2012 11th International Conference on Machine Learning and Applications

Adaptive Extended Local Ternary Pattern (AELTP) for Recognizing Avatar Faces

Abdallah A. Mohamed[1, 2], Roman V. Yampolskiy[1]
[1]Computer Engineering and Computer Science
University of Louisville, Louisville, KY, 40292, USA
[2]Department of Mathematics, Menoufia University, Shebin El-Koom, Menoufia Egypt
Email: {aamoha04, rvyapm01}@louisville.edu

***Abstract*—Many face recognition techniques have been developed during the past decades but the problem remains challenging, especially recognizing non-biological entities or avatars. Local Binary Pattern (LBP) method is one of these techniques which has shown its superiority in recognizing faces. The original LBP operator mainly thresholds pixels in a specific predetermined window based on the gray value of the central pixel of that window. As a result the LBP operator becomes more sensitive to noise especially in near-uniform or flat area regions of an image. To deal with this problem a generalization of the LBP descriptor, Local Ternary Patterns (LTP), came to the presence. In this paper we introduce a new local adapted texture features for efficient avatar face recognition based on the original LTP operator. The proposed technique, Adaptive Extended Local Ternary Pattern (AELTP), shares with the original LTP descriptor being less sensitive to noise. However AELTP is better as it determines the local pattern threshold automatically based on local statistics. Experiments conducted on two virtual world avatar face image datasets show that our technique performs better than original LBP, original LTP and Extended LTP (ELTP) in terms of accuracy.**

Index Terms*— *Avatar, face recognition, LBP, LTP, ELTP, AELTP.

I. INTRODUCTION

Face recognition is one of the biometric traits that received a great attention from many researchers during the past few decades because of its potential applications in a variety of civil and government-regulated domains. It usually involves: initial image normalization, preparing an image for feature extraction by detecting the face in that image, extracting facial features from appearance or facial geometry, and finally classifying facial images based on extracted features.

Face recognition however is not only concerned with recognizing human faces, but also with recognizing faces of non-biological entities or avatars. To address the need for a decentralized, affordable, automatic, fast, secure, reliable, and accurate means of identity authentication for avatars, the concept of Artimetrics has emerged [1, 2]. Artimetrics is a new area of study concerned with visual and behavioral recognition and identity verification of intelligent software agents, domestic and industrial robots, virtual world avatars and other non-biological entities [1, 2]. People often complain about the insufficient security system in popular virtual worlds such as Second Life which motivates our research on security in virtual worlds [1, 3]. Extracting discriminant information from a facial image is one of the key components for any face recognition system [4]. There are many different algorithms proposed in the past to extract features, such as principal component analysis (PCA) [5], linear discriminant analysis (LDA) [6].

Local binary pattern (LBP) operator has proven itself as a powerful texture descriptor providing excellent results in terms of accuracy in many applications such as, motion detection, image retrieval, remote sensing and biomedical image analysis. Among all these applications, LBP method has shown its superiority in recognizing faces [4]. LBP is one of the most popular local feature-based methods. It was first proposed by Ojala et al. [7] as a powerful method for describing textures and it was applied to face recognition for the first time by Ahonen et al. [8]. But the original LBP method worked as a local descriptor to capture only local information [9]. It thresholds all pixels in the neighborhood based on the gray value of the central pixel. As a result the original LBP becomes more sensitive to noise especially in near uniform or flat areas [10]. One solution to this problem is to extract the features based on 3-valued texture operators as in [11].

Most of the work done so far on face recognition has focused on humans. Research on recognizing virtual world avatars has not received due attention yet. Some methods further developed LBP for either recognizing human faces or avatar faces. For example, Yang et al. [12] applied LBP for face recognition with Hamming distance constraint. Chen et al. [13] used Statistical LBP for face recognition. Mohamed et al. [1] applied hierarchical multi-scale LBP with wavelet transform to recognize avatar faces. Mohamed et al. [14] applied discrete wavelet transform with adapted local binary patterns with direction statistical features to recognize avatar faces.

