2014 22nd International Conference on Pattern Recognition

Adaptive Median Binary Patterns for Texture Classification

Adel Hafiane
INSA CVL, Univ. Orléans, PRISME, EA 4229
88 Boulvard Lahitolle
F-18022, Bourges, France

Kannappan Palaniappan
Department of Computer Science
University of Missouri
Columbia, M0, 65211, USA

Guna Seetharaman
Information Directorate
Air Force Research Laboratory
Rome, NY, 13441, USA

***Abstract*—This paper addresses the challenging problem of recognition and classification of textured surfaces under illumination variation, geometric transformations and noisy sensor measurements. We propose a new texture operator, Adaptive Median Binary Patterns (AMBP) that extends our previous Median Binary Patterns (MBP) texture feature. The principal idea of AMBP is to hash small local image patches into a binary pattern texton by fusing MBP and Local Binary Patterns (LBP) operators combined with using self-adaptive analysis window sizes to better capture invariant microstructure information while providing robustness to noise. The AMBP scheme is shown to be an effective mechanism for non-parametric learning of spatially varying image texture statistics. The local distribution of rotation invariant and uniform binary pattern subsets extended with more global joint information are used as the descriptors for robust texture classification. The AMBP is shown to outperform recent binary pattern and filtering-based texture analysis methods on two large texture corpora (CUReT and KTH_TIPS2-b) with and without additive noise. The AMBP method is slightly superior to the best techniques in the noiseless case but significantly outperforms other methods in the presence of impulse noise.**

I. Introduction

Texture is present in most natural surfaces and manufactured materials. Texture provides an important cue in computer vision and pattern recognition as it conveys information about the nature of objects and surfaces. Although texture analysis and classification have been extensively investigated over the past few decades it is still an open problem, particularly for complex textured surfaces under varying luminance, viewing conditions and noise processes. Many types of texture features have been proposed [1], [2], but there is still no unique mathematical definition of texture that is consistent with perceptual properties of the human visual system. Good texture features that are reliable for texture classification and object recognition tasks need to retain similarity relationships under varying viewing geometries and illumination levels. The importance of improving existing texture features and discovering new ones with various invariance properties for constructing texture dictionaries and classification systems has spurred new research in texture analysis, emphasizing rotation, scale, and intensity-shift invariance.

Methods developed over the last decade based on histograms of local primitives and related properties show remarkable accuracy, reliability and generality [3], [4], [5], [6], [7], [8], [9]. Among these approaches, the Local Binary Patterns (LBP) operator [5] and its variations have emerged as a leading texture descriptor for many applications including texture classification, object recognition and face biometrics, due to its discriminative power, simplicity and computational efficiency. Zhang *et al.* [10] combined Gabor filters and LBP to improve face recognition performance. Heikkilä and Pietikäinen [11] used adaptive local binary pattern histograms to model video scene backgrounds for change detection. Zhao and Pietikäinen [12] extended the LBP to incorporates temporal information for dynamic textures. Liao *et al.* [13] used a histogram of the most frequently occurring LBP textons combined with the Gabor filter to improve classification tasks. Guo *et al.* [14] introduced the Completed LBP (CLBP) that incorporates both the sign and magnitude of local differences compared to the mean difference over the entire image encoded as a joint histogram resulting in improved performance. Chen et al. [15] use histograms of relative difference ratios and orientations based on the Weber Law. The trend in current LBP-based methods has been to focus on additional features and extending the descriptor space to increase discriminative power using Gabor filters, gradient orientations, global information, *etc.* [13], [14], [15], [16]. However, few papers focus on feature fusion without inflating dimensionality or the effects of noise on micro-texture and the distribution of local binary patterns. A recent example of the latter is the robust LBP which remaps selective non-uniform patterns to improve noise resilience [17].

