Local Quaternary Patterns and Feature Local Quaternary Patterns

Jiayu Gu and Chengjun Liu
The Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA

Abstract - *This paper presents a new local texture descriptor, Local Quaternary Patterns (LQP) and its extension, Feature Local Quaternary Patterns (FLQP). The LQP, which encodes four relationships of local texture, includes more information of local texture than the Local Binary Patterns (LBP) and Local Ternary Patterns (LTP). The FLQP which encodes both local and feature information is expected to perform better than the LQP for texture description and pattern recognition. To reduce the size of feature dimensions and histograms of both LQP and FLQP, a new coding schema is proposed to split the LQP and FLQP into two binary codes: the upper and lower binary codes. As a result, the total possible values of split LQP and FLQP are reduced to 512. The feasibility of the proposed LQP and FLQP methods is demonstrated on an eye detection problem. Experimental results using the BioID database show that both the FLQP and the LQP methods archive better performance than the feature LTP, the LTP, the feature LBP and the LBP methods. Specifically, the FLQP method achieves the highest eye detection rate among all the competing methods.*

Keywords: Local Quaternary Patterns (LQP), Feature Local Quaternary Patterns (FLQP), Local Binary Patterns (LBP), Feature Local Binary Patterns (FLBP) and Local Ternary Patterns (LTP).

1 Introduction

Local Binary Patterns (LBP) [1] has recently become a popular method in texture description for content based image search and feature extraction for pattern recognition and computer vision. The most important properties of the LBP operator are its tolerance against illumination and computational simplicity, which makes it possible to analyze images in real-world in real-time. The Local Binary Patterns (LBP) has been widely applied in many applications, such as face recognition [2-4], face detection [5], [6], and facial expression analysis [7-10].

Tan and Triggs [2] argued that the original LBP tends to be sensitive to noise, especially in near-uniform image regions, because it thresholds exactly at the value of the central pixel. To solve the problem, they proposed 3-valued codes, called Local Ternary Patterns (LTP). In LTP, neighbor pixels are compared with an interval [-r, +r] around the value of the center pixel. A neighbor pixel is assigned 1, 0 or -1, if its value is above +r, in the interval [-r, +r] or below –r, respectively. Because the radius r is not changed with the gray scale, the LTP is no longer a strictly gray-scale invariant texture description, and is less tolerance against illumination than LBP. The LTP has 6561 possible values, which not only poses a computational challenge but also leads to sparse histograms. To solve these problems, a coding scheme is introduced to split a LTP code into two binary codes, the positive one (PLTP) and the negative one (NLTP). Therefore the total number of possible values of two split binary codes is reduced to 512. Some of experiments in [2], [10] show that LTP and LBP achieved similar, although LTP doubles the size of feature dimensions and histograms, and has a higher computational cost than LBP.

To improve the performance of LTP, we present in this paper a new local texture descriptor, Local Quaternary Patterns (LQP) and its extension, Feature Local Quaternary Patterns (FLQP). The LQP encodes four relationships of local texture, and therefore it includes more information of local texture than the LBP and the LTP. To reduce the size of feature dimensions and histograms of LQP, a coding scheme is introduced to split each LQP code into two binary codes, the upper LQP (ULQP) and the lower LQP (LLQP). The possible LQP values are reduced to 512. We [11] have introduced a new Feature Local Binary Patterns (FLBP) method to improve upon the LBP approach. In this paper, we further extend LQP to FLQP, and demonstrate that FLQP improves upon LQP and other competing methods, such as LBP, FLBP, LTP, and Feature LTP (FLTP). The FLQP which encodes both local and feature information is expected to perform better than the LQP for texture description and pattern analysis. We further show that the FLQP code can be split into two binary codes as well, the upper FLQP (UFLQP) and the lower FLQP (LFLQP). To demonstrate the feasibility of the proposed LQP and FLQP methods, we apply them to eye detection on the BioID database. Experimental results show that both FLQP and LQP achieve better eye detection performance than FLTP, LTP, FLBP and LBP. The FLQP method has the best performance among all the methods.

2 Local Binary Patterns and Local Ternary Patterns

Before we introduce our Local Quaternary Patterns (LQP) and Feature Local Quaternary Patterns (FLQP), we briefly review LBP and LTP. LBP define a gray-scale invariant texture description by comparing a center pixel used as a threshold, with those pixels in its local neighborhood [1]. Specifically, for a 3 × 3 neighborhood of a pixel **p** = $[x, y]^t$, each neighbor is labeled by a number from 0 to 7 shown in Fig.1. The neighbors of the pixel **p** thus may be defined as follows:

$$\mathbf{N}(\mathbf{p},i) = [x_i, y_i]^t, \quad i = 0, 1, 2, \ldots, 7 \tag{1}$$

where i is the number used to label a neighbor. The value of the LBP code of a pixel **p**(x, y) is calculated as follows:

$$LBP(\mathbf{p}) = \sum_{i=0}^{7} 2^i S_{lbp}\{G[\mathbf{N}(\mathbf{p},\ i)],\ G(\mathbf{p})\} \tag{2}$$

where G(**p**) and G[**N**(**p**, i)] are the gray levels of the pixel **p** and its neighbor **N**(**p**, i), respectively. S_{lbp} is a threshold function that is defined as follows:

$$S_{lbp}(g_i,\ g_c) = \begin{cases} 1, & if\ \ g_i \ge g_c; \\ 0, & otherwise. \end{cases} \tag{3}$$