In this paper, we propose a novel LTP face recognition technique to recognize avatar faces from different virtual worlds. In this approach, we define a new operator from the original LTP operator, Adaptive Extended Local Ternary Pattern (AELTP). In our proposed technique, AELTP operator computes a weight to each pixel in the neighborhood of a central pixel and provides a new value for each pixel in that neighborhood. Based on the pixels new values and the standard deviation of all pixels in an image patch, AELTP will provide a new binary code for the central pixel. The efficacy of our proposed method is demonstrated by the experiments on two different avatar datasets from Second Life and Entropia Universe virtual worlds.

The rest of this paper is organized as follows; Section 2 briefly provides an introduction to local binary pattern. In Section 3, an overview of the LTP is presented. Section 4

978-0-7695-4913-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ICMLA.2012.19

presents the extended local ternary (ELTP) operator. The adaptive extended local ternary pattern (AELTP) operator is descripted in section 5. Section 6 reports experimental results which are followed by conclusions in Section 7.

II. LOCAL BINARY PATETRN

A. LBP Operator

LBP operator, proposed by Ojala et al. [7], is a very simple and efficient local descriptor for describing textures. It labels the pixels of an image by thresholding the pixels in a certain neighborhood of each pixel with its center value, multiplied by powers of two and then added together to form the new value (label) for the center pixel [15]. The output value of the LBP operator for a block of 3x3 pixels can be defined as follows [15]:

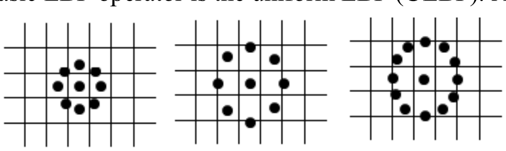
$$LBP_{P,R} = \sum_{p=0}^{7} 2^p S(g_p - g_c) \tag{1}$$

where g_c corresponds to the gray value of the central pixel, g_p (p = 0,1,2,..,7) are the gray values of its surrounding 8 pixels and $S(g_p - g_c)$ can be defined as follows:

$$S(g_p - g_c) = \begin{cases} 1, & g_p \geq g_c \\ 0, & otherwise \end{cases} \tag{2}$$

Later new versions of LBP operator have emerged as an extension to the original one. These versions used neighborhoods of different sizes to be able to deal with large scale structures that may be the representative features of some types of textures [9, 16]. (Fig. 1 gives examples of different LBP operators where P is the neighborhood size and R is its radius).

One of the most important and successful extensions to the basic LBP operator is the uniform LBP (ULBP). An LBP



(P=8, R=1) (P= 8, R=1.5) (P=12, R=1.5)

Figure 1. Three different LBP operators[8, 9]

is called uniform if the binary pattern contains at most two different transitions from 0 to 1 or 1 to 0 when the binary string is viewed as a circular bit string [9]. For example, 11000011, 00111110 and 10000011 are uniform patterns [8].

B. LBP Histogram

Suppose the given image is of size N x M. To represent the whole texture image after computing the LBP pattern value for each pixel in that image, a histogram is built using [16]:

$$H_i = \sum_{x,y} I(LBP(x,y) = i), i = 0,1,.., n-1 \tag{3}$$

where $n=2^P$ is the number of different labels produced by the LBP operator, and I (A) is a decision function with value 1 if the event A is true and 0 otherwise.

III. LOCAL TERNARY PATTERNS (LTP)

Local binary pattern is a 2-valued (binary) code that is successfully used in many applications .The LBP operator idea is based on just two bit values either 1 or 0. This basis does not allow the LBP operator to discriminate between multiple patterns. The LBP operator has two main points of weakness:

- The LBP operator cannot distinguish between two pixel values if the first one is near the central pixel but a little bit below that pixel and the second undistinguishable one is far below the center pixel value [17].
- In flat image areas, such as in face images, where all pixels nearly have the same gray value, if a slight amount of noise were added to these areas the LBP operator will give some bits the value 0 and others the value 1. So the LBP feature will be unstable and thus the LBP operator will not be suitable for analyzing these areas [17].

