Enhancement of Textural Differences based on Morphological Component Analysis

Jianning Chi, Mark Eramian

***Abstract*—This paper proposes a new texture enhancement method which uses an image decomposition that allows different visual characteristics of textures to be represented by separate components in contrast with previous methods which either enhance texture indirectly or represent all texture information using a single image component. Our method is intended to be used as a preprocessing step prior to the use of texture-based image segmentation algorithms. Our method uses a modification of Morphological Component Analysis (MCA) which allows texture to be separated into multiple morphological components each representing a different visual characteristic of texture. We select four such texture characteristics and propose new dictionaries to extract these components using MCA. We then propose procedures for modifying each texture component and recombining them to produce a texture-enhanced image. We applied our method as a preprocessing step prior to a number of texture-based segmentation methods and compared the accuracy of the results, finding that our method produced results superior to comparator methods for all segmentation algorithms tested. We also demonstrate by example the main mechanism by which our method produces superior results, namely that it causes the clusters of local texture features of each distinct image texture to mutually diverge within the multidimensional feature space to a vastly superior degree versus the comparator enhancement methods.**

***Index Terms*—texture, enhancement, segmentation, morphological component analysis, non-linear transform;**

I. Introduction

THE subject of this paper is the preprocessing of image texture to improve the differentiability of textures with respect to texture features. The motivation for this work is to extract greater performance from any texture-based segmentation method by establishing a general-purpose texture enhancement algorithm. The method we propose method is similar in concept to the manner in which chemical staining is used in histopathology to enhance the appearance of differences in tissue composition to aid a pathologist's assessment of a sample. Our method of enhancing texture differences is analogous to the stain and the effect of this enhancement is to make it easier for different textures to be distinguished by segmentation algorithms (analogous to aiding the pathologist). An important property of our method is that it makes no assumptions about the number and type of textures present in an image.

Some early texture enhancement methods reduce noise or artifacts in the image to highlight the textures indirectly, for example, the median filter [17] and the Weiner filter [18]. However, these conventional filters degrade the textures in addition to removing noise because of their lowpass-filter-like qualities. Discontinuity-preserving filters were developed to mitigate this issue to a certain extent, for example, the non-local means filter [7] which can smooth the noise and artifacts in the image while preserving image detail as much as possible. Wavelet-based methods, e.g. VISUShrink [9], BayesShrink [11], SUREShrink [12], were proposed to remove noise by shrinking coefficients in high-frequency sub-bands not exceeding certain thresholds, while preserving the image textures which are represented by coefficients in high-frequency sub-bands that exceed these thresholds.

Other methods enhance the textures in the image directly. Unsharp masking (UM) was proposed to improve the visual appearance of an image by emphasizing its high frequency contents [23]. However, the highpass-filter-like nature of UM causes enhancement of noise and artifacts in the image as well. The same is true of histogram equalization methods [22]. Therefore, some non-linear methods have been proposed to enhance the textures. Hong et. al. [16] proposed a texture enhancement algorithm that can improve the clarity of ridge and valley structures of fingerprint textures based on the estimated local ridge orientation and frequency. Coherence-enhancing anisotropic diffusion [30] is based on the modifications of partial differential equations (PDEs) [8] which can preserve strong discontinuities at edges while removing artifacts from smooth regions. Shock filtering [31] is a transformation of anisotropic diffusion which smooths along the coherent texture flow orientations, and reduces diffusivity at non-coherent structures which enhances textural detail.

All of the above methods enhance or suppress all of the "textural" components and "non-textural" components of the image to the same extent, thus, the quantitative difference between texture descriptors for different textures is not much altered. This occurs because all of the different textures in an image are treated as a single "texture" component alongside other non-texture components which results in any transformation to texture being applied to all textures uniformly.

