

# MATCHING TEXTURE UNITS FOR FACE RECOGNITION

*Bangpeng YAO, Haizhou AI*

Computer Science and Technology Department,  
Tsinghua University, Beijing 100084, China

*Shihong LAO*

Sensing and Control Technology Laboratory,  
Omron Corporation, Kyoto 619-0283, Japan

## ABSTRACT

For an image, texture unit (TU) is a small complete unit which characterizes the local texture of a given pixel and its neighborhood. Recently, TU-based approaches have been widely used in face recognition. This paper proposes a novel face representation and recognition approach based on TU. We make three major contributions: (1) we introduce a novel TU feature, Local Gabor Quarternary Pattern (LGQP), which incorporates both Gabor magnitude and phase information in a single TU code; (2) similarity measure of two TU images is treated as a tracking problem between two images, and we present a novel point-to-point matching (PPM) approach for TU similarity measure; (3) based on an integral histogram technique, the PPM similarity can be computed very rapidly. Experimental results on CMU-PIE and FERET data sets show that our method is able to reach very promising results.

*Index Terms*— Texture Unit, Local Gabor Quarternary Pattern, Point-to-Point Matching, Face Recognition

## 1. INTRODUCTION

Due to its wide range of commercial and law enforcement applications, face recognition received various attentions during the past decade. Although great advances have been achieved, recognition of face images with large variations of pose, illumination and expression is still a challenging task.

There are mainly two approaches to represent face images. One is the holistic methods such as PCA and LDA. The other is local features, such as Gabor wavelet and *texture unit* (TU) [2]. TU is a small unit which characterizes the local texture of a given pixel and its neighborhood. Recently TU features have drawn increasing attention in face recognition because they can capture small appearance details as well as describing the overall face structure. Methods in this category include Local Binary Pattern (LBP) [1], Local Gabor Texton (LGT) [3], Local Gabor Binary Pattern (LGBP) [11], etc.

TU face similarity is usually measured by histogram operations, which divide the face image into many grid of cells and compute the histogram similarity in each cell. Such methods, although have been widely used in face recognition, inevitably cause some loss of discriminative power, because position within each cell is not coded. In [8], a Distance Trans-

form (DT) method [8] was presented to alleviate this problem. Given two images  $X$  and  $Y$ , DT first generates a DT image for each TU code in  $X$ , and then uses these DT images to match each code in  $Y$ . Pixel deviations between the two images are penalized with Gaussian metric or truncated linear distance. Although DT takes more advantage of spatial information, it is still sensitive to image spatial deviations due to misalignment or pose variations. For example, if  $Y$  is obtained by slightly shifting  $X$ , the DT similarity of  $X$  and  $Y$  will be greatly decreased because of the penalty function.

In this paper, we propose a novel Point-to-Point Matching (PPM) method to measure TU based face similarity. The PPM metric can consider detailed face spatial information and is also robust to image spatial deviations. It is based on the idea that, two TU images can be regarded as two consecutive frames in an image sequence, and thus the approaches in object tracking, especially correspondence based point tracking [9] can be used for TU similarity measure. Moreover, with integral histogram [5] for an intermediate face representation, PPM similarity can be calculated fast.

Furthermore in this paper, we introduce a novel TU code, Local Gabor Quarternary Pattern (LGQP). LGQP combines the discriminative information in both Gabor magnitude and phase in one code, and therefore is more powerful than the other Gabor filter based face representations that separately treat magnitude and phase.

In the remaining part of this paper, we will firstly introduce the LGQP feature and PPM method in Section 2 and 3 respectively, and then give experimental results in Section 4.

## 2. LOCAL GABOR QUARTERNARY PATTERN

Gabor filter is one of the most successful features in face representation. There are mainly two ways to use it. One is to use only Gabor magnitude features [3] or Gabor phase features [10], the other is to concatenate magnitude and phase features, such as the Extended LGBP in [11], which shows that the discriminative information in Gabor magnitude and that in Gabor phase are complementary to each other.