Recently, we extended the LBP approach to the MBP using the local median instead of the center pixel [18], [19] and applied it to appearance-based vehicle tracking in wide area video [20], [21]. The MBP preserves the overall characteristics of LBP and increases robustness especially against impulse noise processes. The MBP was independently shown to be one of the best performing non-parametric texture description operators [22]. We observed that the MBP can produce more diverse binary patterns if it adapts to local context in the image. In this paper we propose the Adaptive Median Binary Patterns (AMBP) operator that uses local adaptive region analysis to hash binary patterns. The principle is based on adaptive median filtering used to increase the noise tolerance of the median filter [23]. The AMBP analysis window size can vary from one pixel location to another, whereas the binary pattern size remains constant to keep the descriptor computationally manageable. The local patterns are hashed using the 3×3 local neighborhood of each pixel. The AMBP can use thresholds from beyond the 3×3 neighborhood. AMBP uses either the central pixel or the median value within the adaptive window as the threshold. The former case is the same as the LBP whereas the latter one yields the MBP. The AMBP histogram fuses information from both LBP and MBP operators depending on

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.205

the microstructure and noise. The benefit of such operators is to diversify the binary patterns by adapting the analysis window to local scene content and context in an automated manner.

II. Texture analysis using binary patterns

A. Local Binary Patterns

The Local Binary Patterns (LBP) [5] is one of the best performing texture descriptors with high discriminative classification behavior. The computation of the LBP descriptor consists of two major steps including, first, extracting textons (*i.e.* binary patterns or texels) of local elementary spatial structures in the texture image; and, second, analyzing the histogram distribution of the textons. The first step is illustrated in Fig. 1 where the central pixel gray value of 127 is used as the threshold for the 3×3 patch to produce the 8-bit binary pattern 00001110 or 14. More formally, the basic LBP using a non-circular neighborhood for pixel (i, j) is defined as [24]:

$$LBP(i,j) = \sum_{k=0}^{7} 2^k H(b_k - b_c),$$
$$H(b) = \begin{cases} 1, & \text{if } b \geq 0 \\ 0, & \text{if } b < 0 \end{cases} \quad (1)$$

for a 3×3 pixel neighborhood, with b_k being the pixel gray value at position k in the neighborhood, b_c the gray value of the central pixel and $H(\cdot)$ the Heaviside (discrete) unit step function also referred to as a binary thresholding function. The resulting pattern is captured as an 8-bit binary number representing one of 256 distinct binary patterns. The histogram of these binary patterns is used as a texture descriptor. The LBP method tries to characterize the local properties of texture by using spatial operators to transform image pixels and patches into a new representation. Then, it seeks to code, characterize and quantify the spatial arrangement of structures observed in the transformed domain.

127	130	135
120	127	88
90	30	43

LBP

1	1	1
0		0
0	0	0

Fig. 1: LBP operator applied to a 3×3 neighborhood. The central pixel is not encoded in the output pattern. $LBP = 00001110_2 \equiv 14$

B. Median Binary Patterns

The Median Binary Patterns (MBP) operator maps from the intensity space to a localized binary pattern by thresholding the pixels against their median value within a (typically 3×3) neighborhood, instead of always using the central pixel, to provide more sensitivity to microstructure and noise robustness [18], [19]. An example is shown in Fig. 2, where the median value in the patch is 120. The MBP at pixel (i, j) is given by:

$$MBP(i,j) = \sum_{k=0}^{L-1} 2^k H(b_k - \tau) \quad (2)$$

where L is the patch size or pixel neighborhood (*ie.* $L = 9$ for a 3×3 patch), and τ is the median computed over the local patch (*ie.* 3×3). By constraining the threshold τ to be the median within the patch means that there will be at least 5 one bits in the resulting binary pattern and only 256 possible binary patterns can occur. The MBP in this case does not generate the full 9 bit range of binary patters (i.e. [0, 511]) but only a 256 binary pattern subset. The histogram of MBP is used to measure the distribution of these patterns over the image and forms the texture descriptor. Thus, the entire image is hashed to a 256×1 vector, i.e., the MBP histogram using the median within the local image patch.

127	130	135
120	127	88
90	30	43

MBP

1	1	1
1	1*	0
0	0	0

Fig. 2: MBP operator for a 3×3 patch (*i.e.* $L = 9$) with a median value of 120 (*i.e.* $\tau = 120$). Output binary pattern $MBP = 100011110_2 \equiv 286$. Ordering of bits in the binary pattern uses b_0 as the least significant bit (Fig.3).