0	1	2
7	*p*	3
6	5	4

Fig. 1 The 3 × 3 neighborhood of a pixel **p** and the label of its neighbors

Tan and Triggs proposed a Local Ternary Pattern or LTP operator [2]. In LTP the threshold function S_{ltp} is defined as follows:

$$S_{ltp}(g_i,\ g_c,\ r) = \begin{cases} 1, & if\ g_i \ge g_c + r \\ 0, & if\ |g_i - g_c| < r \\ -1, & if\ g_i \le g_c - r \end{cases} \tag{4}$$

where r is the radius of the interval around the grey level of the central pixel. Fig. 2 shows an example of the computation of the LTP. The grey level of the central pixel is 40 and r is 5. A neighbor pixel is assigned to 1, 0 or -1, if its grey level is greater than or equal to 45, between 44 and 36, or less than or equal to 35, respectively. The total number of the possible LTP codes is 3^8 = 6561, which leads to a large size for the feature dimension and sparse histograms of the LTP codes. To solve the problem, an LTP code is split into two binary codes: the positive and negative halves as shown in Fig. 2. The positive half of LTP (PLTP) is obtained by replacing -1 with 0. The negative half of LTP (NLTP) is obtained by first replacing the 1 with 0 and then changing -1 to 1. Thus an LTP code can be represent by two binary codes. As a result, the total number of the split LTP codes is reduced to 512.

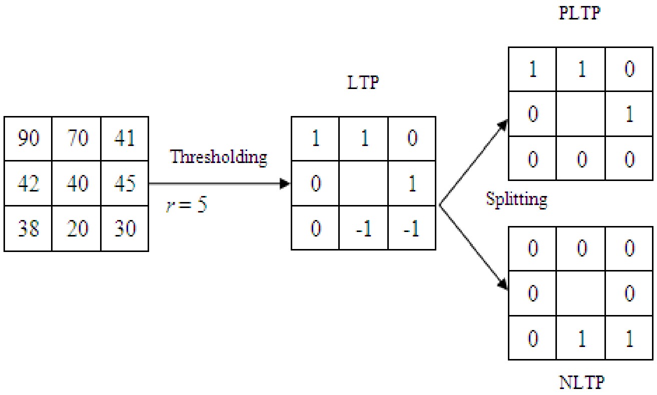


Fig. 2 Computing the LTP and splitting it to two binary codes, PLTP and NLTP

3 Local Quaternary Patterns

We now present our new Local Quaternary Patterns (LQP) which encodes four relationships of local texture, and therefore it includes more information of local texture than LBP and LTP. The threshold function of LQP is defined using two binary digits as follows:

$$S_{lqp}(g_i, g_c, r) = \begin{cases} 11, & \text{if } g_i \ge g_c + r \\ 10, & \text{if } g_c \le g_i < g_c + r \\ 01, & \text{if } g_c - r \le g_i < g_c \\ 00, & \text{if } g_i < g_c - r \end{cases} \tag{5}$$

where r is the radius of the interval around the value of the central pixel and may be defined as follows:

$$r = c + \tau g_c \tag{6}$$

where c is a constant and τ is a parameter to control the contribution of g_c to r. To reduce the total number of codes, an LQP code can be split into two binary codes, the upper and lower halves. The upper half of LQP (ULQP) is obtained by extracting the first digit of LQP code. The lower half of LTP (LLQP) is obtained by extracting the second digit of LQP code. Thus the total number of LQP codes is reduced to 512.