Rather than using the above two approaches that separately treat magnitude and phase features, in this work we introduce a new TU feature, Local Gabor Quarternary Pattern (LGQP), which makes use of both magnitude and phase

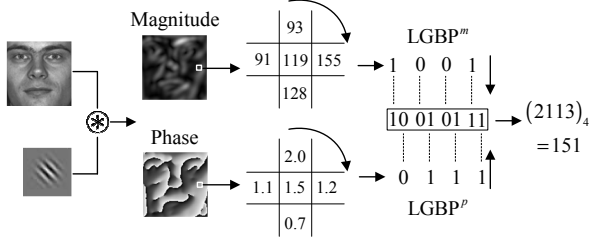


Fig. 1. Example of LGQP feature extraction.

information in one code. For an image point  $I_0$ , we firstly extract LBP codes,  $LGBP^m$  and  $LGBP^p$ , in its associated Gabor magnitude and phase images respectively. Then, the LGQP code is formed by applying bit insertion operation to the  $LGBP^m$  and  $LGBP^p$  codes. More formally, the extraction of LGQP can be described as follows.

Given a neighborhood of  $(P+1)$  values in a Gabor filtered image:  $\{\rho_0 \cdot e^{i\theta_0}, \rho_1 \cdot e^{i\theta_1}, \dots, \rho_P \cdot e^{i\theta_P}\}$ , where  $\rho_0 \cdot e^{i\theta_0}$  is the central value, and  $\{\rho_p \cdot e^{i\theta_p}\}_{p=1}^P$  are its neighborhoods. The LGQP code is defined by

$$LGQP = \sum_{p=1}^P s_p 4^{p-1} \quad (1)$$

where

$$s_p = \begin{cases} 0, & \text{if } \rho_p \geq \rho_0 \ \& \ \theta_p \geq \theta_0 \\ 1, & \text{if } \rho_p \geq \rho_0 \ \& \ \theta_p < \theta_0 \\ 2, & \text{if } \rho_p < \rho_0 \ \& \ \theta_p \geq \theta_0 \\ 3, & \text{if } \rho_p < \rho_0 \ \& \ \theta_p < \theta_0 \end{cases}$$

From Equation 1 we can see that LGQP has  $4^P$  possible code values. This is a large number and thus it will require more space and time to store and match LGQP features, compared with LGBP [11] in the same setting. Recall that in LGBP, the neighborhood sampling number  $P$  is usually set to 8, which generates  $2^8$  different code values. In order to reduce the computation cost for LGQP representation and matching, here we set  $P = 4$ . Thus, LGQP will generate the same number of code values with that in LGBP [11]. Although here we use a smaller  $P$  value, in experiment part we show that the LGQP still performs better than LGBP for face recognition.

An example of LGQP extraction is shown in Figure 1.

### 3. POINT-TO-POINT MATCHING MEASURE

#### 3.1. Point to Point Matching Approach

Two TU images  $X$  and  $Y$  can be regarded as two consecutive frames in an image sequence, and thus the approaches in object tracking, especially correspondence based point tracking [9], can be used to measure TU image similarity. Inspired by this idea, we propose a Point-to-Point Matching (PPM) metric, which can take advantage of image spatial information in pixel level, and is not sensitive to spatial deviations. PPM calculates the similarity of  $X$  and  $Y$  by tracking interest points from  $X$  to  $Y$  (or from  $Y$  to  $X$ ) in a three-step procedure:

- Detect *interest points* in  $X$ .
- For each interest point in  $X$ , find its *target point* in  $Y$  and get their *correspondence score*.
- Combine the correspondence scores to get the similarity between  $X$  and  $Y$ .

In our approach, a TU code  $x$  in  $X$  is an interest point iff one or more codes at a nearby position in  $Y$  take the same value to  $x$ . Therefore, for each point at position  $(i, j)$  in  $X$ , we can determine whether it is an interest point by searching a  $n \times n$  neighborhood centered at  $(i, j)$  in  $Y$ , which is denoted as  $\Omega_{(i,j)}$ . In this work, where the normalized image size is  $120 \times 120$ , we empirically set  $n = 11$ .

