SMOOTH-IN Variant GAUSSIAN FEATURES FOR DYNAMIC TEXTURE RECOGNITION
Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara

To cite this version:
Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara. SMOOTH-IN Variant GAUSSIAN FEATURES FOR DYNAMIC TEXTURE RECOGNITION. ICIP, Sep 2019, Taipei, Taiwan. hal-02133547

HAL Id: hal-02133547
https://hal.archives-ouvertes.fr/hal-02133547
Submitted on 18 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
SMOOTH-IN Variant GAUSSIAN FEATURES FOR DYNAMIC TEXTURE RECOGNITION

Thanh Tuan Nguyen¹,²,³, Thanh Phuong Nguyen¹,², Frédéric Bouchara¹,²

¹ Université de Toulon, CNRS, LIS, UMR 7020, 83957 La Garde, France
² Aix-Marseille Université, CNRS, ENSAM, LIS, UMR 7020, 13397 Marseille, France
³ HCMC University of Technology and Education, Faculty of IT, HCM City, Vietnam

ABSTRACT
An efficient framework for dynamic texture (DT) representation is proposed by exploiting local features based on Local Binary Patterns (LBP) from filtered images. First, Gaussian smoothing filter is used to deal with near uniform regions and noise which are typical restrictions of LBP operator. Second, the receptive field of Difference of Gaussians (DoG), which is exploited in DT description for the first time, allows to make the descriptor more robust against the changes of environment, illumination, and scale which are main challenges in DT representation. Experimental results of DT recognition on different benchmark datasets (i.e., UCLA, DynTex, and DynTex++), which give outstanding performance compared to the state of the art, verify the interest of our proposal.

Index Terms— Dynamic Texture, Dynamic Texture Recognition, DoG, Gaussian Filter, LBP, CLBP.

1. INTRODUCTION
A dynamic texture is a repetition of its features in a spatio-temporal domain, such as fountain, fire, clouds, blowing flag, trees, waves, etc. Efficiently describing DTs is one of the crucial factors in computer vision applications. Many efforts, which have addressed different techniques for DT representation, can be mainly grouped into six categories as follows. Firstly, optical-flow-based methods [16, 14] effectively capture the characteristics of chaotic motions in natural ways for DT recognition. Secondly, model-based methods [25, 30] are mostly based on Linear Dynamical System [25] to model spatial-temporal features. Thirdly, filter-based methods [3, 24] have taken various filters into account dealing with the problems of noise and illumination in encoding videos. Fourthly, geometry-based methods [31, 32, 21] are focused on fractal analysis of dynamic properties for DT description. Fifthly, learning-based methods are in two main trends: deep learning techniques (e.g., Convolutional Neural Network (CNN)) are primarily utilized for learning features [18, 1, 2] while the kernel sparse coding is taken into account dealing with the problems of noise and illumination in encoding videos. Fourthly, geometry-based methods [31, 32, 21] are focused on fractal analysis of dynamic properties for DT description. Fifthly, learning-based methods are in two main trends: deep learning techniques (e.g., Convolutional Neural Network (CNN)) are primarily utilized for learning features [18, 1, 2] while the kernel sparse coding is taken into account dealing with the problems of noise and illumination in encoding videos. Lastly, local-feature-based methods [12, 26, 27, 29, 28, 13] take advantages of LBP operator in simple and efficient computation for DT representation. Most of them are based on two following variants: Volume LBP (VLBP) [33] and LBP on three orthogonal planes (LBP-TOP) [33].

In spite of obtaining the promising rates on DT recognition tasks, several restrictions in local-feature-based methods have been addressed in following works: illumination and noise problems [3, 11] in filter-based methods; near uniform regions, sensitivity to noise [27, 12]; and large dimensional puzzles [33, 23, 26]. To mitigate these shortcomings, we propose in this paper an efficient framework to encode a DT video in two main stages. First, Gaussian-based kernels (Gaussian and DoG) are applied on each orthogonal plane of the video sequence to capture their smooth and invariant features against illumination and noise problems. Second, a local feature extractor (e.g., CLBP [7]) is exploited to encode the relationships of local patterns in these filtered images. Finally, a prominent descriptor is formed by concatenating and normalizing the obtained probability distributions. Evaluations of the proposed framework on different benchmark datasets in DT classification task validate that our method significantly outperforms in comparison with the state-of-the-art results.

2. PROPOSED METHOD
The typical local-feature-based methods are often limited by the problems of sensitivity to illumination and noise. To deal with those, hereafter, we firstly take a look of LBP and its variants as well as Gaussian-based filtering kernels. We then propose a robust descriptor based on smooth-invariant Gaussian features to enhance discrimination power. It should be noted that DoG is involved in DT encoding for the first time.