In the method presented herein, it is assumed that textures consist of several different components representing different visual characteristics. By modifying these components in different ways, distinct textures become more different in terms of the descriptors used to differentiate between them. Morphological component analysis (MCA) has proven successful in decomposing images into morphologically distinct components, e.g., a smooth component, an oscillating component and noise component [4], [13], [14], [25], [32]. By filtering out

Mark Eramian is with the Department of Computer Science, University of Saskatchewan, SK, S7N 5C9 CA e-mail: eramian@cs.usask.ca

Jianning Chi is with the Department of Computer Science, University of Saskatchewan, SK, S7N 5C9 CA e-mail:chi.jianning@gmail.com

the noise component and recombining the smooth component and the oscillating component, the image variations due to noise is reduced [5]. However, these works only decompose images according to the "cartoon+texture+noise" model [5] or otherwise express texture using only a single morphological component.

In this paper, we first select some textural characteristics based on human visual perception. Then, a novel form of MCA is used to decompose textures into *multiple* morphological components according to these characteristics by introducing several dictionaries. The morphological components of different textures are then modified in different ways so that textures become more different with respect to these textural characteristics.

The paper is organized as follows. In section 2, we describe our MCA decomposition by texture characteristics. The specific characteristics used for our texture enhancement algorithm and their dictionaries are detailed in section 3. The modification of textures by manipulating their morphological components using local non-linear operators is explained in section 4. In section 5 the proposed method is compared with prior methods from the literature: unsharp masking (UM), VISUShrink (VISU), shock filtering, coherence-enhancing diffusion filtering and the standard "cartoon+texture" MCA (MCA-CT).

II. Characteristic-based Texture Decomposition using Morphological Component Analysis

A *texture characteristic* is broadly defined any property of a texture that can be quantified. Fig. 1 shows a schematic of the proposed method to enhance textural differences by manipulating certain texture characteristics in certain ways. Intuitively, each step functions as follows:

1) the image is decomposed to several pairs of components where each pair consists of a component that strongly exhibits a particular texture characteristic and a component that weakly exhibits it, or exhibits opposite characteristics, e.g. a "coarse" component and a "fine" component;
2) the components are manipulated to enhance the texture characteristics they are meant to capture, e.g. a high-coarseness component is manipulated so that so it becomes coarser, a low-coarseness (fine) component is manipulated so that it becomes finer;
3) the manipulated components are recombined to obtain an image in which textures are more different from each other than in the original image with respect to the chosen texture characteristics.

A. MCA: Standard "Cartoon+Texture" Decomposition

In [25], morphological component analysis (MCA) was proposed to seek components of the image x by solving:

$$\left\{s_1^{opt},\dots,s_K^{opt}\right\} = \arg\min_{\{s_1,\dots,s_K\}} \sum_{k=1}^{K} \|T_k s_k\|_1 + \lambda \left\| x - \sum_{k=1}^{K} s_k \right\|_2^2, \quad (1)$$