To solve these problems a new 3-valued texture operator, Local Ternary Patterns (LTP), that can be considered as an extension to LBP was introduced recently [11]. Instead of a thresholding that is based only on the central pixel value of the neighborhood, the user will define a threshold say t and any pixel value within the interval of $-t$ and $+t$, thus assigns the value 0 to that pixel, while the user assigns the value 1 to that pixel if it is above this threshold and a value -1 if it is below it when compared to the central pixel value. The following equation shows how to compute the LTP operator [11]:

$$LTP(i) = \begin{cases} 1 & if\ p_i - p_c \geq t \\ 0 & if\ |p_i - p_c| < t \\ -1 & if\ p_i - p_c \leq -t \end{cases} \tag{4}$$

where t is a user specified threshold, p_i is a pixel value in the neighborhood and p_c is the central pixel value.

This definition leads to obtaining a texture operator that is less sensitive to noise (since it is no longer mainly based on the value of the central pixel) but no longer strictly invariant to gray-level transformations. Fig. 2 shows an example of how the LTP operator works by using a threshold value t = 5:

12	34	45
38	35	55
11	65	23

(a)

-1	0	1
0		1
-1	1	-1

(b)

Figure 2. LTP computation

To get rid of the negative values in Fig. 2, the LTP values are divided into two LBP channels, the upper LTP (LTPU) and the lower LTP (LTPL) as in Fig. 3. The LTPU is obtained by replacing the negative values in the original LTP by zeros. The LTPL is obtained in two steps: first, we replaced all the value of 1's in the original LTP to be zeros then we changed the negative values to be 1's [10].

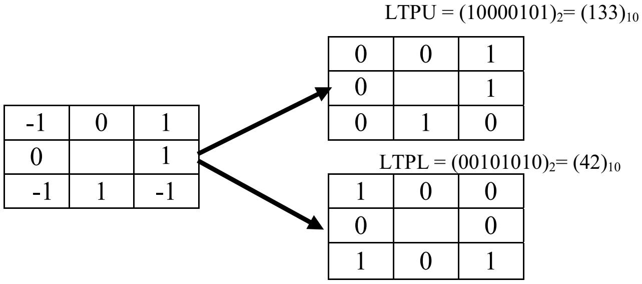


Figure 3. Splitting LTP into two LBP channels [11]

IV. EXTENDED LOCAL TERNARY PATTERNS (ELTP)

Liao figured out that when he applied the original LTP in his experiments the result became worse than the original LBP in the presence of noise [18]. He proposed a new definition to the LTP in [18]. Actually his definition (Extended Local ternary Pattern) is the same as the original one but he did not apply a fixed threshold. He converted an image regions to its ELTP representation based on a threshold employs the local statistics of the neighborhood of a central pixel [18].

$$ELTP\ (i) = \begin{cases} 1 & if\ p_i - p_c \geq (\alpha * \sigma) \\ 0 & if\ |p_i - p_c| < (\alpha * \sigma) \\ -1 & if\ p_i - p_c \leq -(\alpha * \sigma) \end{cases} \quad (5)$$

where σ is the standard deviation of the region around the central pixel , α is a scaling factor and $0 < \alpha \leq 1$.

This LTP representation is the same as the original one except for the definition. In the original one the threshold value t is fixed while in the ELTP t is not fixed but its value is based on the local statistics of the region around the central pixel p_c.

Let us consider that the value of α is 0.5, Fig. 4 gives an explanation and comparison between how LTP and ELTP works. The result of applying the base 3 system approach is that the feature dimension size increases drastically to be 3^P where P is the number of the sample points in the neighborhood [18].

