Pattern Recognition 48 (2015) 2609–2620

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Joint Adaptive Median Binary Patterns for texture classification

Adel Hafiane [a,*], Kannappan Palaniappan [b], Guna Seetharaman [c]

[a] INSA Centre Val de Loire, Université d'Orléans, Laboratoire PRISME, EA 4929, F-18000, Bourges, France
[b] Department of Computer Science, University of Missouri, Columbia, MO 65211, USA
[c] Information Directorate, Air Force Research Laboratory, Rome, NY 13441, USA

ARTICLE INFO

Article history:
Received 10 November 2013
Received in revised form
28 October 2014
Accepted 12 February 2015
Available online 20 February 2015

Keywords:
Adaptive median
Local Binary Pattern
Impulse noise
Rotation invariance
Multiscale
Texture classification

ABSTRACT

This paper addresses the challenging problem of the recognition and classification of textured surfaces given a single instance acquired under unknown pose, scale and illumination conditions. We propose a novel texture descriptor, the Adaptive Median Binary Pattern (AMBP) based on an adaptive analysis window of local patterns. The principal idea of the AMBP is to convert a small local image patch to a binary pattern using adaptive threshold selection that switches between the central pixel value as used in the Local Binary Pattern (LBP) and the median as in Median Binary Pattern (MBP), but within a variable sized analysis window depending on the local microstructure of the texture. The variability of the local adaptive window is included as joint information to increase the discriminative properties. A new multiscale scheme is also proposed in this paper to handle the texture resolution problem. AMBP is evaluated in relation to other recent binary pattern techniques and many other texture analysis methods on three large texture corpora with and without noise added, CUReT, Outex_TC00012 and KTH_TIPS2. Generally, the proposed method performs better than the best state-of-the-art techniques in the noiseless case and significantly outperforms all of them in the presence of impulse noise.

1. Introduction

Texture is universally present in natural objects and manufactured materials. In computer vision and pattern recognition, it provides an important cue as it conveys physical information about the characteristics of objects and surfaces. Although many types of texture features have been proposed [1,2], there is still no unique mathematical definition of texture that is consistent with the perceptual properties of the human visual system. One recent class of powerful texture measures that are widely applied is known as Local Binary Patterns (LBPs) [3]. In this paper we focus on this class of binary pattern or texton-based texture descriptors and their variations to investigate performance across a wide variety of scene contents. The current methods based on LBP tend to focus more on additional features to increase the discriminative properties, employing supplementary descriptors besides the LBP such as Gabor filters and gradient orientations [4–6]. However, few studies focus on the local pattern properties and the noise effect. We have proposed a scheme based on LBP that explores the order of the local pixel values using the median approach, called Median Binary Pattern (MBP) [7]. 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 [8]. 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 Pattern (AMBP) operator that uses local adaptive analysis in the process of extracting binary patterns or textons. The AMBP uses the principle put forward for adaptive median filtering [9] to preserve image detail even in the presence of high levels of noise by varying the size of the local median window. In a similar manner, AMBP adaptively changes the analysis window size in a spatially varying manner, to obtain a better threshold depending on the local context. AMBP produces either the central pixel or the median value as threshold. The first case yields LBP whereas the second one yields MBP. In fact, the AMBP histogram can combine both LBP and MBP depending on the local structures and noise. The interest of such a process is to capture significant patterns by adapting the analysis window to the local context. Moreover, we propose a new scheme to incorporate the joint information with AMBP using the adaptive window variations and global thresholding. Our contribution can be summarized as follows:

- Adaptive selection of the type of pattern.
- Joint information with the adaptive neighborhood variation.
- Efficient multiresolution descriptor that combines the circular and pyramidal schemes.

* Corresponding author. Tel.: +33 248 484 078.
E-mail address: adel.hafiane@insa-cvl.fr (A. Hafiane).

http://dx.doi.org/10.1016/j.patcog.2015.02.007
0031-3203/

The paper is organized as follows: the next section presents the related literature. Section 3 gives a brief recap of LBP and MBP, then describes the AMBP and the scheme employed for rotation invariants. Section 4 details the proposed joint AMBP method. Section 5 presents the multiscale technique. The experimental results are given in Section 6, and the conclusions in Section 7.

2. Related work

Before presenting our method, we briefly discuss a number of texture models designed to achieve texture classification under various geometric transformations. Early approaches tried to solve the invariant features using statistical, structural or spectral texture measures [10–15]. These models are good for simple classification tasks, but they are unable to solve the general problems of natural material representation and recognition under varying lighting and viewing conditions. The subsequent generation of texture classification methods over the last decade resulted in significant improvements in texture analysis. These methods are essentially based on extracting local primitives or textons and measuring their distribution using histograms [3,16–20]. Among the most successful techniques, we can distinguish two main ones, both based on the local analysis of the texture elements: the Bag of Words (BoW) and Local Binary Pattern (LBP).

The BoW employs a learning phase to build the descriptor. First local responses to a bank of filters or neighborhood properties are used to generate a feature vector in each location of the training images. Then vectors are clustered to build a texton dictionary model. This model is then used to build the descriptor for each image. Various schemes have been developed with BoW, Leung and Malik [17] used local geometric and photometric properties to encode the local structures called 3D textons, obtained with a bank of filters and *k*-means algorithm to build a texton vocabulary. Cula and Dana [18] modeled the texture as a function of viewing and illumination directions represented by a histogram that encodes the statistical distribution of local structures. Varma and Zisserman [19] developed a statistical model using a bank of rotationally invariant filters (MR8) to create a texton library encoded with histograms; in recent work the same authors [21] argued that the bank of filters is not necessary to generate textons and that it can be replaced by image patches of the local neighborhood. In a similar manner, Liu and Fieguth [22] used compressed sensing and the random projection of local patches to increase the descriptor's ability to separate classes. These studies are based on dense representation (i.e. computing features in each pixel). Another approach is to use sparse representation, such as the one developed by Lazebnik et al. [20], using affine invariant descriptors extracted from affine Harris and Laplacian regions in the image. The results of these methods are very satisfactory for the tested databases, but the performance drops for different applications; moreover, they are computationally expensive.

On the other hand, most Local Binary Pattern based approaches do not use learning to build the texture descriptor, which makes them attractive in terms of computational complexity and efficiency. Despite the great success of the original LBP in many applications, it also has several limitations, like BoW methods. Numerous methods that fall into the LBP category have been adapted to new situations in different applications. Zhang et al. [23] combined Gabor filters and LBP to achieve a better performance for face recognition. Heikkilä and Pietikäinen [24] used adaptive Local Binary Pattern histograms to model a video scene background and demonstrated better performances compared to the state-of-the-art. Zhao and Pietikäinen [25] extended LBP to temporal aspects, showing interesting properties for dynamic textures. Liao et al. [4] extracted the most frequent patterns in LBP histograms to detect the dominant patterns that form the new descriptor, and also used Gabor filter as a second feature to enforce the classification task. Chen et al. [6] also improved performance by using histograms of relative differences and of orientations using the concept of the Weber Law. The two histograms are concatenated to form the texture descriptor. Qian et al. [26] combined the pyramidal and LBP multiscale schemes to increase the classification accuracy. While these LBP variants have improved the performances, they are not sufficient for emerging applications. Recent work attempts to use a learning phase to build a descriptor based on the LBP approach; Guo et al. [27] proposed to incorporate the learning phase with LBP using dominant patterns, and achieved better results with this technique. However, the learning phase increases the time complexity of the descriptor which is one of the drawbacks of BoW techniques. Another type of LBP based method uses joint information, aiming to increase the discriminative properties of the LBP. For instance, Guo et al. [5] extended the LBP scheme by adding the signs and magnitudes of differences between local pixels, which were all encoded with a joint histogram, and achieved good performances. It is clear that including joint information is an important strategy to increase the classification rate, as demonstrated by Crosier and Griffin [28], where the joint information with simple features have improved drastically the performance for texture classification.

Despite the great success of the various approaches proposed particularly the LBPs in the standard databases, there are several limitations with these kinds of methods. Textures present complex local structures in which the local patterns are not necessarily regular in terms of size, orientation, etc. Generally the operators employ a fixed neighborhood size over the image to extract local structure. This approach may fail to handle the variability of all the texture elements. In this work we use the adaptive analysis of local pattern with joint information as a new approach for the texture classification problem, aiming for a better representation of the local structure.