C. Adaptive Median Binary Patterns (AMBP)

The AMBP algorithm is described in Algorithm 1, where S is the square region around a given pixel, k_{max} is the maximum size of S, Z_{med}, Z_{min} and Z_{max} are the median, the minimum and the maximum values inside S respectively. The AMBP considers larger analysis windows around the central pixel for computing the local median to establish the threshold gray value for converting to a binary pattern (see Figure 3). In this case the resulting 9-bit binary pattern can take on the full range of 512 values since the median gray value may not be contained within the 3×3 binary pattern image patch. The classical median filter uses a constant square (or circular) neighborhood window around the processed pixel of size 3×3, 5×5, etc. The median value depends on all of the pixels within the filtering window which can be of different sizes. Small regions may miss important information while a large median window may yield a biased median value due to the larger number of pixels.

There is no single optimal fixed window size that can be determined *a priori* due to the changing local image context and scene variability. A constant size analysis window cannot handle all the scene variations in imagery. An automatically adaptive window presents a better alternative since the analysis region can optimize a desired criteria at each (central) pixel location. The adaptive median filter uses this idea to compare the relationship between the median value, the central pixel and the extreme values (minimum and maximum) in the neighborhood to avoid replacing the central pixel with the value from a flat region or impulse noise. The adaptive median filter has been shown to be more effective in many image processing and computer vision applications compared to the simple median filter [23]. Similarly, instead of using a fixed window to compute the median value we adapt the AMBP neighborhood size based on the local scene context.

III. Multi-scale binary pattern histograms

In order to reduce scale sensitivity, the image is decomposed into several frequency bands using a specific pyramidal subsampling scheme and the binary pattern histogram is computed for each of these images at each level of the pyramid. Let $\mathcal{I}$ be the original image of size $M \times N$, then, the subsampling process can be described as follows:

$$\mathcal{I}_1(i,j) = \mathcal{I}(2i, 2j), \quad \mathcal{I}_2(i,j) = \mathcal{I}(2i+1, 2j) \quad (3)$$
$$\mathcal{I}_3(i,j) = \mathcal{I}(2i, 2j+1), \quad \mathcal{I}_4(i,j) = \mathcal{I}(2i+1, 2j+1)$$

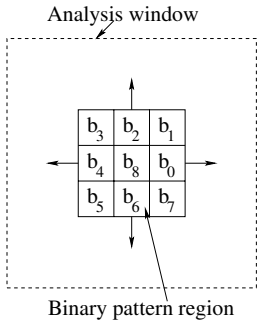

Fig. 3: AMBP terminology is illustrated showing the central pixel (b_8), associated image patch for creating *binary patterns* or *textons* (which is always 3×3), and the adaptive *analysis window* around the central pixel for which statistics such as the median can be computed depending on the local texture microstructure (see Algorithm 1). The median for MBP always comes from the image patch but the AMBP median can come from anywhere within the larger analysis window including the image patch.

Algorithm 1 Adaptive Median Binary Patterns (AMBP)

Input: Gray scale Image I; maximum analysis window k_{max}; patch size $L = 9$
Output: Adaptive median binary pattern image M
for all i, j **do**
  $k \leftarrow 1$
  repeat
    $S \leftarrow I[i-k:i+k, j-k:j+k]$
    $Z_{med} \leftarrow$ median(S)
    $Z_{min} \leftarrow$ min(S)
    $Z_{max} \leftarrow$ max(S)
    if $Z_{min} < Z_{med} < Z_{max}$ **then break**
    $k \leftarrow k + 1$
  until $k \leq k_{max}$
  if $Z_{min} < I[i,j] < Z_{max}$ **then**
    $\tau = I[i,j]$
  else
    $\tau = Z_{med}$
  end if
  $M[i,j] \leftarrow L$-bit binary pattern using τ and Eq. 2
end for

where $i = 0, 1, 2, \cdots, \lfloor M/2 \rfloor$, $j = 0, 1, 2, \cdots, \lfloor N/2 \rfloor$, and $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3, \mathcal{I}_4$ are a set of four subimages each at half-resolution covering all four-phases. The four-phase images at each resolution helps to capture microstructure relationships between pixels that are not immediate neighbors in a scale independent fashion.