To reduce the size of the feature dimension Liao [18] computed the similarity between any two ELTP strings. This will transfer the dimension reduction problem to be a graph partitioning problem which can be solved by using spectral clustering.

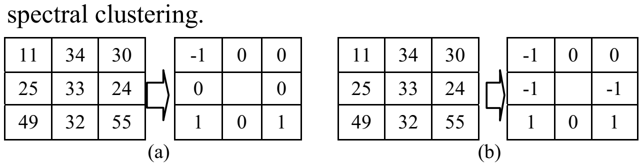


Figure 4. Comparison between LTP and ELTP:
a) Using LTP with fixed threshold t = 5.
b) Using ELTP to represent a region of an image with α = 0.5.

V. ADAPTIVE EXTENDED LOCAL TERNARY PATTERN (AELTP)

Using the LTP operator allows to overcome some of the weaknesses found in applying the LBP operator. Instead of matching based on 2-valued codes in LBP now we can work with 3-valued codes in LTP which increases the available number of patterns. Importantly in order to obtain good result after using the LTP operator we have to be very careful in choosing the system threshold. Obviously, for face recognition systems to gain a good results, there is no fixed threshold and the best performing threshold has to vary depending on the facial dataset. So the ideal solution is to find ways that compute the threshold automatically based on the available facial dataset [10]. To this end, we defined a weight for each pixel in a neighborhood around a central one. We used these weights to compute a new value for each pixel in this neighborhood. Local statistics of all pixels in that neighborhood will be used to compute the new threshold. This threshold value changes automatically from one patch of pixels to another during the whole image based on pixels' values in theses patches. We used this new threshold in the definition of new LTP operator which we called Adaptive Extended Local Ternary Pattern (AELTP).

A. Computing the Local Weight for Each Pixel

To compute the local weight for each pixel in any local patch of pixels we have to define the following equation:

$$J = (g_c(i,j) - \sum_{q=1}^{P} w_q g_q(i,j))^2 \quad (6)$$

where g_c corresponds to the central pixel, g_q to the surrounding pixels, w_q is the weight for any pixel q and $W = [w_1, w_2, w_3, \ldots, w_P]$ and $\sum_{q=1}^{P} w_q = 1$

This equation minimizes the overall differences between the central pixel in any neighborhood and all pixels in that neighborhood based on the size of that neighborhood. By deriving both sides of equation 6 with respect to w_p we get:

$$\frac{\partial J}{\partial w_p} = -2g_p(i,j)(g_c(i,j) - \sum_{q=1}^{P} w_q g_q(i,j)) = 0$$

then:

$$g_c(i,j) = \sum_{\substack{q=1 \\ q \neq p}}^{P} w_q g_q(i,j) + w_p g_p(i,j) \quad (7)$$

From equation 7 we can obtain the value of w_p using the following equation:

$$w_p = \frac{g_c(i,j) - \sum_{\substack{q=1 \\ q \neq p}}^{P} w_q g_q(i,j)}{g_p(i,j)} \quad (8)$$

The following are the steps we followed to compute w_p for different pixels:

1- Initialization $w_p = 1/P$ for every $p = 1,2,\ldots,P$
2- Use the updated equation 8
  Repeat
    For p = 1 : P
      Update w_p with the new value. Each time

when we compute the new value of w_p for any pixel use this value to compute the w_p for the next pixels.

end

B. AELTP Operator

By applying these steps we will have the weights for all pixels in the neighborhood. At this point we can define the AELTP operator as following:

The following equation shows the proposed AELTP operator:

$$AELTP\ (i) = \begin{cases} 1 & if\ p_i \geq (m + k\sigma) \\ 0 & if\ (m - k\sigma) < p_i < (m + k\sigma) \\ -1 & if\ \ p_i \leq (m - k\sigma) \end{cases} \tag{9}$$

where σ refers to the standard deviation of the old pixels values in the neighborhood, m refers to the median of the new values of pixels and k is constant.