3. Texture analysis using binary patterns

3.1. Local Binary Pattern

The Local Binary Pattern (LBP) [29,3] is a widely used texture descriptor that has an excellent performance with highly discriminative classification behavior for a variety of applications. The computation of the LBP descriptor consists of two major steps: first, extracting binary patterns (i.e. textons or texels) of spatially local elementary microstructures in the texture image; and, second, analyzing the histogram distribution of these textons. The first step uses the central pixel of a small patch as the threshold value to capture the sign of the differences with neighboring pixels and is illustrated in Fig. 1. More formally the LBP at pixel (i,j) is defined as [3]

$$LBP_{P,R}(i,j) = \sum_{k \in \mathcal{N}_P(i,j)} 2^k H(x_k - x_c); \quad H(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where x_c is the pixel value at the coordinates (i,j) and $\mathcal{N}_P(i,j)$ is the set of the neighborhood at the position (i,j). x_k is the pixel gray value at

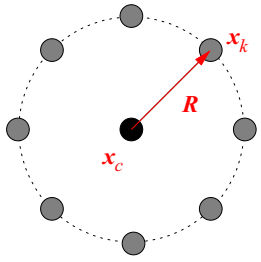

Fig. 1. The LBP circular neighborhood scheme.

position index k which is a simpler lexicographical ordering of the pixel locations in the circular neighborhood. Its coordinates are estimated by $(R\sin(2\pi k/|\mathcal{N}_P|), R\cos(2\pi k/|\mathcal{N}_P|))$, where $|\mathcal{N}_P|$ is the cardinality of $\mathcal{N}_P$ (i.e. the number of neighbors). $H(\cdot)$ is the Heaviside (discrete) unit step function also referred to as a binary thresholding function. The gray values of the neighbors, that are not in the center of the square grid, are computed by interpolation. The resulting pattern is captured as a $|\mathcal{N}_P|$−bit binary number representing $2^{|\mathcal{N}_P|}$ distinct binary patterns. The histogram of these binary patterns in the transformed domain is computed for the transformed image and treated as a texture-descriptor.

3.2. Median Binary Pattern

The Median Binary Pattern (MBP) introduced by us in [30,7] is similar to the LBP but uses the median within the image patch as the local threshold instead of using the central pixel to provide more sensitivity to microstructure and greater noise robustness. In general form MBP can be defined as

$$\mathrm{MBP}_{P,R}(i,j) = \sum_{k \in \mathcal{N}_P(i,j)} 2^k H(x_k - \tau(i,j)) \tag{2}$$

where $\tau(i,j)$ is the median within the image patch that is

$$\tau(i,j) = \mathrm{median}(\mathcal{N}_P(i,j)) \tag{3}$$

means that there will be at least $(|\mathcal{N}_P|)$ bits in the resulting binary pattern and only $2^{|\mathcal{N}_P|-1}$ possible binary patterns can occur. Since the thresholding is not necessarily against the central value (x_c), this value can be either considered or ignored in the binary pattern. In that case, the number of bits is $(|\mathcal{N}_P|+1)$ if the central pixel is included to the pattern or $(|\mathcal{N}_P|)$ if it is not. Fig. 2 depicts an example of the LBP and MBP pattern generation in a 3×3 simple scheme. It can be observed that the patterns are different from each other, which yields different descriptors.

3.3. Adaptive Median Binary Pattern

The median value associated with the center pixel depends on the set of pixels within the local (fixed size) filtering window. However, a small region may miss important information and a large one may yield a biased median value due to the large number of pixels involved. It is not easy to determine an optimal fixed size due to the change in the local context, in images, from one location to another. A constant size of analysis window cannot handle all the variations in the image. An adaptive window presents a good alternative since it can vary its analysis region depending on some criteria at each (central) pixel location. The Adaptive Median Binary Pattern (AMBP) considers larger analysis windows around the central pixel to compute the local median so as to more flexibly select the threshold gray value for converting from intensity space to binary patterns. In this case the resulting $|\mathcal{N}_P|$−bit binary pattern can take on the full range of $2^{|\mathcal{N}_P|}$ values since the median gray value may occur outside the pattern neighborhood. Fig. 3 shows an example of the 3×3 binary pattern image patch; the median can be anywhere within the analysis window.

The adaptive median filter incorporates signal adaptation and adaptive window sizes or neighborhoods for filtering positive and negative impulse noise while preserving local image details. Stage A checks if the median value in the neighborhood is an impulse: in this case, the window size is increased until it reaches the max window size or until the median value is not an impulse. In Stage B, it checks if the center pixel itself is an impulse or not. If neither the median nor the center pixel are impulses then it keeps the center pixel. However, this assumption might be not valid to detect local variation in non-noisy data where the extrema are not significant. In order to tackle this issue we use a sort of robust statistics by introducing a range of values instead of extrema. Let $S(i,j)$ be the square analysis window for given coordinates (i,j) and x_c is the pixel value at this location:

$$\Psi = \frac{1}{|S|}\sum_{m \in S(i,j)} |x_c - x_m|, \quad m \neq c \tag{4}$$

The testing condition of the median value becomes

$$Z_{min} + \alpha\Psi < Z_{med} < Z_{max} - \alpha\Psi \tag{5}$$

where α is a control parameter that determines the range value for the median, if $\alpha=0$ it turns out to the standard adaptive median filtering. For $\alpha \neq 0$, the median range is not necessarily between the extrema values; it is related also to the magnitude of the differences between the center pixel and its neighbors. Ψ is adaptive, as well, since it depends on the local context. Using this concept, the AMPB method is presented in Algorithm 1. There are two stages: the first one determines the optimal window size and the second one produces the local threshold value.

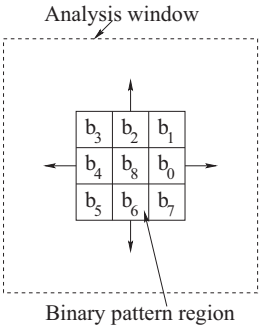

Fig. 3. Adaptive Median Binary Pattern terminology is illustrated showing the central pixel (b_8), its associated image patch for creating *binary patterns* or *textons* (which is always 3×3), and the adaptive *analysis window* around the central pixel for which statistics such as the median can be computed. The analysis window adapts to the local texture microstructure as described in Algorithm 1. The AMBP median can come from anywhere within the larger analysis window which includes the image patch but the median for the MBP operator always comes from the image patch.

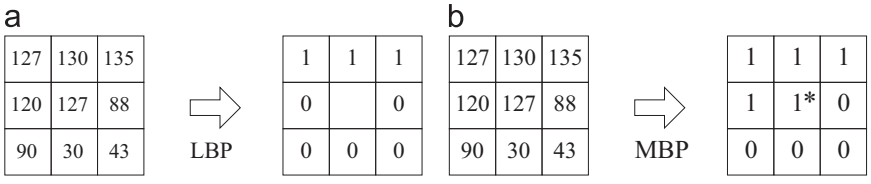

Fig. 2. (a) LBP operator applied to a single (central) pixel using its associated image patch (i.e. 3×3 neighborhood). (b) The MBP operator, (∗) the central pixel is optional; if used, the pattern is encoded with 9-bit, otherwise it is encoded with 8-bit.

Algorithm 1. Adaptive Median Binary Pattern.

Input: Gray scale Image $\mathcal{I}$; Maximum analysis window W_{max}; α; P and R; patterns layout.
Output: Adaptive Median Binary Pattern image $\mathcal{J}$
for all i,j **do**
  $w \leftarrow 1$
  repeat
    $S \leftarrow \mathcal{I}[i-w : i+w, j-w : j+w]$
    $Z_{med} \leftarrow$ median(S)
    $Z_{min} \leftarrow$ min(S)
    $Z_{max} \leftarrow$ max(S)
    if $Z_{min} + \alpha\Psi < Z_{med} < Z_{max} - \alpha\Psi$ **then**
      break
    $w \leftarrow w+1$
  until $w \leq \lfloor W_{max}/2 \rfloor$
  if $Z_{min} - \alpha\Psi < \mathcal{I}[i,j] < Z_{max} - \alpha\Psi$ **then**
    $\tau = \mathcal{I}[i,j]$
  else
    $\tau = Z_{med}$
  end if
  $\mathcal{J}[i,j] \leftarrow$ $|\mathcal{N}_P|$−bit binary pattern using τ and Eq. (2)
end for

Examples of the AMBP, MBP and LBP are illustrated in Fig. 4. The image (b) shows a 3×3 region around the central pixel. The regular LBP or MBP employs this neighborhood size, yielding the patterns shown in Fig. 4(c) and (d). In that case LBP does not match the visual pattern and provides flat output considered as a spot in LBP space. The MBP output produces a more consistent pattern but some parts are still missing. On the other hand, the AMBP increases the analysis region (Fig. 4(a)) to obtain better threshold since the central pixel has maximum value. AMBP method uses two types of threshold values: the median value provided that the central pixel does not lie in the range of values (*i.e* $[Z_{min} + \alpha\Psi, Z_{max} - \alpha\Psi]$), otherwise the central pixel is used as the threshold which is equivalent to LBP thresholding. As a matter of fact, the AMBP combines LBP and MBP, taking advantage of each one depending on the local context. For instance, if the central pixel is impulse noise the MBP is used, if it is not, the LBP is more likely to be used. To study the statistics of the Median vs. Central pixel thresholding, we computed the occurrence of the chosen threshold in the whole CUReT [31] database. Fig. 5 presents the mean percentage of the LBP and MBP over 5612 images. It can be clearly seen that the AMBP is a mixed binary pattern of LBP and MBP using the adaptive local context analysis. Note that the AMBP pattern scheme is realized with the same LBP or MBP scheme (circular neighborhood), whereas the analysis adaptive window has a square neighborhood. The AMBP descriptor consists in hashing the binary string at each pixel location into the corresponding decimal number, then the occurrence histogram of those numbers is computed over the entire image.

