ESKİŞEHİR TECHNICAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
A- APPLIED SCIENCES AND ENGINEERING

2018, 19(3), pp. 784 - 795, DOI: 10.18038/aubtda.441568

HISTOGRAM OF EDGE SEGMENT CURVATURES FOR TEXTURE RECOGNITION

Mehmet KOÇ [1,*], Cihan TOPAL [2]

[1] Department of Electrical and Electronics Engineering, Engineering Faculty, Bilecik Şeyh Edebali University, Bilecik, Turkey
[2] Department of Electrical and Electronics Engineering, Engineering Faculty, Eskişehir Technical University, Eskişehir, Turkey

ABSTRACT

Texture recognition is one of the active fields in pattern recognition. Researchers have been searching for the best representation of a texture image for decades. The majority of methods use appearance-based properties of texture images to generate a feature descriptor. In this paper, we propose novel feature descriptor, namely histogram of edge segment curvatures (HESC) which extracts edge segments of an input image and construct a histogram from quantized curvature values of them. Therefore, HESC unveils geometric information of texture images by utilizing curve strengths for each pixel along the edge segments. We show that the proposed feature descriptor is robust against rotation and translation. We also extend HESC descriptor to emphasis the contribution of small curvature values. We carry out several experiments in UIUC texture dataset and compare the performance of the proposed HESC descriptor to well-known Local Binary Pattern (LBP). The proposed texture descriptor outperforms LBP in terms of recognition accuracy.

Keywords: Texture recognition, Texture descriptor, Curvature, Feature extraction

1. INTRODUCTION

Recognition of texture images is a very popular problem in image processing and computer vision for decades. Due to the wide range of texture recognition applications, there is a very crowded literature on the topic. Among many others, processing of hyperspectral images for classification of urban areas [1] and evaluation of agricultural fields [2], recognition of biometric data [3, 4], detection of local lesions in biomedical images [5], face and facial expression recognition [6-8] and detection of material surface defects [9] are some major applications that benefit from texture recognition studies.

A texture is usually formed by the repetition of one or more micro-patterns that contain information about visual and physical features such as roughness, color, shape, scale, or reflectance of an image. Similar to many pattern recognition problems, texture recognition also consists of two major steps, i.e. feature extraction and classification. In order to obtain discriminative properties of a texture image, a suitable feature extraction method should be used.

There are numerous feature extraction methods in the literature to obtain a discriminative feature vector representation [10-12]. However, the majority of these feature representations aim to utilize appearance-based characteristics of textures. One of the most well-known appearance-based feature descriptors is LBP [13]. LBP is a popular feature extraction method that captures the local properties of an image by encoding the signs of neighbor pixel differences. Then the histograms of the binary codes generated from differences are used as feature descriptor. In this way, LBP utilizes all pixels in the image and extract local intensity changes in a rough quantization process, however it does not take the geometric information into account in an explicit way.

There are a few numbers of texture descriptors that utilize geometric properties for texture representation. These geometric properties can be evaluated from the contours of the shape, i.e.

*Corresponding Author: mehmet.koc@bilecik.edu.tr
Received: 08.07.2018 Accepted:07.09.2018

curvature or interior of the shape, i.e. topological descriptors, moments invariants [14]. Efficient shape-based features should be invariant to translation, rotation, scale [14].

In this work, we propose a novel rotation and translation invariant texture descriptor that directly exploits geometric characteristics of the input image. In our method, we first extract edge segments from the input texture and compute the curvature information of them. Then we calculate the curvature histogram to estimate a good feature representation of the input texture by utilizing shape information encoded in the texture patterns. We run quantitative experiments on a well-known texture dataset, i.e. UIUC and compare the performance of the proposed method with local binary patterns as the baseline algorithm.

The rest of the paper is organized as follows: We give a brief review of LBP and related concept with the proposed algorithm in Sec. 2. In Sec. 3 we provide details on the proposed feature descriptor based on curvature information. In Sec. 4 we present quantitative experimental results and conclude the paper in Sec. 5.

2. RELATED WORK

Local binary pattern was firstly introduced by Ojala et al. [13] for texture recognition. LBP uses a local 3×3 neighborhood of a pixel, thresholds the center pixel and its neighbors. It combines the binary result of the thresholding in a 256-bin histogram and uses as an image descriptor. LBP can be formulized as follows:

$$LBP(x_c, y_c) = \sum_{p=0}^{7} s(g_p - g_c)2^p \tag{1}$$

Here g_c is the grey level value of the center pixel, g_p is the grey level value of the neighbor pixels equally spaced on a circle of radius 1, and $s(x)$ is 1 when $x \geq 0$, otherwise it is 0. A simple illustration of LBP coding is given in Figure 1(a). After thresholding the neighbor pixels with the center pixel, binary code 11100010 is obtained which equals 225 in decimal form. In Figure 1(b), extension of LBP operator is shown. Since a circular neighborhood with unit radius is used, the intensity values of the equally spaced sampling points are calculated using bilinear interpolation when they are not in the center of a pixel. LBP and its variants are used several machine vision problems such as, face recognition [6, 15], texture recognition [16-20], object recognition [21], etc.

