Locating robust patterns based on invariant of LTP-based features

Thanh Tuan Nguyen, Thanh Phuong Nguyen, Nadège Thirion-Moreau

Locating Robust Patterns Based on Invariant of LTP-based Features

Thanh Tuan Nguyen[1,2] and Thanh Phuong Nguyen [1] and Nadège Thirion Moreau [1]
[1]Université de Toulon, Aix Marseille Univ, CNRS, LIS, Marseille, France
[2] HCMC University of Technology and Education, Faculty of IT, Thu Duc City, Ho Chi Minh City, Vietnam
tuannt@hcmute.edu.vn, tpnguyen@univ-tln.fr, thirion@univ-tln.fr

1 Introduction

Dynamic textures (DTs) are textural features repeated in a temporal domain [1]. Effective analysis of DTs can be one of the important tasks in real applications: background subtraction [2, 3], facial expression [4], etc. Many efforts [5] have been made to deal with different aspects for enhancing the performance: i) Model-based methods [6, 7, 8, 9] extracted DT features using the variants of Linear Dynamical System (LDS) [1]; ii) Optical-flow-based techniques [10, 11, 12] considered the chaotic motions of DTs based on the direction attributes of normal flow [13]; iii) Geometry-based approaches utilized geometry theory to evaluate self-similarity characteristics: fractal spectrum [14, 15, 16], lacunarity spectrum [17]; iv) Filter-based methods tried to diminish the negative influence of illumination and noise [18, 19]; v) Learning-based methods learned deep DT patterns in different ways: based on CNNs [20, 21, 22], dictionary learning [23, 24], randomized neural network [25]; vi) In simple frameworks, local-feature-based methods [26, 27, 28] have obtained promising performance by using LBP-based operators to encode spatio-temporal relationships surrounding a center pixel.

Among those works, the local-feature-based methods have recently obtained significant results in simple frameworks. These methods generally take into account Local Binary Pattern (LBP) [29] for DT feature extraction. Zhao *et al.* [30] could be seen as the pioneers in consideration of LBP to propose local discriminative descriptors: LBP-TOP - addressing LBP on three collections of plane-images of a given video $\mathcal{V}$, and Volume-LBP [30] - exploiting LBP on three consecutive frames of $\mathcal{V}$. Motivated by two kinds of these local encodings, many researchers have made efforts to handle the LBP conventional restrictions for improving the DT discrimination: sensitivity to noise and near-uniform regions [31, 26, 32, 33]; rotation-invariant issues [34, 35]. Inspired by the simple and effective computation of the LBP method, many techniques were introduced to capture the principal patterns for textural representation. Bianconi *et al.* [36] stated that the dominant patterns play an important role in the descriptor construction. However, locating these key features can be difficult due to the negative impact of encoding factors. Indeed, realizing the abnormality of 3-adjacent bits in a LBP string, Chen *et al.* [37] introduced robust LBPs (RLBP) by switching on/off the center bit of the sub-string if it is different from the prior and posterior bits. Also, by a switching technique, Tiwari and Tyagi [38] attempted to modify the noisy bits of the non-uniform LBPs so that the achieved results are uniform for robust DT patterns. Furthermore, co-occurrence features were introduced in [39, 40] for texture classification, while Nanni *et al.* [41] proposed to locate the highest-variance dominant patterns using the neighborhood preserving embedding technique. Guo *et al.* [42] built a learning framework for determining dominant patterns, but the high number of features could be a potential drawback in the case of a larger number of classes. In another aspect, Liao *et al.* [43] took a minimum set of pattern labels that represents around 80% of the total occurrences of the LBP patterns to structure a dominant LBP-based descriptor (DLBP). The fact that DLBP does not encapsulate any information of the selected LBPs on the way of descriptor construction, Doshi *et al.* [44] proposed to maintain the discrimination between different LBPs by using the Borda count method to rank the frequencies of the LBP patterns. The top n of the ranked patterns were gathered to form a final descriptor.

It can be realized that there exist two obstacles for locating the dominant local patterns which have to be addressed: *i)* encapsulating the discrimination of local features [43, 44], *ii)* the noise sensibility caused by a small gray-scale change of local neighbors [38, 37]. Different from the above techniques, we propose a novel concept to efficiently locate robust patterns for the local DT encoding. To be more explicit, we consider the invariant features in two scales of LTP computation [45] with a pre-defined pair of thresholds $\{\delta_1, \delta_2\}$. It means that the

robust patterns (called RLTPs) would be satisfied if and only if the LTP_{δ_1} patterns are identical to the LTP_{δ_2} ones. Our idea is simple yet effective because it simultaneously deals with two already mentioned issues. Furthermore, the optical flow estimation and its variants have been used in motion analysis. One of the most efficient algorithms for the motion estimation was introduced by Sun *et al.* [46]. It was named Classic+NL and afterward applied to many applications: motion boundary detection [47], video segmentation [48, 49], etc. Inspired by Classic+NL, in this work, we propose to address its horizontal and vertical components to locate robust DT motions with our RLTP operator. To the best of our knowledge, it is the first time that Classic+NL is used for estimating DT motions, where its outputs will be featured by a local operator for DT description. Experiments have illustrated that our RLTP-based descriptor has very good results compared to the non-robust and original ones, i.e., LTP [45] and CLTP [50]. In short, our major contributions can be listed as

- A novel concept of robust RLTP patterns is introduced by considering the invariance of LTP-based features. RLTP could be a generalization of LTP [45] under the condition of δ_1=δ_2.
- A crucial derivative of CLTP [50] (called RCLTP) is proposed to be in accordance with RLTP.
- An efficient application is presented to take into account RCLTP for local DT representation.
- The local optical-flow-based features are taken into account DT motions for boosting the discrimination.
- Just addressing the robust patterns, our RCLTP descriptor obtains very good results compared to the original LTP, CLTP, and other LBP-based ones.

2 A brief survey of LTP and optical flow estimation

2.1 Local ternary pattern and its completed model

Motivated by LBP [29], Tan and Triggs [45] introduced LTP patterns by extending LBPs to ternary strings as

$$\mathrm{LTP}_{P,R}(\mathbf{q}) = \left\{\xi\big(\mathcal{I}(\mathbf{p}_i) - \mathcal{I}(\mathbf{q})\big)\right\}_{i=0}^{P-1} \tag{1}$$

where $\xi(.)$ is a ternary function defined subject to a pre-defined threshold δ as $\xi(x) = \mathbf{1},\ \text{if}\ x \geq \delta; \xi(x) = \mathbf{0},\ \text{if}\ -\delta < x < \delta; \xi(x) = -\mathbf{1},\ \text{if}\ x \leq -\delta$. To be in accordance with LBP, LTP of $\mathbf{q}$ is deducted in two LBP-based patterns subject to that its code is -1 or 1. Concretely, the upper pattern is defined as $\mathrm{LTP}^{u}_{P,R}(\mathbf{q}) = \mathrm{LTP}_{P,R}(\mathbf{q}) \bigcap \{1\}$, while the lower one is defined as $\mathrm{LTP}^{l}_{P,R}(\mathbf{q}) = \mathrm{LTP}_{P,R}(\mathbf{q}) \bigcap \{-1\}$.
Motivated by the completed LBP model [51], Rassem and Khoo [50] introduced a completed model of LTP [45] (so-called CLTP). Accordingly, CLTP includes three complementary components for each of the LTP-separated patterns: $\mathrm{CLTP\_S}^{\mathrm{u/l}}$, $\mathrm{CLTP\_M}^{\mathrm{u/l}}$, $\mathrm{CLTP\_C}^{\mathrm{u/l}}$, where the superscript *u/l* denotes the upper/lower patterns. In other words, CLTP would be composed of six components in total. Refer to [50] for further formulas and discussions. Additionally, other noticeable LTP-based variants were proposed for textual analysis such as synchronized rotation LTP [52] and feature-based wavelet CLTP [53].