For each interest point  $X(i, j)$ , we should find its associated target point  $Y(i + \Delta i, j + \Delta j)$  in  $Y$  and get the correspondence score. If in  $\Omega_{(i,j)}$  there is only one *matching point* takes the same code value to  $X(i, j)$ , then this point is the target point. If there are many matching points in  $\Omega_{(i,j)}$ , we select the one that minimizes the local first-order constraint error in a small spatial neighborhood as the target point. The constraint we use here is very similar to that in the optical-flow tracking. For two points  $X(i, j)$  and  $Y(i', j')$ , their local first-order constraint error is

$$\varepsilon(X(i, j), Y(i', j')) = \sum_{\delta i, \delta j} [X(i + \delta i, j + \delta j) \ominus Y(i' + \delta i, j' + \delta j)]^2 \quad (2)$$

where “ $\ominus$ ” indicates the distance of two code values.

We use Hamming distance to calculate two TU codes’ distance. For two codes  $P = (b_{p1} \dots b_{pL})$  and  $Q = (b_{q1} \dots b_{qL})$  in binary version, their hamming distance is

$$P \ominus Q = \sum_{l=1}^L |b_{pl} - b_{ql}| \quad (3)$$

where  $L$  is the number of bits. In this work because the number of possible TU code values is limited, we calculate the distance of each two code values in advance and store all the distances in a *distance table*. In this table, the value in the  $p$ -th row,  $q$ -th column is the distance between two code values  $p$  and  $q$ . Thus, distance between any two codes can be obtained immediately through querying the distance table.

The correspondence score for each interest point  $X(i, j)$  is also calculated based on Equation 2, which is

$$s_{X(i,j)} = \begin{cases} 1, & \text{if } \varepsilon(X(i, j), Y(i + \Delta i, j + \Delta j)) < T, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

where  $T$  is a user-defined threshold.

The similarity of  $X$  and  $Y$  can be obtained by adding the correspondence scores of all interest points in  $X$ . Yet it might happen that many points in  $X$  share a same target point in  $Y$ . In order to alleviate this problem, a regularization factor  $\gamma(Y)$  is introduced.  $\gamma(Y)$  is obtained by dividing the number of

target points in  $Y$  by the total number of pixels in  $Y$ . Finally the similarity of  $X$  and  $Y$  is

$$S(X, Y) = \gamma(Y) \sum_{i,j} s(X(i, j)) \quad (5)$$

One thing to mention is that, in the PPM measure,  $S(X, Y) \neq S(Y, X)$ . However we find that, in practical applications, given two images  $X$  and  $Y$ , using  $S(X, Y)$  and  $S(Y, X)$  can both lead to similar good results.

### 3.2. Weighted Correspondence Score for PPM

In Equation 4, all the interest points that successfully find their target points take the same correspondence score 1. However it has been shown that different face regions make different contributions for recognition [1]. It is reasonable that interest points in different areas have different correspondence scores. Therefore, we divide a TU image into many small regions. Correspondence scores might be variant from one region to another. As in [1] where weights are allocated to different regions, the resulted correspondence score here is called “weighted correspondence score”. For the  $r$ -th region, its weighted correspondence score  $s_r$  is defined as follows.

Given a set of training images, we use PPM to calculate the similarity in region  $r$  of each two images. (Set correspondence scores according to Equation 4). Mean and variance for intra- and extra-personal similarities in this region are denoted as  $m_{r,intra}$ ,  $\sigma_{r,intra}^2$ ,  $m_{r,extra}$ , and  $\sigma_{r,extra}^2$  respectively. The weighted correspondence score for region  $r$  is

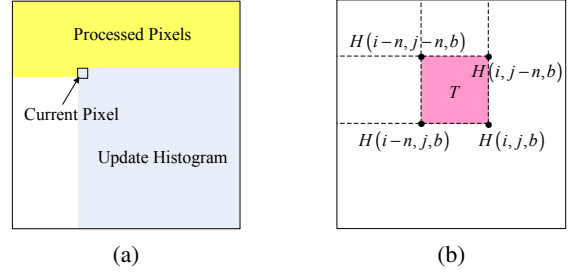
$$s_r = \frac{(m_{r,intra} - m_{r,extra})^2}{\sigma_{r,intra}^2 + \sigma_{r,extra}^2} \quad (6)$$

### 3.3. Fast Interest Points Detection

Interest points can be detected very rapidly using an intermediate image representation called integral histogram [5]. Interest points detection is the most time-consuming step in PPM measure. This is because in the brute-forth approach, we should go through every point in  $\Omega_{(i,j)}$  to determine whether  $X(i, j)$  is an interest point. Thus in our experiment setting, each pixel in  $Y$  should be accessed 121 times in the interest point detection step, which is time-consuming.