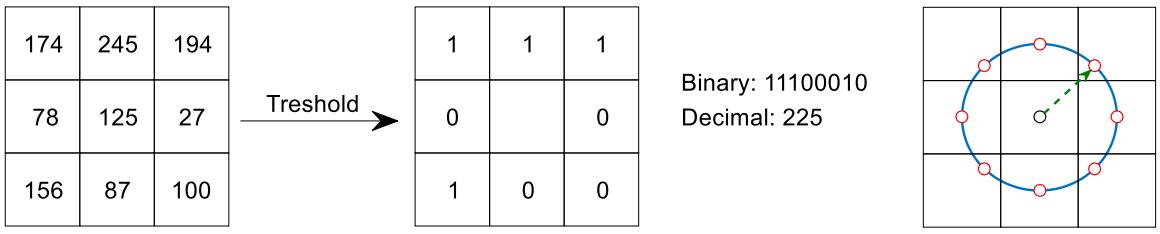


Figure 1. Illustration of **(a)** basic LBP operator, **(b)** circular LBP operator with radius 1 and 8 sampling points

Shape is one of the most basic geometric features that is used to describe the content of the image. Simple geometric features can be used to describe shape of an object. Some of these features are center of gravity, eccentricity, circularity ratio, convexity, hole area ratio, rectangularity, etc. However, they cannot efficiently discriminate the shapes when they used alone as a descriptor [14]. Other types of shape descriptors are derived from the boundary of a shape. These descriptors can get the perceptual properties of the shape [22]. Among these, one of the well-known shape descriptors is contour curvature which has distinct perceptual properties and is shown to be useful for shape recognition [23]. On the other hand, curvature is never used in texture recognition problems to describe the texture image.

3. PROPOSED METHOD

A texture image is formed by the repetition of several visual and physical micro-patterns. Appearance-based texture descriptors such as LBP aim to provide an efficient feature representation by accumulating similar micro-patterns in the same bins of a histogram. In this way, redundant visual information in textures can be transformed into a discriminative feature vector.

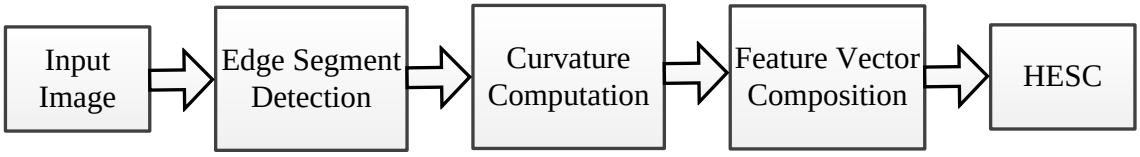


Figure 2. Block diagram of the proposed algorithm

In this paper, we propose a novel texture descriptor that utilizes geometric information in texture images by calculating repetition amounts of curvature values in edge segments. In the first step of the algorithm we extract edge segments from the input image. In the next step we compute curvature of the detected edge segments each of which consists of consecutive edge pixels. Edge detection step must be accurately performed to ensure that the resulting edges efficiently capture the geometric structure of the imaged texture. Once the edge segments are obtained, we compute curvature values of them. To suppress the effect of quantization, we apply Gaussian smoothing along x and y coordinates before the curvature computation. Finally, we accumulate curvature values in a histogram to obtain the final feature vector. Block diagram of the proposed texture descriptor algorithm is presented in Figure 2. We provide more detail about the proposed method in the following subsections.

3.1. Edge Segment Detection

The first step of the proposed texture descriptor is detection of edge segments in the texture image each of which is a connected array of pixels. This step has substantial importance due to the fact that the geometric information from input textures is obtained in this process. To have a discriminative feature vector that efficiently represents image patterns in the feature space, edge segments need to be detected precisely so that they can represent the geometric characteristics of input texture efficiently. To ensure capturing the geometric structure of texture details we need to have contiguous, smooth and one-pixel width edge segments.

For this purpose, we used Edge Drawing (ED) [24] algorithm that is able to extract high quality edge segments in an image. Conventional edge detectors like Canny [25] extracts edge pixels via a subtractive approach that detects a group of edge pixels and eliminates them with respect various criteria such as gradient magnitude and gradient direction. Application of smoothing and derivative filtering (i.e. Sobel) then thresholding are common steps in almost all edge detection algorithms. After these steps, Canny algorithm applies non-maximal suppression step and decides for each pixel whether it is and edgel (edge element) or not without utilizing the status of neighbor pixels. It evaluates each pixel individually and eliminates pixels from the ones that survive thresholding in a subtractive manner. In this way, single pixels can be selected as an edgel or some critical pixels that provide connectivity cannot be selected as an edgel. Therefore, unattended pixel clusters occur in the final edge output and edge quality is degraded.

Instead of the abovementioned subtractive approach, i.e. eliminating non-edge pixels from the thresholded pixels, ED algorithm follows an additive approach where it first locates several edge pixels

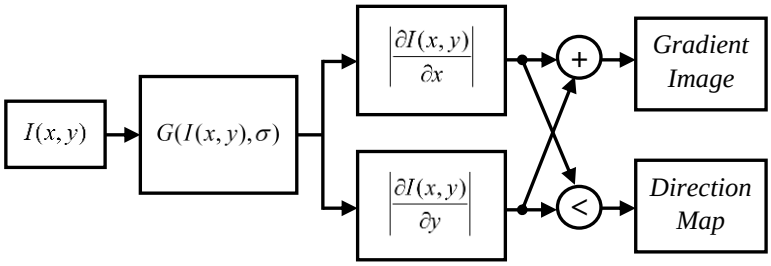


Figure 3. Computation of gradient image and direction map in ED algorithm.

called *anchor points* and extracts the whole edge map by appending edge pixels with respect to their gradient responses. Since the algorithm picks the pixels having the highest gradient response among the neighbor pixels; it ensures both the edge continuity and good localization.