We can summarize the steps for applying the AELTP operator on one facial image in the following steps:

1- Divide the facial image into non-overlapping sub-regions J_0, J1,…J_{t-1}, where J_0 is the first sub-region and t is the number of non-overlapping sub-regions.

2- Decide what the radius is and what is number of pixels in the neighborhood, say R and P.

3- Starting from the first sub-region, map each sub-region to its sub-region of AELTP codes (based on equation 9). Each AELTP code is divided into two separate LBP patterns one for the positive part of AELTP and the second is for the negative part of AELTP. The resulting patterns are concatenated. Using AELTP operator increases the dimension of the features dramatically, as in ELTP, so we used PCA and then LDA to reduce the dimensions. (See Fig. 5 as an example of applying AELTP operator).

4- Finally, for classifying we used the chi-square distance.

11	34	30
25	33	24
49	32	55

(a)

1.375	4.25	4.25
3.125		3
6.125	4	6.875

(b)

-1	0	0
-1		-1
1	0	1

(c)

-1	0	0
-1		-1
1	0	1

AELTPU = $(00001010)_2 = (10)_{10}$

0	0	0
0		0
1	0	1

AELTPL = $(00110001)_2 = (49)_{10}$

1	0	0
1		1
0	0	0

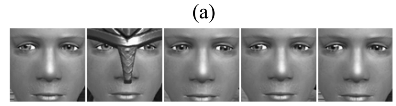
(d)

Figure 5. AELTP computation: a) The original image window b) the result after applying equation 8 c) The result of applying equation 9 d) Splitting the result of AELTP into two LBP channels

VI. EXPERIMENTS

In this section, we verify the performance of the proposed algorithm on two different avatar datasets: the first is Second Life (SL) dataset and the second is Entropia Universe (ENT) dataset (Fig. 6 shows an example of a subject from each dataset). The proposed method is compared with other single scale techniques such as, the original LBP, LTP and ELTP (See Fig. 7 which indicates how robust is our technique when compared to others).

(a)



(b)

Figure 6. Samples of one subject of facial images from:
a) Second Life dataset b) Entropia dataset

A. Experimental Setup

To evaluate our proposed technique, we have used two facial image datasets (we added Gaussian noise with default parameters to both datasets).

The first dataset was acquired from a large collection of SL virtual world avatar face dataset [19]. This dataset contains 847 gray scale images with size 1280 x 1024 pixels each to represent 121 different avatars. Each avatar subject has 7 different images for the same avatar with different frontal pose angles (front, far left, mid left, far right, mid right, top and bottom) and facial expressions.

The second dataset was collected from ENT virtual world [20]. ENT dataset contains 545 gray scale images with size 407 x 549 pixels. These images were organized in 109 subjects (avatars). Each subject has 5 different images for the same avatar with different frontal angle and facial details (wearing a mask or not).

The facial part of each image in SL and ENT datasets was manually cropped from the original images based on the location of the two eyes, mouth and the nose. The new size of each facial image in SL dataset is 260 x 260 pixels while in ENT dataset each facial image was resized to the size of 180 x 180 pixels.

Each one of the two datasets was split into two independent sets. One set is used for training, and the second set is used for testing. During our experiments, we used different sizes of the training and the testing sets. All training images were randomly chosen while the rest of images in both datasets were used for testing.

B. Experimental Results

To gain better understanding on whether our new definition of the LTP operator is advantageous or not we compared AELTP with the original LBP, LTP and ELTP in several experiments. First we got the performance of AELTP with different block size (non-overlapping sub-regions) with R =1, 2, 3 and P = 8, 16, 24 as we can see in Fig. 8. Changing the block size affects the recognition

