Pattern Recognition Letters 51 (2015) 94–100

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Local Directional Texture Pattern image descriptor ☆

Adín Ramírez Rivera [a,*], Jorge Rojas Castillo [b], Oksam Chae [b]

[a] Escuela de Informática y Telecomunicaciones, Universidad Diego Portales, Ejército 441, Santiago 8320000, Chile
[b] Department of Computer Engineering, Kyung Hee University, 1 Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do 446-701, Republic of Korea

ARTICLE INFO

Article history:
Received 17 January 2014
Available online 16 September 2014

Keywords:
Directional number pattern
Expression recognition
Face descriptor
Image descriptor
Local pattern
Scene recognition

ABSTRACT

Deriving an effective image representation is a critical step for a successful automatic image recognition application. In this paper, we propose a new feature descriptor named Local Directional Texture Pattern (LDTP) that is versatile, as it allows us to distinguish person's expressions, and different landscapes scenes. In detail, we compute the LDTP feature, at each pixel, by extracting the principal directions of the local neighborhood, and coding the intensity differences on these directions. Consequently, we represent each image as a distribution of LDTP codes. The mixture of structural and contrast information makes our descriptor robust against illumination changes and noise. We also use Principal Component Analysis to reduce the dimension of the multilevel feature set, and test the results on this new descriptor as well.

1. Introduction

Nowadays several applications need to recognize something through visual cues, such as faces, expressions, objects, or scenes. These applications need robust descriptors that discriminate between classes, but that are general enough to incorporate variations within the same class. However, most of the existing algorithms have a narrow application spectrum, *i.e.*, they focus on some specific task. For example, previous methods [1,2] designed for face analysis, cannot be used to recognize scenes, as these methods have been tuned for fine texture representation. Similarly, descriptors [3,4] tuned for scene recognition under-perform on other tasks. Thereby, in this paper, we design and test a robust descriptor capable of modeling fine textures as well as coarse ones.

A wide range of algorithms have been proposed to describe micro-patterns. The most common ones are the appearance-based methods [5] that use image filters, either on the whole-image, to create holistic features, or some specific region, to create local features, to extract the appearance change in the image. The performance of the appearance-based methods is excellent in constrained environment but their performance degrade in environmental variation [6]. In the literature, there are many methods for the holistic class, such as Eigenfaces and Fisherfaces. Although these methods have been studied widely, local descriptors have gained attention because of their robustness to illumination and pose variations. The Local Binary Pattern (LBP) [1] is by far the most popular one, and has been successfully applied to several problem domains [5,7,8]. Despite LBP's robustness to monotonic illumination, it is sensitive to non-monotonic illumination variation, and shows poor performance in presence of random noise [7]. Tan and Triggs [9] proposed an improvement of LBP by introducing a ternary pattern (LTP) which uses a threshold to stabilize the micro-patterns. A directional pattern (LDP) [2] has been proposed to overcome the limitations of LBP. However, it suffers in noisy conditions, is sensitive to rotations, and cannot detect different transitions in the intensity regions. Similarly, many other methods appeared that extract information and encoded it in a similar way like LBP, such as infrared [10], near infrared [11], and phase information [12,13]. Nevertheless, all these methods inherit the sensitivity problem, *i.e.*, the feature being coded into the bit-string is prone to change due to noise or other variations. Thereby, the directional-number-based methods [14–17] appeared as a solution to the common bit-string representation, as these methods use an explicit coding scheme in which the prominent directions are embedded into the code. However, all these methods still encode only one type of information, *e.g.*, intensity or direction, which limits their description capabilities.

Therefore, in this paper, we propose a novel feature descriptor named Local Directional Texture Pattern (LDTP) which exploits the advantages of both directional and intensity information in the image. The combination of both features outperforms the singled-feature counterparts, *e.g.*, LBP, LTP, LDP, among others. The proposed method identifies the principal directions from the local neighborhood, and then extracts the prominent relative intensity information. Following, LDTP characterizes the neighborhood by mixing these two features in a single code. On the contrary, previous methods rely on one type

☆ This paper has been recommended for acceptance by L. Yin.
* Corresponding author. Tel.: +56 02 26768134.
E-mail address: adin.ramirez@mail.udp.cl (A. Ramírez Rivera).