From Eq. 5 we can derive the threshold functions of ULQP and LLQP, S_{ulqp} and S_{llqp} which may be formulated as follows:

$$S_{ulqp}(g_i, g_c) = S_{lbp}(g_i, g_c) \tag{7}$$

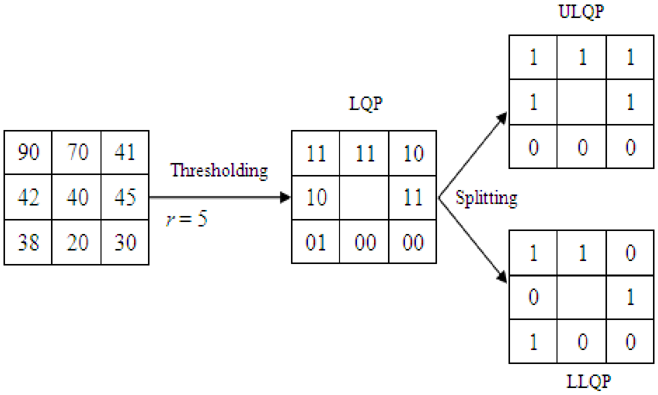
$$S_{llqp}(g_i, g_c, r) = \begin{cases} 1, & if\ g_i \ge g_c + (-1)^{[1 - S_{lbp}(g_i, g_c)]} r; \\ 0, & otherwise. \end{cases} \tag{8}$$

The threshold function of ULQP, S_{ulqp} is equal to the threshold function of LBP and is not depend on the r. The ULQP and LLQP are therefore defined as follows:

$$ULQP(\mathbf{p}) = LBP(\mathbf{p}) \tag{9}$$

$$LLQP(\mathbf{p}) = \sum_{i=0}^{7} 2^i S_{llqp}\{G[\mathbf{N}(\mathbf{p},i)], G(\mathbf{p})\} \tag{10}$$

Note that the ULQP is the same as the LBP. Fig 3 shows an example of the computation of the LQP. The grey level of the central pixel is 40 and r is 5. The ULQP code is 1111001 which is the same as LBP code. For LLQP, a pixel is assigned 1 if it is greater than or equal to 45, or it is less than 40 and greater than or equal to 35, otherwise it is assigned 0.



90	70	41
42	40	45
38	20	30

LQP

11	11	10
10		11
01	00	00

ULQP

1	1	1
1		1
0	0	0

LLQP

1	1	0
0		1
1	0	0

Fig. 3 Computing the LQP and splitting it to two binary codes ULQP and LLQP

4 Feature Local Quaternary Patterns

We [11] have introduced a new Feature Local Binary Patterns (FLBP) method. FLBP generalizes the LBP approach by introducing feature pixels, which may be broadly defined by, for example, the edge pixels, the intensity peaks or valleys in an image. FLBP which encodes both local and feature information, has been shown more effective than LBP for texture description and pattern recognition, such as eye detection. In this paper, we extend LQP to FLQP. Next we briefly review the concepts of distance vector [12] and FLBP method, and then introduce our FLQP method.

In a binary image, each pixel assumes one of two discrete values: 0 or 1. While pixels of value 0 are called the background pixels, pixels of 1 are called feature pixels. Let **p** and **q** represent a pixel and its nearest feature point in a binary image, respectively. The distance vector of **p** pointing to **q** is defined below:

$$\begin{aligned} \mathbf{dv}(\mathbf{p}) &= \mathbf{q} - \mathbf{p}, \\ \mathbf{q} &= \arg \min_{\mathbf{r} \in F} \delta(\mathbf{p},\mathbf{r}) \end{aligned} \tag{11}$$

where F is the set of feature pixels of the binary image. δ is a distance metric.

FLBP is defined on the concepts of True Center (TC) which is the center pixel of a given neighborhood, and Virtual Center (VC) which is a pixel used to replace the center pixel of a given neighborhood. The TC which may be any pixel on the path pointed by **dv(p)** from **p** to **q**, is defined below:

$$\mathbf{C}_t(\mathbf{p}) = \mathbf{p} + \alpha_t \mathbf{dv}(\mathbf{p}) \tag{12}$$

where $\alpha_t \in [0, 1]$ is a parameter that controls the location of the TC. The VC which may be any pixel on the path pointed by **dv(p)** from **p** to **q** as well, is defined below:

$$\mathbf{C}_v(\mathbf{p}) = \mathbf{p} + \alpha_v \mathbf{dv}(\mathbf{p}) \tag{13}$$

where $\alpha_v \in [0, 1]$ is a parameter that controls the location of the VC. The general form of FLBP is defined below:

$$FLBP(\mathbf{p}) = \sum_{i=0}^{7} 2^i S_{lbp}\{G[\mathbf{N}(\mathbf{C}_t(\mathbf{p}),i)], G[\mathbf{C}_v(\mathbf{p})]\} \tag{14}$$

where $\mathbf{N}(\mathbf{C}_t(\mathbf{p}), i)$ defined by Eq. 1 represents the neighbors of the TC. $G[\mathbf{C}_v(\mathbf{p})]$ and $G[\mathbf{N}(\mathbf{C}_t(\mathbf{p}), i)]$ are the gray levels of the VC and the neighbors of the TC, respectively.