In Figure 3 construction of gradient image and direction map is shown in block diagram. As shown in the diagram, gradient map is constructed as the magnitudes of horizontal and vertical partial image derivatives. By comparing the same partial derivatives, we also obtain the direction map which guides composition of edge segments. Then the algorithm picks anchor points several local gradient maxima pixels as initial edge elements and extract edge remaining edge pixels by the help of direction map. In Figure 4 a sample image is shown with extracted anchor points and final edge image.

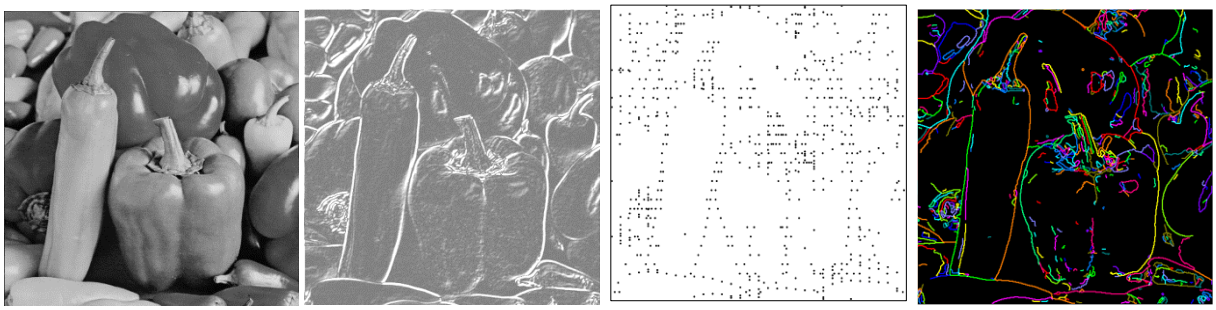


Figure 4. a) Sample image, **b)** gradient magnitudes, **c)** extracted anchor images, **d)** edge segments.

In this way we are able to extract high quality edge segments without compromising geometric info. In Figure 5 edge segments of an example texture image is provided obtained by Canny and ED algorithms. The edge segments shown in Figure 5(c) obtained by Canny followed by a connected component analysis have discontinuities and notches. In Figure 5(d) edge segments obtained by ED algorithm is shown for the same texture patch. It is easily seen that their geometry is better preserved with one-pixel width smooth and contiguous array of pixels.

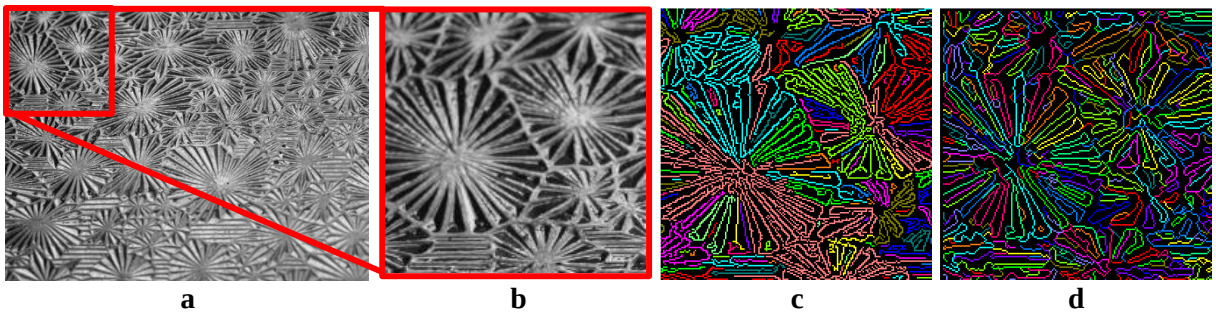


Figure 5. Extracted edge segments from a sample texture image. **a)** sample texture image, **b)** a magnified patch from the image, **c)** edge segments obtained by Canny + 8-way connected component detection, **d)** edge segments obtained by ED algorithm.

3.2 Curvature Computation

After the edge segments are extracted from the input texture, the next step is to compute the curvature values of them by utilizing their consecutive pixel coordinates. Curvature can be defined as a measure of deviation of a geometric curve from being flat. It is generally used for detection of corner points along planar curves due to the fact that corner locations' high curvature values. In addition to that, curvature function can also extract discriminative information for individual curve and shapes. We can expect that curves having similar shapes provide analogous curvature values. In this study, we aim to utilize curvature information in order to compose a shape-based texture descriptor. For this purpose, we compute curvature values of edge segments obtained from texture images.

Even though there are several ways to compute the curvature; the most common calculation method is given by the following formula:

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{[\dot{x}^2(t) + \dot{y}^2(t)]^{\frac{3}{2}}} \tag{2}$$

Here $x(t)$ and $y(t)$ represent the x and y coordinates of the pixel at time t. Also $\dot{x}$ and $\ddot{x}$ denote the first and the second order derivatives of time series x, respectively. In Figure 6, a sample shape (Figure 6(a)) and its curvature plot (Figure 6(b)) are shown. Since the pixel values are quantized in digital images, the output of the curvature function for even a smooth shape can be fluctuating and becomes difficult to interpret. Therefore, coordinate sequences are smoothed by an appropriate Gaussian kernel before the calculation of curvature. In Figure 6(c) the curvature plot is shown after the convolution of edge segment coordinates with a Gaussian kernel of $\sigma = 10$. The effect of smoothing operation is clearly seen in the figure. Note that sharp trajectory changes result higher curvature values along the contour of the sample butterfly shape.