http://dx.doi.org/10.1016/j.patrec.2014.08.012
0167-8655/

of information and use a sensitive coding strategy. In detail, LDTP encodes the structural information in a local neighborhood by analyzing its directional information and the difference between intensity values of the first and second maximum edge's responses. This mechanism is consistent against noise, since the edge response is more stable than intensity, and the use of relative intensity values makes our method more robust against illumination changes and other similar conditions. Moreover, given that we encode only the prominent information of the neighborhood, our method is more robust to changes in comparison to other methods, as we dispose insignificant details that may vary between instances of the image. Consequently, we convey more reliable information of the local texture, rather than coding all the information that may be misleading, not important, or affected by noise. Furthermore, we evaluate the performance of the proposed LDTP feature with a machine learning method, Support Vector Machine, on five different databases for expression recognition and three different databases for scene recognition to demonstrate its robustness and versatility.

2. Local Directional Texture Pattern

LBP feature labels each pixel by thresholding a set of sparse points of its circular neighborhood, and encodes that information in a string of bits. Similarly, LDP encodes the principal directions of each pixel's neighborhood into an eight bit string. However, these methods only mark which neighbor has the analyzed characteristic on or off. Furthermore, this *ad hoc* construction overlooks the prominent information in the neighborhood, as all the information in the neighborhood (regardless of its usefulness) is poured into the code. In contrast, we create a code from the principal directions of the local neighborhood, similar to the directional numbers [14–17]. However, the later has the problem of using only the structural information of the neighborhood. Therefore, we propose to extract the contrast information from the principal directions to enhance the description of our code. In other words, we code the principal direction and the intensity difference of the two principal directions into one number. This approach allows us to encode the prominent texture information of the neighborhood that is revealed by its principal directions.

To compute LDTP, we calculate the principal directional numbers of the neighborhood using the Kirsch compass masks [18]—in eight different directions. We define our directional number as

$$P^1_{\mathrm{dir}} = \arg\max_i \{\mathbb{I}_i \mid 0 \le i \le 7\}, \tag{1}$$

where P^1_{dir} is the principal directional number, $\mathbb{I}_i$ is the absolute response of the convolution of the image, I, with the ith Kirsch compass mask, M_i, defined by

$$\mathbb{I}_i = |I * M_i|. \tag{2}$$

Thus, we compute the absolute value of the eight Kirsch mask's responses, $\{M_0, \ldots, M_7\}$, applied to a particular pixel. More precisely, we take the two greatest responses, P^1_{dir} and P^2_{dir}. Therefore, the second directional number, P^2_{dir}, is computed in the same way, with the difference that we take the second maximum response in Eq. (1) instead. These directions signal the principal axis of the local texture.

In each of the two principal directions, we compute the intensity difference of the opposed pixels in the neighborhood. That is

$$d_n^{(x,y)} = I(x_{P^n_{\mathrm{dir}}+}, y_{P^n_{\mathrm{dir}}+}) - I(x_{P^n_{\mathrm{dir}}-}, y_{P^n_{\mathrm{dir}}-}), \tag{3}$$

where d_n is the nth difference for the pixel (x, y) in the nth principal direction, $I(x_{P^n_{\mathrm{dir}}+}, y_{P^n_{\mathrm{dir}}+})$ corresponds to the intensity value of the pixel $(x_{P^n_{\mathrm{dir}}+}, y_{P^n_{\mathrm{dir}}+})$, which is the next pixel in the given principal direction, and $I(x_{P^n_{\mathrm{dir}}-}, y_{P^n_{\mathrm{dir}}-})$ is the intensity value of the pixel $(x_{P^n_{\mathrm{dir}}-}, y_{P^n_{\mathrm{dir}}-})$, which is the previous pixel in the given principal direction. In other words, the next and previous pixel positions defined by each direction are

$$x_{P^n_{\mathrm{dir}}\pm} = \begin{cases} x \pm 1 & \text{if } P^n_{\mathrm{dir}} \in \{0, 1, 7\}, \\ x & \text{if } P^n_{\mathrm{dir}} \in \{2, 6\}, \\ x \mp 1 & \text{if } P^n_{\mathrm{dir}} \in \{3, 4, 5\}, \end{cases} \tag{4}$$

$$y_{P^n_{\mathrm{dir}}\pm} = \begin{cases} y \pm 1 & \text{if } P^n_{\mathrm{dir}} \in \{1, 2, 3\}, \\ y & \text{if } P^n_{\mathrm{dir}} \in \{0, 4\}, \\ y \mp 1 & \text{if } P^n_{\mathrm{dir}} \in \{5, 6, 7\}. \end{cases} \tag{5}$$