Next, we use the grayscale image shown in Fig. 4(a) to illustrate how to compute the FLBP code. We assume that the upper left pixel is at location (1, 1) in the Cartesian coordinate system with the horizontal axis pointing to the right and the vertical axis pointing downwards. As discussed before, feature pixels are broadly defined. Here we define the feature pixels in Fig. 4(a) to be those with gray level greater than 80. Because the pixel at the coordinates (6, 6) in Fig. 4(a) is the only pixel whose gray level is greater than 80, As a result the pixel becomes the only feature pixel in the binary image shown in Fig. 4(b). And this feature pixel becomes the nearest one for all the pixels in Fig. 4(a).

70	1	2	1	3	2	7
1	50	45	66	1	4	9
3	56	30	77	1	2	8
1	58	62	64	20	31	1
2	1	1	34	60	26	2
4	3	4	22	28	90	1
9	5	8	1	2	1	40

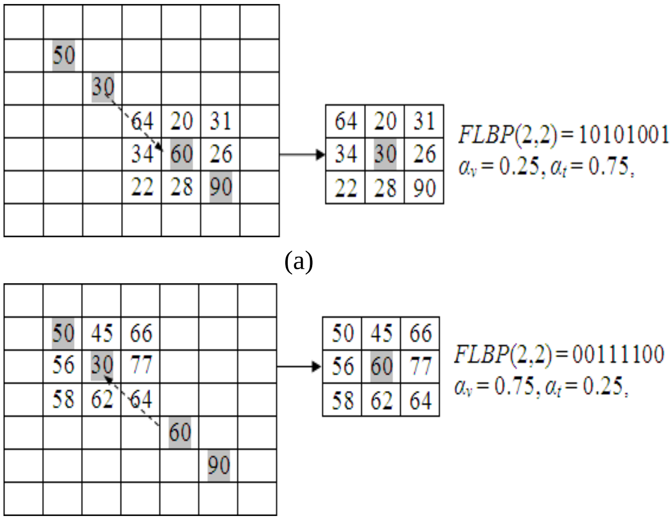
(a)

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	0

(b)

Fig. 4 (a) a grayscale image used in the examples of FLBP computation. (b) The binary feature image derived by extracting feature pixel with gray level greater than 80 from the grayscale image.

We select the pixel **p** at coordinates (2, 2) in Fig. 4(a) as an example to compute the FLBP code. We first compute the **dv**(**p**). Given **p** = $[2, 2]^t$, and **q** = $[6, 6]^t$, we have $\mathbf{dv}(\mathbf{p}) = \mathbf{q} - \mathbf{p} = [4, 4]^t$. Then we determine the locations of TC and VC which are controlled by the parameters α_t and α_v respectively. Fig. 5 shows two examples of the computation of FLBP with different locations of TC and VC. In Fig. 4(a) given $\alpha_t = 0.75$ and $\alpha_v = 0.25$, we have $\mathbf{C}_t(\mathbf{p}) = \mathbf{p} + \alpha_t\, \mathbf{dv}(\mathbf{p}) = [5, 5]^t$, and $\mathbf{C}_v(\mathbf{p}) = \mathbf{p} + \mathbf{dv}(\mathbf{p}) = [3, 3]^t$. Therefore, the TC is the pixel at location (5, 5) and the VC is the pixel at location (3, 3). According to Eq. 14 we replace the gray level 60 of TC at location (5, 5) by the gray level 30 of VC at location (3, 3), and threshed the neighbors of the TC. We have the binary FLBP code: FLBP(2, 2) = 10101001. Fig. 5(b) shows another example of the FLBP(2, 2) computation when $\alpha_t = 0.25$, and $\alpha_v = 0.75$. Similarly, we locate the TC is the pixel at location (3, 3) and the VC is the pixel at location (5, 5). The binary FLBP code becomes: FLBP(2, 2) = 00111100 when $\alpha_t = 0.25$, and $\alpha_v = 0.75$.



(a)

(b)

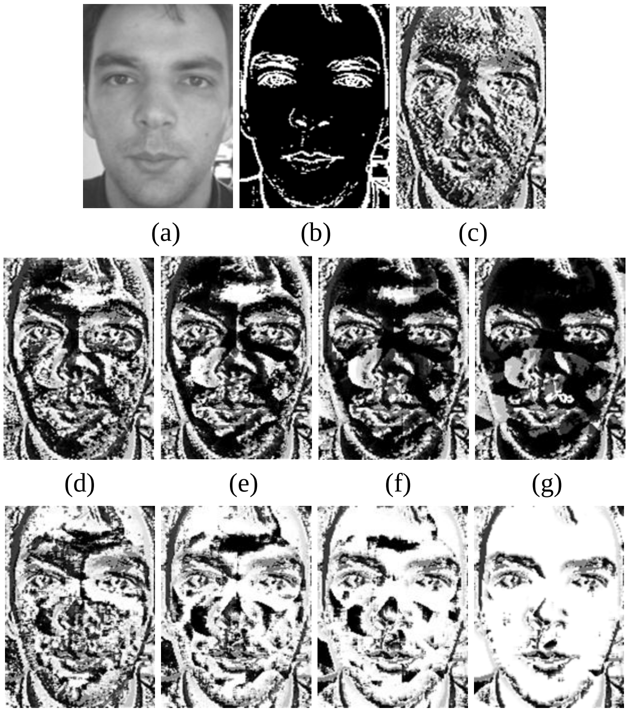
Fig. 5 The computation of FLBP for the pixel at (2, 2). (a) An example when TC ($\alpha_t = 0.75$) is at (5, 5) and VC ($\alpha_v = 0.25$) is at (3, 3) (b) An example when TC ($\alpha_t = 0.25$) is at (3, 3) and VC ($\alpha_v = 0.75$) is at (5, 5)