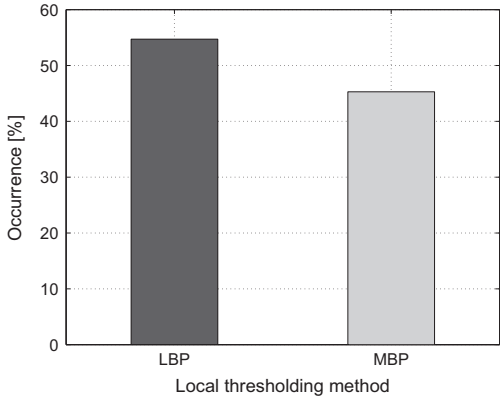

Fig. 5. AMBP is a mixed MBP/LBP based descriptor (Algorithm 1). The figure presents the statistics of (MBP) or center pixel (LBP) threshold occurrence over the CUReT database. The percentage indicates how many times median or central pixel values have been used.

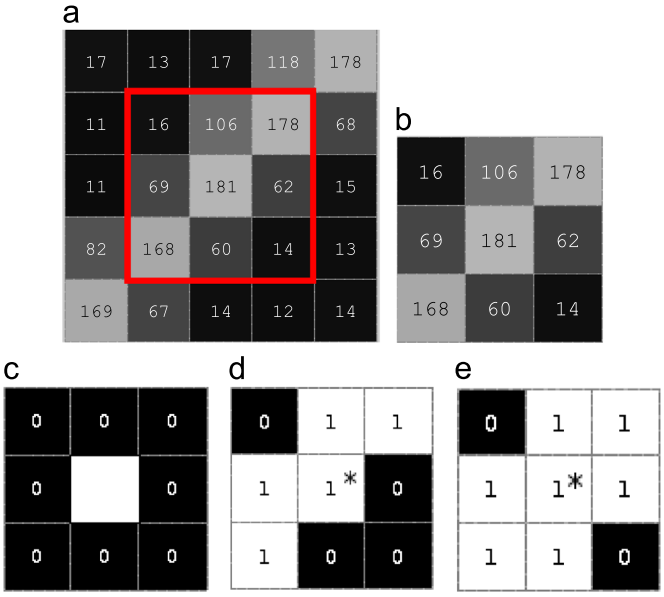

Fig. 4. An example showing how the local threshold value is selected for simple LBP, MBP and AMBP: (a) local 5×5 image region with the 3×3 image patch outlined in red; (b) texture descriptor or texton mapping for the central pixel in (a) uses the 3×3 image patch gray values around the central pixel; (c) LBP of the 3×3 image patch in (b) using the central pixel as the threshold value, with $\tau = 181$; (d) MBP of the image patch in (b) but using the patch median giving $\tau = 69$; (e) AMBP for image patch in (b), when the median from the larger 5×5 analysis window is selected giving $\tau = 60$. (*) For MBP and AMBP, the central pixel is optional and was not used in experiments. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

3.4. Equivalence pattern classes under rotation

We demonstrated that the MBP can be a rotationally invariant descriptor [7]. We extend the MBP-histogram to obtain rotational invariance by following the approach of Ojala et al. [3]. As the AMBP has the same LBP or MBP pattern properties, the rotationally invariant patterns can apply with the same concept to either the MBP or AMBP. When the image is rotated, the set of neighbors is geometrically altered, so the binary string pattern is modified and its value changes since the specific order of "1"s and "0"s is different. When an image is subject to a rigid rotation each local neighborhood undergoes nearly the same amount of rotation. The central pixel remains fixed in a relative sense, while the neighbors are circularly rotated. For example, the string `ABCDEFGHI` transforms to `ACDEFGHIB`, and `A` is the center of rotation that is

ROT : ("ABCDEFGHI"):="ACDEFGHIB"; and,

$\mathcal{L}^{ri}_{P,R}$: "ABCDEFGHI" ≡ "ACDEFGHIB"

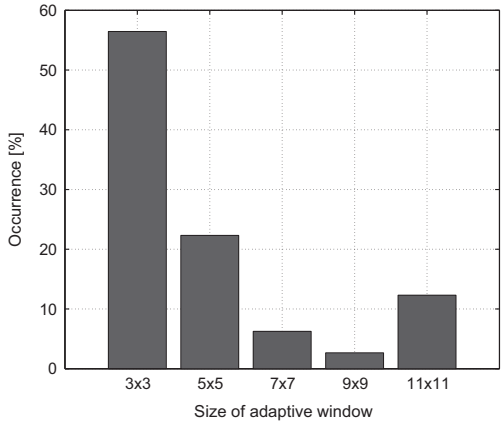

Fig. 6. AMBP statistics of the adaptive window variation in the CUReT database. The maximum size of the adaptive window $W_{max} = 11$, and the parameter $\alpha = 1.1$. The percentage indicates how many times a given window has been selected to produce the threshold for the binary pattern.

where $\mathcal{L}^{ri}_{P,R}$ is a circular shift function applied to the circular neighborhood with radius R and $|\mathcal{N}_P|$ samples, and the $\mathcal{L}$ operator is some labeling mechanism. The members of equivalent classes can be obtained by

$$\mathcal{L}^{ri}_{P,R}(i,j) = \min\left(\underset{0 \le b < |\mathcal{N}_P|}{\mathrm{ROT}}\left[\bigcup_{k \in \mathcal{N}_P(i,j)} H(x_k - x_c)\right]_b\right) \tag{6}$$

where $\bigcup$ is a concatenation operator that produces a binary string of the defined neighborhood scheme. The histogram entries at these indexes have a certain mutual impact. Likewise when the image is rotated by 45° the AMBP, MBP or LBP histograms are modified in a complex manner. However, the trace, defined as the sum of frequency of occurrence of all members within an equivalence group, remains invariant.

3.5. Rotationally invariant uniform patterns

Each group or set of patterns gives a different confidence, and each group has independent likelihoods of occurrence. There are some patterns that occur more frequently than others and using only these frequently occurring or *uniform patterns* was observed to improve the classification performance of rotated textures [3]. Uniform patterns are defined as those containing at most one transition in the binary string. A uniform pattern satisfies the condition

$$U(\mathcal{L}_{P,R}(i,j)) = |H(x_{|\mathcal{N}_P|-1} - x_c) - H(x_0 - x_c)| + \sum_{k=1}^{|\mathcal{N}_P|-1} |H(x_k - x_c) - H(x_{k-1} - x_c)| \tag{7}$$

The uniform scheme can be combined with rotationally invariant patterns as expressed by the following equation:

$$\mathcal{L}^{riu2}_{P,R}(i,j) = \begin{cases} \mathcal{L}^{ri}_{P,R}(i,j) & \text{if } U(\mathcal{L}^{ri}_{P,R}(i,j)) \le 2 \\ B_e & \text{otherwise} \end{cases} \tag{8}$$

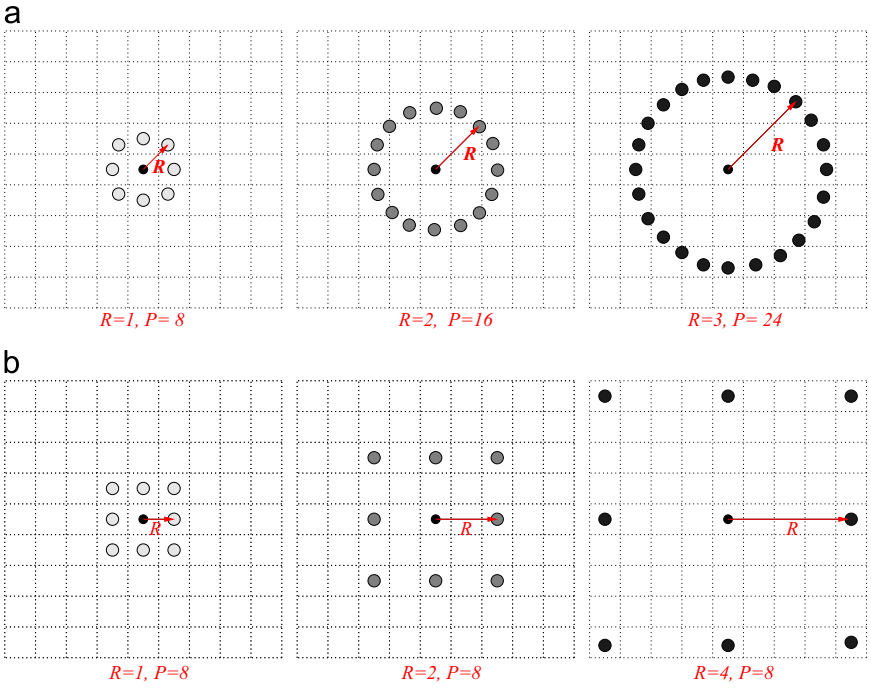

Fig. 7. (a) The circular neighborhood layout, for $R > 1$ the interpolation is required to compute the intensity of the pixels that do not lie in the regular grid. (b) The subsampling scheme for multiresolution does not require interpolation, the highlighted pixels are neighbors, while the other ones are ignored.

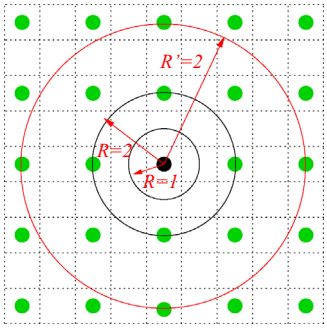

Fig. 8. An example of the proposed combined multiresolution scheme. The black center pixel represents the first resolution (original image), circular multiscale is used at that location (several rings, $R=1$ and $R=2$). The highlighted points (green) are obtained through subsampling, where only the last largest ring, from the previous scale, is used at these points to produce the binary pattern ($R'=2$). For the first pyramidal level the scheme is $(8,1)\cup(16,2)$, for the second one it is (16,2). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

where B_e is an extra "non-uniform" pattern. Note that for AMBP or MBP it is possible to use the central pixel information for the $\mathcal{L}^{riu2}_{P,R}$ operator, which doubles the number of patterns.

