolerant to the border constraints. The overlapping volume is a small fraction of the block size. In our experiments, we take one sixth of the total patch size volume.

On the overlapped volume, in order to obtain smooth transitions and minimize artifacts between two adjacent blocks, we blend the volumes using a feathering algorithm [20]. This algorithm set weights to the pixels for attenuating the intensity around the blocks' boundaries using a ramp style transition. As a result, possible discontinuities are avoided, and soft transitions are achieved.

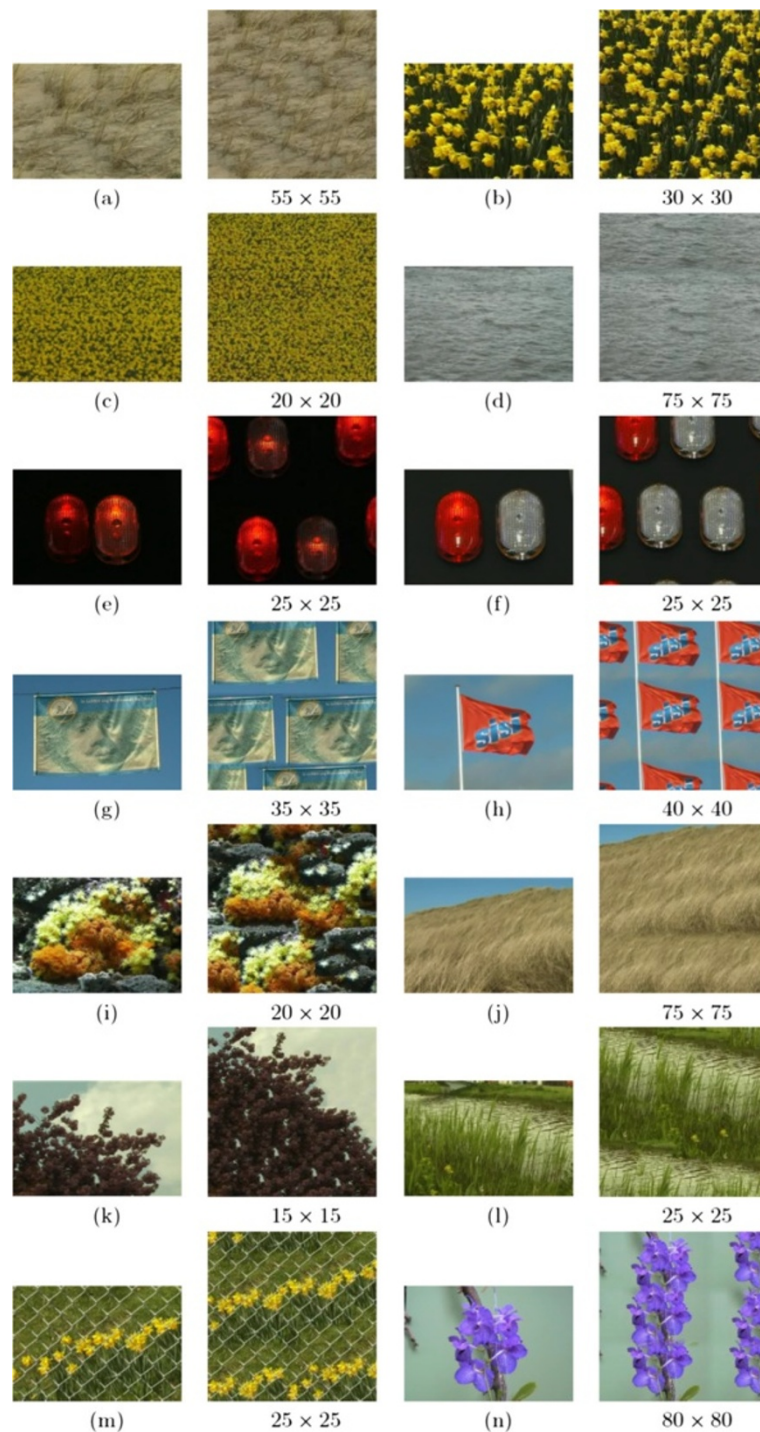
### Experiments and results

In this section, we present series of tests that have been accomplished in order to evaluate the performance of our method. At first, an assessment of performance is made on a variety of dynamic textures. Next, comparisons between the proposed approach with other four state-of-



**Figure 3** Illustration of our patch-based method process for dynamic texture synthesis. The darker zone is the already synthesized portion of video. Implicitly, three possible overlapping zones between the output  $E_{out}$  and the new block  $E_k$  are also shown: (a) the overlapping zone is on the left side of  $E_k$ , (b) the overlapping zone is taken from the upper side of  $E_k$ , and (c) the overlapping zone is taken from the upper and left sides.

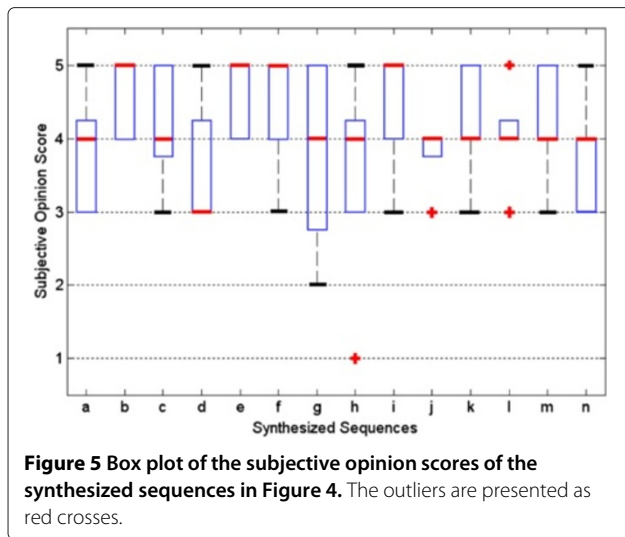




**Figure 4 Results of the synthesis in spatial domain. (a to n)** Frames taken from the original sequence and the corresponding synthesis results. The block size used for obtaining such results is shown for each sequence.

the-art methods are made to validate its application. The comparison is carried out with both parametric and non-parametric methods. The parametric methods presented are proposed by Bar-Joseph et al. [21] and by Costantini et al. [4]. For comparison with non-parametric methods,

we have included the pixel-based technique proposed by Wei and Levoy [9] and the patch-based approach introduced by Kwatra et al. [13]. The selected baseline methods were selected for their impact to the dynamic texture synthesis field. Specifically, the method presented in [13]



is the most popular and recent approach that achieves the synthesis of dynamic textures in both time and space domains using a non-parametric technique. Moreover, it is more similar to our proposal, since it uses a patch-based sampling. Afterwards, we also consider our method for a video completion task. All the resulting videos are available on the website <https://dl.dropboxusercontent.com/u/13100121/SynResults.zip>.