In [11] we present a new feature pixel extraction method, the LBP with Relative Bias Thresholding (LRBT) method. The LRBT method first computes the LBP representation using the relative bias threshold function defined below:

$$S(g_i, g_c, \beta) = \begin{cases} 1, & \text{if } g_i \geq (1+\beta) g_c; \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

where β is a parameter that controls the contribution of g_c to the bias. Then the LRBT method derives the binary LRBT feature image by converting the LBP image to a binary image, whose feature pixels correspond to those whose LBP code is greater than 0, and the background pixels correspond to the pixels in the LBP image with the LBP code 0.

Fig. 6 shows an example of the FLBP representations of a face image. Fig. 6(a) and (b) display a face image and its binary feature image. The feature pixel of the binary image is derived using our LRBT method when $\beta = 0.1$. Fig. 6(c) shows the LBP image of the face image in Fig. 6(a). Fig. 6(d) - (g) exhibit the FLBP images when $\alpha_t = 0.25, 0.5, 0.75, 1$, respectively, and $\alpha_v = 0$. Fig. 6(h)–(k) exhibit the FLBP images when $\alpha_v = 0.25, 0.5, 0.75, 1$, respectively, and $\alpha_t = 0$.



(a) (b) (c)

(d) (e) (f) (g)

(h) (i) (j) (k)

Fig. 6 (a) A face image (b) The binary LRBT feature image of (a) (c) The LBP representation of the face image of (a) (d)–(g) The FLBP image when $\alpha_v = 0$ and $\alpha_t = 0.25, 0.5, 0.75$, 1 respectively (h)–(k) The FLBP image when $\alpha_t = 0$ and $\alpha_v = 0.25, 0.5, 0.75$, 1 respectively

Our new feature local quaternary patterns or FLQP can be split into two binary codes, the upper half of FLQP (UFLQP) and the lower half of FLQP (LFLQP) using the threshold functions defined in Eqs. 7 and 8, respectively. The UFLQP is equivalent to FLBP. The general form of the UFLQP and the LFLQP is defined below:

$$UFLQP(\mathbf{p}) = FLBP(\mathbf{p}) \tag{16}$$

$$LFLQP(\mathbf{p}) = \sum_{i=0}^{7} 2^i S_{llqp}\{G[\mathbf{N}(\mathbf{C}_t(\mathbf{p}), i)], G[\mathbf{C}_v(\mathbf{p})], r\} \tag{17}$$

Fig. 7 shows the computation of FLQP when $r = 5$. The gray level image and the feature pixel are the same as those in Fig. 4. We use the same pixel $\mathbf{p}$ at (2, 2) and the same values of α_v and α_t as those in Fig. 5 to compute

FLQP. Therefore the computation of **dv**(**p**), TC and VC are the same as the examples in Fig. 5. Fig. 7(a) shows the FLQP computation of the pixel **p** at (2, 2), when $\alpha_v = 0.25$, and $\alpha_t = 0.75$. Because the UFLQP is the same as the FLBP shown in Fig 5(a), only the LFLQP is shown in Fig. 7(a). First the grey level 60 of TC at (5, 5) is replaced by the grey level 30 of VC at (3, 3). For LFLQP, a neighborhood pixel is assigned 1 if it is greater than or equal to 35, or it is less than 30 and greater than or equal to 25, and is assigned 0 otherwise. Then we have LFLQP(2, 2) = 10011100. Fig. 7(b) shows another example of the FLQP computation when $\alpha_v = 0.75$, and $\alpha_t = 0.25$. The UFLQP is the same as FLBP shown in Fig 5(b) and not shown in Fig. 7(b). First the grey level 30 of TC at (3, 3) is replaced by the grey level 60 of VC at (5, 5). For LFLQP, a neighborhood pixel is assigned 1 if it is greater than or equal to 65, or it is less than 60 and greater than or equal to 55, and is assigned 0 otherwise. Then we have the LFLQP(2, 2) = 00110011.