2.1. A brief review of LBP and CLBP
Ojala et al. [15] introduced a LBP as a string of binary values to depict the local relationships of textural features in an image. Let $I$ denote a 2D gray-scale image. A LBP code of a center pixel $q_c$ is formed by considering the difference of $q_c$’s gray-level and its local neighbors’ as follows.

$$\text{LBP}_{P,R}(q_c) = \sum_{i=0}^{P-1} f(I(p_i) - I(q_c))2^i$$ (1)

for $i = 0, \ldots, P-1$.

In the CLBP, the function $f$ is replaced by a sigmoid function $f(x) = \frac{1}{1 + e^{-x}}$. The CLBP is computed by convolving the image with Gaussian kernels $g$ and then applying LBP on the resulting image.

$$\text{CLBP}_{P,R}(q_c) = \sum_{i=0}^{P-1} g(i) f(I(p_i) - I(q_c))2^i$$ (2)

where $g$ is a Gaussian kernel with standard deviation $\sigma$. The CLBP is invariant to illumination changes and has a high discriminative power.

In our proposed method, we use a combination of Gaussian and DoG filters to preprocess the input video sequence. First, a Gaussian filter is applied to each orthogonal plane of the video sequence to capture their smooth and invariant features against illumination and noise problems. Second, a local feature extractor (e.g., CLBP) is exploited to encode the relationships of local patterns in these filtered images. Finally, a prominent descriptor is formed by concatenating and normalizing the obtained probability distributions. Evaluations of the proposed framework on different benchmark datasets in DT classification task validate that our method significantly outperforms in comparison with the state-of-the-art results.
in which \( \{ p_i \}_{i=0}^{P-1} \) means \( P \) local neighbors of \( q \) that are sampled by a circle of radius \( R \), \( I(.) \) returns the gray-scale value of a pixel, and function \( f(.) \) is defined as

\[
f(x) = \begin{cases} 
1, & x \geq 0 \\
0, & \text{otherwise.}
\end{cases}
\] (2)

The LBP code has thus large dimension of \( 2^P \) bins. In practice, different mappings are used to reduce the curse of dimensionality: \( u2 \) for capturing uniform patterns, \( riu2 \) for addressing rotation invariant uniform patterns, Local Binary Count [35] - an alternative of uniform features, \( TAP^4 \) mapping [10] for taking into account topological information.

Guo et al. [7] introduced two more complementary parts so that they can be integrated together with LBP to form CLBP model as follows: \( CLBP_S \) that is identical to LBP, \( CLBP_M \) that exploits local variations of magnitudes, and \( CLBP_C \) that addresses the grey-value difference of a center pixel against the average level of the entire image. In practice, the joint of these components (i.e., \( CLBP_S/M/C \)) is utilized because of its effectiveness in performance.

### 2.2. Smooth-Invariant Gaussian features

A Gaussian filter is a convolution filter whose convoluted results comply with the rule of the Gaussian distribution. Its 2D kernel is defined as follows.

\[
G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)
\] (3)

where \( \sigma \) is a standard derivation, and \( x, y \in [-3\sigma, 3\sigma] \) are settings for our implementation. Accordingly, a kernel of the difference of Gaussian filters is defined as

\[
\text{DoG}_{\sigma_1, \sigma_2}(x, y) = G_{\sigma_1}(x, y) - G_{\sigma_2}(x, y)
\] (4)

Two above kernels are used to produce corresponding smooth (\( I_G \)) and invariant (\( I_{\text{DoG}} \)) filtered images as follows.

\[
\begin{align*}
I_G &= G_{\sigma_1}(x, y) * I \\
I_{\text{DoG}} &= |\text{DoG}_{\sigma_1, \sigma_2}(x, y)| * I
\end{align*}
\] (5)

where \( I \) denotes a 2D image texture, \( \sigma_1 < \sigma_2 \), “\( * \)” is the convolution operator. Figure 1 shows an instance of this filtering.