### Performance on a variety of dynamic textures

In the first experiment, a set of 14 videos was selected for evaluating our approach performance on different types of dynamic textures. The videos were selected from the DynTex database [22], which provides a comprehensive range of high-quality dynamic textures and can be used for diverse research purposes. In Figure 4, a frame ( $176 \times 120$  pixel size) taken from each original video is presented. The selected sequences correspond to videos that show spatial and temporal regularities (a to d), constrained objects with temporal regularity (e to h), and videos that show some irregularity in either appearance or motion (i to n). In this context, the term regularity can be interpreted as the repeatability of a given pattern along one dimension, such as of a texture primitive or a movement cycle. Next to each original frame, the resulting synthesized outputs, enlarged to  $200 \times 200$  pixel size, are presented. The spatial dimensions of the block  $W_x \times W_y$  used for synthesis are shown below each image. The temporal dimension  $W_t$  of each block corresponds to the time duration of the given

input sample, so that from now the  $W_t$  parameter will be obviated.

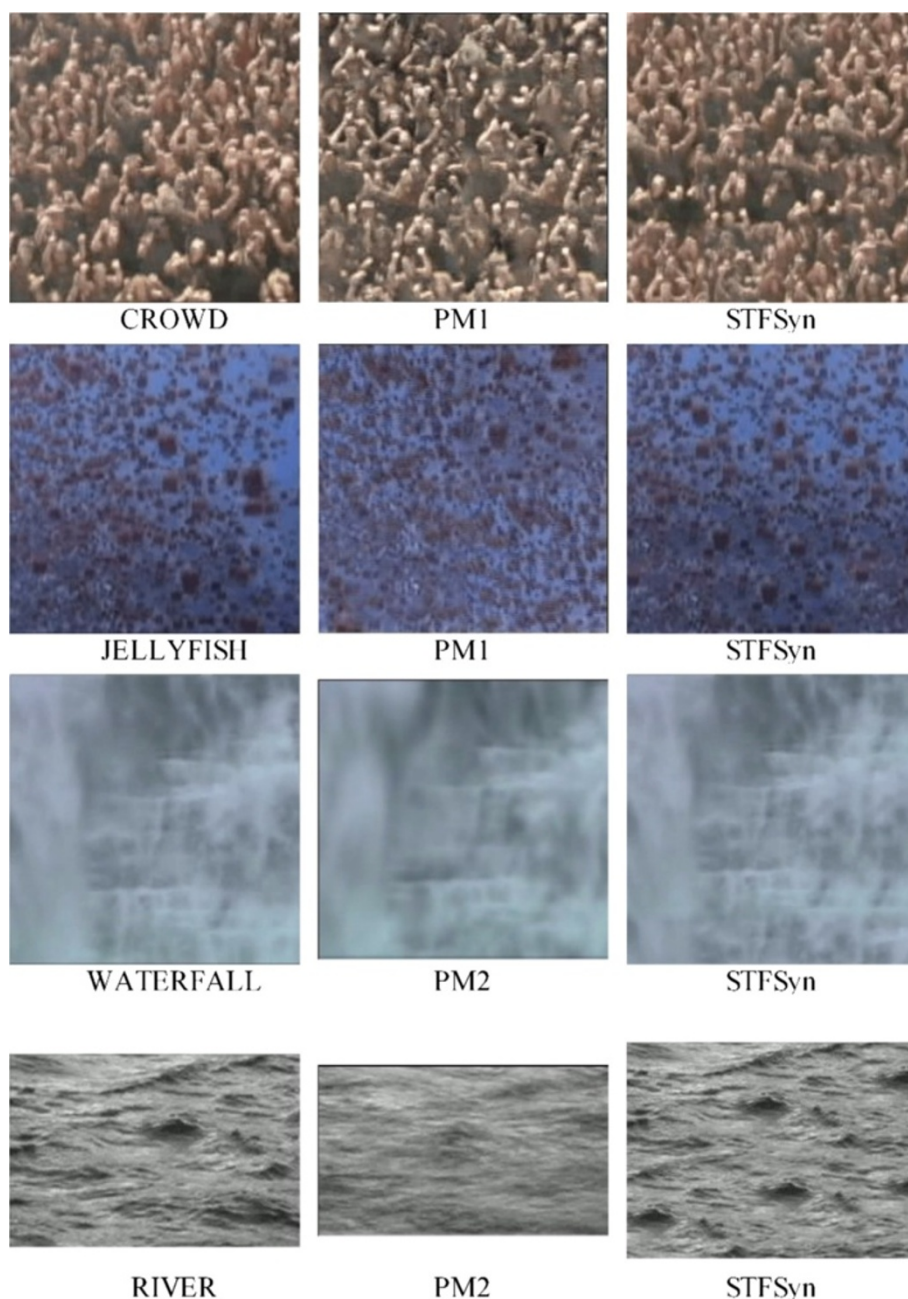
As we can observe in Figure 4a,b,c,d, our method preserves the spatiotemporal regularity of the input, and the borderlines between blocks are practically invisible. It is worth mentioning that in our method, we do not need to compute an additional optimal seam on the borders between the adjacent video volumes to achieve smooth transitions, such as the optimal cut used in [13]. This soft transition in our outcomes is achieved through the proper selection of blocks that have similar spatiotemporal features using the LBP-TOP representation. The sequences shown in Figure 4e,f,g,h are different in the sense that they are composed by constrained objects. In these examples, it is important that the structure of these objects can be maintained in the output, where we aim to generate an array of these objects. As it is pointed out by the results, our method can keep the shape and structure of a given object without generating any discontinuity.