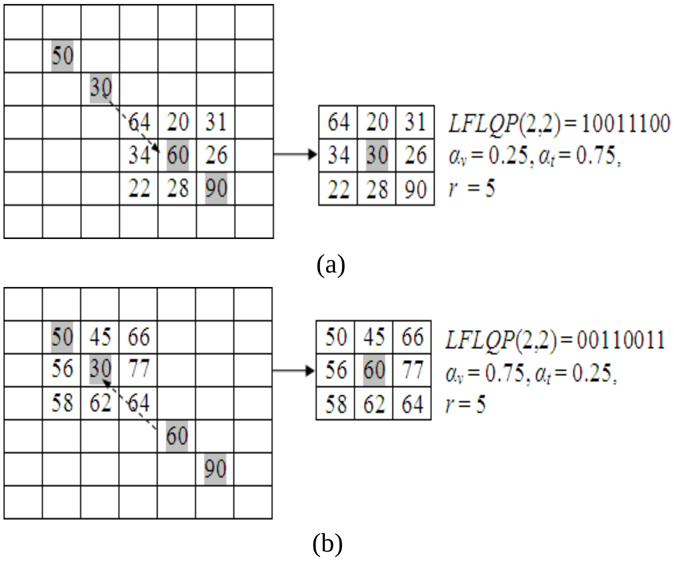


Fig. 7 The computation of FLQP for the pixel at (2, 2). (a) An example when TC ($\alpha_t = 0.75$) is at (5, 5) and VC ($\alpha_v = 0.25$) is at (3, 3) (b) An example when TC ($\alpha_t = 0.25$) is at (3, 3) and VC ($\alpha_v = 0.75$) is at (5, 5)

Fig. 8 shows an example of the FLQP representations when $r = 0.1g_c$. The face image and the binary LRBT feature image are the same as Fig. 6(a) and Fig. 6(b). Fig 8(a) shows the LLQP image. The ULQP image is the same as LBP image in Fig. 6(c). Fig. 8(b) - (e) show LFLQP images when α_t = 0.25, 0.5, 0.75, 1, respectively, and $\alpha_v = 0$. Their corresponding UFLQP are the same as Fig. 6(d) – (g). Fig. 8(f) - (i) show LFLQP images when α_v = 0.25, 0.5, 0.75, 1, respectively, and $\alpha_t = 0$. Their corresponding UFLQP are the same as Fig. 6(h) – (k).

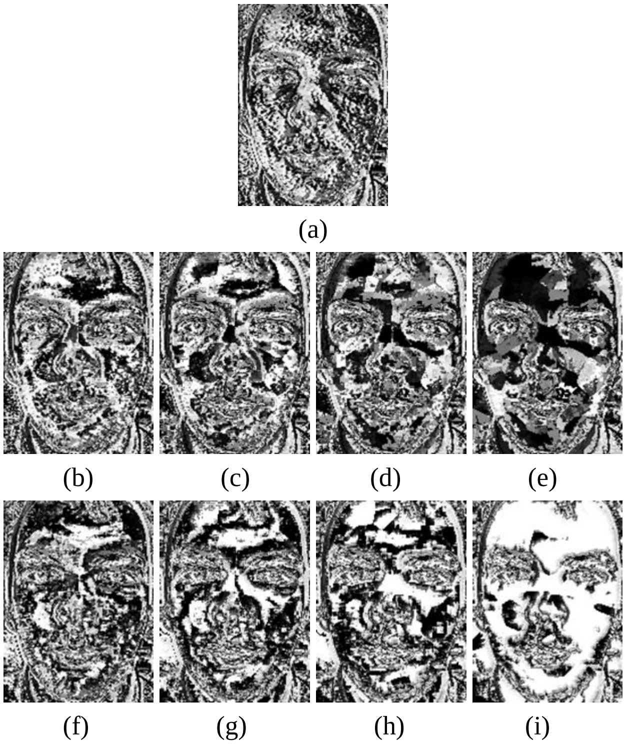


Fig. 8 (a) The LLQP image when $r = 0.1g_c$ (b) - (e) The LFLQP images when $r = 0.1g_c$, α_t = 0.25, 0.5, 0.75, 1, respectively, and $\alpha_v = 0$ (f) - (i) The LFLQP images when $0.1g_c$, α_v = 0.25, 0.5, 0.75, 1, respectively, and $\alpha_t = 0$

5 Experiments

We apply the FLQP methods on eye detection. Fig. 9 shows the system architecture of our FLQP-based eye detection method which is similar to the FLBP method on eye detection introduced in [11]. Fig. 9 consists of three major steps. In first step, a binary image, which contains the feature pixels of the grayscale face image, is derived by applying LRBT feature pixel extraction method. In the second step the FLQP representation of the face image is formed based on the grayscale image and a distance vector field or DVF, which is obtained by computing the distance vector between each pixel and its nearest feature pixel defined in the binary image. The FLQP code is then split to two binary codes, from which two images, UFLQP and LFLQP images are formed. In the finally step, each eye candidate is compared with the eye template based on the UFLQP and LFLQP histograms and similarity measures.