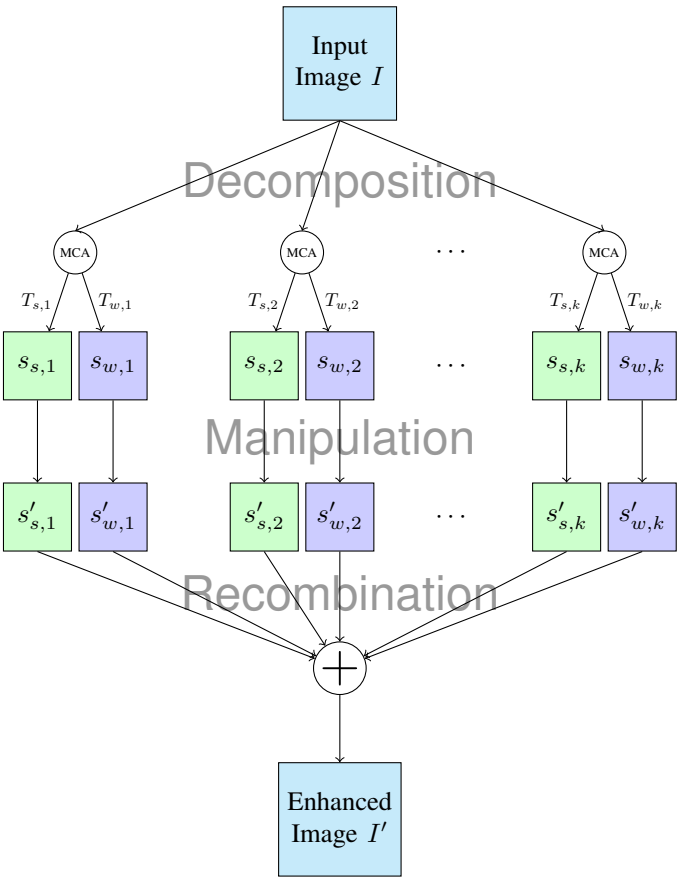


Fig. 1. The proposed texture enhancement process. The input image undergoes k MCA decompositions (Eq. 4) using dictionaries $T_{s,i}$ and $T_{w,i}$ $(i = 1,\dots,k)$ to extract k pairs of components that strongly and weakly exhibit a particular texture characteristic. These components are manipulated (Section IV) to enhance or suppress the characteristic: in strong components, the texture characteristic is further enhanced, in weak components, the characteristic is further suppressed. The manipulated components are then recombined (Eq. 6) to form an image in which the differences between individual textures has been enhanced. Symbols $s_{s,i}$ and $s_{w,i}$ represent strong and weak components of the i-th characteristic respectively $(i = 1,\dots,k)$; $s'_{s,i}$ and $s'_{w,i}$ symbols represent manipulated components.

where $s_1,\dots,s_K$ are image components, $T_k,\ k = 1,2,\dots,K$ are transformations employed by the k-th dictionary to represent the k-th component s_k as a "sparse" linear combination of a small number of basis elements. In [25], decomposition of the image into cartoon and texture components is the special case of Eq. 1,

$$\left\{s_c^{opt}, s_t^{opt}\right\} = \arg\min_{\{s_c,s_t\}} \|T_c s_c\|_1 + \|T_t s_t\|_1 + \lambda \|x - s_c - s_t\|_2^2, \quad (2)$$

where s_c and s_t are the cartoon and texture components, x is the image to be decomposed, T_c and T_t are transformations employed by dictionaries for representing cartoon and texture components respectively.

To solve the optimization problem Eq. 2, the following steps were implemented in [25]:

1) two dictionaries were selected first for the cartoon and texture parts of the image. The dictionaries employed transformations that represent either texture or piecewise smooth (cartoon) behaviours of the image. More details

1. Initialize L_{max}, the number of iterations, and threshold $\delta = \lambda \cdot L_{max}$, s = input image, $s_c = s_t = 0$. T_c, T_t, R_c, R_t are the forward and inverse transforms for the dictionaries, respectively.
2. Perform L_{max} times:
   Update of s_c assuming s_t is fixed:
   - Calculate the residual $r = s - s_t - s_c$.
   - Calculate the transform T_c of $s_c + r$ and obtain $\alpha_c = T_c(s_c + r)$.
   - Soft threshold the coefficient α_c with the δ threshold and obtain $\widehat{\alpha_c}$.
   - Reconstruct s_c by $s_c = R_c\widehat{\alpha_k}$.

   Update of s_t assuming s_c is fixed:
   - Calculate the residual $r = s - s_c - s_t$.
   - Calculate the transform T_t of $s_t + r$ and obtain $\alpha_t = T_t(s_t + r)$.
   - Soft threshold the coefficient α_t with the δ threshold and obtain $\widehat{\alpha_t}$.
   - Reconstruct s_t by $s_t = R_t\widehat{\alpha_t}$.
3. Update the threshold by $\delta = \delta - \lambda$.
4. If $\delta > \lambda$, return to Step 2. Else, finish.