![Fig. 1. A sample of Gaussian filtering. (a): an input gray-scale image; (b), (c): smoothed images of \( \sigma_1 = 0.5, \sigma_2 = 4 \) respectively; (d): the difference of Gaussians of (b) and (c).](image)

**2.3. Proposed DT descriptor**

Given a video \( \mathcal{V} \). For DT representation, we analyze and compute its features in three stages (see Figure 2 for graphical illustration). First, \( \mathcal{V} \) is split into separative images according to its three orthogonal planes \( \{XY, XT, YT\} \). Second, the Gaussian filtering kernels are taken into account on these planes to extract smooth and invariant images against noise and illumination. Finally, a local encoding function is used for each of them to capture spatio-temporal patterns of DTs. The obtained probability distributions are concatenated and normalized to construct a robust descriptor with features of smooth-invariant Gaussians \( \text{FoSIG}(\mathcal{V}) \) as follows.

\[
\text{FoSIG}(\mathcal{V}) = [\psi(I_G, XY), \psi(I_{\text{DoG}, XY}), \psi(I_G, XT), \psi(I_{\text{DoG}, XT}), \psi(I_G, YT), \psi(I_{\text{DoG}, YT})]
\] (6)

where \( \psi(.) \) denotes a local encoding operator (e.g., LBP, CLBP, ...) for capturing local features in an image.

**2.4. Advantages of our descriptor**

FoSIG improves the performance based on several properties:

- **Robust to illumination and environment changes**: It is well-known that DoG filtered images are invariant against illumination changes. Moreover, by considering the difference between two Gaussians of different scales, they are robust against scale changes. This makes our encoding more robust against following main challenges of DT representation: illumination, scale, and environment changes.

- **Robust to noise**: Local DT features captured on \( I_G \) and \( I_{\text{DoG}} \) are more insensitive to noise in comparison to that in the original image. It should be noted that Gaussian filtering has been introduced together with LBP for
2D texture representation [9]. However, contrariwise to [9], the Gaussian filtering is directly calculated on the entire image in our encoding instead of on neighborhoods at various local scale areas of a pixel.

- **Forceful discrimination:** Mentioned as an approximation of Laplacian of Gaussian (LoG), DoG produces beneficial invariant features for recognition. It is along with the Gaussian smooth filtering to form two critical complemented components in order to significantly enhance the discriminant power (see Table 2).

- **Low computational cost:** It only takes less than 0.92s to structure a video of $48 \times 48 \times 75$ dimension with a raw MATLAB code of our algorithm which is run on a Linux laptop of CPU Intel Core i7 1.9 Ghz, 4G RAM.

3. EXPERIMENTS

In this section, we verify our proposed method for recognition issue on various benchmark DT datasets. The obtained results are evaluated in comparison with the existing approaches. A linear multi-class SVM algorithm of the LIBLINEAR\(^1\) library [5] with the default parameters are used for DT classification.

3.1. Experimental settings

We conduct $\sigma_1 = 0.5$ and $\sigma_2 = \{3, 4, 5, 6\}$ for computing smooth-invariant filtered features. To capture local relationships, we use the popular operator CLBP with joint settings of $\psi = CLBP^{5\times5\times2}$, i.e., $\psi = CLBP^{5\times5\times2}$.

As the result, the obtained descriptor has dimension of 1200 bins. The pair of $(\sigma_1, \sigma_2) = (0.5, 6)$ is addressed for comparison with the existing methods due to its outperformance. Empirically, values of $\sigma_1$ and $\sigma_2$ should be in the range of above parameters because the informative appearance of DTs will be sharply diminished when $\sigma_2$ is close to $\sigma_1$ or $\sigma_2 > 6$.

3.2. Datasets and protocols

**UCLA dataset:** It includes 200 DT videos which are divided into 50 categories with four sequences per one group [25]. Each original video is recorded in $110 \times 160 \times 75$ dimension to capture textural motions, such as fountain, waterfall, flower, plant, etc. In experiment, its small version with $48 \times 48 \times 75$ sequences of main appearances of DTs is usually addressed and composed more challenging subsets for DT recognition as follows. Table 1 shows a brief of key features for a glance.

- **50-class:** 50 groups using two protocols: leave-one-out [3, 29] and 4-fold cross validation [12, 27].
- **9-class and 8-class:** A sub-dataset of 9 classes is rearranged from the 50 classes consisting of “boiling water” (8), “plants” (108), “sea” (12), “fire” (8), “flowers” (12), “fountains” (20), “smoke” (4), “water” (12), and “waterfall” (16), where the numbers beside indicate their quantities. The “plants” is removed from 9-class to structure 8-class with more challenging [32]. Following [6, 12], a half of samples is randomly chosen for training and the rest for testing. The final recognition rate is returned from the average of 20 runtimes.

**DynTex dataset:** It includes more than 650 high-quality videos captured in diversities of environmental conditions [17]. Following [3, 27, 33], a challenging sub-dataset (named DynTex35) with 10 categories is composed by cropping 35 videos as follows: 8 non-overlapping sub-sequences obtained by randomly splitting each video with various cutting points but not in the half of X, Y, and T axes; 2 sub-videos by only splitting along its T axis. Other challenging sub-datasets are also assembled from the original DynTex using leave-one-out protocol for DT recognition as follows: Alpha — three categories with 20 videos in each of them, Beta — 162 sequences divided into 10 classes with different numbers of items for each, and Gamma — 10 classes of 264 DT videos [2, 3].