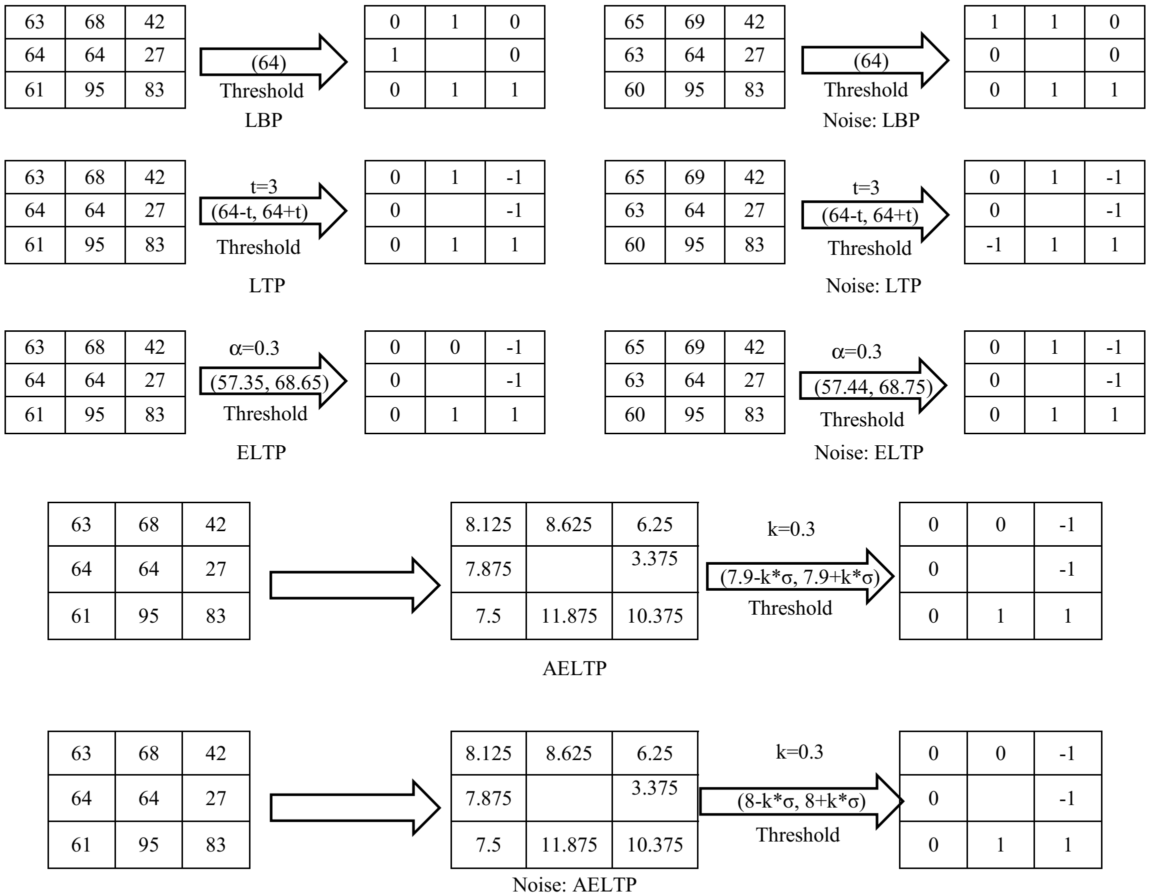


Figure 7. Comparison of LBP, LTP, ELTP and AELTP operators.

rate. Dividing the facial image into a large number of small sub-regions increases the computation time and may reduce the system accuracy while dividing the facial image into a smaller number of large sub-regions increases the loss of spatial information. In our experiments each facial image has been divided into q x q non-overlapping rectangle size sub-regions while the best value of q is obtained by practicing and it differs from one dataset to another.

As a result, we compared the performance of AELTP with that of the original LBP, LTP and ELTP using $q = 7$ for the SL dataset and $q = 5$ for the ENT dataset. The average of the recognition rate with different LBP operators and with h randomly selected images from each class for all techniques can be shown in figures 9 and 10 and the detailed numbers are shown in tables I and II.