Algorithm 1
THE ALGORITHM FOR MINIMIZING EQ. 2 [25].

about candidate dictionaries were described in [24];

2) with the selected dictionaries T_c and T_t, the Algorithm 1 was proposed to minimize Eq. 2 and seek two images s_c and s_t as the cartoon and texture components of the image x.

However, the traditional MCA cartoon + texture method (MCA-CT) described above has shortcomings:

1) the dictionaries only focused on decomposing the image into a single cartoon component and a single texture component, ignoring the fact that textures may be comprised of multiple components exhibiting diverse texture characteristics to varying degrees;
2) the dictionaries were restricted to employ multi-scale and local analysis of the image content, resulting in a quite narrow range of candidate dictionaries.

B. MCA: Decomposition by Multiple Texture Characteristics

In this paper, we improve the traditional MCA method by loosening the restrictions of dictionaries and seeking optimal parameters for those selected dictionaries, so that the image can be decomposed into different components corresponding to k textural characteristics described in Section III-A by solving the following k optimization problems (one for each value of i):

$$\left\{s_{s,i}^{opt}, s_{w,i}^{opt}\right\} = \arg\min_{\{s_{s,i}, s_{w,i}\}} \|T_{s,i}s_{s,i}\|_1 + \|T_{w,i}s_{w,i}\|_1 + \|I - s_{s,i} - s_{w,i}\|_2^2, \quad (3)$$

where $s_{s,i}$ and $s_{w,i}$ are the components having strong and weak aspects of the i-th texture characteristic, $i = 1, 2, \ldots, k$, e.g. a "coarse" component and a "non-coarse" component. $T_{s,i}$ and $T_{w,i}$ are dictionaries for $s_{s,i}$ and $s_{w,i}$ respectively, and I is the original image.

The selection of dictionaries in our work is directly based on the basic assumptions for MCA: for every image to be decomposed, there exists a dictionary that is highly efficient in highlighting one texture characteristic of the image and highly inefficient in highlighting other texture characteristics. Therefore, in this paper, the dictionaries are not restricted to transformations employing multi-scale local analysis and synthesis of the image content. Instead we permit transformations, including local spatial filters, that preserve image regions that strongly exhibit a texture characteristic and remove regions that weakly exhibit that characteristic (or vice versa), e.g. a transformation that filters out areas of fine texture while preserving areas of coarse texture.

The dictionaries used in our method (described in Section III) have adjustable parameters so that the performance of decomposition can be more consistent for different images than the traditional MCA. To allow for the additional optimization of dictionary parameters, Eq. 3 is modified to:

$$\left\{s_{s,i}^{opt}, s_{w,i}^{opt}, T_{s,i}^{opt}, T_{w,i}^{opt}\right\} = \arg\min_{\{s_{s,i}, s_{w,i}, T_{s,i}, T_{w,i}\}} \|T_{s,i}s_{s,i}\|_1 + \|T_{w,i}s_{w,i}\|_1 + \|I - s_{s,i} - s_{w,i}\|_2^2, \quad (4)$$

where $T_{s,i}^{opt}$ and $T_{w,i}^{opt}$ are the transformations or local spatial filters used as dictionaries for components corresponding to strong and weak aspects of i-th characteristic, respectively, whose parameters have been optimized. Algorithm 2 is proposed to solve the optimization problem in Eq. 4 to seek components $s_{s,i}$ and $s_{w,i}$ as well as the parameters of dictionaries $T_{s,i}$ and $T_{w,i}$. In Algorithm 2, $\mu_{s,i}$ and $\mu_{w,i}$ are the parameter sets of the dictionaries $T_{s,i}$ and $T_{w,i}$ respectively. L_{max} is the maximum number of iterations of decomposition. The parameters in $\mu_{s,i}$ and $\mu_{w,i}$ are decreased uniformly over each iteration to 0 till obtaining the optimizing results. The initial values for the parameters of every dictionaries are listed in Table I. Note that the decomposition for each texture characteristic is independent of the others so that any number of texture characteristics may be used as desired. Details about the dictionaries used in our method are discussed in Section III.