2.2 Optical flow estimation

In order to deal with the aperture problem, Horn and Schunck [13] introduced an optical flow computation by addressing the apparent velocities of movements in an image $\mathcal{I}$. For a pixel $\mathbf{p}(x, y) \in \mathcal{I}$ at time t, the optical flow formulation can be generally written as

$$\nabla\mathcal{I}(\mathbf{p}, t).\overrightarrow{\mathbf{v}} + \mathcal{I}_t(\mathbf{p}, t) = 0 \tag{2}$$

where $\nabla\mathcal{I}(\mathbf{p}, t)$ denotes the derivatives at $\mathcal{I}(\mathbf{p}, t)$ subject to directions $\mathcal{I}_x$ and $\mathcal{I}_y$, i.e., $\nabla\mathcal{I}(\mathbf{p}, t) = \big(\mathcal{I}_x(\mathbf{p}, t), \mathcal{I}_y(\mathbf{p}, t)\big)^\top$; $\mathcal{I}_t(\mathbf{p}, t)$ is the temporal derivative of $\mathcal{I}(\mathbf{p}, t)$; $\nabla\mathcal{I}(\mathbf{p}, t).\overrightarrow{\mathbf{v}}$ is the usual dot product of $\nabla\mathcal{I}(\mathbf{p}, t)$ and the 2D velocity $\overrightarrow{\mathbf{v}} = (u, v)^\top$. Accordingly, two unknown components of $\overrightarrow{\mathbf{v}}$ would be estimated in particular computing fields. Sun *et al.* [46] proposed a Classic+NL algorithm to interleave the heuristic median filtering with the approximate minimization. Afterward, Classic+NL has been applied to some applications: detection of motion boundaries [47], segmentation videos [48, 49], etc. For estimating DT motions, it is the first time that the horizontal and vertical components of Classic+NL will be featured by a local operator to extract their robust patterns for DT description.

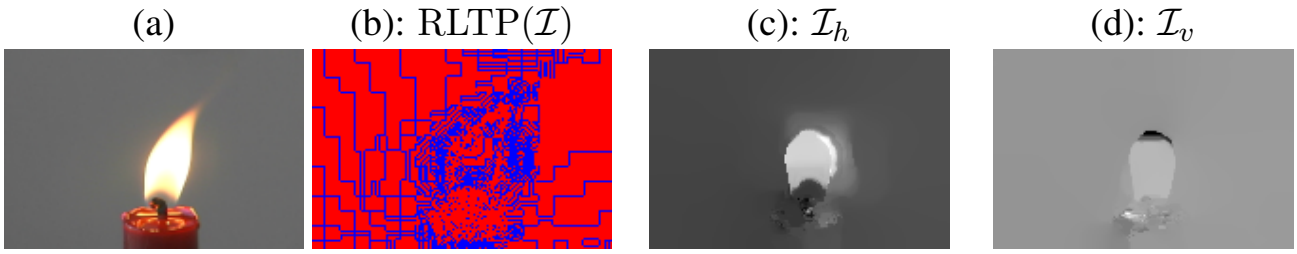

Figure 1: (a) Input image $\mathcal{I}$. (b) Locating the dominant pixels of $\mathcal{I}$ based on its LTP-based features using $\{\delta_1, \delta_2\} = \{1, 2\}$, $(P, R) = (8, 1)$ where the dominant pixels are located by red pixels, the non-dominant ones are in blue. (c) and (d) are Classic+NL's horizontal and vertical components of $\mathcal{I}$.

3 Novel robust patterns and their applications

3.1 Locating robust patterns

Taking into account the variability of LTP-based features for different thresholding levels, we propose an effective method to locate the robust patterns for video representation. Accordingly, a LTP-based pattern is located to be Robust (hence the acronym RLTP) if and only if it is invariant to the changes of those levels. Without loss of generality, let us consider two LTP-based encoding phases corresponding to two different predefined thresholds, i.e., $\{\delta_1, \delta_2\}$ with $(\delta_1 < \delta_2)$. For similarity, let $\mathrm{LTP}_{P,R,\delta_1}(\mathbf{q})$ and $\mathrm{LTP}_{P,R,\delta_2}(\mathbf{q})$ be two patterns computed by the operator LTP [45] with two thresholds δ_1 and δ_2 for a center pixel $\mathbf{q}$, respectively. RLTP of $\mathbf{q}$ would be satisfied with the identical condition as

$$\begin{aligned} \mathrm{RLTP}_{P,R}^{\delta_1,\delta_2}(\mathbf{q}) &= \mathrm{LTP}_{P,R,\delta_1}^{\delta_1}(\mathbf{q}) \\ &\text{so that } \mathrm{LTP}_{P,R}^{\delta_1}(\mathbf{q}) \equiv \mathrm{LTP}_{P,R}^{\delta_2}(\mathbf{q}) \end{aligned} \tag{3}$$

where $\equiv$ is an identical operator. Thereby, a pixel is also named dominant pixel if and only if its corresponding LTP-based pattern is robust. Furthermore, it can be deduced that LTP [45] would be a degeneration case of our RLTP in the case of δ_1=δ_2. Fig. 1(b) shows an instance of locating the dominant pixels of a real image $\mathcal{I}$ subject to their RLTP-based characteristics.

As mentioned in Section 2.1, LTP [45] can be separated into the upper and lower patterns to be compliant with the LBP-based computation. Correspondingly, we introduce two novel kinds of robust patterns: upper-robust (RLTP^u) and lower-robust (RLTP^l), which are formulated as

$$\begin{aligned} \mathrm{RLTP}_{P,R}^{\delta_1,\delta_2,u}(\mathbf{q}) &= \mathrm{LTP}_{P,R}^{\delta_1,u}(\mathbf{q}) \\ &\text{so that } \mathrm{LTP}_{P,R}^{\delta_1,u}(\mathbf{q}) \equiv \mathrm{LTP}_{P,R}^{\delta_2,u}(\mathbf{q}) \end{aligned} \tag{4}$$

$$\begin{aligned} \mathrm{RLTP}_{P,R}^{\delta_1,\delta_2,l}(\mathbf{q}) &= \mathrm{LTP}_{P,R}^{\delta_1,l}(\mathbf{q}) \\ &\text{so that } \mathrm{LTP}_{P,R}^{\delta_1,l}(\mathbf{q}) \equiv \mathrm{LTP}_{P,R}^{\delta_2,l}(\mathbf{q}) \end{aligned} \tag{5}$$

Figure 2 illustrates an intuitive instance of locating RLTP^u for a center pixel $\mathbf{q}$ with a gray-scale level of 19 (in red) and its $P = 8$ local neighbors. Accordingly, with $\{\delta_1, \delta_2\} = \{1, 2\}$, two correspondingly obtained upper LTP patterns are identical. Subject to our proposed definitions above, the local feature at this pixel is upper-robust. However, it would not be for the pairs of thresholds $\{1, 3\}$ or $\{2, 3\}$ due to the difference at the blue neighbor. On the other hand, it can be deduced similarly from Fig. 2 that the red pixel is lower-robust for three scales of these thresholds. As a result, there are many variants of RLTP which are needed to be investigated for their influence on video representation. We will thoroughly discuss this issue in Section 4.3.

It is worthy of note that our novel concept of RLTP has the following beneficial properties compared to the former works [43, 44, 38, 37].

- Our RLTPs maintain the discriminative information of the local patterns. It should be noted that DLBP [43] does not encapsulate any local-based discrimination, while IDLBP [44] was formed straightforwardly by taking top n of the ranked LBPs.
- Thanks to the invariant of LTP-based features in the tolerance intervals of thresholds $\{\delta_1, \delta_2\}$, RLTP can efficiently handle the noise issue. Meantime, the abnormal bits were either synchronized or turned unusually in ATNR-TOP [38] and RLBP [37] so that the outputs would be more insensitive to noise.