The last set of examples, shown in Figure 4i,j,k,l,m,n, is of great interest since it shows videos that contain both static and dynamic elements, showing irregularity in both appearance and motion. This is a very common characteristic in real sequences. As far as we know, there are no proposals to handle this kind of videos since in general, previous methods for dynamic texture synthesis assume some spatial homogeneity. Specifically, the example shown in Figure 4m is interesting because it contains a lattice in front of the flowers in motion. In the resulting video, we can see that the structure of the lattice is completely maintained without any discontinuity. Furthermore, the appearance and dynamics of the rest of the elements, both static and in motion, from the original video are preserved.

A quantitative and reliable evaluation of texture synthesis is not an easy task. The final goal of synthesis is to achieve a high-quality result, able to trick the human visual perception. However, the perception of quality differs from person to person, and moreover, there are usually more than one acceptable outcome. Therefore, in this paper, we consider that a subjective evaluation is the most appropriate. We propose to carry a subjective assessment, where a set of test subjects are asked to give their opinion about the perceived quality of a given video. The synthetic sequences are subjectively evaluated using the DSCQS methodology [23], provided by the International Telecommunication Union through the recommendation ITU-R BT.500-11, for a subjective assessment of

**Table 1** Average performance of our method

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	Average
$\mu$	3.8	4.6	4.1	3.6	4.5	4.5	3.6	3.6	4.4	3.7	4.3	4.1	4.3	3.7	4.1
$\sigma$	0.78	0.5	0.78	0.86	0.52	0.72	1.22	1.22	0.72	0.44	0.70	0.60	0.70	0.66	0.74



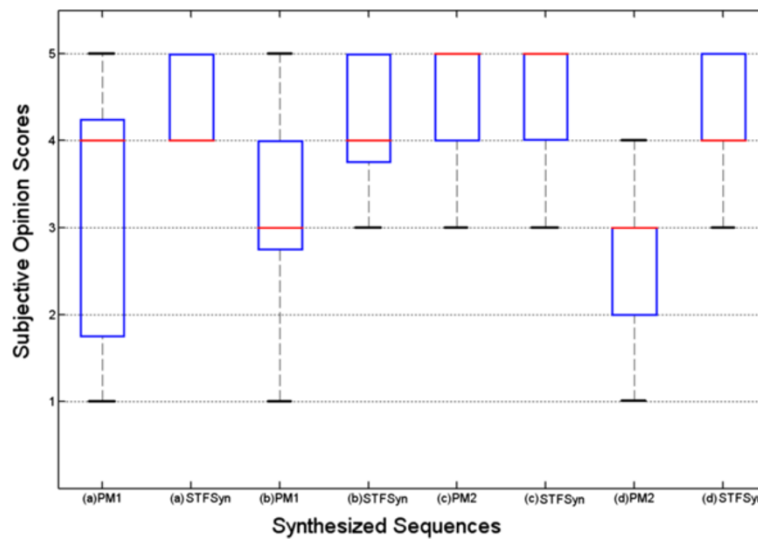
**Figure 6** Visual comparison between the original video and the results of parametric techniques and our method. The original video is presented in the first column, while the resulting clips of the parametric techniques PM1 and PM2 are shown in the second column. The resulting sequence of our method STFSyn is presented in the third column.

the quality of videos. This is a measure given by a number of subjects on how well the image characteristics of the original clip are faithfully preserved in the synthesized video. The measure is presented as a five-point quality scale. The scores correspond to the following: 1 BAD, 2 POOR, 3 FAIR, 4 GOOD, and 5 EXCELLENT. The main advantage of this evaluation method is that it

turns the subjective tests in reproducible and objective evaluations.

The testing protocol is described as follows. We asked for 15 non-expert volunteers to participate in the experiments. All of these subjects neither were part of our team nor are related to texture synthesis work. This is important to mention since non-expert observers yield more





**Figure 7** Box plot of the subjective opinion scores of quality for each synthesized sequence in Figure 6. The clips are (a) JELLYFISH, (b) CROWD, (c) WATERFALL, and (d) RIVER.

critical evaluations about the synthesis quality. We placed the input texture video and the corresponding output on a screen side by side and asked our subjects to rate the quality of the video.

The results for the opinion scores of the synthesized videos are presented in Figure 5, where each box is the result of subjective opinions for each sequence presented in Figure 4. The box plots help indicate the degree of dispersion of the corresponding data samples, and the median and outliers (red cross) of the samples can be easily identified. In this box plot, we can see that the median of the opinion scores of most of our sequences ranges between 4 (GOOD) and 5 (EXCELLENT). It was observed that only the synthetic video (d) has a median on FAIR. The corresponding output of video (g) not only has received the lowest score of 2 (POOR) but also has received the highest of 5 (EXCELLENT). A very low outlier of 1 (BAD) is detected in the synthetic video (h). This means that only one subject considers this video as bad. The opinion scores were also statistically evaluated; the mean value ( $\mu$ ) and standard deviation ( $\sigma$ ) of the opinions are computed to determine the total average results. These data can be consulted on Table 1, where we can see that the total mean quality achieved by our method is of 4.1, interpreted as 4 (GOOD), for all the test sequences. We can also observe that there is a low variation of opinions, with a value of  $\sigma$  of 0.74.

### Performance comparison

The second experiment is a comparison with other state-of-the-art methods. We have compared our approach (called STFSyn from now on, for spatiotemporal

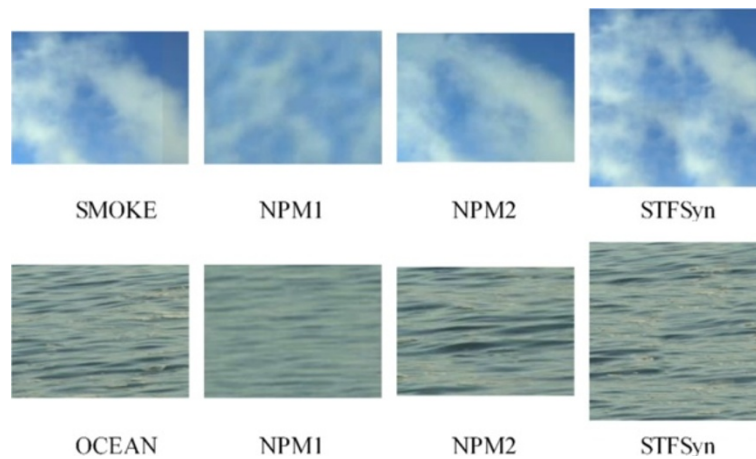
feature-based synthesis) with both parametric and non-parametric approaches. We have borrowed the sequences used by other methods for testing their approaches and feed our method with such inputs. This is to compare the resulting quality achieved by the different methods. All the synthetic videos created for comparison purposes were also assessed with the DSCQS methodology. Each synthetic video is placed next to the original one and submitted for evaluation of the same 15 non-expert subjects who would score the perceived quality in comparison with the original one. In this part of the experiment, each stimulus is randomly presented to the subjects, without telling them the method implemented to obtain such outcome. The parameters used by our method STFSyn are reported in each description.