C. Manipulation of the image components

After decomposing the image into pairs of strong and weak texture characteristic components, these components are manipulated to enhance the texture characteristics they are meant to capture. In general,

$$\begin{aligned} s_{s,i}^{'} &= f_{s,i}(s_{s,i}) \\ s_{w,i}^{'} &= f_{w,i}(s_{w,i}) \end{aligned}, \quad (5)$$

where $s_{s,i}$ and $s_{w,i}$ are the components respectively exhibiting strong and weak aspects of the i-th texture characteristic, $i = 1, 2, \ldots, k$, $s_{s,i}^{'}$ and $s_{w,i}^{'}$ are the manipulated strong and weak characteristic components, and $f_{s,i}$ and $f_{w,i}$ are the manipulation functions used to enhance the texture components $s_{s,i}$ and $s_{w,i}$ respectively. More details about the design of $f_{s,i}$ and $f_{w,i}$ are discussed in Section IV.

D. Re-combination of the manipulated components

After manipulating every component to enhance its own properties, the components are re-combined in to a final texture-enhanced image I' as follows:

$$I' = \frac{1}{k}\sum_{i=1}^{k}\left(s_{s,i}^{'} + s_{w,i}^{'}\right), \quad (6)$$

1. Initialize the number of iterations L_{max}, the parameters $\mu_{s,i}$ and $\mu_{w,i}$ of the dictionaries $T_{s,i}$ and $T_{w,i}$, $\delta = \lambda \cdot L_{max}$, threshold for stopping decomposition, and ϕ is a threshold for updating parameters of dictionaries. $s_{s,i} = s_{w,i} = 0$.
2. Perform L_{max} times:
   Update of $s_{s,i}$ assuming $s_{w,i}$ is fixed:
   - Calculate the residual $r = I - s_{s,i} - s_{w,i}$.
   - Calculate the transformation $T_{s,i}$ of $s_{s,i} + r$ and obtain $s'_{s,i} = T_{s,i}(s_{s,i} + r)$.
   - Calculate $d = \left\| s'_{s,i} - s_{s,i} \right\|_1$.
   - If $d > \phi$, update $T_{s,i}$ by updating $\mu_{s,i}$ with $\mu_{s,i} - \frac{\mu_{s,i}}{L_{max}}$. Else, $\mu_{s,i}$ keep the same values.
   - Update $s_{s,i}$ with $s'_{s,i}$.

   Update of $s_{w,i}$ assuming $s_{s,i}$ is fixed:
   - Calculate the residual $r = I - s_{w,i} - s_{s,i}$.
   - Calculate the transformation $T_{w,i}$ of $s_{w,i} + r$ and obtain $s'_{w,i} = T_{w,i}(s_{w,i} + r)$.
   - Calculate $d = \left\| s'_{w,i} - s_{w,i} \right\|_1$.
   - If $d > \phi$, update $T_{w,i}$ by updating $\mu_{w,i}$ with $\mu_{w,i} - \frac{\mu_{w,i}}{L_{max}}$. Else, $\mu_{w,i}$ keep the same values.
   - Update $s_{w,i}$ with $s'_{w,i}$.
3. Update the threshold by $\delta = \delta - \lambda$.
4. If $\delta > \lambda$, return to Step 2. Else, finish.

Algorithm 2
THE ALGORITHM FOR MINIMIZING EQ. 4.

where $s'_{s,i}$ and $s'_{w,i}$ are calculated in Eq. 5 as the manipulated strong and weak characteristic components respectively, and k is the total number of characteristics used for image decomposition.

In the next section we describe the specific texture characteristics we used in our texture enhancement algorithm and their corresponding dictionaries.

III. PROPOSED TEXTURE CHARACTERISTICS AND THEIR DICTIONARIES

As discussed in Sec. II-B, the two key points in decomposing the image according to texture characteristics are: 1) the selection of texture characteristics and 2) the selection of filters used as dictionaries which we now proceed to describe in detail.