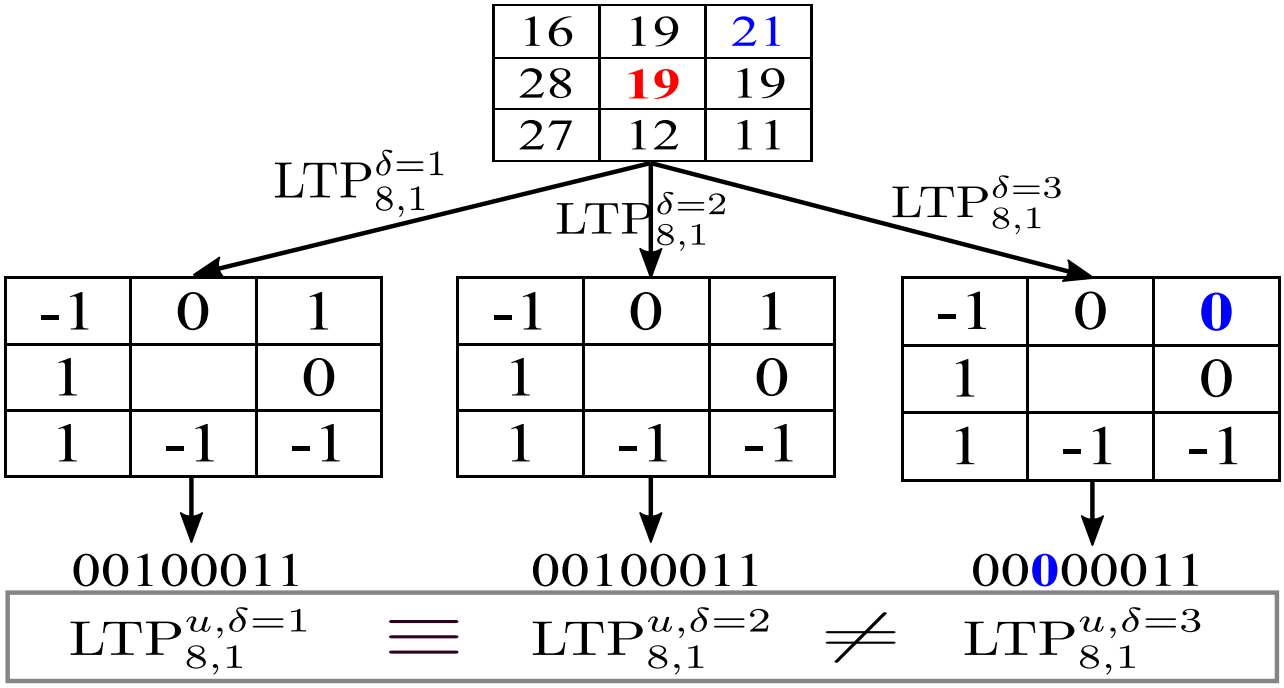

Figure 2: An example of locating RLTP^{u} for a center pixel $\mathbf{q}$ (in red) and $P = 8$ local neighbors.

- Our locating process of robust patterns could be more natural than the other works [37, 38]. Indeed, it is due to still considering the full relationships by thresholding a center pixel with its local neighbors. Meanwhile, several forcing processes were addressed in ATNR-TOP [38] and RLBP [37] to modify the neighbors stipulated for the noise identification.

3.2 Adapting robust features to CLTP operator

To be compliant with the novel concept of robust patterns, we propose to adapt our RLTP to CLTP [50] in order to make a new local encoding way which only addresses different kinds of robust patterns for DT representation. For simplicity, let us make two masks of RLTP's positions for a given image $\mathcal{I}$ corresponding to Eqs. (4) and (5) as

$$\begin{aligned} U_{mask}(\mathcal{I}) &= \left[\mathrm{LTP}_{P,R}^{\delta_1,u}(\mathcal{I}) \equiv \mathrm{LTP}_{P,R}^{\delta_2,u}(\mathcal{I})\right] \\ L_{mask}(\mathcal{I}) &= \left[\mathrm{LTP}_{P,R}^{\delta_1,l}(\mathcal{I}) \equiv \mathrm{LTP}_{P,R}^{\delta_2,l}(\mathcal{I})\right] \end{aligned} \tag{6}$$

As a result, these masks would be binary matrices, where the 1-bits locate the robust patterns while the 0-bits are for the non-robust ones. As mentioned in Section 2.1, CLTP [50] consists of six complementary components subject to the upper/lower encoding (i.e., $\mathrm{CLTP\_S}^{u/l}$, $\mathrm{CLTP\_M}^{u/l}$, and $\mathrm{CLTP\_C}^{u/l}$). Accordingly, CLTP [50] will be in accordance with our proposal of robust patterns to form a robust completed LTP-based operator (named RCLTP) as: For the upper encoding, we have

$$\begin{aligned} \mathrm{RCLTP\_S}_{P,R}^{\delta_1,\delta_2,u}(\mathcal{I}) &= U_{mask}(\mathcal{I}) * \mathrm{CLTP\_S}_{P,R}^{\delta_1,\delta_2,u}(\mathcal{I}) \\ \mathrm{RCLTP\_M}_{P,R}^{\delta_1,\delta_2,u}(\mathcal{I}) &= U_{mask}(\mathcal{I}) * \mathrm{CLTP\_M}_{P,R}^{\delta_1,\delta_2,u}(\mathcal{I}) \\ \mathrm{RCLTP\_C}_{P,R}^{\delta_1,\delta_2,u}(\mathcal{I}) &= U_{mask}(\mathcal{I}) * \mathrm{CLTP\_C}_{P,R}^{\delta_1,\delta_2,u}(\mathcal{I}) \end{aligned} \tag{7}$$

For the lower encoding, we have similarly

$$\begin{aligned} \mathrm{RCLTP\_S}_{P,R}^{\delta_1,\delta_2,l}(\mathcal{I}) &= L_{mask}(\mathcal{I}) * \mathrm{CLTP\_S}_{P,R}^{\delta_1,\delta_2,l}(\mathcal{I}) \\ \mathrm{RCLTP\_M}_{P,R}^{\delta_1,\delta_2,l}(\mathcal{I}) &= L_{mask}(\mathcal{I}) * \mathrm{CLTP\_M}_{P,R}^{\delta_1,\delta_2,l}(\mathcal{I}) \\ \mathrm{RCLTP\_C}_{P,R}^{\delta_1,\delta_2,l}(\mathcal{I}) &= L_{mask}(\mathcal{I}) * \mathrm{CLTP\_C}_{P,R}^{\delta_1,\delta_2,l}(\mathcal{I}) \end{aligned} \tag{8}$$

As the assessments of CLTP [50] and CLBP [51], the full joint integration (i.e., $_{S/M/C}$) of the above components could make the obtained features be more discriminative. Therefore, we have $\mathrm{RCLTP}_{\mathrm{S/M/C}}^{\mathrm{u}}$ (resp. $\mathrm{RCLTP}_{\mathrm{S/M/C}}^{\mathrm{l}}$) for the upper (resp. lower) components. From now on, we refer to $\mathrm{RCLTP}^{u/l}$ for simplicity.

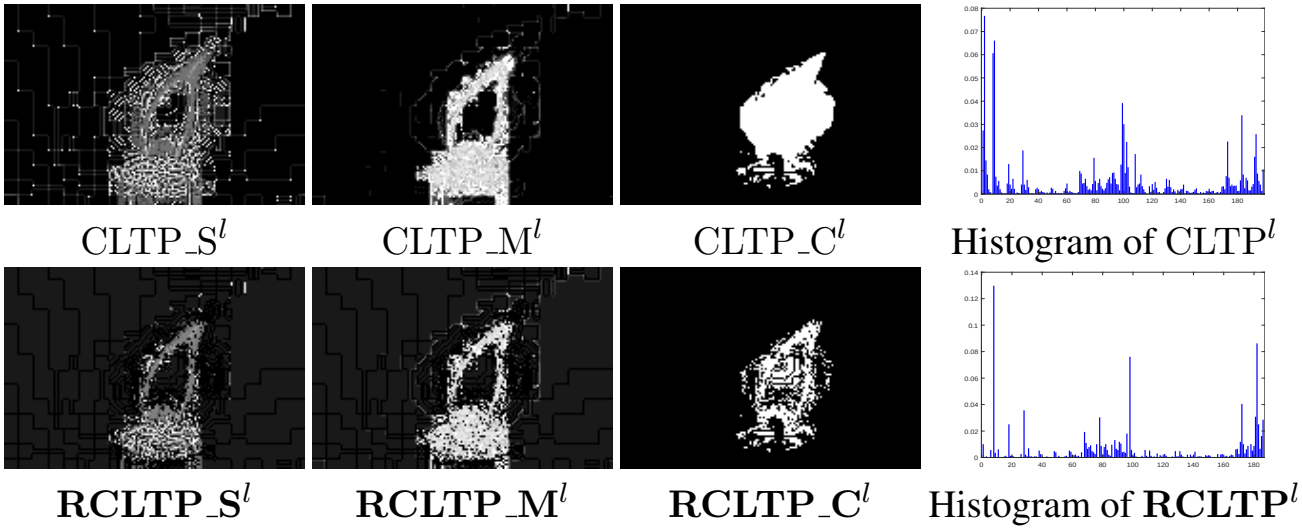

Figure 3: RCLTP of a real image $\mathcal{I}$ in Fig. 1(a) using $\{\delta_1, \delta_2\} = \{1, 2\}$, $R = 1$, $P = 8$, compared to the original CLTP [50]. For a visual view of histograms, some statistical distributions of RCLTP were hidden due to their dominance.