For comparison with parametric methods, we have borrowed the sequences presented by Bar-Joseph et al. [21] and used by Costantini et al. [4] for their experiments, and execute STFSyn algorithm with such inputs. In Figure 6, a

**Table 2** Performance comparison of our method with the parametric approaches

	JELLYFISH	CROWD	WATERFALL	RIVER	Average
PM1 $_{\mu}$	3.23	3.23	-	-	3.23
PM2 $_{\mu}$	-	-	4.41	2.64	3.53
STFSyn $_{\mu}$	4.32	4.0	4.58	4.29	4.32
PM1 $_{\sigma}$	1.56	1.25	-	-	1.40
PM2 $_{\sigma}$	-	-	0.71	0.93	0.82
STFSyn $_{\sigma}$	0.49	0.74	0.61	0.58	0.60

The best average results are italicized.

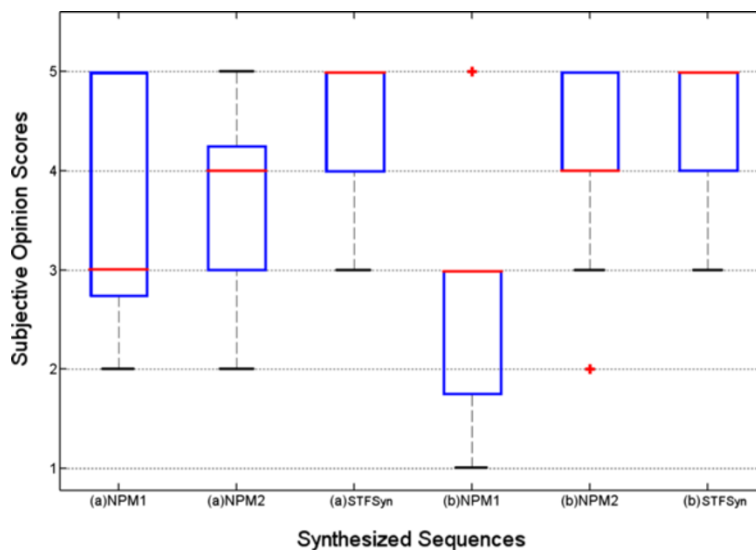


**Figure 8** Visual comparison between the original video, non-parametric techniques and our method. The original video is presented in the first column. The resulting clips of the non-parametric techniques NPM1 and NPM2 are shown in the second and third columns, respectively. Finally, the sequence resulting from our method STFSyn is presented in the fourth column.

frame extracted from the original sequences CROWD and JELLYFISH ( $256 \times 256$  pixel size) used by Bar-Joseph et al. (called as PM1 for parametric method 1) is presented. Next to the original frame, a frame extracted from the sequence obtained from PM1 is also shown. In the third column, a frame from STFSyn results is displayed. In our method, the video CROWD was synthesized with a block of  $70 \times 70$  pixel size, and the JELLYFISH video synthesis required a  $80 \times 80$  pixel size volume. In Figure 6, we also present a frame of the original sequences, WATERFALL and RIVER, reported by Costantini et al. (PM2) in their experiments. A frame from STFSyn results is also displayed. The video WATERFALL was synthesized with

a block of  $85 \times 85$  pixel size, and the RIVER video synthesis required a  $90 \times 90$  pixel size block. Here, it is observed that both parametric methods present artifacts, discontinuities, and are blurred, while the videos generated by the proposed method STFSyn keep a natural look in comparison with the original.

The resulting subjective opinion scores for each synthetic sequence are shown in Figure 7. In this figure, we can see that STFSyn method achieves better performance than the parametric methods. The median of the perceived quality of STFSyn is ranked as 4 (GOOD) and 5 (EXCELLENT), while the median by the Bar-Joseph method (PM1) [21] is only between 3 (FAIR) and 4



**Figure 9** Box plot of the subjective opinion scores of the synthesized sequences in Figure 8. The outliers are presented as red crosses. The clips are (a) SMOKE and (b) OCEAN.