**DynTex++ dataset:** Ghanem et al. [6] selected 345 raw videos of DynTex to compose DynTex++. These videos, which are filtered and fixed in size of $50 \times 50 \times 50$ to capture the major chaotic motions, are categorized into 36 classes with 100 items for each, i.e., 3600 DTs in total. Following [3, 6], a half of samples of each class is randomly selected for training and the rest for testing. The final recognition rate is the mean of 10 repetitions.

3.3. Experimental results

It can be seen in Table 2 that taking smooth-invariant features into account describing DTs points out a robust descriptor with outperformance. Specific evaluations of FoSIG descriptor on the datasets are presented in Table 3, in which the highest recognition rates are in bold. The experimental results are then compared to the state of the art in Tables 4 and 5, in which the highest rates are in bold. Evaluations of VLBP and LBP-TOP are referred to the implementations in [27, 18] while the rest are directed from the original works. In general, FoSIG significantly outperforms in comparison to most of the existing methods on various DT datasets for recognition issue. It just operates on DynTex less efficiently than deep-learning-based methods taking enormous cost of computation with complex algorithms for learning DT features. Hereafter, the performance of our proposed descriptor on the

<p>| Table 1. A summary of main properties of DT datasets. |
| --- | --- | --- | --- | --- | --- |</p>
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Sub-dataset</th>
<th>#Videos</th>
<th>Resolution</th>
<th>#Classes</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCLA</td>
<td>50-class</td>
<td>200</td>
<td>$48 \times 48 \times 75$</td>
<td>50</td>
<td>Leave-one-out and 4-fold validation</td>
</tr>
<tr>
<td>UCLA</td>
<td>9-class</td>
<td>200</td>
<td>$48 \times 48 \times 75$</td>
<td>9</td>
<td>50%/50%</td>
</tr>
<tr>
<td>UCLA</td>
<td>8-class</td>
<td>92</td>
<td>$48 \times 48 \times 75$</td>
<td>8</td>
<td>50%/50%</td>
</tr>
<tr>
<td>DynTex</td>
<td>DynTex35</td>
<td>350</td>
<td>Different dimensions</td>
<td>10</td>
<td>Leave-one-out</td>
</tr>
<tr>
<td>DynTex++</td>
<td>Alpha</td>
<td>60</td>
<td>$32 \times 288 \times 250$</td>
<td>3</td>
<td>Leave-one-out</td>
</tr>
<tr>
<td>DynTex++</td>
<td>Beta</td>
<td>162</td>
<td>$32 \times 288 \times 250$</td>
<td>10</td>
<td>Leave-one-out</td>
</tr>
<tr>
<td>DynTex++</td>
<td>Gamma</td>
<td>264</td>
<td>$32 \times 288 \times 250$</td>
<td>10</td>
<td>Leave-one-out</td>
</tr>
<tr>
<td>DynTex++</td>
<td>DynTex35</td>
<td>350</td>
<td>Different dimensions</td>
<td>10</td>
<td>Leave-one-out</td>
</tr>
</tbody>
</table>

\(^1\)https://www.csie.ntu.edu.tw/~cjlin/liblinear
particular datasets is evaluated in detail.

**UCLA dataset:** It can be verified from Tables 3 and 4 that the proposed descriptor obtains the best results on scenarios of 50-class (99.50%) and 50-fold (100%) compared to all existing methods, including deep-learning-based. In terms of 9-class and 8-class with rates of 98.95% and 98.59%, FoSIG nearly has the same performance as DT-CNN’s [1]. In the meanwhile, those are the best evaluations among the LBP-based methods, except CVLBC [34]. However, it is not better than ours on other datasets (see Table 5). The highest rate of 9-class is 99.15% with setting of $(\sigma_1, \sigma_2) = (0.5, 4)$ and of 8-class is 98.59% at $\sigma_1 = 0.5$ and $\sigma_2 = \{3, 6\}$ (see Table 3).

**DynTex dataset:** With rate of 99.14%, FoSIG is the best performance on DynTex35, except CSAP-TOP [13] (100%) and MEWLP [28] (99.71%). However, MEWLP has not been verified on more challenging datasets (i.e., Alpha, Beta, Gamma) while CSAP-TOP is about 2% lower than ours on Gamma and has been missing on DynTex++ (see Table 5). In respect of Alpha (96.67%), Beta (92.59%), and Gamma (92.42%), our proposed descriptor outperforms significantly compared to most of the state of the art, except st-TCOF [18], D3 [8], DT-CNN [1] (see Table 5). These methods use deep learning techniques with sophisticated algorithms to learn DT features while the cost of time is crucial in real applications.