The integral histogram  $H(i, j, b)$  contains the sum of code values equal to  $b$  that appear above and to the left of  $(i, j)$ . We use wavefront scan to obtain the integral histogram. In propagation, it requires updating the histogram for such points that their left, upper and upper-left neighbors are already scanned, as shown in Figure 2(a). Using integral histogram, whether a region contains a code value can be computed with only three arithmetic operations, rather than going through every point in the region. Figure 2(b) shows the computation of the number of code values that are equal to  $b$  in region  $T$ .

The integral histogram based detection approach cannot return the accurate positions of matching points, and thus we



**Fig. 2.** (a) Propagation of integral histograms. (b) The number of code values that equal to  $b$  in  $T$ , a  $n \times n$  region:  $H(T, b) = H(i, j, b) - H(i - n, j, b) - H(i, j - n, b) + H(i - n, j - n, b)$ .

still need to apply brute-forth search to the detected interest points so that the positions of their matching points can be obtained. Yet we argue that this limitation is not serious in the face matching problem, because we find that interest points only take up a small proportion (about 15%). Moreover, for a recognition task which contains one probe image and a large number of gallery images, we only need to compute the integral histogram for the probe image, and then perform matching from each gallery image to the probe image to get the PPM similarity. Therefore with the integral histogram representation, PPM can be computed very rapidly.

## 4. EXPERIMENTS

Experiments are carried on CMU-PIE [6] and FERET [4] data sets. All the face images are normalized to  $120 \times 120$  pixels and rectified according to eye-centers. Gabor filters used here are with 5 scales and 8 orientations, as that in [11].

### 4.1. Results on CMU-PIE dataset

There contains 40,000+ face images of 68 persons in PIE database. We randomly select 40 images for each person that cover variations of pose ( $\pm 30^\circ$  in both upper-down and left-right), illumination and expression. For each person, we select one image with the most neutral expression, lighting and pose for gallery and the remaining images are used as probes.

We first compare the proposed LGQP feature with some other TUs, including LBP [1], LGBP [11] and LGT [3]. The results are shown in Table 1. Results for each feature are obtain by using  $\chi^2$  similarity measure, and we do not allo-



**Fig. 3.** Example of PIE face images used in our experiment.

**Table 1.** Comparison of LBP, LGBP, LGT, and LGQP. In the bracket is the number of cells in each method.

Method	Result
LBP (144)	80%
LGT (11520)	89%
LGBP (11520)	92%
LGQP (5760)	94%

**Table 2.** Comparison of  $\chi^2$ , DT, and PPM similarity measure.

Method	Result
LBP+ $\chi^2$	80%
LBP+DT	81%
LBP+PPM	85%
LGQP+ $\chi^2$	94%
LGQP+DT	95%
LGQP+PPM	97%

cate weights to different regions. As suggested in [3], the cell size for histogram estimation is set to  $10 \times 10$ . Table 1 shows that LGQP yields the best recognition accuracy. Moreover, comparing with other two Gabor filtering based features, LGT and LGBP, which separately treat Gabor magnitude and phase, LGQP uses only a half number of cells. Therefore the computation time for LGQP is also reduced. (As mentioned in Section 2, we set  $P = 4$  for LGQP, and therefore the histogram length in LGQP is the same to that in LGBP.)

We then compare our PPM similarity measure with Chi-square ( $\chi^2$ ) distance [1] and Distance Transform (DT) [8]. The parameters of DT are set as in [8]. Experimental results are shown in Table 2. Due to space limitation we only list the results using LBP and LGQP representation. The results show that, while DT only slightly performs better than  $\chi^2$ , the PPM method significantly improves recognition accuracy for both LBP and LGQP feature. Considering this data set covers large variations in pose, illumination and expression, we can see that PPM works better than  $\chi^2$  and DT in these situations.