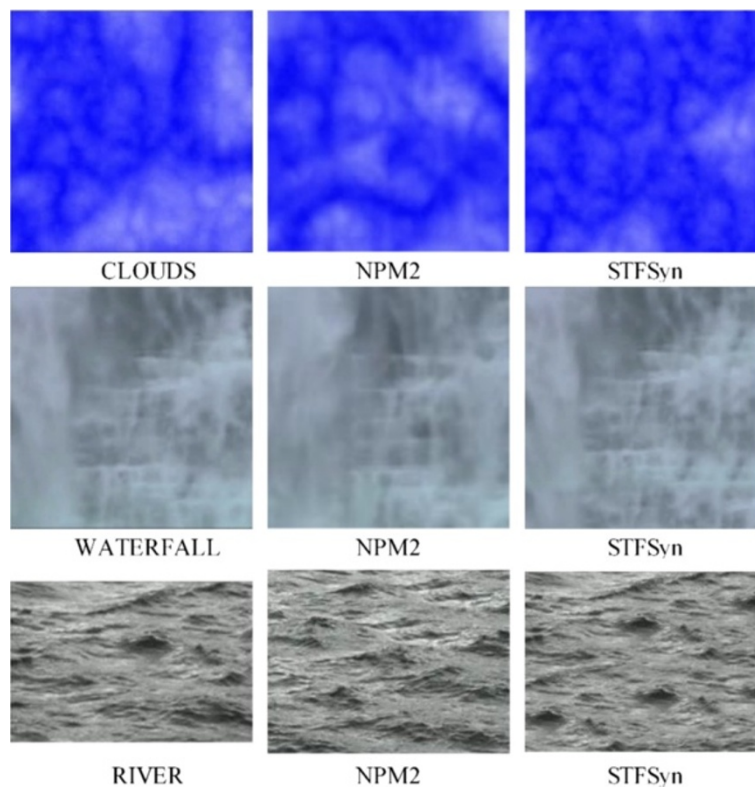
(GOOD). The sequence WATERFALL synthesized by the PM2 [4] has received the same median punctuation than our method as 5 (EXCELLENT). For the video RIVER, the PM1 received a median ranking of 3 (FAIR), while STFSyn was punctuated as 4 (GOOD). The mean ( $\mu$ ) and the standard deviation ( $\sigma$ ) were also computed from the data obtained by each method (PM1, PM2, and our STFSyn) for each video clip. The results are presented in Table 2, where the best results are highlighted in italics. In this table, we can see that the STFSyn approach gets the best average score of 4.32, in comparison to the 3.23 and 3.53 for the PM1 and PM2, respectively. Besides, our approach presents a lower variation in the perceived quality by the subjects, with a  $\sigma$  value of 0.6, in comparison with the 1.40 and 0.82 presented by the PM1 and PM2, respectively.

We have also compared our approach with non-parametric methods. The selected proposals were the two most representative methods, the pixel-based technique proposed by Wei and Levoy [9] and the well-known patch-based method proposed by Kwatra et al. [13]. We have borrowed the sequences named OCEAN and SMOKE used by both Wei and Levoy (NPM1) and Kwatra (NPM2) in their experiments for spatiotemporal synthesis and made a comparison of their quality with our results. The

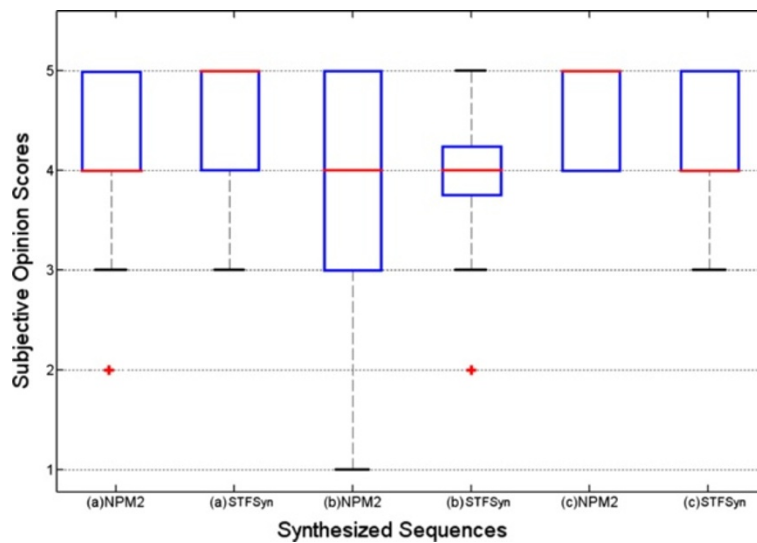
video OCEAN was synthesized with the STFSyn using a block of  $75 \times 75$  pixel size, and the SMOKE video synthesis required a  $95 \times 95$  pixel size block. In Figure 8 frames extracted from the original sample, the result from Wei and Levoy (NPM1) [9], the outcomes by Kwatra et al. (NPM2) [13], and our results are presented. Here, it is observed that the videos obtained by NPM1 are considerably blurred, while the videos generated by NPM2 and by our method keep a natural appearance and motion of the two phenomena.

The corresponding assessment of the results by NPM1 [9] and NPM2 [13] with the DSCQS methodology is presented in Figure 9. In this figure, we can see that for the two sequences, the NPM1 receives a median score of 3 (FAIR); the NPM2 resulting sequences have their median ranked as 4 (GOOD), while our method achieves a median score of 5 (EXCELENT) in both cases.

In a third comparison (see Figure 10), we have borrowed more video sequences reported only by Kwatra et al. (NPM2) for spatiotemporal synthesis. Here, the tested sequences are CLOUDS, WATERFALL, and RIVER. The video CLOUDS was synthesized with a video patch of  $80 \times 80$  pixel size; WATERFALL and RIVER, as it was mentioned before, required a block of  $85 \times 85$  and  $90 \times 90$  pixel



**Figure 10** Visual comparison between the original sequence, the non-parametric technique NPM2 and our method. The original sequence is shown in the first column. The resulting outcome of the non-parametric method NPM2 is presented in the second column, and our method resulting video is shown on the third column.



**Figure 11** Box plot of the subjective opinion scores of the synthesized sequences in Figure 10. The clips are (a) CLOUDS, (b) WATERFALL, and (c) RIVER. The outliers are presented as red crosses.

size, respectively. As it can be seen from the results shown in Figure 10, both NPM2 and our method have generated very competitive and pleasant results for the dynamic texture synthesis. It is very difficult to see if any of the methods generates artifacts or discontinuities. However, the assessment carried out with the DSCQS methodology (see Figure 9 and Figure 11) highlights the differences in quality achieved by each method. In Figure 11, we can observe that both approaches performed with median rankings of 4 (GOOD) and 5 (EXCELENT). However, the NPM2 in the WATERFALL sequence has received evaluations as low as 1 (POOR). It is important to highlight that the clip RIVER was the only one considered by NPM2 for increasing the spatial resolution, process that we have done to every video presented in these comparisons. From all these comparisons, we can observe that the only method that achieves similar quality results is the one presented by Kwatra et al. (NPM2) [13]. The main difference and advantage of our method in comparison with NPM2 is that because of the proper representation of building blocks, our method does not need to compute an optimal seam. This characteristic allows us to simplify the synthesis method.