3.3 DT representation with two input streams of RCLTP

We propose to exploit the RLTP-based patterns in a multimodal approach by addressing two input streams which are together complementary for DT representation: *i)* a raw input video $\mathcal{V}$ and *ii)* its horizontal and vertical optical-flow-based components $\mathcal{L}_h$ and $\mathcal{L}_v$, which are calculated by the Classic+NL algorithm [46]. Traditionally, LBP-based methods [30, 32, 26] for DT description often partition the video $\mathcal{V}$ into plane-image collections f_{XY}, f_{XT}, and f_{YT} subject to its three orthogonal planes $\{XY, XT, YT\}$. Let us consider an orthogonal plane z ($z \in \{XY, XT, YT\}$) of $\mathcal{V}$. The RCLTP-based features for all plane-images $\mathcal{I} \in f_z$ of the plane z are then formed as

$$\Upsilon_{P,R}^{\delta_1,\delta_2,z} = \frac{1}{\mathcal{N}_z} \sum_{\mathcal{I} \in f_z} \left[\mathrm{RCLTP}_{P,R}^{\delta_1,\delta_2,u}(\mathcal{I}), \mathrm{RCLTP}_{P,R}^{\delta_1,\delta_2,l}(\mathcal{I}) \right] \tag{9}$$

where $\mathcal{N}_z$ denotes the number of f_z's plane-images.

Figure 3 intuitively illustrates a computing instance of RCLTP^l compared to CLTP^l for an image of f_{XY}. It can be observed that RCLTP would represent more apparent DT features than CLTP. Our idea is to exploit multimodal input streams by using the RCLTP operator to boost the discrimination. For the first input stream using the RCLTP-based features of the raw video $\mathcal{V}$, we form $\Upsilon_{P,R}^{\delta_1,\delta_2,XY}$, $\Upsilon_{P,R}^{\delta_1,\delta_2,XT}$, and $\Upsilon_{P,R}^{\delta_1,\delta_2,YT}$ by addressing this computation on f_{XY}, f_{XT}, and f_{YT} of $\mathcal{V}$, respectively. For the second stream, we only address the spatial plane-images to compute the optical-flow-based components. Concretely, for each $\mathcal{I} \in f_{XY}$, we have the corresponding Classic+NL-based images: $\mathcal{I}_h = \mathcal{L}_h(\mathcal{I})$ and $\mathcal{I}_v = \mathcal{L}_v(\mathcal{I})$, which present the appearance DT motions (see Figs. 1(c,d) for visual results of this computation). As a result, we have two corresponding collections of Classic+NL-based images, named $f_{\mathcal{L}_h}$ and $f_{\mathcal{L}_v}$, respectively. It should be noted that Classic+NL is not included for the temporal collections due to the less meaningful DT motions. In addition, the values of $\mathcal{I}_h$ and $\mathcal{I}_v$ will be normalized in [0, 255] to use the same parameters δ_1 and δ_2 as the first input stream of the construction. Finally, based on Eq. (9), a proposed DT descriptor for $\mathcal{V}$ would be defined as

$$\begin{aligned} \mathrm{RCLTP}_{P,R}^{\delta_1,\delta_2}(\mathcal{V}) = \big[& \Upsilon_{P,R}^{\delta_1,\delta_2,XY}, \Upsilon_{P,R}^{\delta_1,\delta_2,XT}, \Upsilon_{P,R}^{\delta_1,\delta_2,YT}, \\ & \Upsilon_{P,R}^{\delta_1,\delta_2,\mathcal{L}_h}, \Upsilon_{P,R}^{\delta_1,\delta_2,\mathcal{L}_v} \big] \end{aligned} \tag{10}$$

Algorithm 1 illustrates the main steps of this construction.

To better assess the true contributions of our proposed robust patterns, we also structure other DT descriptors using the original operators LTP [45] and CLTP [50]. These LTP-based descriptors are also constructed in the same ways as RCLTP. Accordingly, it could be pointed out the following beneficial properties of our RCLTP to enhance the discrimination power in comparison with others.

- Our RCLTP could be a crucial derivative of CLTP [50]. Indeed, because ours is constructed by the robust pixels encoded by CLTP, the CLTP's features for a center pixel are also encapsulated in RCLTP.

Algorithm 1: Structuring RCLTP descriptor.

Input: A video $\mathcal{V}$; a pair of thresholds $\{\delta_1, \delta_2\}$;
P local neighbors and an interpolating radius R.
Output: A RCLTP descriptor of $\mathcal{V}$.

$\Upsilon_{P,R}^{\delta_1,\delta_2,z} =$ a zero vector
Calculate optical-flow components $f_{\mathcal{L}_h}$ and $f_{\mathcal{L}_v}$
for each $\mathcal{I} \in f_{XY}$ **do**
//Computing binary masks of robust patterns
$U_{mask} = \left[\mathrm{LTP}_{P,R,\delta_1}^{u}(\mathcal{I}) \equiv \mathrm{LTP}_{P,R,\delta_2}^{u}(\mathcal{I})\right]$
$L_{mask} = \left[\mathrm{LTP}_{P,R,\delta_1}^{l}(\mathcal{I}) \equiv \mathrm{LTP}_{P,R,\delta_2}^{l}(\mathcal{I})\right]$
//Computing CLTP's components for $\mathcal{I}$
$S_{\mathcal{I}}^{u} = \mathrm{CLTP\_S}_{P,R}^{\delta_1,\delta_2,u}(\mathcal{I}); S_{\mathcal{I}}^{l} = \mathrm{CLTP\_S}_{P,R}^{\delta_1,\delta_2,l}(\mathcal{I})$
$$M_{\mathcal{I}}^{u} = \mathrm{CLTP\_M}_{P,R}^{\delta_1,\delta_2,u}(\mathcal{I}); M_{\mathcal{I}}^{l} = \mathrm{CLTP\_M}_{P,R}^{\delta_1,\delta_2,l}(\mathcal{I})$$
$$C_{\mathcal{I}}^{u} = \mathrm{CLTP\_C}_{P,R}^{\delta_1,\delta_2,u}(\mathcal{I}); C_{\mathcal{I}}^{l} = \mathrm{CLTP\_C}_{P,R}^{\delta_1,\delta_2,l}(\mathcal{I})$$
//Histograms of $\Psi^{u/l} = \mathrm{RCLTP}_{P,R}^{\delta_1,\delta_2,u/l}$ *with a full joint integration* Γ *of components*
$\Psi_{\mathcal{I}}^{u} = \Gamma(U_{mask} * S_{\mathcal{I}}^{u}, U_{mask} * M_{\mathcal{I}}^{u}, U_{mask} * C_{\mathcal{I}}^{u})$
$\Psi_{\mathcal{I}}^{l} = \Gamma(L_{mask} * S_{\mathcal{I}}^{l}, L_{mask} * M_{\mathcal{I}}^{l}, L_{mask} * C_{\mathcal{I}}^{l})$
$\Upsilon_{P,R}^{\delta_1,\delta_2,XY} = \Upsilon_{P,R}^{\delta_1,\delta_2,XY} + \left[\Psi_{\mathcal{I}}^{u/l}\right]$
end for
$\Upsilon_{P,R}^{\delta_1,\delta_2,XY} = \frac{1}{|\mathcal{N}_{XY}|}\Upsilon_{P,R}^{\delta_1,\delta_2,XY}$; *//Normalization*
Repeat steps 3 and 4 on the sets f_{XT}, f_{YT}, $f_{\mathcal{L}_h}$, $f_{\mathcal{L}_v}$ for $\Upsilon_{P,R}^{\delta_1,\delta_2,XT}$, $\Upsilon_{P,R}^{\delta_1,\delta_2,YT}$, $\Upsilon_{P,R}^{\delta_1,\delta_2,\mathcal{L}_h}$, $\Upsilon_{P,R}^{\delta_1,\delta_2,\mathcal{L}_v}$
//Concatenating all the obtained histograms
$\mathrm{RCLTP}_{P,R}^{\delta_1,\delta_2}(\mathcal{V}) = \left[\Upsilon_{P,R}^{\delta_1,\delta_2,XY}, \Upsilon_{P,R}^{\delta_1,\delta_2,XT}, \Upsilon_{P,R}^{\delta_1,\delta_2,YT}, \Upsilon_{P,R}^{\delta_1,\delta_2,\mathcal{L}_h}, \Upsilon_{P,R}^{\delta_1,\delta_2,\mathcal{L}_v}\right]$

- RCLTP only uses robust patterns for its histogram construction (see Fig. 3). It means the non-robust ones can be noisy. The thorough assessments in Section 4.3 will have proved this issue.
- Our proposed descriptor is very different from the previous approaches [43, 44]. Indeed, RCLTP consists of the local features on which LTP patterns were selected, while DLBP [43] is not. In addition, the robust features for forming RCLTP descriptor are pointed out by the strictly invariant condition with the changes of thresholds, while IDLBP [44] simply took the top of the ranked LBP patterns.
- Our RCLTP is already very descriptive and a preponderance of the overall patterns, while the other robust descriptors [37, 38] addressed the whole local features on the way of the descriptor construction.
- Profitable features of optical-flow-based and local-feature-based methods are exploited thanks to using RCLTP for encoding the Classic+NL's components.