#### 4.2. Results on FERET dataset

Here we use the standard FERET test protocol [4] to test the performance of our method: using LGQP for face representation and PPM for similarity measure. FERET *fa* is used as gallery and *fb*, *fc*, *dupI* and *dupII* as the probes. The training set contains 1002 images of 429 persons, which are used to compute the mean and variance in Equation 6.

We compare our result with the best result in [4], and some recently published results [11, 7, 3]. The results show that, our method significantly improves recognition accuracy on the *fc*, *dupI* and *dupII* data sets, which contain large age and illumination variations and are much more challenging to recognize than *fb*. Compared with [7], in which the best results on FERET database are reported to our best knowledge, our method achieves comparable accuracy on *fb* and *dupI*, and improves the recognition accuracy on *fc* and *dupII*.

### 5. CONCLUSION

In this paper, we introduced a new TU based face representation, LGQP, and proposed a novel PPM method to measure TU similarity. The PPM similarity measure can be obtained

**Table 3.** Rank-1 accuracy on FERET database. “(LGFP+PPM)<sub>w</sub>” uses the method in Section 3.3 to allocate different tracking scores to different face regions.

Method	<i>fb</i>	<i>fc</i>	<i>dupI</i>	<i>dupII</i>	
Result in [4]	96%	82%	59%	52%	
Result in ICPR06 [11]	99%	95%	79%	75%	
Result in FGR06 [7]	99%	97%	87%	79%	
Result in ICB07 [3]	97%	90%	71%	67%	
Our Method	LGFP+PPM	99%	98%	82%	81%
	(LGFP+PPM) <sub>w</sub>	99%	98%	87%	85%

very rapidly with integral histograms. Experimental results on CMU-PIE and FERET database showed that our method significantly improves recognition accuracy. Our future work will focus on reducing the feature length generated by LGQP by using feature selection techniques.

### 6. ACKNOWLEDGEMENTS

This work is supported in part by National Science Foundation of China under grant No.60673107, and it is also supported by a grant from Omron Corporation.

### 7. REFERENCES

- [1] T. Ahonen, A. Hadid, and M. Pietikäinen. “Face Recognition with Local Binary Patterns”. In *ECCV 2004*, pages 469–481.
- [2] D. He and W. Li. “Texture Unit, Texture Spectrum, and Texture Analysis”. *IEEE T. Pattern Anal.*, 28:590–512, 1990.
- [3] Z. Lei, S. Li, R. Chu, and X. Zhu. “Face Recognition with Local Gabor Textons”. In *ICB 2007*, pages 49–57.
- [4] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. “The FERET Evaluation Methodology for Face Recognition Algorithms”. *IEEE T. Pattern Anal.*, 22(10):1090–1104, 2000.
- [5] F. Porikli. “Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces”. In *CVPR 2005*, pages 829–836.
- [6] T. Sim, S. Baker, and M. Bsat. “The CMU Pose, Illumination, and Expression Database”. *IEEE T. Pattern Anal.*, 25(12):1615–1618, 2003.
- [7] Y. Su, S. Shan, X. Chen, and W. Gao. “Hierarchical Ensemble of Gabor Fisher Classifier for Face Recognition”. In *FGR 2006*, pages 91–96.
- [8] X. Tan and B. Triggs. “Enhanced Local Texture Feature Sets for Face Recognition under Different Lighting Conditions”. In *AMFG 2007*, pages 469–481.
- [9] A. Yilmaz, O. Javed, and M. Shah. “Object Tracking: A Survey”. *ACM Comput. Surv.*, 38(4), 2006.
- [10] B. Zhang, S. Shan, X. Chen, and W. Gao. “Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition”. *IEEE T. Image Process.*, 16(1):57–68, 2007.
- [11] W. Zhang, S. Shan, X. Chen, and W. Gao. “Are Gabor Phases Really Useless for Face Recognition?”. In *ICPR 2006 Supplemental*, pages 39–42.