A histogram of AMBPs or MBPs is computed over the whole image, at each resolution and for each phase, using the appropriate number of bins to handle all possible textons. The histograms for the four-phases are combined into a single histogram at each level (using an order-statistic). The optimal number of resolutions or pyramid levels is application dependent and is empirically chosen as $d = 3$ levels, based on a trade-off between complexity and performance. The rotational and uniform pattern groupings for MBP and AMBP follow the same definitions as used in LBP.

IV. Joint information

The local scale binary hash patterns convey valuable information about the nature of texture since they encode information about the micro-structure variations in an image. But texture cues are more complex involving larger scale interactions. This more global discriminative information is missed by local binary pattern operators. In order to improve classification performance the distribution of hash patterns is often combined with additional information such as variance, using joint distributions [5]. Recently, Guo *et al.* [14] combined LBP with the magnitude of the local differences using an 8-bit binary pattern generated by using a global mean difference threshold, and a one-bit pattern based on comparison with the global mean intensity. The joint histogram of these three components showed better performance than the LBP combined with variance. In order to increase the AMBP and MBP discriminative properties we use the signed magnitude and global thresholding. The signed magnitude is similar to the LBP operator but instead of using intensities it applies the LBP operator to local differences:

$$M_{P,R} = \sum_{p=0}^{P-1} 2^p H(m_p - c) \tag{4}$$

where m_p is the magnitude of the local difference between the central pixel and its neighbor p. The threshold c is taken as the mean of m_p over the full texture *i.e.* whole image. The same descriptor coding scheme as LBP rotation invariant uniform patterns is adopted for $M_{P,R}$. In our tests we combine AMBP (or MBP) with $M_{1,8}$ resulting in 10 uniform patterns for $M_{P,R}$. Our aim is to have small descriptors by keeping only the essential information. For capturing multiscale information we use our scheme (see Section III). Let AMBP,*l*/M/C or MBP,*l*/M/C be the joint histogram of AMBP or MBP uniform patterns (riu), the signed magnitude of local differences encoded as uniform patterns and the central pixel having a binary value obtained by global thresholding against the mean value of the image gray levels. AMBP or MBP is used with rotationally invariant uniform patterns and l indicates the scale. The descriptor includes one bin for non-uniform patterns in the AMBP or MBP histogram which results in 19 bins for AMBP and 10 bins for MBP. For instance AMBP,1/M/C joint histogram contains $19 \times 10 \times 2 = 380$ bins.

V. Results and discussion

Experiments were conducted using two texture databases CUReT and KTH_TIPS2-b. The k nearest neighbor (k-NN fixed to one) classifier was trained using a subset of textures from each class, then tested for recognition accuracy of the same class. The k-NN is used with the χ^2 distance:

$$\chi^2(\mathbf{u}, \mathbf{v}) = \frac{1}{2} \sum_i \frac{(u_i - v_i)^2}{u_i + v_i} \tag{5}$$

where $\mathbf{u}$ and $\mathbf{v}$ are feature descriptor vectors. The general approach we used was to compare the simple schemes of AMBP, MBP and LBP first since these methods differ from each other in the thresholding method. In the second step we use the joint AMBP and MBP as described in Section IV to evaluate our method with recent state-of-the art operators like VZ_MR8 [7], DLBP [13] and CLBP [14]. The Gabor filter

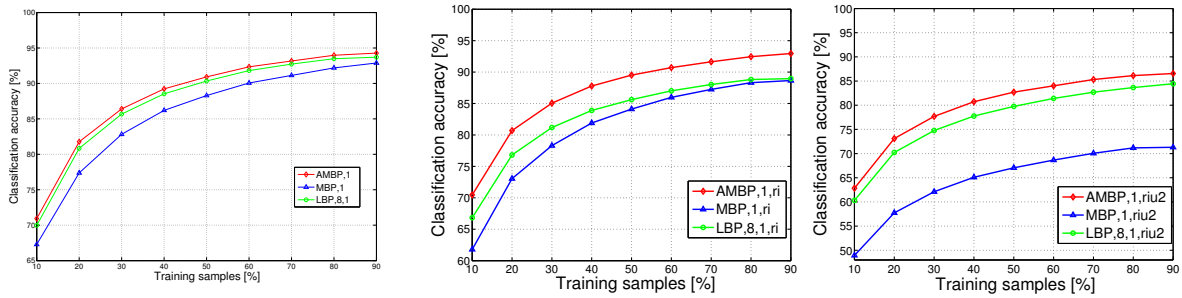

Fig. 4: Comparison of performance versus percentage of training samples used for the CUReT database. Each classification rate estimate is based on an average of 100 repetitions of randomly selected percentage of training samples for each class. a) (Left) Full descriptor AMBP, MBP and LBP. b) (Middle) Rotationally invariant descriptors. c) (Right) Rotationally invariant and uniform descriptors.