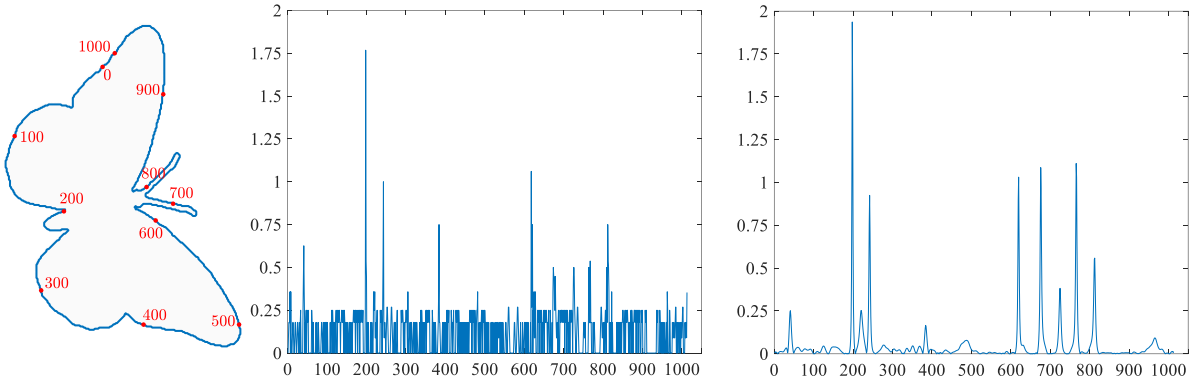


Figure 6. **a)** Edge segment extracted from a sample image. The curvature of the edge segment is computed starting from the pixel marked with 0 along the increasing values of the marked numbers, **b)** curvature of the edge segment, **c)** curvature of the edge segment after smoothing.

3.3. Feature Vector Composition

In the previous section, we show that curvature values obtained from edge segments contain distinctive shape information. After the curvature values are computed from high quality edge segments, the next step is composing a feature vector by utilization of curvature values. Because we want our descriptor invariant against rotation changes, we quantize curvature values and construct a histogram from them. Due to the fact that the curvature function calculates derivatives of x and y coordinate values, it gives the same output regardless orientation and rotation of the shape. Consequently, we title the proposed descriptor as HESC that stands for *histogram of edge segment curvatures*.

Several sample shapes and their histograms of quantized curvature values for various scale and rotations are shown in Figure 7. It is clearly seen that we can obtain similar curvature histograms for different inputs regardless their orientations. Since the histograms in Figure 7 (a), (b) and (c) columns have very close shapes for different rotations, we can assure that the proposed descriptor is rotation invariant. Also, since the histograms of each shape have a diversified distribution, we can say that HESC has potential to provide discriminative features for different input textures.

Note that the sample images consist of synthetic objects having only one edge segment obtained from the object's contour. These objects are employed for illustration purposes to qualitatively show efficiency of the proposed descriptor. In real texture images there are large number of edge segments and the histogram will be constructed from all of them.