This local difference, is equivalent to the local threshold that LBP does. Unlike the LBP binary encoding, we encode the difference using three levels (negative, equal, and positive), which creates a more distinctive code for the neighborhood. Then each difference is encoded as

$$D_f(d) = \begin{cases} 0, & \text{if } -\varepsilon \le d \le \varepsilon \\ 1, & \text{if } d < -\varepsilon \\ 2, & \text{if } d > \varepsilon, \end{cases} \tag{6}$$

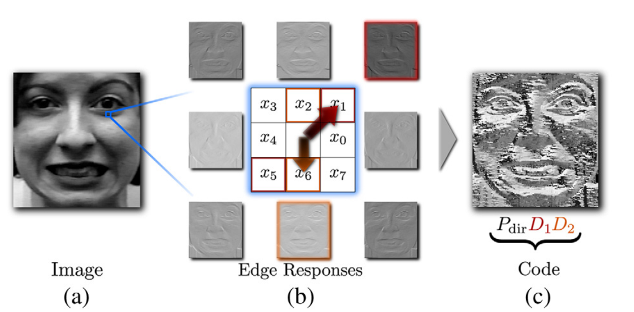
where D_f is the encoded intensity difference, d is the actual intensity difference, ε is a threshold value (in our experiments we use $\varepsilon = 15$).

Consequently, the code is created by concatenating the binary form of the principal direction, and the two differences. This concatenation can be represented by the following operation

$$\mathrm{LDTP}(x, y) = 16P^{1(x,y)}_{\mathrm{dir}} + 4D_f(d_1^{(x,y)}) + D_f(d_2^{(x,y)}), \tag{7}$$

where $\mathrm{LDTP}(x, y)$ is the code for the pixel (x, y), $P^{1(x,y)}_{\mathrm{dir}}$ is the principal directional number (from 0 to 7) of the neighborhood of the pixel (x, y), and $D_f(d_1^{(x,y)})$ and $D_f(d_2^{(x,y)})$ are the first and second coded differences of the neighborhood of the pixel (x, y), respectively. The length of the code is $72 = 8 \times 3 \times 3$, as the possible values for the directional number is 8, and 3 for each difference.

For example, consider the neighborhood shown in Fig. 1(d), first we compute the Kirsch mask responses in the neighborhood—we show them in their respective orientation in Fig. 1(e). The principal, M_1, and the secondary, M_6, directions are shown in red and blue, respectively. Then, we compute the intensity difference of the corresponding pixel intensities in these directions [as shown by the colored pairs in Fig. 1(d)]. In this case, the differences are: $d_1 = 143 - 133 = 10$, and $d_2 = 137 - 141 = -4$, which are transformed with Eq. (6) into $D_f(d_1) = 0$, and $D_f(d_2) = 0$, assuming a threshold $\varepsilon = 15$. Finally, we create the LDTP code by concatenating the binary form of the principal direction index, and the two differences as shown in Fig. 1(f).



$$\begin{bmatrix} 139 & 141 & 143 \\ 134 & 140 & 145 \\ 133 & 137 & 133 \end{bmatrix} \quad \begin{bmatrix} -3 & 69 & 117 \\ -67 & x & 53 \\ -83 & -91 & 5 \end{bmatrix} \quad \overbrace{001}^{P_{\mathrm{dir}}}\ \overbrace{00}^{D_1}\ \overbrace{00}^{D_2}$$

(d) (e) (f)

Fig. 1. Example of the LDTP code computation (details are in the text). (a) Original image. (b) Edge responses after applying Kirsch masks. (c) Coded image. (d) Sample neighborhood (intensity values). (e) Edge responses of the sample neighborhood shown in (d). (f) Code of the neighborhood shown in (d). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)