is also evaluated as a standard method. The rotational and uniform patterns were used for MBP and AMBP.

A. CUReT Texture Dataset

The Columbia-Utrecht (CUReT) database [25] captures 3D views of textures. It covers 61 different material surfaces, each one observed with 205 combinations of viewing and illumination directions. In general only 92 samples per class are used for the classification task [26]. The samples are fixed 200×200 pixel cropped subimages from the original. We evaluate the performance of AMBP and related methods using different amounts of training samples ranging from 10% to 90% (see Figure 4). At each percentage, we repeat the random selection of texture samples 100 times which gives 100 training/testing subsets for each class. For instance, using 10% training data produces 9 labeled samples and 81 unlabeled samples for a given class. Figure 4 shows the classification rate averaged over all classes for AMBP, MBP and LBP methods, versus the percentage of training samples. The AMBP technique shows better classification accuracy than the MBP and LBP methods. The rotationally invariant patterns provide good classification accuracy with a descriptor size that is much smaller than the full descriptor. This shows a good tradeoff between computational complexity and classification performance. After evaluating the proposed method using the simple scheme we compare it to several methods in the literature, using joint histogram information with AMBP or MBP (*i.e.* AMBP,*l*/M/C, MBP,*l*/M/C). Table I shows the global classification performance. We observe that incorporating the joint histogram scheme increases the discriminative capabilities of AMBP outperforming CLBP which uses the same scheme with LBP. AMBP outperforms VZ_MR8 in our experimental setup. Slightly higher scores have been reported for VZ_MR8 with different experimental parameters; however, it is beyond the scope of this work to test all combinations for VZ_MR8. Overall the accuracy of the proposed AMBP texture operator is competitive with the most successful techniques.

B. Noisy Textures

In many applications including imaging surface textures, noise is inherent in the imaging system due to device sensitivity, lighting conditions, exposure settings, etc. Consequently, the texture recognition and classification tasks become tougher. In general, images may be corrupted by three types of noise, including additive, multiplicative or impulse noise (salt and pepper noise). Such types of noise affect both the local

Method	Percentage of training samples				
	10%	30%	50%	70%	90%
AMBP,1/M/C	77.65	91.48	95.12	96.52	97.23
AMBP,2/M/C	**80.33**	**92.94**	**96.26**	**97.48**	**98.21**
MBP,1/M/C	74.93	89.62	93.86	95.71	96.58
MBP,2/M/C	78.07	91.52	95.09	96.60	97.52
CLBP_S/M/C	78.77	91.99	95.38	96.79	97.67
DLBP	65.60	79.25	84.04	86.75	88.23
VZ_MR8	78.13	91.96	95.67	97.30	98.09
GABOR	49.81	67.47	75.38	80.39	82.17

TABLE I: Comparison of AMBP and MBP with four well regarded texture operators versus different percent of training samples using CUReT. Each classification rate is the average of 100 trials of randomly selected samples.

microstructures and the global visual properties decreasing separability of texture classes, particularly among fine textures. An example of impulse noise added to a texture is shown in Figure 5 illustrates the impact of noise density on the perceptual visual quality of the texture. In this work we focus on impulse noise due to its relationship with the median filter. We carried out a series of experiments by synthetically adding noise to the CUReT dataset with the noise ratio increasing from 2% to 20% in increments of 2%. The texture classification performance was evaluated for each noise density. The training set includes the original CUReT images and the test set contains the same images but with noise added. Each set incorporates 5612 sample with 92 images per class.