4. Joint AMBP

The Local Binary Pattern conveys valuable information about the nature of texture since it captures local structures. However, as the texture cue is very complex, the binary patterns cannot handle all the discriminative information. To achieve high performances the binary pattern is in general combined with different types of information such as the variance, using a joint distribution [3].

As described previously, the adaptive window changes size according to the local gray level distribution to select the appropriate threshold. One can examine the statistics of the window sizes over the whole database. Fig. 6 depicts the frequency of each window size on the CUReT database (i.e. the parameter $\alpha=1.1$). From these statistics it can be concluded that the window size changes can provide additional useful information about the local neighborhood. At each pixel we can consider which adaptive window has been selected. It can be encoded as joint information of the AMBP pattern. In other words, at each pixel we get a binary pattern and the window size $S(i,j)$ used to obtain this pattern (AMBP/W). The number of bins of W is computed as an index of the window size; $W=\lfloor S(i,j)/2\rfloor$, where $S(i,j)=3,5,\ldots,W_{max}$. For instance, if the maximum window size is set to 9×9, this results in four possible bins: $W=1$, 2, 3 and 4. The parameter W_{max} determines the number of bins which impact the size of the joint histogram. However, increasing this number does not necessarily provide the best performance. In the experimental part we show the influence of the variation of the W_{max} on the classification accuracy.

The joint histogram of the selected window helps us to enrich the descriptor with more discriminative features. Yet, the global information is also important for that purpose. In recent work, Guo et al. [5] combined LBP with the magnitude of the local differences encoded with binary pattern and global thresholding. The joint histogram of these three components showed nice properties demonstrating better performances than the LBP combined with local variances (i.e. LBP/VAR). To capture global information we also used global thresholding:

$$\Gamma(i,j)=\begin{cases}1 & \text{if } I(i,j)\geq\mu\\ 0 & \text{otherwise}\end{cases} \tag{9}$$

where μ is the global mean of the original image $I(i,j)$, that is $\mu=(1/MN)\sum_i^M\sum_j^N I(i,j)$. Here, Γ generates two bins that indicate at each pixel if it belongs to one of the two segmented regions. This information can also be encoded with the AMBP/W as a 3D joint histogram. Algorithm 2 summarizes the different steps to compute the joint AMBP/W/Γ. Note that the calculation of the joint 2D histogram AMBP/W is straightforward, by removing the variable z from Algorithm 2, which discards the Γ information.

Algorithm 2. Joint AMBP/W/Γ.

Input: Gray scale Image $\mathcal{I}$; Maximum analysis window W_{max}; α; P and R; patterns layout; Γ.
Output: Joint Histogram $\mathcal{H}$
initialize: $\mathcal{H}\leftarrow 0$
for all i,j **do**
  $x\leftarrow$ AMBP(i,j)
  $y\leftarrow W(i,j)$
$z\leftarrow\Gamma(i,j)$
$\mathcal{H}(x,y,z)\leftarrow\mathcal{H}(x,y,z)+1$
end for
$\mathcal{H}\leftarrow\frac{\mathcal{H}}{\|\mathcal{H}\|_1}$

5. Multiscale Binary Pattern

Multiscale information is important for texture classification, since the patterns can vary in different image resolutions. To capture this information LBP uses a circular scheme that combines patterns from different rings around the center pixel [3]. Fig. 7(a) depicts the LBP multiresolution layout. Then, it concatenates histograms of different values of (P, R), typically ($LBP_{8,1}\cup LBP_{16,2}\cup LBP_{24,3}$) for three scales. In [7], the multiscale MBP is utilized with a pyramidal subsampling scheme (see Fig 7(b)). Recently, the two approaches have been combined, showing better performances than the use of a standalone scheme [26]. It computes the LBP circular multiresolution at each pyramidal level. This can lead, however, to redundant information and scale sensitivity of the descriptor. To tackle this problem, we combine the circular and pyramidal schemes in a partial manner.

Let $\mathcal{M}_{P,R,L_n}$ be a Multiscale Binary Pattern for n pyramidal levels of resolution and $\mathcal{I}^n$ the image at the level n. $\mathcal{I}^0$ is the original image, then, the subsampling process can be described as follows:

$$\begin{aligned}&\mathcal{I}^n_1(i,j)=\mathcal{I}^{n-1}(2i,2j);\quad \mathcal{I}^n_2(i,j)=\mathcal{I}^{n-1}(2i+1,2j)\\ &\mathcal{I}^n_3(i,j)=\mathcal{I}^{n-1}(2i,2j+1);\quad \mathcal{I}^n_4(i,j)=\mathcal{I}^{n-1}(2i+1,2j+1)\end{aligned} \tag{10}$$

where $\mathcal{I}^n_1,\mathcal{I}^n_2,\mathcal{I}^n_3,\mathcal{I}^n_4$ are a set of four subimages each at half-resolution covering all four-phases. The four-phase images at each resolution help us to capture microstructure relationships between pixels that are not immediate neighbors in a scale independent fashion. In order to reduce the computation complexity only one phase ($\mathcal{I}^n_1$) was considered for the multi-scale scheme. Using the subsampling procedure, we can observe from Fig. 7, that further neighbors can be captured than with the circular scheme. As subsampling keeps the number of pixels involved at each level constant, the descriptor size remains constant over different scales. The goal is to take advantage of both mutliresolution schemes. One can use the circular multiscale layout at each subsampled points, but this introduces a great deal of redundancy, as some pixel locations are included in both schemes many times. For that reason, we propose to use only the last circular scale for the upper levels. This helps us to capture further pixels and reduces the number of those that are already included. That is,

$$\begin{aligned}\mathcal{M}_{P,R,L_n}=&\{\mathcal{L}_{P_1,R_1}(\mathcal{I}^0)\}\cup\{\mathcal{L}_{P_2,R_2}(\mathcal{I}^0)\}\cup\cdots\\ &\cup\{\mathcal{L}_{P_{max},R_{max}}(\mathcal{I}^0)\}\cup\{\mathcal{L}_{P_{max},R_{max}}(\mathcal{I}^1)\}_{n>0}\cup\\ &\cdots\cup\{\mathcal{L}_{P_{max},R_{max}}(\mathcal{I}^n)\}\end{aligned} \tag{11}$$

where $\mathcal{L}_{P,R}\{\mathcal{I}^i\}$ can be the AMBP, MBP, LBP or joint AMBP operator applied to an image at scale i. Fig. 8(a) illustrates an example of the use of two levels of pyramidal multiresolution; $P_{max}=16, R_{max}=2$ so, $\mathcal{M}_{P,R,L_1}=\{\mathcal{L}_{8,1}(\mathcal{I}^0)\}\cup\{\mathcal{L}_{16,2}(\mathcal{I}^0)\}\cup\{\mathcal{L}_{16,2}(\mathcal{I}^1)\}$. Note that, for $n=0$, $\mathcal{M}_{P,R,L_n}$ is equivalent to the $\mathcal{L}_{P,R}$ operator.

Algorithm 3. Multiscale Binary Pattern.

1: **Input**: Gray scale Image $\mathcal{I}$; $\mathcal{L}_{P,R}$ operator's parameter; number of levels (n).
2: **Output**: Histogram of Multiscale Binary Pattern
3: Compute for level 0
$H_0 \leftarrow \{\mathcal{L}_{P_1,R_1}(\mathcal{I}^0)\}\cup\cdots\cup\{\mathcal{L}_{P_{max},R_{max}}(\mathcal{I}^0)\}$
4: **for** each level $1<i<n$ **do**
5: Compute subsampling $\mathcal{I}^i$
6: Compute $\mathcal{H}_i \leftarrow \mathcal{L}_{P_{max},R_{max}}(\mathcal{I}^i)$
7: $\mathcal{H}_i \leftarrow \frac{\mathcal{H}_i}{\|\mathcal{H}_i\|_1}$
8: **end for**
9: $\mathcal{H}\leftarrow(\mathcal{H}_0,\cdots,\mathcal{H}_{n-1})$

The multiscale histogram computation is shown in Algorithm 3. Fig. 9 presents the general scheme for the multiresolution descriptor. The optimal number of pyramid levels is application dependent, and can be seen as based on a trade-off between complexity and performance. Increasing the number of levels or frequency subbands does not necessarily lead to improved texture classification performance. In the experimental part we show that for standard texture databases classification $n=1$ is the optimal value and for noisy textures $n=2$ showed better results.

6. Results and discussion

The experiments were conducted mainly on three databases: Outex_TC_00012 [32], CUReT [31] and KTH_TIPS2-(a,b) [33]. The k nearest neighbors classifier was trained with a subset of textures from each class, and its ability to recognize textures of the same class at different alterations was then evaluated. The k-NN is used with χ^2 distance defined as

$$\chi^2(\mathbf{x},\mathbf{y})=\frac{1}{2}\sum_i\frac{(x_i-y_i)^2}{x_i+y_i} \tag{12}$$

where $\mathbf{x}$ and $\mathbf{y}$ are feature vectors. The number of k-NN was fixed to one for all experiments.