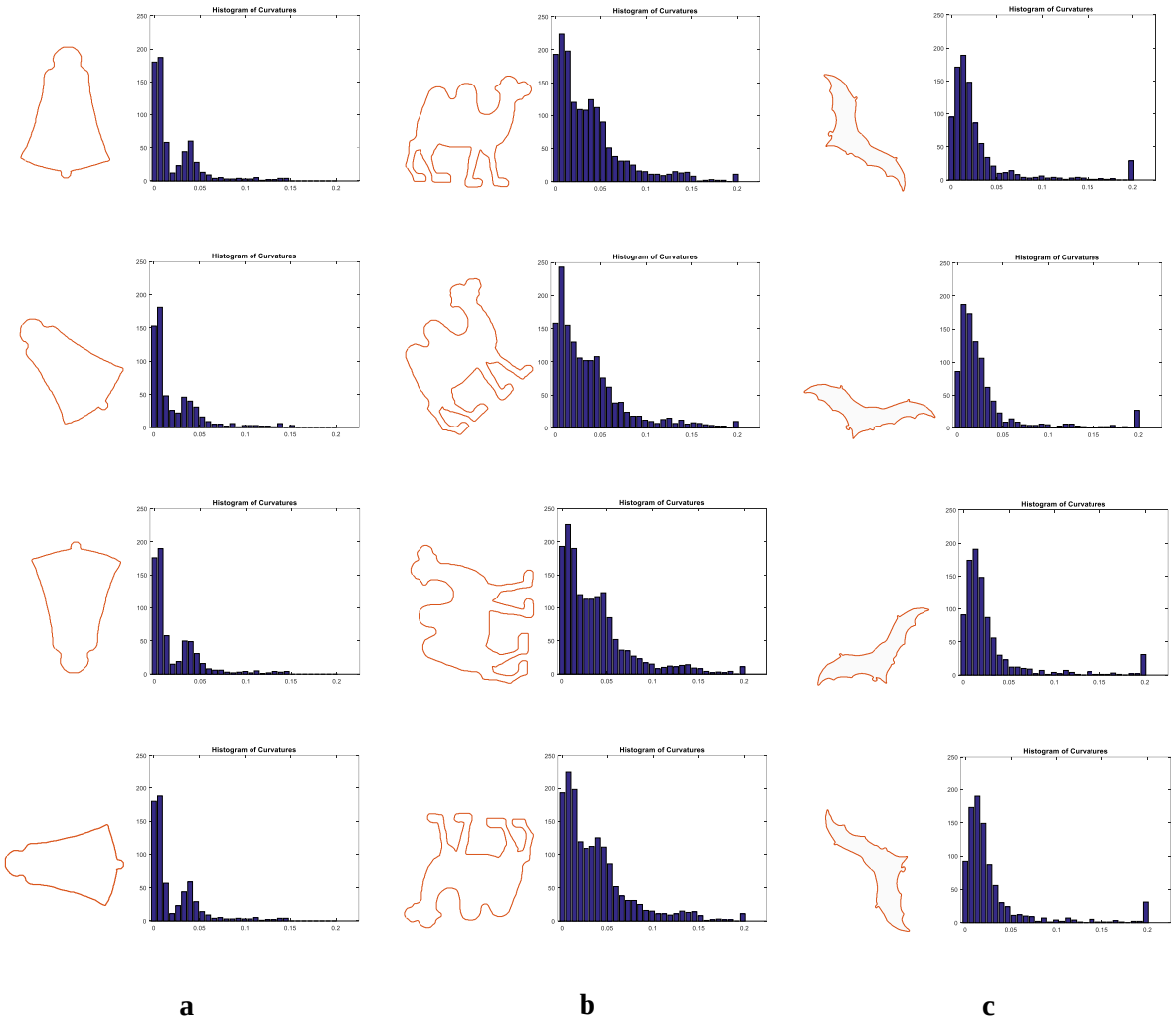
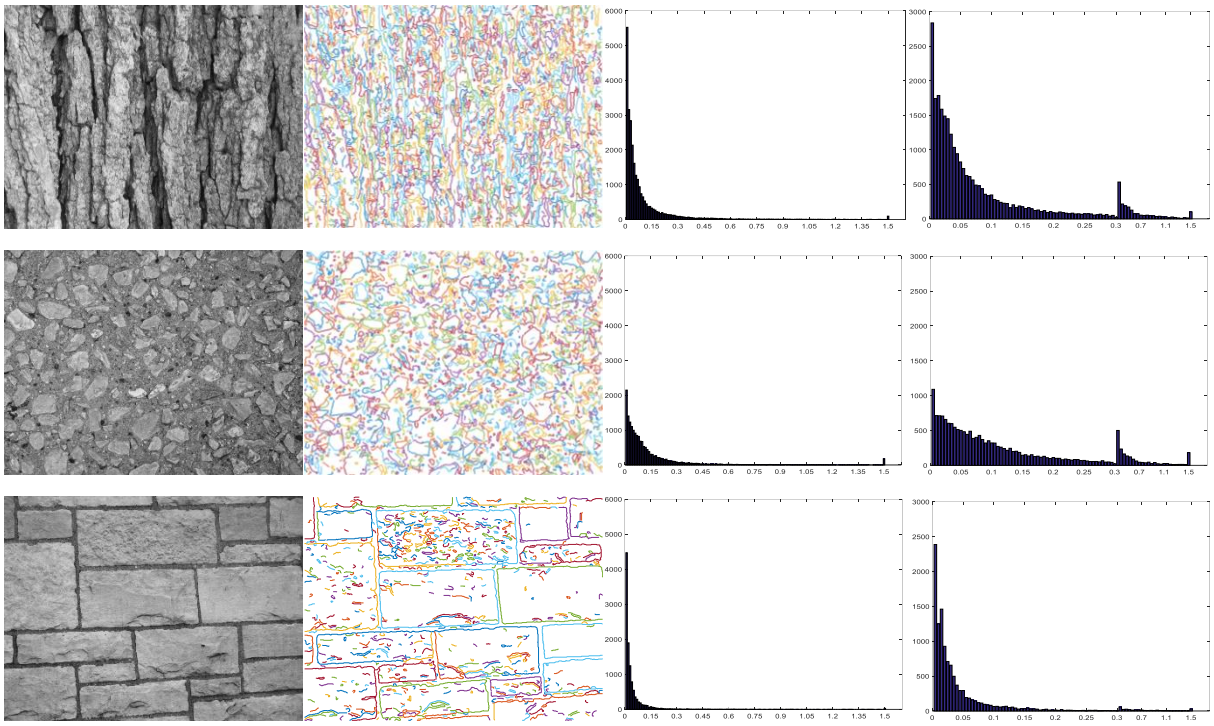


Figure 7. The comparison of the histogram of contour curvatures for three different shapes and their rotated variants.

Range of the curvature function starts from 0 for pixels lying along flat segments and increases with respect to the magnitude of the turning amount of the contour. Although we rarely encounter values greater than 2, it can have positive values up to 9 in our experiments for edge segments obtained from real images. Since the curvature function is continuous, its output is also continuous. For this reason, we quantize curvature values into bins to construct a histogram.

As clearly seen from Figure 8(b) that the most of the contour information is accumulated in the first few bins of the histogram. This is not unexpected when considering that real world shapes usually consist of straight lines and weak curves and they have a few numbers of strong curves and corners. Because of this reason we decided to use different quantization step size for low and high curvature values in a feature vector. In this way, we aim to increase the resolution of the histogram's first bins where curvature values from straight and smooth details are accumulated. We entitle the version that we divide the feature vector into two pieces HESC+. The HESC and HESC+ histograms for various textures are shown in Figure 8.



a **b** **c** **d**

Figure 8. a) Sample texture images from UIUC dataset, **b)** edge segments of the texture images derived using ED algorithm, **c)** HESC, and d) HESC+ histograms.