A. Texture Characteristics

Human visual perceptual characteristics were selected for our MCA framework because they describe the texture closely to how human beings perceive the texture leading to intuitive morphological components. The following characteristics from Tamura descriptor [26] were selected as the basis of decomposition:

1) coarseness: a measure of the number of edges in a local square neighbourhood of radius;
2) contrast: a measure of the variance of the intensities of the pixels in a local window;
3) directionality: a measure of the mean edge direction in a local window;
4) line-likeness: a measure of the variance of the edge direction in a local window;

For each of the above characteristics, we propose a pair of dictionaries to represent the strong and weak characteristic components, e.g., for the characteristic of coarseness, the image is decomposed to coarse (strong) and fine (non-coarse, weak), components.

B. Dictionaries of each characteristic

The dictionaries $T_{s,i}$ and $T_{w,i}$ in Eq. 4 are proposed in this section. With the same principle of the dictionary selection in the standard MCA, the dictionaries proposed in this paper should satisfy: 1) they can highlight the components corresponding to a certain texture characteristic, and 2) they are insensitive to the other texture characteristics.

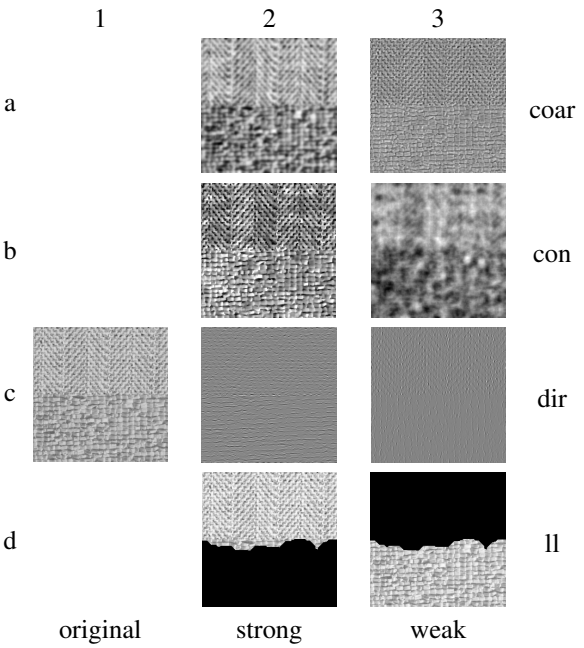


Fig. 2. Decomposition of a textural image by different characteristics. Rows: pairs of textures with strong and weak characteristics – coarseness (coar), contrast (con), directionality (dir) and line-likeness (ll). Columns: input image (left), components of input images with strong characteristics (middle), components of input images with weak characteristics (right).

1) Dictionaries for coarseness: Coarseness quantifies the number of edges in the local texture. Therefore, decomposing a texture into the coarse component and the fine component is to look for dictionaries corresponding to regions with few strong texture edges and regions with many weak texture edges, respectively.

Bilateral filtering [27] is used as the dictionary $T_{s,1}$ to highlight the coarse component since it can remove small-scale elements from the image:

$$T_{s,1}(I(x)) = \frac{1}{K} \sum_{\xi \in S} I(\xi) \cdot e^{-\frac{1}{2}\left(\frac{\|\xi - x\|}{\sigma_d}\right)^2} \cdot e^{-\frac{1}{2}\left(\frac{I(\xi) - I(x)}{\sigma_r}\right)^2}, \tag{7}$$

where x is the center pixel in the neighbourhood S, $I(x)$ and $I(\xi)$ denote the intensity of center and the nearby pixels in the neighbourhood. The spatial and spectral bandwidths, σ_d and σ_r, comprise the parameter set $\mu_{s,1} = [\sigma_d, \sigma_r]$ for this dictionary.