The experiments and evaluation are divided into four main parts. The first part deals with the Outex database, using well defined testing and training samples. The second one tests the CUReT database, with random selection of the training and testing samples. The third measures the performances of the proposed method under the noise condition, where the testing samples are the noisy data and the training are the noiseless ones. The fourth part addresses the KTH_TIPS2-(a,b) database, using the already separate sets, which leads to a sort of K-folds cross validation procedure. Note that in all experiments, the color information is not used. Abbreviations and symbols used in the figures and tables are defined in Table 1. For all experiments, unless otherwise specified, the parameter $\alpha=1.1$ and $W_{max}=5$ which results in 2 bins. The central pixel is not encoded in AMBP and MBP, in order to use lower dimensional descriptors. The symbols ✓ and × mean used and not used respectively.

6.1. Experiments on Outex_TC_00012

The Outex database includes different test suites of material classification. Among these test suites, Outex_TC_00012 provides a popular benchmark for illumination and rotation invariance, particularly for LBP based methods. It consists of 24 texture materials and 20 images for each texture. Materials are imaged under nine different physical rotations (0°, 5°, 10°, 15°, 30°, 45°, 60°, 75°, and 90°). Three simulated illuminations were used for each texture, namely 2300 K horizon sunlight, 2856 K incandescent and 4000 K fluorescent tl84. The challenge is to train the classifier under 0° rotation and incandescent illumination, then classify correctly the other images. It is divided into two problems, '000' refers to nine rotations with tl84 and '001' refers to nine rotations with horizon light illumination.

The results of the proposed algorithm with different multiresolution schemes are listed in Table 2. We can observe that, in general, the AMBP alone improves the classification accuracy compared to the same LBP scheme. This proves that the combined thresholding (Central/Median pixel value) through the adaptive neighborhood is better than using either LBP or MBP. As we can observe, the proposed multiresolution scheme increases the scores of all the methods, where we can notice the difference between the resolution levels $n=0$ and $n=1$. The variation of the adaptive window is also important. It provides the information of the neighborhood statistics by analyzing the median and the range gray values. As a result, the joint histogram of AMBP and the selected neighborhood size (AMBP/W) significantly improves the performances compared to AMBP alone. Moreover, the 3D joint histogram with global thresholding (AMBP/W/Γ) further increases the classification rate. Finally, the best results were obtained with two levels of pyramidal resolution $n=1$ and the 3D AMBP joint histogram. Compared to the different state-of-the-art techniques including the joint histogram and learning based techniques, the proposed method yields the best results with a smaller number of features as shown in Table 3. Note that all the results referenced in Table 3 are reproduced from the cited papers.

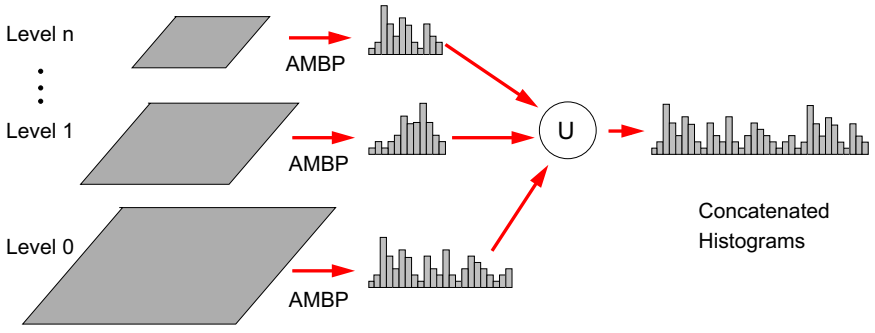

Fig. 9. The multiresolution scheme for the binary pattern. The figure shows an example where at each level an AMBP histogram is generated, then concatenated to form one descriptor. The sizes of descriptors in the upper levels are smaller than in the bottom level.

Table 1
Description of abbreviations of methods used in the comparison of performances and their descriptor sizes for n levels of pyramidal resolution. The case of $n=0$ the $\mathcal{M}$ operator is applied to the original image. T_i is the neighborhood size, for instance $\mathcal{M}^{riu2}_{(8,1)\cup(16,2),L_0} = \mathcal{L}^{riu2}_{8,1} \cup \mathcal{L}^{riu2}_{16,2}$, gives $T^{riu2}_1 + T^{riu2}_2$, where $T^{riu2}_1 = 10$ and $T^{riu2}_2 = 18$. As a result, for n levels, the descriptor size is $28 + n \cdot 18$.

Method	Description	Descriptor size
$\mathcal{M}_{P,R,L_n}$	Operator AMBP, MBP or LBP applied to 3×3 neighborhood	$(n+1)\cdot 256$
$\mathcal{M}^{ri}_{P,R,L_n}$	$\mathcal{M}_{P,R,L_n}$ with rotationally invariant scheme '*ri*'	$T^{ri}_1+\cdots+T^{ri}_M+n\cdot T^{ri}_M$
$\mathcal{M}^{riu2}_{P,R,L_n}$	$\mathcal{M}^{ri}_{P,R,L_n}$ and uniform patterns scheme '*riu2*'	$T^{riu2}_1+\cdots+T^{riu2}_M+n\cdot T^{riu2}_M$
$\text{AMBP}^{riu2}_{P,R,L_n}/\text{W}$	2D Joint histogram of $\text{AMBP}^{riu2}_{P,R}$ with W	$(T^{riu2}_1+\cdots+T^{riu2}_M+n\cdot T^{riu2}_M)\cdot W$
$\text{AMBP}^{riu2}_{P,R,L_n}/\text{W}/\Gamma$	3D Joint histogram of $\text{AMBP}^{riu2}_{P,R}$ with W and Γ	$(T^{riu2}_1+\cdots+T^{riu2}_M+n\cdot T^{riu2}_M)\cdot 2W$

Table 2
The classification accuracy in % of Joint AMBP, AMBP, MBP and LBP with different circular resolution schemes and 2 levels of pyramidal resolution (i.e. $n=1$). Descriptors were applied to Outex_TC_00012 texture suite, '000' corresponds to the tl84 and '001' to horizon light illumination. The bold values represent the best score of each category of result.

Method	(8,1)		(16,2)		(24,3)		(8, 1) ∪ (16, 2)		(8, 1) ∪ (16, 2) ∪ (24, 3)	
	000	001	000	001	000	001	000	001	000	001
$\text{LBP}^{riu2}_{P,R,L_0}$	67.5	64.2	81.7	74.7	83.9	79.1	84.2	79.7	88.8	84.0
$\text{MBP}^{riu2}_{P,R,L_0}$	53.8	55.8	82.7	76.6	85.8	83.3	83.2	79.1	89.7	84.1
$\text{AMBP}^{riu2}_{P,R,L_0}$	64.4	63.2	84.2	78.6	87.0	84.6	87.5	82.7	91.6	86.9
$\text{AMBP}^{riu2}_{P,R,L_0}/\text{W}$	74.7	75.5	88.3	86.8	92.1	90.0	91.1	90.1	94.9	91.6
$\text{AMBP}^{riu2}_{P,R,L_0}/\text{W}/\Gamma$	**87.3**	**91.0**	**94.3**	**93.0**	**95.6**	**94.0**	**95.3**	**94.7**	**96.9**	**95.3**
$\text{LBP}^{riu2}_{P,R,L_1}$	81.9	76.0	88.8	85.9	89.0	86.0	92.2	87.5	93.4	88.8
$\text{MBP}^{riu2}_{P,R,L_1}$	69.4	69.9	89.6	85.6	91.8	90.1	91.5	89.1	94.3	90.9
$\text{AMBP}^{riu2}_{P,R,L_1}$	81.0	77.5	89.1	86.5	91.5	90.1	92.0	89.4	93.0	91.3
$\text{AMBP}^{riu2}_{P,R,L_1}/\text{W}$	86.0	83.6	92.7	90.6	94.0	93.0	93.9	93.0	96.4	94.6
$\text{AMBP}^{riu2}_{P,R,L_1}/\text{W}/\Gamma$	**92.2**	**92.2**	**95.8**	**93.7**	**97.2**	**95.4**	**97.1**	**95.1**	**98.0**	**96.7**

6.1.1. The effect of the adaptive window maximum size

In this section we study the influence of the W_{max} parameter (see Algorithm 1). This parameter determines how many window sizes can be used in the local neighborhood and, therefore, the number of bins in the joint histogram. Fig. 10 shows the classification scores in the Outex_TC_00012 database for different values of W_{max}, and for $n=0,1$. As can be seen, the performance increases with different adaptive neighborhood sizes. From the size of 3×3 to 5×5 the improvement is 4% and the best performances were obtained with $W_{max}=5\times5$. This means that the optimal thresholds for producing a reliable binary patterns lie between two patches 3×3 and 5×5, and that it is not necessarily the central pixel; it could be the median or the central pixel as described in Section 3.3. The optimal value of W_{max} is, actually, computationally interesting since it makes the algorithm run faster and it reduces the descriptor size.

6.1.2. The effect of scale change

In this section we study the influence of using the multiscale to represent the texture at different levels of resolution. Images are down-sampled, then the descriptors are computed at each resolution as described in Section 5. In this experiment we evaluate the classification performance as a function of the number of resolution levels with the pyramidal scheme. Fig. 11 presents the results obtained from one to four scales. For instance, two scales means the combination of the first and the second down-sampled images with histogram concatenation as described in Section 5. The best performance was obtained with two scales (i.e. $n=1$). Increasing the multiresolution levels after the optimal level dramatically decreases the performance. Two scales are appropriate for this type of relatively small images. However, the multiresolution optimal value may change for different types of images. For instance in the noise case,

Table 3
Comparison of the proposed method with the state-of-the-art methods on the Outex_TC_00012 test suite.