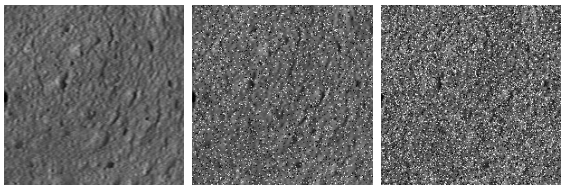

Fig. 5: Example of impulse or salt & pepper noise in a CUReT sample. a) (Left) Original image. b) (Middle) Image corrupted with 10% density impulse noise. c) (Right) Image corrupted with higher 20% density noise.

Figure 6 shows the classification performance of AMBP, MBP, VZ_MR8, CLBP, DLBP and GABOR versus the amount of impulse noise. The graphs show the failure of well known methods to maintain good classification accuracy even for small amounts of noise. Despite the filtering process utilized by VZ_MR8 or GABOR, it is not sufficient to handle the impulse noise. The CLBP and DLBP methods also cannot handle noisy textures, because they use thresholding based on the central pixel. These methods lead to saturation of the center pixel

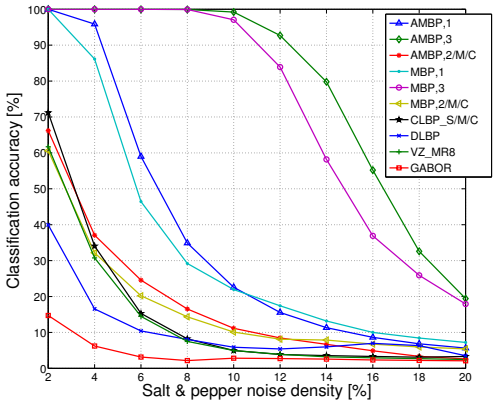

Fig. 6: Effects of impulse noise (*salt-and-pepper*) on classification accuracy for the CUReT database. The training set is noise free and the test set uses the same images, but with varying amounts of impulse noise added.

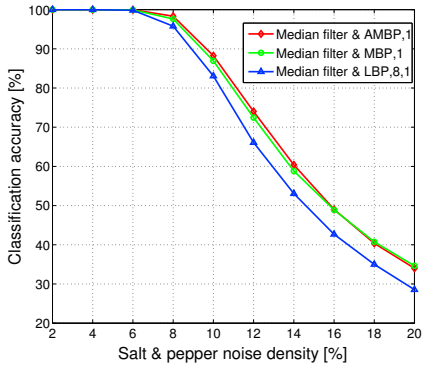

Fig. 7: Classification performance after preprocessing the CUReT dataset (impulse noise) using 3×3 median pre-filtering.

for noisy textures resulting in extreme minimum or maximum values in the LBP schemes. So the LBP output will be quite different from the noise-free training data.

Pattern differences due to noisy pixels influence the bin frequency in the histogram and reduce the percent of uniform patterns [17]. Moreover, the noise affects different pixel locations in different images since the noise is randomly distributed and independent. This modifies the texture descriptor histograms which distorts the similarity measure and decreases the likelihood that the LBP operator will fall in the correct class. However, the MBP-based methods show a greater robustness to noise. They start at a high level of accuracy with less noise, then the performance decreases with each additional 2 percent increase. From 2% to 8% noise the AMBP and MBP with three scales show perfect results (100%). AMBP,3 maintains high accuracy until 14%, beyond which the classification rate decreases significantly as the noise level increases to 20%. At this level of impulse noise, outliers are denser in the local neighborhood. The AMBP tries to enlarge the window size to find a better median, but the number of noisy pixels involved skews the statistics which increases the probability of estimating a biased median value. Using only one scale leads to a rapid decrease in performance, whereas using three scales the performance curves show a more linear decrease, more consistent results and robustness against noise.

In the presence of noise the additional information using joint histograms (AMBP,l/M/C, MBP,l/M/C) for classification does not appear to improve the discriminative properties and drastically decreases their robustness as shown in Figure 6.

These experiments demonstrate the capabilities of MBP and AMBP to remove impulse noise without any pre-filtering. We compare this to a more traditional approach of adding anise reduction pre-filtering step prior to binary pattern extraction. A 3×3 median filter is applied to the whole CUReT database (noisy and non-noisy images), then we compute LBP, MBP and AMBP. As in previous experiments the training set is the noise free images and the testing set are the noisy ones. The results shown in Figure 7 indicate an overall improvement in performance, particularly for LBP, compared to the results in Figure 6. But the AMBP and MBP still outperform LBP, demonstrating the same tendency as in the non-noisy case.