By applying $T_{s,1}$ to the image, the coarse textures will be enhanced while fine textures will be suppressed because the

weak edges ($|I(\xi) - I(x)| \approx 0$) are removed while strong edges ($|I(\xi) - I(x)|$) are preserved.

Comparing an example image and its strongly-coarse component (Fig. 2(a2) and 2(c1), respectively.), it can be seen that the strong edges are preserved while the regions surrounded by these edges are smoothed to preserve only basic texture structure.

Conversely, the dictionary $T_{w,1}$ corresponding to the low coarseness (fine) texture component needs to weaken the strong edges while preserving the weak edges. We use wavelet thresholding [11] by transforming the coefficients as follows:

$$\begin{array}{c} T_{w,1}(I) = \text{idwt}(\Psi') \\ \Psi = \text{dwt}(I) \end{array}, \tag{8}$$

where $\text{dwt}(\cdot)$ and $\text{idwt}(\cdot)$ are forward and inverse discrete wavelet transform respectively, using the Symlet 4 wavelet. The wavelet coefficients $[\Psi_a, \Psi_h, \Psi_v, \Psi_d] \in \Psi$ (approximation, horizontal, vertical, diagonal) of the input image I are filtered with the soft thresholding method in [11] to $[\Psi'_a, \Psi'_h, \Psi'_v, \Psi'_d] \in \Psi'$ as follows:

$$\Psi(x,y)' = \begin{cases} \Psi(x,y) & \text{if } |\Psi_h(x,y)| < \delta_h \ \& \ |\Psi_v(x,y)| < \delta_v \\ & \quad \& \ |\Psi_d(x,y)| < \delta_d \\ \Psi(x,y) - \lambda & \text{otherwise} \end{cases} \tag{9}$$

where $\Psi_h(x,y)$, $\Psi_v(x,y)$ and $\Psi_d(x,y)$ are coefficients values at (x,y) in horizontal, vertical and diagonal sub-bands before wavelet thresholding. The thresholds for each wavelet sub-band, δ_h, δ_v and δ_d comprise the parameter set $\mu_{w,1} = [\delta_h, \delta_v, \delta_d]$. The fine textures are preserved because only the weak edges with small magnitudes of high-frequency coefficients are not weakened, while coarse textures are removed because the magnitudes of wavelet coefficients corresponding to strong edges are decreased. The low-coarseness (fine) component of our example image is shown in Fig. 2(a3).

2) Dictionaries for contrast: Contrast measures the variance of the grey scale intensities in a local area. Two dictionaries were chosen to represent either textures with high or low intensity variance, respectively.

Fig. 2(c1), 2(b2) and 2(b3) show the decomposition of our example image based on contrast. For the low-contrast component, we make use of anisotropic diffusion (AD) [29]:

$$\begin{array}{c} T_{w,2}(I(x)) = I(x) + \frac{1}{|S|}\sum_{\xi \in S} c(\nabla I(\xi)) \nabla I(\xi) \\ c(\nabla I(\xi)) = e^{-\left(\frac{\nabla I(\xi)}{k}\right)^2} \end{array}, \tag{10}$$

where $I(x)$ is the pixel intensity of image I at x, S is the neighbourhood centred at x, ξ denotes the neighbouring pixels in S, $|S|$ is the number of pixels in the neighbourhood, and $\nabla I(\xi) = I(\xi) - I(x)$, $\forall \xi \in S$. k is the only parameter, thus, $\mu_{w,2} = [k]$.

By applying $T_{w,2}$ to the image, the low-contrast parts where $|\nabla I(\xi)|$ quite small won't change much because $c(|\nabla I(\xi)|) \approx 1$, while the high-contrast parts will be smoothed and reduced in contrast because in such regions the AD behaves as isotropic diffusion (Gaussian filtering).