Method	Descriptor size	Learning phase	000	001
$\text{LBP}^{riu2}_{P,R}/\text{VAR}$ [3]	864	×	87.3	86.4
VZ_MR8 [19]	610	✓	92.6	92.8
VZ_Joint [21]	610	✓	91.4	92.1
PLBP [26]	162	×	88.9	86.6
$\text{DLBP}^{ri}_{P,R}$ [4]	$K_{80\%}$	×	93.2	90.4
$\text{CLBP}^{riu2}_{P,R}/\text{M/C}$ [5]	1352	×	95.3	94.5
$\text{Dis(S+M)}^{ri}_{P,R}$ [27]	1211	✓	97.0	96.5
$\text{AMBP}^{riu2}_{P,R,L_1}/\text{W}/\Gamma$	320	×	**98.0**	**96.7**

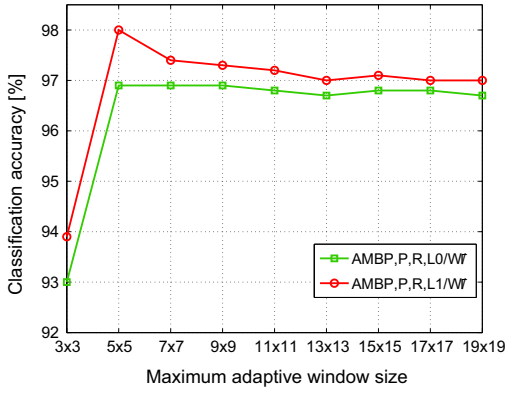

Fig. 10. The classification accuracy versus the maximum size of the adaptive window (W_{max} on Outex_TC_00012, using the tl84 (000) dataset for two levels of resolution.

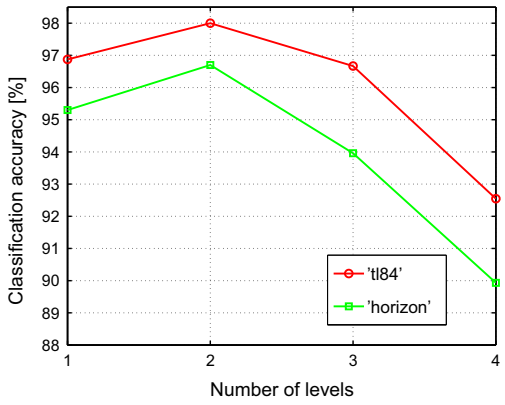

Fig. 11. Evaluation of the effect of the multiresolution levels n for testing the 'tl84' (000) and horizon (001) datasets from Outex_TC_00012 database. Training used 480 images under incandescent illumination.

Table 4
Comparison with the state-of-the-art methods. The AMBP '*riu2*' and '*ri*' layouts are used with $(8,1)\cup(16,2)\cup(24,3)$ and $(8,1)\cup(8,3)\cup(8,5)$ neighborhood schemes respectively. Each classification rate is the average of 100 trials of randomly selected samples.

Method	Descriptor size	Number of training images			
		46	23	12	6
$AMBP^{riu2}_{P,R,L_0}$	54	92.8	86.8	78.3	65.3
$AMBP^{riu2}_{P,R,L_0}/W$	108	94.3	88.7	80.1	66.4
$AMBP^{riu2}_{P,R,L_0}/W/\Gamma$	216	95.7	90.6	82.1	68.3
$AMBP^{ri}_{P,R,L_0}/W/\Gamma$	432	**96.7**	**92.2**	**84.3**	**71.0**
$AMBP^{riu2}_{P,R,L_1}$	80	94.7	89.5	81.7	69.2
$AMBP^{riu2}_{P,R,L_1}/W$	160	95.5	90.4	82.3	68.9
$AMBP^{riu2}_{P,R,L_1}/W/\Gamma$	320	96.0	91.0	82.7	69.0
$AMBP^{ri}_{P,R,L_1}/W/\Gamma$	576	**97.0**	**92.4**	**84.6**	**71.1**

Table 5
Comparison with state-of-the-art methods with different percentages of the training samples on the CUReT database. The AMBP '*ri*' scheme is $(8,1)\cup(8,3)\cup(8,5)$.

Method	Descriptor size	Learning phase	Number of training images 46
VZ_MR8 [19]	2440	✓	97.8
VZ_Joint [21]	2440	✓	97.7
$DLBP^{ri}_{P,R}$ [4]	$K_{80\%}$	×	84.1
$CLBP^{riu2}_{P,R}$/M/C [5]	2200	×	97.4
$Dis(S+M)^{ri}_{P,R}$ [27]	1211	✓	98.3
CS [22]	2440	✓	98.5
BIF [28]	1296	×	**98.6**
$AMBP^{ri}_{P,R,L_1}/W/\Gamma$	**576**	×	97.0

it is found that $n=2$ is better as shown in Section 6.3. Generally, a good value of the number of scales results in a tradeoff between computation complexity and classification accuracy.

6.2. Experiments on CUReT

The Columbia-Utrecht (CUReT) database [31] deals with 3D view of texture. The appearance of 3D texture depends on the viewing direction and the illumination. These conditions make some textures of the same class appear completely different. The CUReT database covers 61 different material surfaces, each one is observed with 205 combinations of viewing and illumination directions, but in general only 92 samples per class are selected for the classification task [19]. The samples are cropped from the original images with a fixed size 200×200 pixels.

Table 4 depicts the classification scores for different AMBP configurations. The AMBP joint histogram of rotationally invariant and uniform patterns ($AMBP^{riu2}_{P,R}/W/\Gamma$) shows good scores reaching 96% of correct classification for two levels of resolution with a histogram of 320 bins. The use of the joint histogram with only rotational invariant patterns ($AMBP^{ri}_{P,R}/W/\Gamma$) increases the classification accuracy by 1% with 576 bins (i.e. 97%). Compared to several methods in the literature, the proposed method features are competitive with the most successful techniques (see Table 5). Essentially, learning techniques such as VZ_MR8, VZ_Joint and CS are more successful in the CUReT database. This is due to the compensation of the biased features during the learning phase. Nevertheless, the BIF feature does not use the learning phase, but it uses multiscale histogram matching which helps us to reduce the matching errors. We expect that using such matching with the proposed method would increase the classification rate. However, it is beyond the scope of the present study to test different matching techniques. Note that the results in Table 5 are reported directly from the cited reference, except for DLBP which is our implementation.

6.3. Experiments on noisy textures

In many applications including surface imaging, noise is inherent to the output images due to many factors such as the acquisition devices and the lighting conditions. Consequently, the texture recognition and classification tasks become tougher. In general, images are corrupted by three types of noises: additive noise, multiplicative noise or impulse noise (salt and pepper noise). Such types of noise affect both the local structures and the global visual properties. They can introduce considerable confusion in the separation of texture classes, particularly the fine-textured surfaces. In this work, we focus on impulse noise because it is especially related to the median filter. To measure its effect we carried out a series of experiments on synthetically added noise to the CUReT dataset. Thus, noisy pixels can have only the maximum or the minimum value in the dynamic range. The corrupted pixel location is random. If the probability of a noisy pixel is p then

$$\mathcal{G}(i,j)=\begin{cases} N(i,j) & \texttt{with probability } p \\ \mathcal{I}(i,j) & \texttt{with probability } 1-p \end{cases} \tag{13}$$

where $N(i,j)$ is a noisy pixel, $I(i,j)$ is the pixel value of the original image at the location i,j and $\mathcal{G}(i,j)$ is the output image of the same pixel location. An example of impulse noise added to a texture is presented in Fig. 12 which shows the impact of noise density on the visual aspect of the texture.

In order to study the effect of noise, we gradually varied the noise ratio from 2% to 20% with step of 2%. Then, the performance of texture classification 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. Therefore, each set incorporates 5612 samples with 92 images per class.

Fig. 13 shows the classification performance of AMBP, MBP, VZ_MR8 (Varma and Zisserman MR8 descriptor), LBP, CLBP (Completed LBP), DLBP (Dominant LBP) and GABOR (Gabor filter) versus the density of impulse noise. The well known state-of-the-art methods failed to provide a good classification rate even for a low noise intensity. Despite the filtering process utilized by VZ_MR8 or GABOR, it is not efficient to handle impulse noise. The LBP, CLBP and DLBP methods cannot handle as well noisy textures either, because they use thresholding on central pixel. As these methods

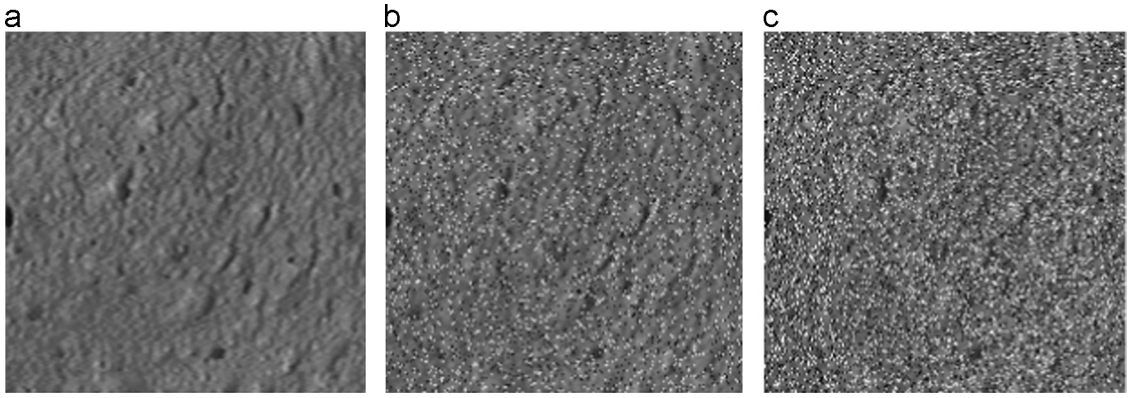

Fig. 12. Example of impulse noise in a CUReT sample. (a) presents the original image, (b) shows the same image as in (a) corrupted with noise density ratio of 10%. (c) The noise density is higher set to 20%.