In the first column of Figure 8, we present three sample images from different subjects of UIUC dataset. The edge segments of the images with one-pixel width are extracted using ED algorithm. As clearly seen from Figure 8(c), most of the curvature values are piled up in the few first histogram bins of HESC. This is an expected result since generally most of the parts of edge segments are simple flat curves like lines and low degree polynomials. In Figure 8(d), we give the HESC+ histograms of the sample images. HESC+ histogram is obtained by splitting the HESC histogram into two parts. The first part of the histogram includes low curvature values that compose the most of HESC. To increase the importance of the low curvature values we increase the resolution of the first part of HESC. In Figure 8(d), the first part includes the curvature values from the interval $[0, 0.3]$. Width of the histogram bins in this interval is 0.005. The second part of the HESC+ histogram consists the curvature values higher than 0.3. Since the edge segments in the real-world objects are mostly flat and high curvature values appear at corner or cusp points of the edge segments the second part of histogram is sparse. So, we set the number of the bins to 24.

4. EXPERIMENTAL RESULTS

In this section we evaluated performance of the proposed HESC descriptor by performing comprehensive set of texture recognition experiments. We ran experiments on well-known UIUC [26] texture dataset that

consist of 25 texture classes each of which includes 40 samples in 640×480 resolution. UIUC is a very challenging dataset that contains diverse scale, rotation, viewpoint variations and non-rigid deformations. Sample texture images from five classes of UIUC database are given in [26]. We employed naïve LBP [13] algorithm as the baseline method to compare accuracy of HESC and HESC+.

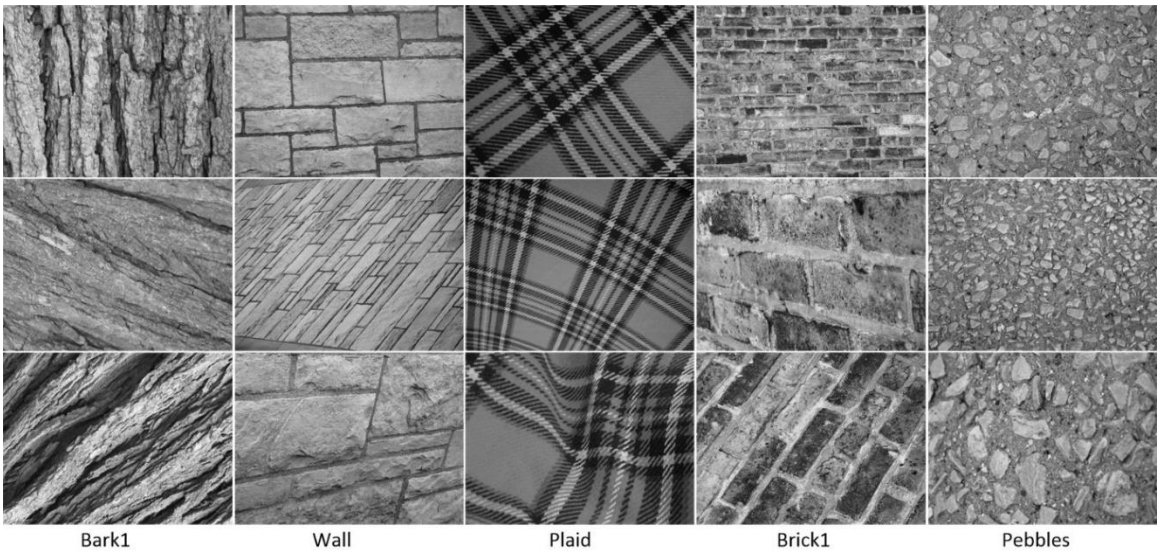


Figure 9. Sample texture images from five different classes of UIUC database. Each column has samples from different classes with different viewpoint, scale, and non-rigid deformations.

In all experiments we randomly chose 20 images from each class for training and used rest of the images for test. In the classification stage, we used Chi-square which is a popular dissimilarity measure to compute the distance between two histograms. We followed the random subsampling procedure and run each experiment 10 times. We obtained the final accuracy by averaging the accuracy of each run.

In the first experiment, we analyzed the effect of quantization range of the curvature values to the recognition accuracy. We set the quantization step size to a reasonable value (0.01) and run the experiments for different quantization ranges starting from 0.5 to 5. In this manner, dimensions of our feature vectors are ranging from 50 to 500, respectively. For each experiment, we quantized the curvature values exceeding the allowed range to the maximum curvature value.

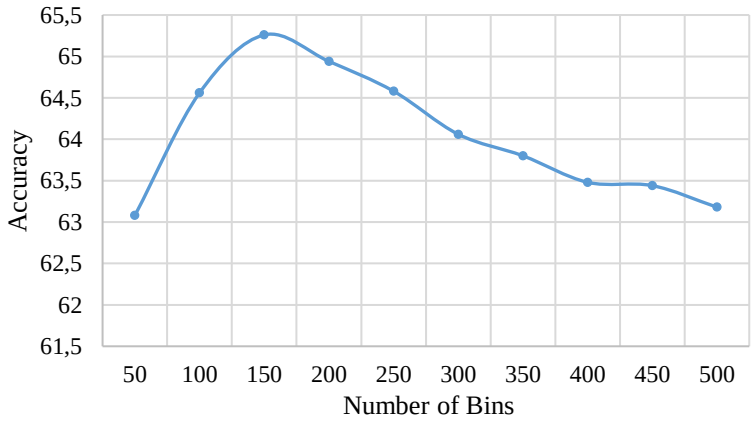


Figure 9. Recognition accuracies of HESC for different curvature ranges.