The corresponding mean ( $\mu$ ) values and standard deviation ( $\sigma$ ) for each video achieved by each non-parametric method (NP1, NP2, and and STFSyn) are detailed in Table 3, where the best average performance is highlighted in italics. In this table, we can see that our method STFSyn in most cases receives higher opinion values than the other non-parametric approaches. Only the NP2 with the clip WATERFALL achieves a slightly superior mean value of 4.58 in comparison with our 4.29 mean opinion score.

The total average for the NP1 is a low 2.99; the average for NP2 is 4.1, while our STFSyn method reaches a total average mean value of 4.32. In the same manner for the standard deviation, the STFSyn approach has the lower value of variation in the scores with a 0.72 total average, in comparison with the 1.23 value by NP1 and 0.91 obtained by NP2.

#### Dynamic texture synthesis in both space and time domains

An extension of the proposed method to achieve a final synthesis in both spatial and temporal domains was also considered in this study. Our method provides the flexibility to be integrated with other existing approaches for synthesis in the temporal domain, like those previously mentioned in the introduction section [13,24,25].

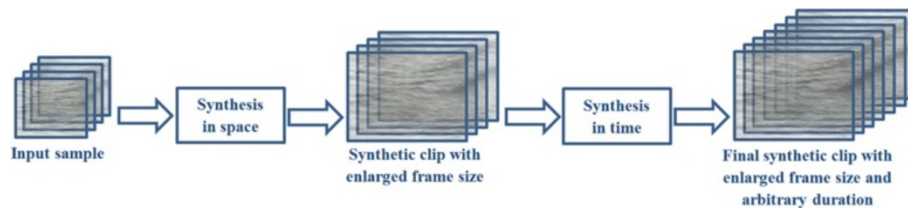
The idea behind these techniques for extending the duration of a video is straightforward yet very effective. The general proposal is to find sets of similar frames that

**Table 3** Performance comparison of our method with non-parametric proposals

	SMOKE	OCEAN	CLOUDS	WATERFALL	RIVER	Average
NPM1 $_{\mu}$	3.47	2.52	-	-	-	2.99
NPM2 $_{\mu}$	3.82	4.05	4.58	3.82	4.23	4.1
STFSyn $_{\mu}$	4.23	4.58	4.29	3.82	4.58	4.32
NPM1 $_{\sigma}$	1.23	1.23	-	-	-	1.23
NPM2 $_{\sigma}$	0.88	0.96	0.50	1.38	0.83	0.91
STFSyn $_{\sigma}$	0.66	0.71	0.58	1.07	0.61	0.72

The best average results are italicized.





**Figure 12** Final procedure to achieve the synthesis in both spatial and temporal domains.

could work as transitions and then loops in the original video to generate a new video stream, making jumps between such matching frames. In the first step of these type of algorithms, the video structure is analyzed for the existence of matching frames within the sequence. The frame similarity is measured in a given image representation, e.g., the intensities of all pixels [13,25] or the intrinsic spatial and temporal features [24]. Depending on the nature of the input video texture, the chosen frame representation method, and the similarity restrictions, the number of matching frames can be either large or small. The implication of this number of frames is that if it is large, a varied number of combinations/transitions can be reached; otherwise, the variability of the transferred clips to the output is compromised. Moreover, under certain motion circumstances, as it is pointed out in [26], some dynamic textures can be pleasantly synthesized in the temporal domain, while others may be not. The reasons include the motion speed and motion periodicity of the input sample. In general, texture synthesis methods in temporal domain are more capable to synthesize dynamic texture that shows repetitive motions, motions with regularity or random motions with fast speed.

The extension presented here is executed by using the algorithm proposed by Guo et al. [24], which has shown to provide high-quality synthesis besides applying LBP-TOP

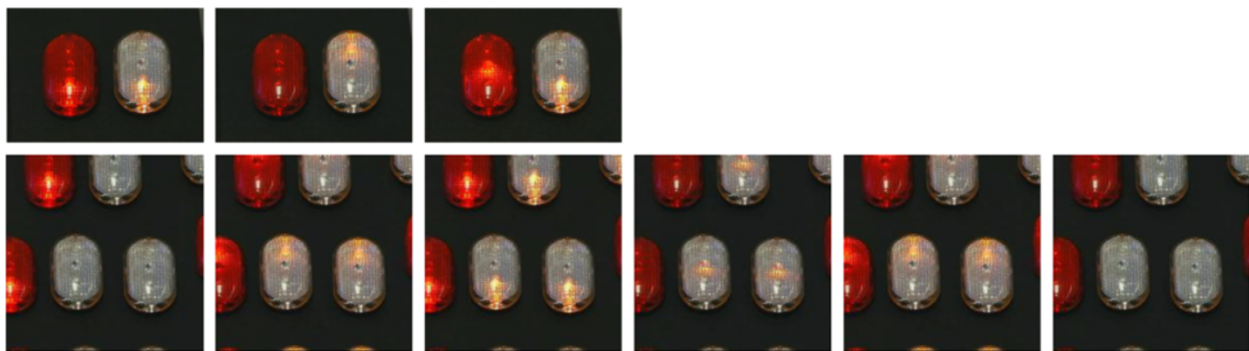
features in the frame representation. The use of LBP-TOP features in the method by Guo et al. proves to capture the characteristics of each frame more effectively, and thus, the most appropriate pairs of frames are found to be stitched together. Additionally, this method is able to preserve temporal continuity and motion without adding annoying jumps and discontinuities.

As is illustrated in Figure 12, for the complete spatiotemporal synthesis, we apply our method and the method of Guo et al. [24] in cascade. We first execute our method for spatial synthesis and the enlargement of the frame size; after that, we perform the extension in the temporal domain.

In the experiments, we take sample videos from the DynTex database that have a duration of 10 s (150 frames), with a frame size of  $176 \times 120$  pixel size. The final result after the spatiotemporal synthesis is a video with 20 s (300 frames) of duration and a frame of  $200 \times 200$  pixel size, noticing that any duration and size can be achieved. In Figure 13, we show an example of the result. More results can be consulted in the results repository at the web page previously cited.