An eye template is constructed from a number of training eye samples. Each eye sample is divided into a grid of $u \times v$ cells. The occurrences of the UFLQP codes in a cell are collected into a UFLQP histogram. The occurrences of the LFLQP codes in a cell are collected into a LFLQP histogram. The eye template is thus defined by *uv* UFLQP mean histograms and *uv* LFLQP mean histograms of the training eye samples. The similarity measure to compare the

UFLQP and LFLQP histograms of an eye template *T* and an eye candidate *C* is defined as follows:

$$M(C,T) = -\sum_{i=1}^{g}\sum_{j=1}^{b}\frac{(C_{i,j}-T_{i,j})^2}{C_{i,j}+T_{i,j}} \tag{18}$$

where $C_{i,j}$ represents the *j*-th bin of the histogram of the *i*-th cell of the eye candidate window, $T_{i,j}$ represents the *j*-th bin of the histogram of the *i*-th cell of the eye template, $g = uv$ is the total number of cells of the u × v grid, and *b* is the number of bins of a histogram. The final similarity measure is the sum of similarity values of the UFLQP and LFLQP histograms. The eye candidate has the largest similarity value with the eye template is the location of the detected eye. We have used a fast algorithm in histogram and similarity measure computation in [11]. The idea of the fast algorithm is to update only two columns or two rows corresponding to the two consecutive eye candidate windows for the histogram and similarity computation instead of repeating the computation for the whole new window. As a result, the fast algorithm significantly improves the computational efficiency of the eye detection method.

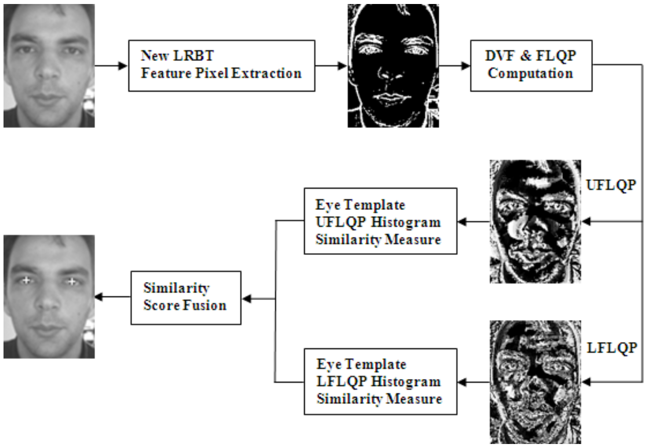


Fig. 8 The system architecture of our FLQP-based eye detection method

We assess the eye detection performance of FLQP and LQP method using the BioID databases which contains 1,521 grayscale frontal face images with spatial resolution of 384 × 286. All facial images in our experiments are cropped and normalized to the size of 132 × 178. To construct the eye template, we collected 70 pairs of eye samples that are not from the BioID database. Eye samples are cropped to 37 × 17. Eye detection performance is determined by a relative distance error and is defined as follows:

$$\gamma = d_1 / d_2 \tag{19}$$

where d_1 is the Euclidean distance between the detected eye center and the ground truth eye center, and d_2 is the intraocular distance between the two ground truth eye centers.

Table 1 compares the performance of the FLQP-based, the LQP-based, the FLTP-based, the LTP-based, the FLBP-based, and the LBP-based eye detection methods. The best experiments are selected from each method to make the comparison. The eye detection success rate when $\gamma \leq 0.25$, 0.1 and 0.05 and the average γ are shown in Table 1. In [11] the experiments show that the 5×5 neighborhood size is better than the 3×3 neighborhood size, and 3 × 4 grid size of eye candidate window yields the best eye detection performance. We apply 5×5 neighborhood size and 3 × 4 grid size in all experiments in Table 1. The experimental results lead to the following findings.