**DynTex++ dataset:** It can be observed from Table 5 that our result (95.99%) is the best in comparison with all approaches, except MEWLP [28], HLB [27], and DT-CNN [1]. However, MEWLP and HLB are not better than ours on UCLA as well as have not been verified on more challenging sub-datasets of DynTex (i.e., Alpha, Beta, Gamma). In the meanwhile, DT-CNN with complicated frameworks of AlexNet and GoogleNet takes much time for learning DTs to achieve over 2% better than ours.

### Table 2. Contribution of $I_{DoG}$ on DynTex++.

| $(\sigma_1, \sigma_2)$ | $|\sigma_1|$ | $|\sigma_2|$ |
|------------------------|-------------|-------------|
| $[0.6, 3]$             | 95.73       | 95.73       |
| $[0.5, 4]$             | 98.88       | 98.88       |
| $[0.5, 5]$             | 98.57       | 98.57       |
| $[0.5, 6]$             | 98.57       | 98.57       |

Note: $|\sigma_1|$ and $|\sigma_2|$ are determined using leave-one-out and four cross-fold validations. Dyn35 and Dyn++ are abbreviated for DynTex35 and DynTex++ sub-datasets respectively.

### Table 3. Recognition rates (%) on DT benchmark datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>UCLA</th>
<th>DynTex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>50-Lo</td>
<td>50-fold</td>
</tr>
<tr>
<td>A</td>
<td>99.50</td>
<td>99.50</td>
</tr>
<tr>
<td>B</td>
<td>99.00</td>
<td>100</td>
</tr>
<tr>
<td>C</td>
<td>99.00</td>
<td>99.50</td>
</tr>
<tr>
<td>D</td>
<td>99.50</td>
<td>100</td>
</tr>
</tbody>
</table>

Note: 50-Lo and 50-fold denote results on 50-class breakdown using leave-one-out and four cross-fold validations. Dyn35 and Dyn++ are abbreviated for DynTex35 and DynTex++ sub-datasets respectively.

### Table 4. Comparison of recognition rates (%) on UCLA.

<table>
<thead>
<tr>
<th>Group</th>
<th>Encoding method</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>FD-MAP [14]</td>
</tr>
<tr>
<td>B</td>
<td>AK-LDS [25]</td>
</tr>
<tr>
<td>C</td>
<td>3D-DoG [31]</td>
</tr>
<tr>
<td>D</td>
<td>MBSS-TOP [3]</td>
</tr>
<tr>
<td>E</td>
<td>VLB [33]</td>
</tr>
<tr>
<td>F</td>
<td>FoSIG</td>
</tr>
</tbody>
</table>

### Table 5. Comparison of rates (%) on DynTex and DynTex++.

<table>
<thead>
<tr>
<th>Group</th>
<th>Encoding method</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>FD-MAP [14]</td>
</tr>
<tr>
<td>B</td>
<td>AK-LDS [25]</td>
</tr>
<tr>
<td>C</td>
<td>3D-DoG [31]</td>
</tr>
<tr>
<td>D</td>
<td>MBSS-TOP [3]</td>
</tr>
<tr>
<td>E</td>
<td>VLB [33]</td>
</tr>
<tr>
<td>F</td>
<td>FoSIG</td>
</tr>
</tbody>
</table>

### Table 6. Comparison of rates (%) on DynTex and DynTex++.

<table>
<thead>
<tr>
<th>Group</th>
<th>Encoding method</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>FD-MAP [14]</td>
</tr>
<tr>
<td>B</td>
<td>AK-LDS [25]</td>
</tr>
<tr>
<td>C</td>
<td>3D-DoG [31]</td>
</tr>
<tr>
<td>D</td>
<td>MBSS-TOP [3]</td>
</tr>
<tr>
<td>E</td>
<td>VLB [33]</td>
</tr>
<tr>
<td>F</td>
<td>FoSIG</td>
</tr>
</tbody>
</table>


4. CONCLUSIONS

We have presented an effective framework for DT description in which the advantages of Gaussian and DoG filtering kernels are taken into account feature encoding to make the proposed descriptor more robust against noise and illumination changes. Assessments of DT recognition on various datasets have validated that our proposal significantly outperforms in comparison with the state-of-the-art results. Analysis in multi-scale solutions can be considered in future works.
5. REFERENCES