The experimental results show that the performance of AELTP technique is better than the performance of all other compared techniques. For the SL dataset, AELTP achieves about 87% accuracy rate with 3 training samples from each class while its closer competitor (ELTP) achieves about 85%. This percentage goes up to more than 96% when the

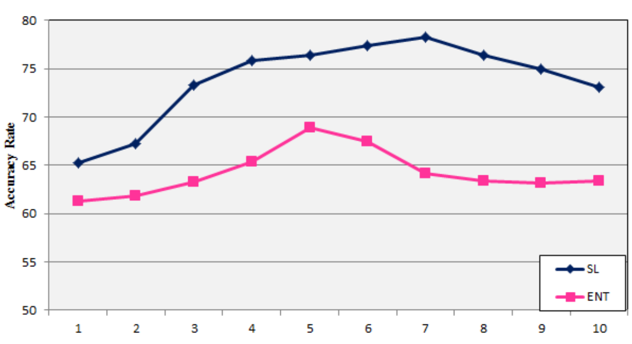


Figure 8. Performance of AELTP with different block size.

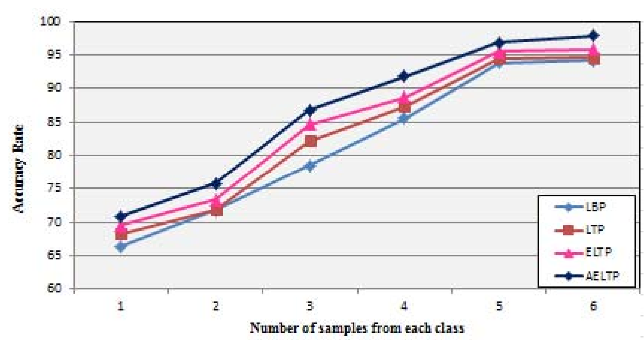


Figure 9. Performance of different techniques with SL dataset.

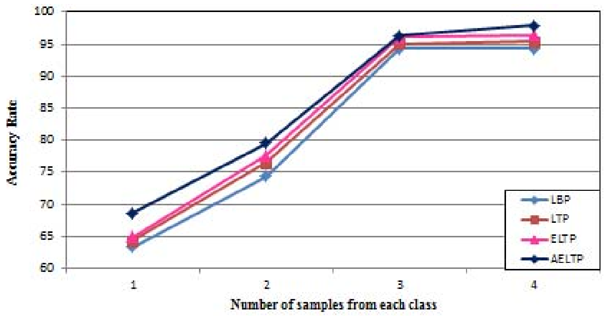


Figure 10. Performance of different techniques with ENT dataset

Table I. Average recognition rate for different techniques with SL dataset

Technique	Number of samples					
	1	2	3	4	5	6
LBP	66.39	71.78	78.45	85.49	93.89	**94.12**
LTP	68.28	71.9	82.23	87.34	94.49	**94.59**
ELTP	69.56	73.29	84.67	88.68	95.48	**95.89**
AELTP	70.89	75.78	86.78	91.78	96.89	**97.89**

Table II. Average recognition rate for different techniques with ENT dataset

Technique	Number of samples			
	1	2	3	4
LBP	63.34	74.12	**94.29**	94.06
LTP	64.39	76.45	95.12	**95.45**
ELTP	64.89	77.59	96.06	**96.49**
AELTP	68.59	79.58	96.35	**97.38**

training samples per class are 5. The best accuracy rate for AELTP is about 98% when the training samples from each class are 6. For the ENT dataset, also the performance of AELTP is better than that of other techniques. The accuracy rate goes up from 68.59% when there is only one sample for training from each class to be more than 97% when the training samples are 4 from each class.

All these results demonstrate the effectiveness of our new technique in comparison to others.