4 Experiments and evaluations

4.1 Datasets and protocols

UCLA dataset: Saisan *et al.* [1] composed 200 DT videos containing various sorts of textural motions: fountain, waterfall, flower, plant, etc. These sequences were sized in $110\times160\times75$ dimension and divided into 50 categories. For DT classification issue, three schemes are usually addressed as follows. *i) 50-class* utilizes 50 categories with two protocols: *leave-one-out* (50-LOO) [18, 54] and *4-fold cross validation* (50-4fold) [32, 26]. *ii) 9-class* has 9 labels composed of 50 categories. *iii) 8-class* is formed from *9-class* by eliminating the "plants" class due to its dominance of video quantities (108 DT videos). Both schemes use 50%/50% protocol, i.e., half of the samples are randomly chosen for training and the rest for testing. A final rate is returned by the mean of 20 trials.
DynTex dataset: Péteri *et al.* [55] captured more than 650 high-quality videos in variant conditions of environmental elements. For DT classification issue, three challenging schemes are usually addressed with the LOO protocol as follows. *i) Alpha* has three categories with 20 videos for each of them. *ii) Beta* has 162 DT

Table 1: Rates of LTP-based descriptors using $\{\delta_1\ [,\delta_2]\} = \{1\ [,2]\}$ with and without Classic+NL's components.

	Without Classic+NL				With Classic+NL			
Descriptor	50-Loo	4fold	Beta	Gamma	50-Loo	4fold	Beta	Gamma
LTP [45]	60.00	93.00	87.04	89.02	96.00	96.00	90.12	92.42
Our RLTP	80.00	94.00	89.51	91.29	97.50	97.50	92.59	93.18
CLTP [50]	**100**	**100**	93.21	92.80	**100**	**100**	**95.68**	95.83
Our RCLTP	**100**	**100**	**93.83**	**93.56**	**100**	**100**	**95.68**	**96.21**

Note: 50-Loo and 4fold denote results on the schemes 50-LOO and 50-4fold, respectively.

videos arranged into 10 categories. *iii) Gamma* has 264 DT videos, also arranged into 10 categories [22, 18].
DynTex++ dataset: Ghanem *et al.* [56] composed DynTex++ by taking out 345 DynTex's videos to be clipped and fixed in size of $50 \times 50 \times 50$. The obtained results were categorized into 36 categories with 100 sub-videos for each category, i.e., 3600 DTs in total. For DT classification, the 50%/50% protocol [18, 56] is used for each trial, and a final rate is then reported by the mean of 10 runtimes.

4.2 Experimental settings

For locating robust patterns, we investigate the influence of thresholds in an amplitude interval from 1 to 5, as specified by different pairs $\{\delta_1, \delta_2\}$ in Table 6. For encoding the RCLTP-based features, we address a 2-scale local region $\{(P,R)\} = \{(8,1),(8,2)\}$ for capturing more further local features. Thereby, it takes $t = 8(P+2)^2 = 800$ bins for structuring $\Upsilon_{P,R}^{\delta_1,\delta_2,z}$ on the plane z. As a result, it leads to taking $5 \times t = 4000$ bins for 5 image collections (i.e., $\{f_{XY}, f_{XT}, f_{YT}, f_{\mathcal{L}_h}, f_{\mathcal{L}_v}\}$) to form the final RCLTP-based descriptor. It should be noted that the other LTP-based descriptors (i.e., LTP [45] and CLTP [50]) are also addressed with the above settings for objective evaluations. For classifying DTs, we utilize the linear multi-class SVM classifier of LIBLINEAR [57] with the default parameters.

4.3 Performing analyses

It can be observed from Table 1 that addressing the Classic+NL-based components has significantly improved the discrimination of LTP-based DT descriptors. Indeed, for the completed LTP-based descriptors (i.e., CLTP [50] and our RCLTP), the performances have boosted up $2 \sim 3\%$. In the meanwhile, the original LTP-based ones (i.e., LTP [45] and our RLTP) have increased by up to 17.5%. Those have validated the interest of the integration of local-feature-based and optical-flow-based approaches for the DT representation. Henceforward, the LTP-based descriptors including the Classic+NL-based components will be addressed in default on the following evaluations.

4.3.1 Effectiveness of robust RLTP patterns

It can be verified that our novel idea of RLTP is quite simple, but it works very well in comparison with the original LTP [45] and non-RLTP. Indeed, based on the experimental results of DT classification in Tables 2 and 4, we could assert several substantial statements as follows.

- Addressing the robust LTP-based patterns can improve the performance compared to the use of all patterns of DTs. Indeed, the rates in Table 4 indicate that our RLTP is better than the original LTP [45] on both simple and challenging schemes. It should be noted that the original LTP is composed of RLTP and non-RLTP. It means RLTP has a fruitful preponderance of the overall DT features.
- It can be stated that non-RLTP patterns could be noisy and negatively impact the local DT encoding. Indeed, Table 4 shows the solidly better performance of RLTP compared to non-RLTP. For instance, RLTP with $\{\delta_1, \delta_2\} = \{1, 2\}$ obtained 92.59% on *Beta*, about 4% higher than non-RLTP. In other words, the experiments have proven the less significance of the non-robust patterns.
- Thanks to the preponderance, RLTP has a little forceful ability of noise resistance with different levels ρ of the salt-and-pepper noise, compared to LTP [45] (see Table 2). It could be the reason why the original CLTP consists of the non-robust patterns. It has consolidated the contributions of our proposal.

Table 2: Performances (%) using $\{\delta_1\ [,\delta_2]\} = \{3\ [,4]\}$ in various levels of salt-and-pepper noise on *Gamma* of DynTex.

Descriptor	ρ=10%	ρ=20%	ρ=30%	ρ=40%	ρ=50%	ρ=60%
LTP [45]	89.77	88.26	87.88	87.12	**87.12**	82.95
Our RLTP	90.15	88.26	87.88	**87.88**	86.74	82.58
CLTP [50]	92.42	**90.91**	89.02	86.74	84.47	79.55
Our RCLTP	**93.18**	90.53	**89.39**	87.12	86.74	**85.98**

Note: It should be noted that for objective comparisons, these results are based on the raw DT plane-images, i.e., without Classic+NL's components included.

Table 3: Rates (%) on Gaussian noise levels of two schemes: *50-4fold* of UCLA and *Gamma* of DynTex.

			Noise for *50-4fold*			Noise for *Gamma*			Time(s)
Descriptor	(P,{R})	Mapping	dB=1	dB=3	dB=5	dB=1	dB=3	dB=5	
VLBP [30]	(4,1)	-	91.00	92.00	94.00	87.12	89.02	90.53	≈ 0.22
LBP-TOP [30]	(8,1)	u2	97.50	99.50	98.50	77.65	84.47	87.12	$\approx$ **0.15**
CLSP-TOP [32]	(8,1)	riu2	98.00	99.50	99.00	82.95	84.47	87.50	≈ 0.27
HILOP [33]	(8,{1,2})	u2	99.50	99.50	99.50	88.64	90.91	91.29	≈ 0.50
CLBP [51]	(8,1)	riu2	99.50	99.50	99.50	85.98	87.88	89.39	≈ 0.23
HoGF [58]	(8,1)	riu2	**100**	**100**	**100**	90.53	90.15	90.53	≈ 0.54
LTP [45]	(8,1)	riu2	97.50	97.50	98.00	86.74	90.15	90.91	≈ 0.21
Our RLTP	(8,1)	riu2	97.50	97.50	98.00	87.50	89.78	89.78	≈ 0.23
CLTP [50]	(8,1)	riu2	98.50	99.00	**100**	90.91	**92.80**	93.18	≈ 0.43
Our RCLTP	(8,1)	riu2	99.00	99.50	**100**	**91.67**	**92.80**	**93.56**	≈ 0.78

Note: "-" means "not available".