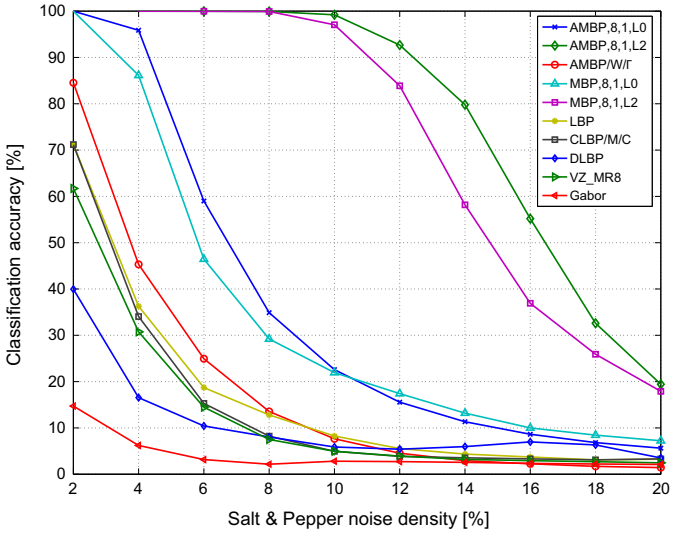

Fig. 13. Effects of impulse noise (*salt-and-pepper*) on the classification accuracy, for the CUReT database. The training set are noise free and the test set represents the same images, but with impulse noise at a specific noise ratio. AMBP and MBP were used with $P=8, R=1$, L_0 and L_2 schemes.

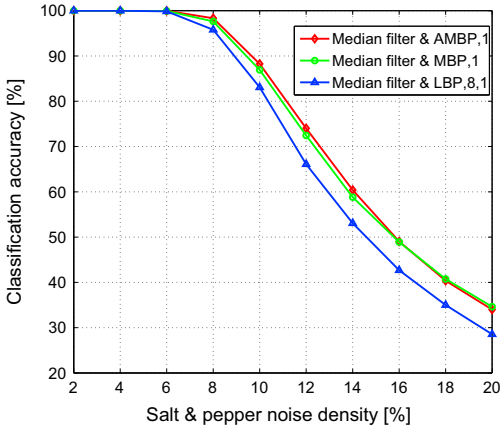

Fig. 14. Classification performance after preprocessing the CUReT dataset (impulse noise) using 3×3 median pre-filtering.

are based on LBP scheme, in the event of noise, the central pixel has a minimum or a maximum value, so, the LBP output will be quite different from the original one. Patterns that occur in noisy pixel influence the bins frequency in the histogram. Moreover, from one image to another the noise alters different pixel locations because it is randomly distributed and independent from each image. This modifies the histograms, inducing distortion in the similarity measure, which makes LBP fail to find the correct class. However, the Median Binary Pattern based methods show good robustness in such a situation. They start at a high level with less noise, then the performances of these methods decrease at each additional ratio of 2%. From 2% to 8% of noise $AMBP_{8,1,L_2}$ and $MBP_{8,1,L_2}$ with three scales show perfect results (100%). $AMBP_{8,1,L_2}$ keeps very high performances until 14%, then the classification rate decreases significantly at the ratio of 20%. At this ratio the noise is denser in the local neighborhood. Furthermore, the use of three scales seems to be more consistent and robust against noise.

These experiments demonstrate the capabilities of MBP and AMBP to remove impulse noise without any pre-filtering. We compared this to a more traditional approach of adding a noise reduction pre-filtering step prior to binary pattern extraction. A 3×3 median filter was applied to the whole CUReT database (noisy and non-noisy images). We then computed LBP, MBP and AMBP. As in previous experiments the training set is the noise free images and the testing set is the noisy ones. The results shown in Fig. 14 indicate an overall improvement in performance, particularly for LBP, compared to the results in Fig. 13. But the AMBP and MBP still outperform LBP, demonstrating the same tendency as the previous results.

6.4. Experiments on KTH_TIPS2

The major drawback of the CUReT database is that materials are imaged at a constant scale. In contrast, the KTH_TIPS2-b database [33] uses several distances from the camera. It includes four physical samples of 11 different materials similar to those used in CUReT database. These samples were imaged with variations in scale as well as variations in pose and illumination. Images were taken in combination of three poses, four illuminations and nine scales yielding 108 images per each physical sample. The images are cropped at size of 200×200 pixels and the four physical samples are categorized into one texture class. In result the database contains 4752 images ($11\times108\times4$). Samples of one material category can have different surface coarseness or roughness which makes them look different even if they are imaged with the same resolution. This is an additional difficulty combined with scale change for sample imaging. Fig. 15 shows the nine resolutions of one sample. The resolution was varied gradually from the second scale till the tenth scale. The visual aspect of the texture also changes gradually with the resolution, which impacts

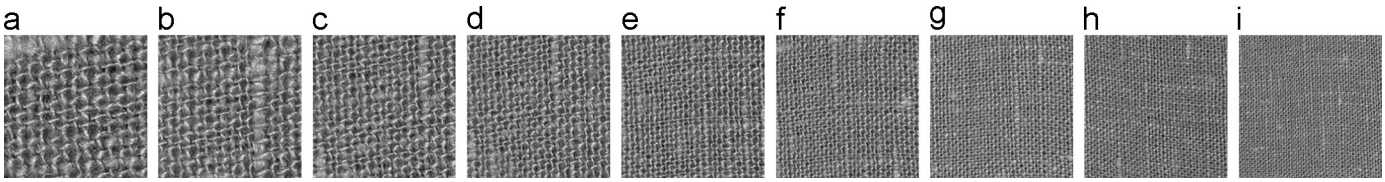

Fig. 15. Example of sample with 9 scales in the KTH_TIPS2-b database. (a) Scale 2, (b) Scale 3, (c) Scale 4, (d) Scale 5, (e) Scale 6, (f) Scale 7, (g) Scale 8, (h) Scale 9, and (i) Scale 10.

Table 6
Classification accuracy of the proposed method and the state-of-the-art on the KTH_TIPS2-b database. The LBP, CLBP and AMBP '*riu2*' are used with $(8,1) \cup (16,2) \cup (24,3)$ scheme.

Method	Descriptor size	Learning phase	Number of training sets		
			1	2	3
$LBP^{riu2}_{P,R}$	54	×	52.0	56.5	60.2
VZ_MR8 [19]	610	✓	46.1	52.0	55.3
VZ_Joint [21]	610	✓	53.5	60.0	61.0
$DLBP^{ri}_{P,R}$ [4]	$K_{80\%}$	×	49.3	55.1	58.0
$CLBP^{riu2}_{P,R}$/M/C [5]	2200	×	55.0	61.1	67.7
Gabor	24	×	45.0	48.5	50.1
$AMBP^{riu2}_{P,R,L_0}$	54	×	57.2	60.2	66.4
BIF [28]	1296	×	50.9	60.0	61.8
$AMBP^{riu2}_{P,R,L_0}$/W	108	×	59.3	61.2	66.7
$AMBP^{riu2}_{P,R,L_0}$/W/Γ	216	×	60.1	64.0	**70.3**
$AMBP^{riu2}_{P,R,L_1}$	80	×	58.3	63.1	67.7
$AMBP^{riu2}_{P,R,L_1}$/W	160	×	59.3	63.4	68.7
$AMBP^{riu2}_{P,R,L_1}$/W/Γ	320	×	**60.2**	**64.2**	**70.3**

the local structures. The aim is to study the descriptors' robustness in such conditions.

In this experiment, each class is associated with one material, which contains four physical samples (a set of 108 images per sample). The goal is to learn a given physical sample of each class, then recognize the remaining ones. For that purpose we used three configurations, with one, two and three training sets per class. The sets 1, 2, 3 and 4 were combined as (training, testing) couples with all possible combinations; for one training set we have the couples: $(1,2 \cup 3 \cup 4)$, $(2,1 \cup 3 \cup 4)$, $(3,1 \cup 2 \cup 4)$ and $(4,1 \cup 2 \cup 3)$. For two training sets: $(1 \cup 2,3 \cup 4)$, $(1 \cup 3,2 \cup 4)$, $(1 \cup 4,2 \cup 3)$, $(2 \cup 3,1 \cup 4)$, $(2 \cup 4,1 \cup 3)$ and $(3 \cup 4,1 \cup 2)$. Finally, three training sets yield $(2 \cup 3 \cup 4,1)$, $(1 \cup 3 \cup 4,2)$, $(1 \cup 2 \cup 4,3)$ and $(1 \cup 2 \cup 3,4)$. The classification score is the average of all combinations scores of each configuration.