VII. Conclusion

In this paper, a novel LTP face recognition approach (AELTP) is proposed based on the original definition of LTP. This approach uses an adaptive threshold computed automatically from the available data by simple statistical operations. The effectiveness of this method is demonstrated on recognizing faces from two different virtual worlds. Compared with LBP, LTP and ELTP with different LBP operators and different number of sub-regions q, our proposed technique improved the recognition rate of the SL and ENT datasets. Recognizing avatar faces using multi-scale adaptive extended local ternary pattern will be the focus of our future work.

References

[1] A. A. Mohamed, D. D'Souza, N. Baili, and R. V. Yampolskiy, "Avatar Face Recognition using Wavelet Transform and Hierarchical Multi-scale LBP," *10th International Conference on Machine Learning and Applications (ICMLA'11)*, Honolulu, Hawaii, 2011, pp. 194-199.

[2] M. L. Gavrilova and R. V. Yampolskiy, "Applying Biometric Principles to Avatar Recognition," *International Conference on Cyberworlds (CW2010)*, Singapore, 2010, pp. 179-186.

[3] J. N. Oursler, M. Price, and R. V. Yampolskiy, "Parameterized Generation of Avatar Face Dataset," *14th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games*, Louisville, KY, 2009, pp. 17-22.

[4] G. Zhenhua, Z. Lei, D. Zhang, and M. Xuanqin, "Hierarchical Multiscale LBP for Face and Palmprint Recognition," *International Conference on Image Processing (ICIP)* Hong Kong, 2010, pp. 4521-4524.

[5] M. Turk and A. Pentland, "Face Recognition using Eigenfaces," *IEEE Conference on Computer Vision and Pattern Recognition*, Maui, HI 1991, pp. 586-591.

[6] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face Recognition Using LDA Based Algorithms," *IEEE Transactions on Neural Networks,* vol. 14, pp. 195-200, 2003.

[7] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition,* vol. 29, pp. 51-59, 1996.

[8] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invriant Texture Classification with Local Binary Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 971-987, 2002.

[9] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 28, pp. 2037-2041, 2006.

[10] M. A. Akhloufi and A. Bendada, "Locally Adaptive Texture Features for Multispectral Face Recognition," *IEEE International Conference on Systems Man and Cybernetics (SMC)*, Istanbul, Turkey, 2010, pp. 3308-3314.

[11] X. Tan and B. Triggs, "Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions," *IEEE Transactions on Image Processing,* vol. 19, pp. 1635-1650, 2010.

[12] H. Yang and Y. D. Wang, "A LBP-based Face Recognition Method with Hamming Distance Constraint," *4th International Conference on Image and Graphics*, 2007, pp. 645-649.

[13] L. Chen, Y.-H. Wang, Y.-D. Wang, and D. Huang, "Face Recognition with Statistical Local Binary Patterns," *8th International Conference on Machine Learning and Cybernetics*, Baoding, 2009, pp. 2433-2438.

[14] A. A. Mohamed, M. L. Gavrilova, and R. V. Yampolskiy, "Artificial Face Recognition using Wavelet Adaptive LBP with Directional Statistical Features," *12th International Conference on Cyberworlds (CW 2012)*, Darmstadt, Germany, 2012, in press.

[15] J. Meng, Y. Gao, X. Wang, T. Lin, and J. Zhang, "Face Recognition based on Local Binary Patterns with Threshold," *IEEE International Conference on Granular Computing*, San Jose, CA, 2010, pp. 352-356.

[16] W. Wang, F. Chang, J. Zhao, and Z. Chen, "Automatic facial expression recognition using local binary pattern," *8th World Congress on Intelligent Control and Automation*, Jinan, China, 2010, pp. 6375-6378.

[17] C. Zhu and R. Wang, "Local multiple patterns based multiresolution gray-scale and rotation invariant texture classification," *Information Sciences,* vol. 187, pp. 93-108, 2011.

[18] W.-H. Liao, "Region Description Using Extended Local Ternary Patterns," International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 1003-1006.

[19] *Second Life*. Available: www.secondlife.com

[20] *Entropia Universe*. Available: www.entropiauniverse.com