4.3.2 Complexity of encoding RCLTP-based patterns

Let us consider a given image $\mathcal{I}$ with dimension $\mathcal{H} \times \mathcal{W}$. Subject to Alg. 1, the complexity for locating the robust upper-lower patterns is $\mathcal{Q}_{\text{UL}} \approx 2 \times \mathcal{H} \times \mathcal{W} + \mathcal{Q}_{\text{LTP}}$, where $\mathcal{Q}_{\text{LTP}} \approx \mathcal{O}(P \times \mathcal{H} \times \mathcal{W})$ denotes the complexity of the LTP encoding [45] with P local neighbors. As presented in Section 2.1, the complexity of the CLTP encoding [50] can be deduced as $\mathcal{Q}_{\text{CLTP}} \approx 3 \times \mathcal{Q}_{\text{LTP}}$ due to the separate calculation of CLTP components. Meanwhile, that of the histogram computation is estimated as $\mathcal{Q}_{\Psi} \approx 6 \times \mathcal{H} \times \mathcal{W}$. So we have $\mathcal{Q}_{\mathcal{I}} \approx (\mathcal{Q}_{\text{UL}} + \mathcal{Q}_{\text{CLTP}} + \mathcal{Q}_{\Psi}) \approx \mathcal{O}(P \times \mathcal{H} \times \mathcal{W})$.

For a video $\mathcal{V}$ with $\mathcal{T}$ frames which is split into three plane-image collections $\{f_{XY}, f_{XT}, f_{YT}\}$, the complexity of computing the robust patterns on f_{XY} is approximated as $\mathcal{Q}_{\Upsilon}^{XY} \approx \mathcal{T} \times (\mathcal{Q}_{\mathcal{I}} + 2 \times \mathcal{Q}_{\mathcal{L}})$, where $\mathcal{Q}_{\mathcal{L}} \approx \mathcal{H} \times \mathcal{W} \times \mathcal{N}_v$ is the complexity of calculating the optical-flow components with the best $\mathcal{N}_v$ flow vectors [59], ($\mathcal{N}_v = 3$ in practice). Meanwhile, $\mathcal{Q}_{\Upsilon}^{XT} = \mathcal{Q}_{\Upsilon}^{YT} \approx \mathcal{T} \times \mathcal{Q}_{\mathcal{I}}$ is for the computation on f_{XT} and f_{YT}. As a result, $\mathcal{Q}_{\text{RCLTP}} \approx \mathcal{Q}_{\Upsilon}^{XY} + \mathcal{Q}_{\Upsilon}^{XT} + \mathcal{Q}_{\Upsilon}^{YT}$, i.e., $\mathcal{Q}_{\text{RCLTP}} \approx \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$. So it can be seen that our $\mathcal{Q}_{\text{RCLTP}}$ is equivalent to that of other LBP-based descriptors: MEWLSP [28], VLBP [30], CSAP-TOP [60], FoSIG [61], CVLBP [31], V-BIG [62], HILOP [33], MDP [27], etc. (refer to those works for more detail), whilst the proposed descriptor achieved better results compared to theirs (see Table 7). In terms of the processing time, we measured the runtime of some of them for encoding a DT video. The last column of Table 3 shows that RCLTP would take more time in processing a Dyntex++'s video due to the extraction and encoding of optical-flow components. It is noteworthy that all those methods were implemented in raw MATLAB codes with single-threading on a 64-bit Linux desktop with CPU Core i7 3.4GHz 16G RAM.

4.3.3 Assessments of RCLTP descriptors

For the experimental results in Table 6, it can be stated that our RCLTP based on the following major factors in order to boost the discrimination, compared to CLTP [50].

Table 4: Performance (%) of RLTP compared to non-RLTP and the original LTP [45].

	Original LTP [45]				**Our RLTP**			non-RLTP		
$\{\delta_1[,\delta_2]\}$	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{2,3\}$	$\{3,4\}$	$\{1,2\}$	$\{2,3\}$	$\{3,4\}$
50-LOO	96.00	97.00	96.00	97.00	**97.50**	97.00	97.00	95.50	96.50	96.50
50-4fold	96.00	97.00	96.50	97.00	**97.50**	97.00	97.00	95.50	96.50	97.00
Beta	90.12	90.12	90.74	90.12	**92.59**	91.36	91.98	88.89	90.12	88.89
DynTex++	91.58	91.72	90.86	90.33	93.67	93.20	**93.93**	92.83	92.72	91.49

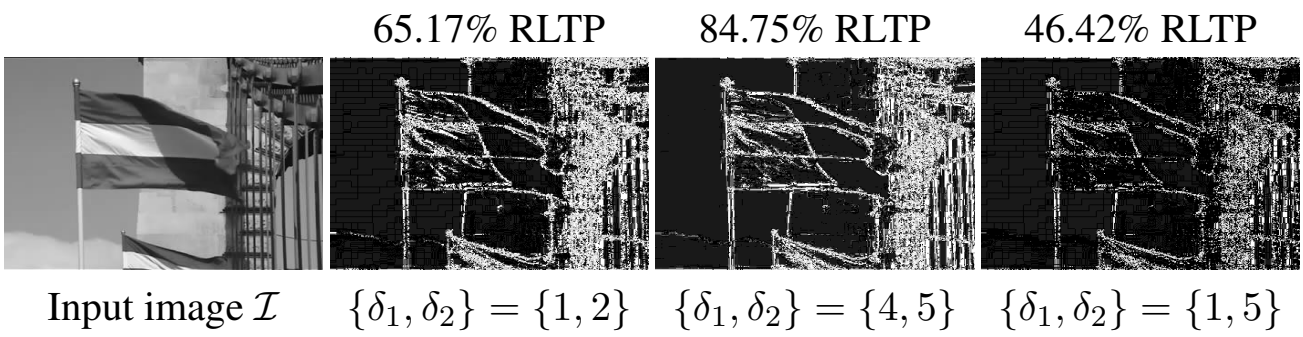

Figure 4: $\mathrm{RCLTP\_S}_{8,1}^{\delta_1,\delta_2,u}$ for image $\mathcal{I}$ with pairs of thresholds.

- As RCLTP is derived from CLTP [50], it would maintain the properties of CLTP in the way of the histogram construction.
- Thanks to addressing RLTP in the encoding, RCLTP is very descriptive and has a solid preponderance of the overall CLTP features. The experiments have validated that locating the robust patterns with the deviation of thresholds can lead to better performance. For instance of $\{\delta_1,\delta_2\}=\{4,5\}$, RCLTP pointed out 96.91% on *Beta*, ≈0.6% higher than CLTP encoded by each threshold (see Table 5).
- RCLTP would be affected by the amplitude of threshold deviation. Indeed, Table 6 shows RCLTP with the high deviations has downward performance. It is because the higher amplitude is addressed, the fewer robust patterns can be located. Fig. 4 shows the dominant quantity (84.75%) of robust patterns with $\{\delta_1,\delta_2\}=\{4,5\}$ compared to the other deviations. This also explains why RCLTP with this setting achieved such good rates (see Table 6).
- Also, thanks to the preponderance, RCLTP is better than the original CLTP in the ability of noise resistance. Indeed, the performance of CLTP was dramatically reduced to 79.55% at the salt-and-pepper noise level $\rho=60\%$, while that of RCLTP was decreased by less than 1%, i.e., 85.98% (see Table 2). In terms of Gaussian noise, it can be observed from Table 3 that whatever the noise level is, RCLTP is more generally stable than non-RLTP, CLTP, and other LBP-based methods. Particularly, RCLTP has the best noise robustness on the challenging scheme "*Gamma*", while HoGF [58] is more stable but does not obtain better performance in DT recognition, e.g., ours is over 1% higher than HoGF's in the heaviest noise level (i.e., dB=1), and 3% better in dB=5.

Since the setting of $\{\delta_1,\delta_2\}=\{4,5\}$ pointed out the best RCLTP's results on most datasets (see Table 6), it should be recommended for real applications as well as for the below comprehensive comparisons.

4.4 Comparison with local-feature-based methods

It can be observed from Table 7 that our proposed descriptor of the robust patterns obtained very good performance compared to the existing local-feature-based methods. Concretely, RCLTP mostly achieved the best rates compared to other LTP-based approaches, i.e., LTP-Lac [66], LTP [45], CLTP [50]. For comparing with the LBP-based ones, rates of RCLTP are respectively 100%, 100%, 99.25%, and 99.46% on *50-LOO*, *50-4fold*, *9-class*, and *8-class*, the best rates among all existing local-based methods. Also, ours obtained competitive the highest rates on the challenging schemes: 100% on *Alpha*, 96.91% on *Beta*, 95.83%, and 96.67% on DynTex++. It can be observed that MDP [27] has the same rate on *Beta* (96.91%), but it is not better than ours on the remaining schemes. Similarly, MEWLSP [28] is superior on DynTex++ (98.48%), but it is not higher than ours on all UCLA schemes, and it has not been verified on the challenging ones. It should be emphasized that the

Table 5: Rates (%) of RCLTP compared to CLTP [50].