#### Video completion with dynamic texture synthesis

Video completion is the task of filling in the missing or damaged regions (in either spatial or temporal domain),



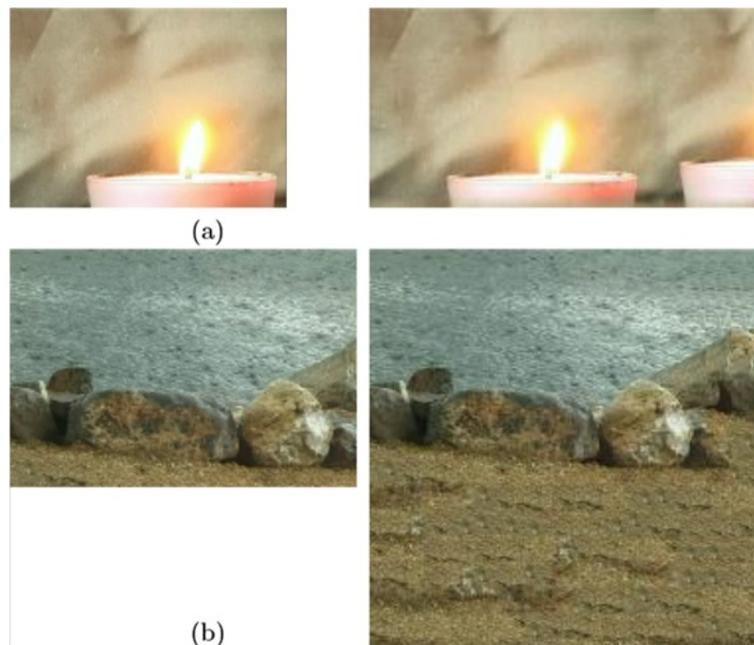
**Figure 13** Example of the spatiotemporal synthesis. First row: three frames of the original sequence (frames 30, 60, and 90). Second row: six frames of the synthesized sequence (frames 30, 60, 120, 180, 240, and 300).

with the available information from the same video. This, with the goal of generating a perceptually smooth and satisfactory result. There is a number of applications for this task. Such applications may include the following: video post-production and to fill large holes in damaged videos for restoration. Also, it can be applied to the problem of dealing with boundaries after a convolution process. The most common approach to deal with these problems is the tiling and reflection of the information in the border; however, this may introduce discontinuities not present in the original image. In many cases, texture synthesis can be used to extrapolate the image by sampling from itself, in order to fill these missing areas.

For testing purposes, we execute our method with simple examples for video completion. The goal of these examples consists on fulfill the boundaries (missing parts) on videos. The boundary constraint is related to the texture that is surrounding the area to be synthesized. This constraint is taken into account in order to avoid subjectively annoying artifacts at the transition between synthetic and natural textures. Two different examples for this task are shown in Figure 14. In these cases, the process is conducted to start the synthesis only for the black holes. In this way, the original video is preserved, while the missing parts are completed with information available in the same sequence. The results show that our method can be also considered for video completion tasks.

## Conclusions

In this paper, the use of local spatiotemporal features for dynamic texture synthesis has been studied. This method explores a patch-based synthesis approach, where the video patch selection is accomplished by taking the corresponding LBP-TOP features, instead of just making use of the intensity of pixels. The LBP-TOP features have the capability of describing the appearance and dynamics of local texture, and thus, a better video block representation can be achieved. The use of this representation leads us to a better matching of adjacent building blocks; as a result, the visual characteristics of the input can be successfully preserved. Moreover, the proposed method is neither difficult to implement nor intricate, since no additional seam optimization is needed to get smooth transitions between video blocks. A final extension to the synthesis in both spatial and temporal domains has been also considered. This extension can be achieved, applying first the synthesis in space and after that, the elongation in the temporal domain. As the experimental results show, this method produces good synthetic clips on a wide range of types of natural videos. We tested video sequences that show spatiotemporal regularity, videos conformed by constrained objects and those constituted by both static and dynamic elements. According to the results of the evaluation, the performance of the proposed method has shown to be better than other parametric and non-parametric methods. We have also shown that this proposal can be considered for applications requiring video completion.



**Figure 14** Two examples of the use of our method for video completion tasks. (a) The frame of the original video of  $140 \times 100$  pixels size is completed, in order to be of  $200 \times 100$  pixels size. (b) The frame of  $176 \times 120$  pixels size is completed to  $200 \times 200$  pixels size.

### Abbreviations

DSCQS: double-stimulus continuous quality scale methodology; LBP: local binary patterns; LBP-TOP: local binary patterns from three orthogonal planes; NPM1: non-parametric method 1; NPM2: non-parametric method 2; PM1: parametric method 1; PM2: parametric method 2; STFsyn: spatiotemporal features-based synthesis.

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

RAL-M carried out the experiments and drafted the manuscript. YG, GZ and MP conceived of the study and participated in the design of the proposal and the experiments. In addition, they developed the temporal synthesis approach. RES-Y contributed in the design of the experiments and in the revision of the manuscript. All authors read and approved the final manuscript.

### Acknowledgements

The authors would like to thank the Academy of Finland, Infotech Oulu, and the Finnish CIMO for the financial support. In addition, Lizarraga-Morales would like to thank the Mexican CONCyTEG and CONACyT for the grants provided and to the Universidad de Guanajuato through the PIFI 2013 for the financial support.

### Author details

<sup>1</sup>Universidad de Guanajuato DICIS, Salamanca, Guanajuato 36885, Mexico.  
<sup>2</sup>Center for Machine Vision Research, Department of Computer Science and Engineering, University of Oulu, P.O. Box 4500, Oulu FI-90014, Finland.