In Figure 9 the recognition results for various quantization ranges of curvature function is shown. Since the quantization step size is kept constant, the dimension of feature vector size increases as the range increases. As it is seen on the plot, we obtained the best accuracy result when we quantize the curvature function into a range of [0, 1.5]. As we increase the curvature range, the accuracy decreases even though the feature vector size also increases.

Besides the quantization range, another significantly important parameter that might affect the accuracy of the proposed feature descriptor is quantization step size. After we had determined the best performing range, then we performed experiments to find an optimal value for quantization step size. For this purpose, we ran experiments by fixing the quantization range to [0, 1.5] and we construct feature vector with various quantization step sizes by splitting that range into different number of bins. In Figure 10 and Table 1, accuracy results for this experiment are presented. As seen in the figure we obtained the best accuracy when quantization step size is $5x10^{-3}$.

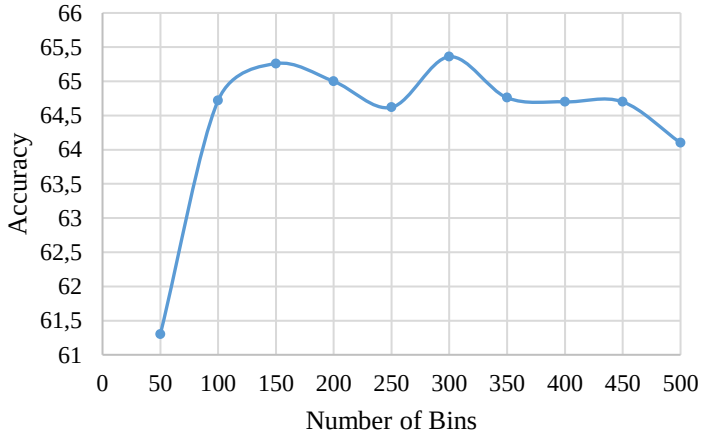


Figure 10. Recognition accuracy of HESC for various quantization step sizes.

By separately testing hyper parameters such as quantization range and step size, we determined suboptimal values that work best. In the final experiment, we used those values to determine where we divide the histogram into two pieces, i.e. low and high curvature parts, to obtain HESC+. Hence, we ran experiments by dividing the feature vector into two parts, i.e. low curvature and high curvature where we used fine and coarse quantization step size, respectively. Besides using different quantization step sizes, we also tried various locations to divide the feature vector into two parts.

We present experimental results for HESC+ descriptor in Table 2. In the table we give details of each feature vector for low and high curvature parts such as width, number of bins and quantization step size. According to quantitative experiments, we obtained best result when we separate the low and high curvature parts at 0.3. In that experiment, we divide the low and high curvature parts into 60 and 24 bins, respectively. Hence, we used an 84-dimensional feature vector and obtained 66.06% accuracy.

Table 1. Recognition accuracies for various quantization step sizes and corresponding number of bins.

Step size	3.0E-2	1.5E-2	1.0E-2	7.5E-3	6.0E-3	5.0E-3	4.3E-3	3.8E-3	3.3E-3	3.0E-3
No. bins	50	100	150	200	250	300	350	400	450	500
Accuracy	61.30	64.72	65.26	65.00	64.62	**65.36**	64.76	64.70	64.70	64.10

Table 2. Recognition accuracies for HESC+ descriptor on UIUC dataset. QSS stands for quantization step size.

Low Curvature Part (QSS = 0.005)		**High Curvature Part (QSS = 0.05)**			
Width	**# of Bins**	**Width**	**# of Bins**	**Total # of Bins**	**Accuracy (%)**
0.1	20	1.4	28	48	65.84
0.2	40	1.3	26	66	65.68
0.3	**60**	**1.2**	**24**	**84**	**66.06**
0.4	80	1.1	22	102	65.46
0.5	100	1.0	20	120	65.50
0.6	120	0.9	18	138	65.36
0.7	140	0.8	16	156	65.54

In the last part of experimental work, we compare the proposed feature descriptor to the well-known texture feature LBP [13] as the baseline method. In Figure 11, recognition accuracies are presented for LBP and proposed descriptors. According to the results, HESC and HESC+ outperformed naïve LBP up to %9 and %10, respectively, with a smaller feature vector and by utilizing only the geometric information. LBP feature vectors are calculated using 8 sampling point on a circle with unit radius. From the definition of LBP coding, the LBP code of a pixel can take 256 different values, thus the dimension of LBP feature vector is 256. When compared with LBP, we not only increase the recognition accuracy but also decrease the feature vector size.

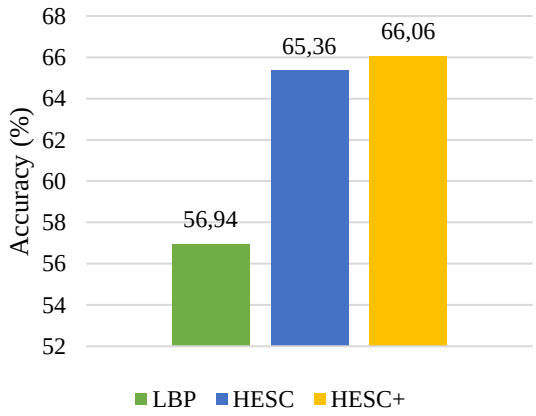


Figure 11. Recognition accuracies of LBP, HESC, and HESC+ in UIUC texture database.

5. CONCLUSIONS

In this study, we propose a novel texture descriptor that utilizes curvature of edge segments extracted from input image. Since output of the curvature function is independent from the orientation of the contour, proposed descriptor becomes invariant to rotation by default. We extract edge segments as array of consecutive pixels and smooth them to remove the aliasing effect of edge pixels. Then we compute curvature function from each edge segment and quantize them according to curvature responses. In the final step we simply accumulate curvature values in a histogram.