In this section we also study the performance of the proposed algorithm and compared it with recent and well known methods from the literature. From Table 6 it can be seen that the highest scores are obtained by the AMBP 3D joint histogram as in Outex_TC_00012, whereas the simple $AMBP^{riu2}_{P,R,L_1}$ shows better scores than the other techniques including VZ_MR8, VZ_Joint and CLBP/M/C (joint histogram). All the presented methods are our implementation, except for the CLBP [5] and BIF [28] where we used the original codes provided.

7. Conclusions

We have presented a novel method based on local adaptive analysis for texture description. The proposed approach uses either the median or the central pixel to generate local patterns, taking into account the variation in the local context in images, and associates the joint information based on adaptive local analysis. It is computationally efficient, which enables real time tasks. We have evaluated the proposed descriptor over a wide range of textures, using Outex, CUReT and KTH TIPS2. Different classifiers may lead to different classification accuracies; our focus is feature extraction. To allow a fair evaluation, only *k*-NN classifier is used to measure the classification accuracy. We have studied different problems related to texture recognition such as the influence of number of the training samples, the effect of noise, and the effect of scale change and illumination changes. Extensive experiments demonstrate that the AMBP approach consistently outperforms numerous state-of-the-art methods. In the presence of impulse noise our algorithm is noticeably better than the most successful ones. The proposed AMBP method shows a new kind of properties for texture descriptors, for instance adaptation of the window size and handling noise suppression without prefiltering the images. AMBP introduces a new type of operator that automatically selects between using the Local Binary Pattern or the Median Binary Pattern . This type of approach can be generalized and extended to other classes of texture operators in a similar fashion, opening up a new direction for combining texture measures. In future work we will extend the proposed approach to handle more types of noise and will focus also on additional information to improve performances even further.

Conflict of interest

None declared.

References

[1] M. Tuceryan, A. Jain, Texture analysis, in: C.H. Chen, L.F. Pau, P.S.P. Wang (Eds.), Handbook of Pattern Recognition and Computer Vision, World Scientific Publishing Co., Singapore, 1998, pp. 235–276.
[2] M. Petrou, P.G. Sevilla, Image Processing: Dealing with Texture, Wiley, Chichester, UK, 2006.
[3] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 971–987.
[4] S. Liao, M.W.K. Law, A.C.S. Chung, Dominant local binary patterns for texture classification, IEEE Trans. Image Process. 18 (2009) 1107–1118.
[5] Z. Guo, L. Zhang, D. Zhang, A completed modeling of local binary pattern operator for texture classification, IEEE Trans. Image Process. 19 (2010) 1657–1663.
[6] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, W. Gao, WLD: a robust local image descriptor, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1705–1720.
[7] A. Hafiane, G. Seetharaman, K. Palaniappan, B. Zavidovique, Rotationally invariant hashing of median patterns for texture classification, in: Lecture Notes in Computer Science (ICIAR), vol. 5112, 2008, pp. 619–629.
[8] A. Fernández, M.X. Álvarez, F. Bianconi, Texture description through histograms of equivalent patterns, J. Math. Imaging Vis. 45 (2013) 76–102.
[9] R.C. Gonzales, R.E. Woods, Digital Image Processing, 3rd edition, Prentice Hall, Upper Saddle River, New Jersey, USA, 2008.
[10] R.M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. 3 (1973) 610–621.
[11] L.S. Davis, Polarograms: a new tool for image texture analysis, Pattern Recognit. 13 (1981) 219–223.
[12] R. Kashyap, A. Khotanzad, A model-based method for rotation invariant texture classification, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1986) 472–481.
[13] F.S. Cohen, Z. Fan, M.A. Patel, Classification of rotated and scaled textured images using Gaussian Markov random field models, IEEE Trans. Pattern. Anal. Mach. Intell. 13 (1991) 192–202.
[14] M.M. Leung, A.M. Peterson, Scale and rotation invariant texture classification, in: The 26th Asilomar Conference on Signals, Systems and Computation, 1992, pp. 461–465.

[15] G.M. Haley, B.S. Manjunath, Rotation-invariant texture classification using a complete space-frequency model, IEEE Trans. Image Process. 8 (1999) 255–269.
[16] K.J. Dana, S.K. Nayar, Correlation model for 3D texture, in: International Conference on Computer Vision, 1999, pp. 1061–1066.
[17] T. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, Int. J. Comput. Vis. 43 (2001) 29–44.
[18] O.G. Cula, K.J. Dana, 3D texture recognition using bidirectional feature histograms, Int. J. Comput. Vis. 59 (2004) 33–60.
[19] M. Varma, A. Zisserman, A statistical approach to texture classification from single images, Int. J. Comput. Vis. 62 (2005) 61–81.
[20] S. Lazebnik, C. Schmid, J. Ponce, A sparse texture representation using local affine regions, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1265–1278.
[21] M. Varma, A. Zisserman, A statistical approach to material classification using image patch exemplars, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 2032–2047.
[22] L. Liu, P. Fieguth, Texture classification from random features, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 574–586.
[23] W. Zhang, S. Shan, W. Gao, X. Chen, H. Zhang, Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition, in: International Conference on Computer Vision, vol. 1, 2005, pp. 786–791.
[24] M. Heikkilä, M. Pietikäinen, A texture-based method for modeling the background and detecting moving objects, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 657–662.
[25] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 915–928.
[26] X. Qian, X.-S. Hua, P. Chen, L. Ke, Plbp: an effective local binary patterns texture descriptor with pyramid representation, Pattern Recognit. 44 (2011) 2502–2515.
[27] Y. Guo, G. Zhao, M. Pietikäinen, Discriminative features for texture description, Pattern Recognit. 45 (2012) 3834–3843.
[28] M. Crosier, L.-D. Griffin, Using basic image features for texture classification, Int. J. Comput. Vis. 88 (2010) 447–460.
[29] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognit. 29 (1996) 51–59.
[30] A. Hafiane, G. Seetharaman, B. Zavidovique, Median binary pattern for textures classification, in: Lecture Notes in Computer Science (ICIAR), vol. 4633, 2007, pp. 387–398.
[31] K. Dana, B. Van-Ginneken, S. Nayar, J. Koenderink, Reflectance and texture of real world surfaces, ACM Trans. Graph. 18 (1999) 1–34 ⟨http://www.cs.columbia.edu/CAVE//exclude/curet/⟩.
[32] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllonen, S. Huovinene, Outex—a new framework for empirical evaluation of texture analysis algorithms, in: International Conference on Pattern Recognition, vol. 1, 2002, pp. 701–706, ⟨http://www.outex.oulu.fi⟩.
[33] P. Mallikarjuna, M. Fritz, A.T. Targhi, E. Hayman, B. Caputo, J. Eklundh, The KTH-TIPS and KTH-TIPS2 Databases, ⟨http://www.nada.kth.se/cvap/databases/kth-tips⟩.

Adel Hafiane Received the MS degree in Embedded Systems & Information Processing from the University of Paris-Sud, France, in 2002 and the Ph.D. from the same university, in 2005. After that, he worked one year in Teaching and Research at Paris-Sud and later for one year at INSA Centre Val de Loire (Former ENSI de Bourges). He was Postdoctoral Fellow in the Computer Science Department at the University of Missouri during 2007-2008. Since September 2008, he is Assistant Professor at INSA Centre Vald de Loire, he is a Member of "Images and Vision" group at PRISME Laboratory (University of Orléans). He was also an invited researcher at the university of Missouri on multiple periods between 2009-2013. His research interests include theory and methods of image processing, computer vision and machine learning for medical and robotic applications.

Kannappan Palaniappan is an associate professor in the Computer Science Department at the University of Missouri. He has received several notable awards including the NASA Public Service Medal for pioneering contributions to scientific visualization, the Air Force Summer Faculty Fellowship, the Boeing Welliver Summer Faculty Fellowship and the William T. Kemper Fellowship for Teaching Excellence. At NASA's Goddard Space Flight Center, he co-founded the Visualization and Analysis Lab that has produced a number of spectacular Digital Earth visualizations used by search engines (BlueMarble), museums, magazines and broadcast television. He is co-inventor of the Interactive Image SpreadSheet for handling large multispectral imagery, and he developed the first massively parallel semi-fluid cloud motion analysis algorithm using geostationary satellite imagery. In 2013-2014, he was on sabbatical as a National Academy of Science Jefferson Science Fellow at the Department of State in Washington, DC. His current interests include computer vision, multicore image processing, machine learning, data visualization, biomedical and microscopy image analysis.

Guna Seetharaman is a Principal Engineer for Computer Vision and Video Exploitation, at the Information Directorate, Air Force Research Laboratory, Rome, NY. He served as an associate professor of computer science and engineering, at the Air Force Institute of Technology (2003-2008) and University of Louisiana at Lafayette (1988-2003). He was also a CNRS Invited professor at University of Paris-Sud (France) on multiple tenures between 1998-2005; and, held a visiting distinguished professorship at Indian Institute of Technology, Mumbai, in 2006. He is currently focused on high performance computing, computer vision, machine learning, content-based image retrieval, persistent surveillance and computational science and engineering.