Descriptor	Original CLTP [50]					**Our RCLTP**			
$\{\delta_1[,\delta_2]\}$	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{5\}$	$\{1,2\}$	$\{2,3\}$	$\{3,4\}$	$\{4,5\}$
Beta	95.68	**96.91**	96.30	96.30	96.30	95.68	96.30	96.30	**96.91**
Gamma	95.83	95.45	95.83	95.83	95.83	**96.21**	95.83	95.45	95.83

Table 6: Classification rates (%) of RCLTP on DT datasets.

Dataset	UCLA				DynTex			DynTex++
$\{\delta_1,\delta_2\}$	50-LOO	50-4fold	9-class	8-class	Alpha	Beta	Gamma	
$\{1,2\}$	**100**	**100**	99.15	99.13	**100**	95.68	**96.21**	96.31
$\{2,3\}$	**100**	**100**	99.00	98.48	**100**	96.30	95.83	96.42
$\{3,4\}$	**100**	**100**	**99.40**	98.75	**100**	96.30	95.45	95.59
$\{4,5\}$	**100**	**100**	99.25	**99.46**	**100**	**96.91**	95.83	**96.67**
$\{1,3\}$	**100**	**100**	98.90	98.60	**100**	93.83	95.45	96.20
$\{2,4\}$	**100**	**100**	98.95	98.48	**100**	95.68	95.83	95.86
$\{1,4\}$	**100**	**100**	99.05	98.04	**100**	95.06	95.45	96.09
$\{2,5\}$	**100**	**100**	98.45	98.15	**100**	96.30	95.45	95.68
$\{1,5\}$	**100**	**100**	98.80	98.04	**100**	95.68	94.70	95.76

above existing methods addressed the whole local features of the analysis to form their descriptors, while ours just took advantage of the key features. This has consolidated the efficacy of our proposal.

5 Conclusions

We have suggested a novel concept for the location of robust patterns based on the invariant of LTP-based features. It can handle two issues of the dominant patterns which the existing local methods [43, 44, 38, 37] have either missed or dealt with inadequately: *i)* maintaining the local features on the way of descriptor construction, *ii)* the noise sensibility caused by small gray-scale changes of local neighbors. RLTP has been adapted to CLTP [50] to efficiently locate the robust CLTP-based patterns of the spatial-temporal features and optical-flow-based motions for DT representation. Experiments have validated that ours has very good performance compared to the non-robust descriptor, the original ones (i.e., LTP [45] and CLTP [50]), as well as the local-feature-based methods.

For perspectives, instead of using $\mathrm{RLTP}^{\mathrm{u/l}}$ for the locating processes, it can consider other dominant patterns based on the invariance of $\mathrm{CLTP\_M}^{\mathrm{u/l}}$ and $\mathrm{CLTP\_C}^{\mathrm{u/l}}$. Also, the concepts of n-ary codes [67] in various significant conditions can be taken into account for our locating robust patterns to have for further enhancement.

Acknowledgments

We would like to express our sincere appreciation for the editors and the reviewers, who pointed out the valuable and insightful remarks allowing us to clarify the presentation of this work. Thanh Phuong Nguyen was supported by ANR ASTRID ROV-Chasseur. Also, we would like to send many thanks to those in Faculty of IT, HCMC University of Technology and Education, Thu Duc City, Ho Chi Minh City, Vietnam, who gave us enthusiastic support.

References

[1] P. Saisan, G. Doretto, Y. N. Wu, S. Soatto, Dynamic texture recognition, in: CVPR, 2001, pp. 58–63.

[2] I. Ali, J. Mille, L. Tougne, Space-time spectral model for object detection in dynamic textured background, Pattern Recognit. Lett. 33 (13) (2012) 1710–1716.

[3] A. C. Cruz, B. Bhanu, N. S. Thakoor, Background suppressing gabor energy filtering, Pattern Recognit. Lett. 52 (2015) 40–47.

Table 7: Comparison of classification rates (%) on datasets.

Dataset	UCLA				DynTex			DynTex++
Method	50-LOO	50-4fold	9-class	8-class	Alpha	Beta	Gamma	
VLBP [30]	-	89.50	96.30	91.96	-	-	-	94.98
LBP-TOP [30]	-	94.50	96.00	93.67	98.33	88.89	84.85	94.05
CVLBP [31]	-	93.00	96.90	95.65	-	-	-	-
HLBP [26]	95.00	95.00	98.35	97.50	-	-	-	96.28
CLSP [32]	99.00	99.00	98.60	97.72	95.00	91.98	91.29	95.50
MEWLSP [28]	96.50	96.50	98.55	98.04	-	-	-	**98.48**
WLBPC [54]	-	96.50	97.17	97.61	-	-	-	95.01
CVLBC [63]	98.50	99.00	99.20	99.02	-	-	-	91.31
CSAP [60]	99.50	99.50	96.80	95.98	96.67	92.59	90.53	-
FoSIG [61]	99.50	**100**	98.95	98.59	96.67	92.59	92.42	95.99
V-BIG [62]	99.50	99.50	97.95	97.50	**100**	95.06	94.32	96.65
HILOP [33]	99.50	99.50	97.80	96.30	96.67	91.36	92.05	96.21
MDP [27]	**100**	**100**	98.70	98.70	98.33	**96.91**	92.05	95.86
RUBIG [64]	**100**	**100**	99.20	99.13	**100**	95.68	93.56	97.08
VSCR [65]	-	99.43	-	-	92.24	-	-	-
LTP-Lac [66]	-	99.70	96.80	**99.20**	89.60	80.90	79.90	94.80
LTP* [45]	97.00	97.00	96.60	95.65	98.33	90.12	93.56	91.72
Our RCLTP	**100**	**100**	**99.25**	**99.46**	**100**	**96.91**	**95.83**	96.67

Note: "-" means "not available". "*" means results of local DT descriptor $\text{LTP}^{\delta=2}$ [45] that are reported by this work with the Classic+NL's components included [46].

[4] P. Yang, Q. Liu, D. N. Metaxas, Boosting encoded dynamic features for facial expression recognition, Pattern Recognit. Lett. 30 (2) (2009) 132–139.

[5] T. T. Nguyen, T. P. Nguyen, A comprehensive taxonomy of dynamic texture representation, ACM Computing Surveys 55 (1) (2023) 1–39.

[6] A. B. Chan, N. Vasconcelos, Modeling, clustering, and segmenting video with mixtures of dynamic textures, IEEE Trans. PAMI 30 (5) (2008) 909–926.

[7] A. Mumtaz, E. Coviello, G. R. G. Lanckriet, A. B. Chan, Clustering dynamic textures with the hierarchical EM algorithm for modeling video, IEEE Trans. PAMI 35 (7) (2013) 1606–1621.

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

[9] Y. Qiao, Z. Xing, Dynamic texture classification using multivariate hidden markov model, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 101-A (1) (2018) 302–305.

[10] C. Peh, L. F. Cheong, Synergizing spatial and temporal texture, IEEE Trans. IP 11 (10) (2002) 1179–1191.

[11] S. Fazekas, D. Chetverikov, Analysis and performance evaluation of optical flow features for dynamic texture recognition, Sig. Proc.: Image Comm. 22 (7-8) (2007) 680–691.

[12] T. T. Nguyen, T. P. Nguyen, F. Bouchara, X. S. Nguyen, Directional beams of dense trajectories for dynamic texture recognition, in: ACIVS, Vol. 11182, 2018, pp. 74–86.

[13] B. K. P. Horn, B. G. Schunck, Determining optical flow, Artificial Intelligence 17 (1-3) (1981) 185–203.

[14] Y. Xu, Y. Quan, Z. Zhang, H. Ling, H. Ji, Classifying dynamic textures via spatiotemporal fractal analysis, Pattern Recognition 48 (10) (2015) 3239–3248.

[15] Y. Xu, S. B. Huang, H. Ji, C. Fermüller, Scale-space texture description on sift-like textons, CVIU 116 (9) (2012) 999–1013.