On the contrary to conventional texture descriptors, proposed HESC and HESC+ descriptors utilize only geometric information extracted from input image. We ran comprehensive set of experiments on UIUC dataset and quantitatively show that HESC gives promising results. We compare recognition results to well-known LBP method and show that HESC and HESC+ outperform it up to 10% even with a lower dimensional feature vector.

In future work, we plan to extend the proposed HESC feature to a multi-scale descriptor which compounds information obtained via several curvature scale spaces (CSS). We also plan to utilize a transformation function to spread the histogram bins more evenly instead of dividing the feature histogram into two distinct parts. Another future study is combining HESC with other appearance- based texture descriptors. Due to the fact that HESC utilizes geometric information, we believe that fusion of appearance-based and geometric-based features might yield better recognition rather than combination of two same-type features such as two appearance-based methods.

REFERENCES

[1] Rellier G, Descombes X., Falzon F, Zerubia J. Texture feature analysis using a gauss-Markov model in hyperspectral image classification. IEEE T Geosci Remote 2004; 42(7):1543-1551.

[2] Mirzapour F, Ghassemian H. Improving hyperspectral image classification by combining spectral, texture, and shape features. Int J Remote Sens 2015; 36(4): 1070-1096.

[3] Guo Z, Zhang L, Zhang D, Mou X. Hierarchical multiscale LBP for face and palmprint recognition. In: IEEE International Conference on Image Processing, 2010; Hong Kong,China. pp. 4521-4524.

[4] Nguyen K, Fookes C, Jillela R, Sridharan S, Ross A. Long range iris recognition: A survey. Pattern Recognit 2017; 72: 123-143.

[5] Kassner A, Thornhill R. Texture analysis: a review of neurologic MR imaging applications. AJNR Am J Neuroradiol 2010; 3(5); 809-816.

[6] Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition. IEEE Trans Pattern Anal Mach Intell 2006; 28(12): 2037-2041.

[7] Shan C, Gong S, McOwan P. Robust facial expression recognition using local binary patterns. In IEEE International Conference on Image Processing, 2005; Genova, Italy. pp. II-370-3.

[8] Gao Y, Ma J, Yuille A. Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. IEEE Trans Image Process 2017; 20(5): 2545-2560.

[9] Xie X. A review of recent advances in surface defect detection using texture analysis techniques. Electronic Letters on Computer Vision and Image Analysis 2008; 7(3): 1-22.

[10] Ojala T, Pietikäinen M, Mäenpää T. Multi-resolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 2002; 7(24): 971-987.

[11] Crosier M, Griffin L. Using Basic Image Features for Texture Classification. Int J Comput Vis 2010; 88(3): 447-460.

[12] Guo Y, Zhao G, Pietikäinen M. Discriminative features for texture description. Pattern Recognit 2012; 45(10): 3834-3843.

[13] Ojala T, Pietikäinen M, Harwood D. A comparative study of texture measures with classification based on feature distributions. Pattern Recognit 1996; 29: 51-59.

[14] Yang M, Kpalma K, Ronsin J. A survey of shape feature extraction techniques. Pattern Recogniti, P. Yin, Ed., IN-TECH, 2008.

[15] Pei S, Chen M, Yu Y, Tang S, ZhongC. Compact LBP and WLBP descriptor with magnitude and direction difference for face recognition In: *IEEE International Conference on Image Processing (ICIP)* 2017; Beijing, China.

[16] Kazak N, Koc M. Some variants of spiral LBP in texture recognition. IET Image Process 2018; 12(8): 1388-1393.

[17] Gou Z, Zhang L, Zhang D. A completed modelling of local binary pattern operator for texture classification. IEEE Trans. Image Process 2010; 19(6): 1657-1663.

[18] Kazak N, Koc M. Performance analysis of spiral neighbourhood topology based local binary patterns in texture recognition. International Journal of Applied Mathematics, Electronics and Computers 2016; 4: 338-341.

[19] Kazak N, Koc M, Benligiray B, TopalC. A comparison of classification methods for local binary patterns. In: IEEE Signal Processing and Communication Application Conference 2016; Zonguldak, Turkey. pp. 805-808.

[20] Liu L, Fieguth P, GuoY, Wang X, Pietikäinen M. Local binary features for texture classification: Taxonomy and experimental study. *Pattern Recognit 2017; 62: 135-160.*

[21] Shang J, Chen C, LiangH. Object recognition using rotation invariant local binary pattern of significant bit planes. IET Image Process 2016; 10(9): 662-670.

[22] Yadav R, Nishchala N, Gupta A, Rastogi V. Retrieval and classification of shape-based objects using Fourier, generic Fourier, and wavelet-Fourier descriptors technique: A comparative study. Opt Lasers Eng. 2007; 45(6): 695-708.

[23] Wang Y, Lee K, Toraichi K. Multiscale curvature-based shape representation using B-spline wavelets. IEEE Trans. Image Process 1999; 8(10): 1586-1592.

[24] Topal C, Akinlar C. Edge Drawing: A combined real-time edge and segment detector. J Vis Commun Image Represent 2012; 23(6): 862-872.

[25] Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 1986; 8(6): 679-698.

[26] Lazebnik S, Schmid C, Ponce J. A Sparse Texture Representation Using Local Affine Regions. IEEE Trans Pattern Anal Mach Intell 2005; 27(8):1265-1278.