C. KTH_TIPS2-b Texture Dataset

The major drawback of the CUReT database is that materials are imaged at a constant scale. In contrast, KTH_TIPS2-b database [27] images objects at several distances from the camera. It includes four physical samples of 11 different materials similar to those used in the CUReT database. These samples were imaged with variations in scale as well as variations in pose and illumination. Images were taken across a combination of three poses, four illuminations and nine scales yielding 108 images for each physical sample. The images were cropped to a size of 200×200 pixels and the four physical samples are categorized into one texture class. The resulting database contains 4752 images ($11 \times 108 \times 4$).

Unlike the previous experiments there is no random selection of the training sets. Each set of physical sample (108 images) can represent a testing or training set. The goal is to learn images from given physical material, under different imaging conditions, in order to recognize other images of the same kind of material. For that purpose we use all possible combinations of viewing parameters. A k-fold cross-validation method is applied to compute the classification rate of the selected sets. In our experiments we use 4-fold and 6-fold combinations, where each fold can contain one or several sets. The number of experiments is determined by the number of possible combinations. The classification performance was measured by computing the average classification accuracy for each configuration. We also evaluated the performance of the proposed texture operators in comparison to recent methods from the literature. In Table II we observe that the highest scores are obtained by AMBP,2/M/C followed by MBP,2/M/C and CLBP. VZ_MR8 does not show good performance on this more challenging dataset compared to CUReT and Gabor filter operator has the worst performance on both datasets.

VI. Conclusions

The Adaptive Median Binary Pattern (AMBP) is the first local binary pattern operator to automatically fuse information from LBP and MBP in a context aware fashion providing noise resilience, to better encode the scene microstructure information. The AMBP uses either the local median or the central pixel to generate local patterns taking local scene context and cues into account. The hybrid AMBP fusion approach is simple, robust to noise and computationally efficient

Method	Number of training sets		
	1	2	3
AMBP,3	54.25	61.31	64.69
MBP,3	53.14	59.83	63.51
LBP,3	52.95	59.36	62.12
MBP,1/M/C	57.46	62.56	64.90
MBP,2/M/C	57.69	62.91	65.87
AMBP,1/M/C	57.88	62.75	65.19
AMBP,2/M/C	**59.06**	**64.10**	**69.44**
CLBP_S/M/C	55.40	62.69	68.18
DLBP	49.31	55.08	58.04
VZ_MR8	45.69	50.22	52.34
GABOR	44.96	48.52	50.11

TABLE II: Comparison of different methods and training sets KTH_TIPS2-b.

for realtime implementation. We evaluated the AMBP texture descriptors over a wide range of textures using the popular CUReT and KTH TIPS2-b databases and characterized the influence of training sample size, and the effects of changes in scale, pose, lighting and noise on recognition accuracy. The AMBP is competitive with the best results on CUReT and significantly better on the much more challenging KTH_TIPS2-b where performance improves by almost 5 percent with each additional training set indicating that with even more training data accuracy could reach above 70 percent. In the case of noisy textures, the hybrid AMBP descriptor outperformed the LBP or MBP only descriptors as well as other state-of-the-art methods. Pre-filtering textures using the median filter increased accuracy compared to no filtering, but was still significantly lower than AMBP fusion without pre-filtering. The Gabor texture operator performed the worst dropping below 10% accuracy even for a small amount of impulse noise ($\leq$ 4%), whereas AMBP performance was above 90% accuracy even with 12% noise (for CUReT). The AMBP approach introduces two new mechanisms for developing novel texture feature analysis methods, including self-adaptive window sizes and automatically selecting the best local texture operator. The principal motivation to consider median based thresholding in MBP, instead of local thresholding (current pixel) in LBP was guided by the classic results that favor median filters versus mean filters in certain noise processes. The adaptive behavior that we seek to integrate by increasing the spatial neighborhood under consideration inherently brings spatial smoothness into consideration. This type of fusion and adaptation mechanisms can be further generalized and extended to other classes of texture operators in a similar fashion opening up new directions for combining texture measures.