[16] H. Ji, X. Yang, H. Ling, Y. Xu, Wavelet domain multifractal analysis for static and dynamic texture classification, IEEE Trans. IP 22 (1) (2013) 286–299.

[17] Y. Quan, Y. Sun, Y. Xu, Spatiotemporal lacunarity spectrum for dynamic texture classification, CVIU 165 (2017) 85–96.

[18] S. R. Arashloo, J. Kittler, Dynamic texture recognition using multiscale binarized statistical image features, IEEE Trans. Multimedia 16 (8) (2014) 2099–2109.

[19] T. T. Nguyen, T. P. Nguyen, F. Bouchara, A novel filtering kernel based on difference of derivative gaussians with applications to dynamic texture representation, SPIC 98 (2021) 116394.

[20] X. Qi, C.-G. Li, G. Zhao, X. Hong, M. Pietikainen, Dynamic texture and scene classification by transferring deep image features, Neurocomputing 171 (2016) 1230 – 1241.

[21] V. Andrearczyk, P. F. Whelan, Convolutional neural network on three orthogonal planes for dynamic texture classification, Pattern Recognition 76 (2018) 36 – 49.

[22] S. R. Arashloo, M. C. Amirani, A. Noroozi, Dynamic texture representation using a deep multi-scale convolutional network, J. Vis. Commun. Image Represent. 43 (2017) 89–97.

[23] Y. Quan, Y. Huang, H. Ji, Dynamic texture recognition via orthogonal tensor dictionary learning, in: ICCV, 2015, pp. 73–81.

[24] Y. Quan, C. Bao, H. Ji, Equiangular kernel dictionary learning with applications to dynamic texture analysis, in: CVPR, 2016, pp. 308–316.

[25] L. C. Ribas, J. J. de Mesquita Sá Junior, A. Manzanera, O. M. Bruno, Learning graph representation with randomized neural network for dynamic texture classification, Appl. Soft Comput. 114 (2022) 108035.

[26] D. Tiwari, V. Tyagi, A novel scheme based on local binary pattern for dynamic texture recognition, CVIU 150 (2016) 58–65.

[27] T. T. Nguyen, T. P. Nguyen, F. Bouchara, X. S. Nguyen, Momental directional patterns for dynamic texture recognition, CVIU 194 (2020) 102882.

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

[29] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. PAMI 24 (7) (2002) 971–987.

[30] G. Zhao, M. Pietikäinen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Trans. PAMI 29 (6) (2007) 915–928.

[31] D. Tiwari, V. Tyagi, Dynamic texture recognition based on completed volume local binary pattern, MSSP 27 (2016) 563–575.

[32] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Completed local structure patterns on three orthogonal planes for dynamic texture recognition, in: IPTA, 2017, pp. 1–6.

[33] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Dynamic texture representation based on hierarchical local patterns, in: ACIVS, 2020, pp. 277–289.

[34] G. Zhao, T. Ahonen, J. Matas, M. Pietikäinen, Rotation-invariant image and video description with local binary pattern features, IEEE Trans. IP 21 (4) (2012) 1465–1477.

[35] M. M. Feraidooni, D. Gharavian, A new approach for rotation-invariant and noise-resistant texture analysis and classification, Mach. Vis. Appl. 29 (3) (2018) 455–466.

[36] F. Bianconi, E. González, A. Fernández, Dominant local binary patterns for texture classification: Labelled or unlabelled?, Pattern Recognit. Lett. 65 (2015) 8–14.

[37] J. Chen, V. Kellokumpu, G. Zhao, M. Pietikäinen, RLBP: robust local binary pattern, in: BMVC, 2013.

[38] D. Tiwari, V. Tyagi, An auto tuned noise resistant descriptor for dynamic texture recognition, MTA 76 (20) (2017) 21225–21246.

[39] E. González, A. Fernández, F. Bianconi, General framework for rotation invariant texture classification through co-occurrence of patterns, J. Math. Imaging Vis. 50 (3) (2014) 300–313.

[40] F. Bianconi, A. Fernández, Rotation invariant co-occurrence features based on digital circles and discrete fourier transform, Pattern Recognit. Lett. 48 (2014) 34–41.

[41] L. Nanni, S. Brahnam, A. Lumini, Selecting the best performing rotation invariant patterns in local binary/ternary patterns, in: IPCV, 2010, pp. 369–375.

[42] Y. Guo, G. Zhao, M. Pietikäinen, Discriminative features for texture description, PR 45 (10) (2012) 3834–3843.

[43] S. Liao, M. W. K. Law, A. C. S. Chung, Dominant local binary patterns for texture classification, TIP 18 (5) (2009) 1107–1118.

[44] N. P. Doshi, G. Schaefer, S. Hossain, Improved dominant local binary pattern texture features, in: ICIEV, 2016, pp. 1157–1160.

[45] X. Tan, B. Triggs, Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions, IEEE Trans. IP 19 (6) (2010) 1635–1650.

[46] D. Sun, S. Roth, M. J. Black, Secrets of optical flow estimation and their principles, in: CVPR, 2010, pp. 2432–2439.

[47] P. Weinzaepfel, J. Revaud, Z. Harchaoui, C. Schmid, Learning to detect motion boundaries, in: CVPR, 2015, pp. 2578–2586.

[48] F. Li, T. Kim, A. Humayun, D. Tsai, J. M. Rehg, Video segmentation by tracking many figure-ground segments, in: ICCV, 2013, pp. 2192–2199.

[49] M. Narayana, A. R. Hanson, E. G. Learned-Miller, Coherent motion segmentation in moving camera videos using optical flow orientations, in: ICCV, 2013, pp. 1577–1584.

[50] T. H. Rassem, B. E. Khoo, Completed local ternary pattern for rotation invariant texture classification, TSWJ 2014 (2014) 10.

[51] Z. Guo, L. Zhang, D. Zhang, A completed modeling of local binary pattern operator for texture classification, IEEE Trans. IP 19 (6) (2010) 1657–1663.

[52] H.-C. Shih, H.-Y. Cheng, J.-C. Fu, Image classification using synchronized rotation local ternary pattern, IEEE Sensors Journal 20 (2020) 1656–1663.

[53] A. M. Shamaileh, T. H. Rassem, S. Liew, O. N. A. Sayaydeh, A new feature-based wavelet completed local ternary pattern (feat-wcltp) for texture image classification, IEEE Access 8 (2020) 28276–28288.

[54] D. Tiwari, V. Tyagi, Improved weber's law based local binary pattern for dynamic texture recognition, Multimedia Tools Appl. 76 (5) (2017) 6623–6640.

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

[56] B. Ghanem, N. Ahuja, Maximum margin distance learning for dynamic texture recognition, in: ECCV, 2010, pp. 223–236.

[57] R. Fan, K. Chang, C. Hsieh, X. Wang, C. Lin, LIBLINEAR: A library for large linear classification, JMLR 9 (2008) 1871–1874.

[58] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Prominent local representation for dynamic textures based on high-order gaussian-gradients, IEEE Trans. on Multimedia 23 (2021) 1367–1382.

[59] J. Xiang, Z. Li, D. T. Blaauw, H. Kim, C. Chakrabarti, Low complexity optical flow using neighbor-guided semi-global matching, in: ICIP, 2016, pp. 4483–4487.

[60] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Completed statistical adaptive patterns on three orthogonal planes for recognition of dynamic textures and scenes, JEI 27 (05) (2018) 053044.

[61] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Smooth-invariant gaussian features for dynamic texture recognition, in: ICIP, 2019, pp. 4400–4404.

[62] T. T. Nguyen, T. P. Nguyen, F. Bouchara, N. Vu, Volumes of blurred-invariant gaussians for dynamic texture classification, in: CAIP, 2019, pp. 155–167.

[63] X. Zhao, Y. Lin, J. Heikkilä, Dynamic texture recognition using volume local binary count patterns with an application to 2d face spoofing detection, Trans. Multim. 20 (3) (2018) 552–566.

[64] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Rubik gaussian-based patterns for dynamic texture classification, Pattern Recognition Letters 135 (2020) 180–187.

[65] J. Xie, Y. Fang, Dynamic texture recognition with video set based collaborative representation, IVC 55 (2016) 86–92.

[66] Y. Sun, Y. Xu, Y. Quan, Characterizing dynamic textures with space-time lacunarity analysis, in: ICME, 2015, pp. 1–6.

[67] F. Bianconi, E. González, Counting local *n*-ary patterns, Pattern Recognit. Lett. 117 (2019) 24–29.