Received: 1 December 2012 Accepted: 27 February 2014

Published: 26 March 2014

### References

1. LY Wei, S Lefebvre, V Kwatra, G Turk, State of the art in example-based texture synthesis, Eurographics 2009 State of the Art Report (EG-STAR), European Association for Computer Graphics, 2009, pp. 93–117
2. B Ghanem, N Ahuja, Phase PCA for dynamic texture video compression. Paper presented at the IEEE international conference on image processing, San Antonio, TX, USA, 16 Sept–19 Oct 2007, vol. 3, pp. 425–428. doi: 10.1109/ICIP.2007.4379337
3. G Doretto, E Jones, S Soatto, Spatially homogeneous dynamic textures, in *Computer Vision - ECCV'04*, ed. by T. Pajdla and J. Matas. 8th European Conference on Computer Vision, Prague, Czech Republic, May 2004. Lecture Notes in Computer Science. vol. 3022 (Springer, Heidelberg, 2004), pp. 591–602. doi: 10.1007/978-3-540-24671-8\_47
4. R Costantini, L Sbaiz, S Susstrunk, Higher order SVD analysis for dynamic texture synthesis. *IEEE Trans. Image Process.* **17**(1), 42–52 (2008). doi: 10.1109/TIP.2007.910956
5. CB Liu, RS Lin, N Ahuja, MH Yang, Dynamic textures synthesis as nonlinear manifold learning and traversing. Paper presented in the British machine vision conference, Edinburgh, 4–7 Sept 2006, pp. 88.1–88.10. doi: 10.5244/C.20.88
6. L Yuan, F Wen, C Liu, HY Shum, Synthesizing dynamic texture with closed-loop linear dynamic system, in *Computer Vision-ECCV'04*, ed. by T. Pajdla, J. Matas, vol. 3022 (Springer, Heidelberg, 2004), pp. 603–616. doi: 10.1007/978-3-540-24671-8\_48
7. B Ghanem, N Ahuja, Phase based modelling of dynamic textures. Paper presented at the 11th IEEE international conference on computer vision, Rio de Janeiro, Brazil, 14–21 Oct 2007, pp. 1–8. doi: 10.1109/ICCV.2007.4409094
8. A Efros, T Leung, Texture synthesis by non-parametric sampling. Paper presented at the 7th IEEE international conference on computer vision, Kerkyra, Greece, 20–27 Sept 1999, vol. 2, pp. 1033–1038. doi: 10.1109/ICCV.1999.790383
9. LY Wei, M Levoy, Fast texture synthesis using tree-structured vector quantization. Paper presented at the 27th annual conference on computer graphics and interactive techniques, New Orleans, LA USA, 23–28 July 2000 pp. 479–488. doi: 10.1145/344779.345009
10. Y Gui, M Chen, Z Xie, L Ma, Z Chen, Texture synthesis based on feature description. *J. Adv. Mech. Des. Syst. Manuf.* **6**(3), 376–388 (2012). doi: 10.1299/jamdsm.6.376
11. L Liang, YQ Xu, B Guo, HY Shum, Real-time texture synthesis by patch-based sampling. *ACM Trans. Graph.* **20**(3), 127–150 (2001). doi: 10.1145/501786.501787
12. AA Efros, WT Freeman, Image quilting for texture synthesis and transfer. Paper presented at the 28th annual conference on computer graphics and interactive techniques, Los Angeles, CA USA, 12–17 Aug 2001, pp. 341–346. doi: 10.1145/383259.383296
13. V Kwatra, A Schödl, I Essa, G Turk, A Bobick, Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph. (TOG)* **22**(3), 277–286 (2003). doi: 10.1145/882262.882264
14. Q Wu, Y Yu, Feature matching and deformation for texture synthesis. *ACM Trans. Graph. (TOG)* **23**(3), 364–367 (2004). doi: 10.1145/882262.882264
15. G Zhao, M Pietikäinen, Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(6), 915–928 (2007). doi: 10.1109/TPAMI.2007.1110
16. RA Lizarraga-Morales, Y Guo, G Zhao, M Pietikäinen, Dynamic texture synthesis in space with a spatio-temporal descriptor, in *Computer Vision-ACCV'12 Workshops*, ed. Jong-Il Park and Junmo Kim. ACCV 2012 International Workshops, Daejeon, Korea, Nov 2012. Part I, LNCS 7728, 2013, pp. 38–49. doi: 10.1007/978-3-642-37410-4\_4
17. T Ojala, M Pietikäinen, T Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002). doi: 10.1109/TPAMI.2002.1017623
18. V Asha, P Nagabhushan, N Bhajantri, Automatic extraction of texture-periodicity using superposition of distance matching functions and their forward differences. *Pattern Recognit. Lett.* **33**(5), 629–640 (2012). doi: 10.1016/j.patrec.2011.11.027
19. RA Lizarraga-Morales, RE Sanchez-Yanez, V Ayala-Ramirez, Fast texel size estimation in visual texture using homogeneity cues. *Pattern Recognit. Lett.* **34**(4), 414–422 (2013). doi: 10.1016/j.patrec.2012.09.022
20. R Szeliski, HY Shum, Creating full view panoramic image mosaics and environment maps. Paper presented at the 24th annual conference on computer graphics and interactive technique, Los Angeles, CA USA, 3–8 Aug 1997, pp. 251–258. doi: 10.1145/258734.258861
21. Z Bar-Joseph, R El-Yaniv, D Lischinski, M Werman, Texture mixing and texture movie synthesis using statistical learning. *IEEE Trans. Vis. Comput. Graph.* **7**(2), 120–135 (2001). doi: 10.1109/2945.928165
22. R Péteri, S Fazekas, MJ Huiskes, DynTex : a comprehensive database of dynamic textures. *Pattern Recognit. Lett.* **31**, 1627–1632 (2010). doi: 10.1016/j.patrec.2010.05.009
23. JR Ohm, *Multimedia Communication Technology* (Springer, Berlin, 2004)
24. Y Guo, G Zhao, J Chen, M Pietikäinen, Z Xu, Dynamic texture synthesis using a spatial temporal descriptor. Paper presented at the 16th IEEE international conference on image processing, Cairo, Egypt, 7–10 Nov 2009, pp. 2277–2280. doi: 10.1109/ICIP.2009.5414395
25. A Schödl, R Szeliski, DH Salesin, I Essa, Video textures. Paper presented at the 27th annual conference on computer graphics and interactive techniques, New Orleans, LA, USA, 23–28 July 2000, pp. 489–498. doi: 10.1145/344779.345012
26. Y Guo, G Zhao, Z Zhou, M Pietikäinen, Video texture synthesis with multi-frame LBP-TOP and diffeomorphic growth model. *IEEE Trans. Image Process.* **22**(10), 3879–389 (2013). doi: 10.1109/TIP.2013.226314

doi:10.1186/1687-5281-2014-17

Cite this article as: Lizarraga-Morales et al.: Local spatiotemporal features for dynamic texture synthesis. *EURASIP Journal on Image and Video Processing* 2014 **2014**:17.