- LQP performs better than LTP and LBP. FLQP perform better than FLTP and FLBP. These results demonstrate that the proposed LQP and FLQP, which encode four relationships of local texture, are more effective than the LTP, FLTP, LBP, and FLBP for texture description and pattern recognition, such as eye detection.
- FLQP achieves the best eye detection performance. Specifically, the average γ of the LQP, FLTP, LTP, FLBP, and LBP-based eye detection methods are 6.1%, 7.5%, 9.7%, 6.9%, and 125.6% higher than the average γ of the FLQP-based eye detection method. The results indicate that FLQP improves upon FLTP, LTP, FLBP, and LBP for eye detection.
- FLQP and FLBP perform better than LQP and LBP methods, respectively. FLTP methods archive better results than LTP methods except LTP obtains higher success rate than FLTP when $\gamma \leq 0.1$. The results illustrate that the feature local methods (FLQP, FLTP, and FLBP), which encode both local and feature information, perform better than the local methods (LQP, LTP, and LBP) which do not encode feature information.
- Our experiments show that LTP methods improve upon the LBP methods. However the FLBP methods archive better results than FLTP except FLTP is better for success rates when $\gamma \leq 0.05$. The FLTP method does not outperform the FLBP method. Our results are consistent with the experimental results reported in [2], [10] which showed that LTP and LBP achieved similar results for face and facial expression recognition, although LTP has a higher computational cost than LBP.

Table 1 The eye detection success rates when $\gamma \leq 0.25$, 0.1 and 0.05 and average γ using the FLQP-based, the LQP-based, the FLTP-based, the LTP-based, the FLBP-based and the LBP-based eye detection methods

Method	$\gamma \leq 0.25$	$\gamma \leq 0.1$	$\gamma \leq 0.05$	Average γ
FLQP, $\beta = 0.2$, $\alpha_v = 0.25$, $\alpha_t = 0$, $r = 0.18g_c$	98.75	96.19	89.71	0.0360
LQP, $r = 0.07g_c$	98.39	95.63	89.38	0.0382
FLTP, β=0.2, $\alpha_v = 0.25$, $\alpha_t = 0$, $r = 3$	98.29	95.17	89.12	0.0387
LTP, $r = 4$	98.03	95.50	88.95	0.0395
FLBP, $\beta = 0.2$, $\alpha_v = 0.25$, $\alpha_t = 0$,	98.65	95.23	87.84	0.0385
LBP	92.34	90.34	83.14	0.0812

6 Conclusions

We present in this paper Local Quaternary Patterns (LQP) and Feature Local Quaternary Patterns (FLQP). The FLQP and LQP which encodes four relationships of the local texture include more information of the local texture than the local binary patterns or LBP and the local ternary patterns or LTP. The FLQP, which encodes both local and feature information, is expected to perform better than the LQP for texture description and pattern recognition. To reduce the feature dimension of LQP and FLQP, a new coding scheme is proposed to split the LQP into two binary codes: the Upper LQP (ULQP) and the Lower LQP (LLQP), and the FLQP into two binary codes: the Upper FLQP (UFLQP) and the Lower FLQP (LFLQP). Experimental results using the BioID database show that (i) LQP and FLQP perform better than LTP, FLTP, LBP, and FLBP for eye detection. (ii) FLQP achieves the best eye detection performance. (iii) FLQP, FLTP, and FLBP perform better than LQP, LTP, and LBP, respectively.

7 References

[1] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[2] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635-1650, 2010.

[3] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and MachineIntelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[4] Z. Liu and C. Liu, "Fusion of color, local spatial and global frequency information for face recognition," *Pattern Recognition*, vol. 43, no. 8, pp. 2882–2890, 2010.

[5] A. Hadid, M. Pietikainen, and T. Ahonen, "A discriminative feature space for detecting and recognizing faces," in *Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, June 27 – July 2, 2004, pp. 797–804.

[6] H. Zhang and D. Zhao, "Spatial histogram features for face detection in color images," in *Proc. Advances in Multimedia Information Processing: 5th Pacific Rim Conference on Multimedia*, Tokyo, Japan, November 30 - December 3, 2004, pp. I: 377-384

[7] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: a comprehensive study," *Image and Vision Computing*, Vol. 27, No 6, pp. 803-816, 2009.

[8] G. Zhao and M. Pietikainen, "Experiments with facial expression recognition using spatiotemporal local binary patterns," in *Proc. Int. Conf. Multimedia and Expo (ICME)*, Beijing, China, July 2-5, 2007, pp. 1091-1094.

[9] S. Moore and R. Bowden, "Local binary patterns for multi-view facial expression recognition," *Computer Vision and Image Understanding*, vol. 115, no. 4, pp. 541–558 , 2011

[10] T. Gritti, C. Shan, V. Jeanne, and R. Braspenning, "Local features based facial expression recognition with face registration errors," in *Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition (FG)*, Amsterdam, The Netherlands, Sept. 17-19, 2008.

[11] J. Gu and C. Liu, "A New Feature Local Binary Patterns (FLBP) Method," in *Proc. 16th International Conference on Image Processing, Computer Vision, and Pattern Recognition*, Las Vegas, Nevada, USA, July 16-19, 2012,

[12] P. Danielson, "Euclidean distance mapping," Computer Graphics and Image Processing, vol. 14, no. 3, pp. 227–248, 1980.