REFERENCES

[1] M. Tuceryan and A. Jain, "Texture analysis," in *Handbook of Pattern Recognition and Computer Vision*, 2nd ed. World Scientific Publishing, 1998, pp. 207–248.

[2] M. Petrou and P. G. Sevilla, *Image Processing: Dealing with Texture*. Wiley, 2006.

[3] K. J. Dana and S. K. Nayar, "Correlation model for 3D texture," in *Int. Conf. Computer Vision*, 1999, pp. 1061–1066.

[4] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *Int. J. Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[5] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[6] O. G. Cula and K. J. Dana, "3D texture recognition using bidirectional feature histograms," *Int. J. Computer Vision*, vol. 59, no. 1, pp. 33–60, 2004.

[7] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. J. Computer Vision*, vol. 62, no. 1-2, pp. 61–81, 2005.

[8] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278, Aug 2005.

[9] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032–2047, Nov 2009.

[10] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local Gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition," in *Int. Conf. Computer Vision*, vol. 1, 2005, pp. 786–791.

[11] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.

[12] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, 2007.

[13] S. Liao, M. W. K. Law, and A. C. S. Chung, "Dominant local binary patterns for texture classification," *IEEE Trans. Image Processing*, vol. 18, no. 5, pp. 1107–1118, May 2009.

[14] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. Image Processing*, vol. 19, no. 6, pp. 1657–1663, Jun 2010.

[15] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, and W. Gao, "WLD: A robust local image descriptor," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1705–1720, Sep 2010.

[16] E. Keramidas, D. Iakovidis, and D. Maroulis, "Fuzzy binary patterns for uncertainty-aware texture representation," *Electronic Letters on Computer Vision and Image Analysis*, vol. 10, no. 1, 2011.

[17] J. Chen, V. Kellokumpu, G. Zhao, and M. Pietikäinen, "RLBP: Robust local binary pattern," in *Proc. British Machine Vision Conference*, 2013.

[18] A. Hafiane, G. Seetharaman, and B. Zavidovique, "Median binary pattern for textures classification," in *Lecture Notes in Computer Science (ICIAR)*, vol. 4633, 2007, pp. 387–398.

[19] A. Hafiane, G. Seetharaman, K. Palaniappan, and B. Zavidovique, "Rotationally invariant hashing of median patterns for texture classification," *Lecture Notes in Computer Science (ICIAR)*, vol. 5112, pp. 619–629, 2008.

[20] K. Palaniappan and *et al.*, "Efficient feature extraction and likelihood fusion for vehicle tracking in low frame rate airborne video," in *13th Conf. Information Fusion*, 2010, pp. 1–8.

[21] K. Palaniappan, R. Rao, and G. Seetharaman, "Wide-area persistent airborne video: Architecture and challenges," in *Distributed Video Sensor Networks: Research Challenges and Future Directions*, B. Banhu and *et al.*, Eds. Springer, 2011, ch. 24, pp. 349–371.

[22] A. Fernández, M. X. Álvarez, and F. Bianconi, "Texture description through histograms of equivalent patterns," *Journal of Mathematical Imaging and Vision*, vol. 45, no. 1, pp. 76–102, 2013.

[23] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, 3rd ed. Prentice Hall, 2008.

[24] T. Ojala, M. Pietikäinen, 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.

[25] K. Dana, B. Van-Ginneken, S. Nayar, and J. Koenderink, "Reflectance and texture of real world surfaces," *ACM Transactions on Graphics*, vol. 18, no. 1, pp. 1–34, Jan 1999. [Online]. Available: http://www.cs.columbia.edu/CAVE/software/CUReT/

[26] M. Varma, A. Zisserman, and J. M. Geusebroek. Texture classification benchmark using cropped images from the CUReT database. [Online]. Available: http://www.robots.ox.ac.uk/ vgg/research/texclass

[27] P. Mallikarjuna, M. Fritz, A. T. Targhi, E. Hayman, B. Caputo, and J. Eklundh. The KTH-TIPS and KTH-TIPS2 databases. [Online]. Available: http://www.nada.kth.se/cvap/databases/kth-tips