For the high-contrast components, we propose a modification of AD, which we call "anisotropic shrink", as the dictionary representing high-contrast components:

$$\begin{array}{c} T_{s,2}(I(x)) = I(x) + \frac{1}{|S|}\sum_{\xi \in S} c(\nabla I(\xi)) |\nabla I(\xi)| \\ c(\nabla I(\xi)) = 1 - e^{-\left(\frac{\nabla I(\xi)}{k}\right)^2} \end{array}, \tag{11}$$

where the notations denote the same as those in Eq. 10. Again k is the only parameter, so $\mu_{s,2} = [k]$.

By applying $T_{s,2}$ to the image, the high-contrast parts where $|\nabla I(\xi)|$ are large will be enhanced because $c(\nabla I(\xi)) \approx 1$ which implies $T_{s,2}(I(x)) \approx I(x) + |\nabla I(\xi)|$, while the low-contrast parts where $|\nabla I(\xi)|$ is small remain relatively unchanged same because $c(\nabla I(\xi)) \approx 0$ which implies $T_{s,2}(I(x)) \approx I(x)$. The result is that the high-contrast regions of the input image are highlighted.

3) Dictionaries for directionality: Directionality of texture measures the orientation of the local texture. Though the orientation of texture could be anywhere in the range from 0 to π, we decompose the texture into only the horizontal component and the vertical component because any orientation can be considered as combination of horizontal and vertical direction. There is no obvious notion of "strong" or "weak" directionality, but for the sake of consistent nomenclature, we call the horizontal component the "strong" component, and the vertical component the "weak" component, or equivalently, "strongly horizontal" and "weakly horizontal". The decomposition by directionality of our example image is shown in Fig. 2(c1), 2(c2) and 2(c3).

To decompose the texture into horizontal and vertical components, we apply the wavelet thresholding based on the stationary wavelet transform (SWT) [28] since it can preserve more details in the high frequency subbands so that more small texture edges can be retained. As shown as Eq. 12 and Eq. 13, coefficients in different sub-bands are preserved to different extents so that the texture is decomposed into components representing different directions:

$$\begin{array}{c} T_{s,3}(I) = \text{iswt}(\Psi') \\ \Psi = \text{swt}(I) \end{array}, \tag{12}$$

where $\text{swt}(\cdot)$ and $\text{iswt}(\cdot)$ are forward and inverse stationary wavelet transform respectively, the wavelet coefficients $[\Psi_a, \Psi_h, \Psi_v, \Psi_d] \in \Psi$ (approximation, horizontal, vertical, diagonal) of the input image I are filtered with the soft thresholding method [11] to $[\Psi'_a, \Psi'_h, \Psi'_v, \Psi'_d] \in \Psi$ as follows:

$$\Psi(x,y)' = \begin{cases} \Psi(x,y) & \text{if } |\Psi_h(x,y)| > \theta_h \\ 0 & \text{otherwise} \end{cases}, \tag{13}$$

where a, h, v, d represents the wavelet coefficients of the SWT. θ_h comprises the parameter set $\mu_{s,3} = [\theta_h]$, controlling which coefficients need to be removed. Only the horizontal textures where the magnitudes of the horizontal wavelet coefficients are larger than the threshold ($|\Psi_h(x,y)| > \theta_h$) are highlighted while other textures are removed from the image. Similar methods are applied to highlight the vertical textures in the image as the dictionary for "weak" component:

$$\begin{array}{c} T_{w,3}(I) = \text{iswt}(\Psi') \\ \Psi = \text{swt}(I) \end{array}, \tag{14}$$

$$\Psi(x,y)' = \begin{cases} \Psi(x,y) & \text{if } |\Psi_v(x,y)| > \theta_v \\ 0 & \text{otherwise} \end{cases}, \tag{15}$$

where all the parameters are defined same as those in Eq. 12 and Eq. 13. And similarly θ_v is the parameter in the parameter set $\mu_{w,3} = [\theta_v]$, controlling which coefficients need to be removed. Only the vertical textures where the magnitudes of the vertical wavelet coefficients are larger than the threshold ($|\Psi_v(x,y)| > \theta_v$) are highlighted while other textures are removed from the image.

4) Dictionaries for line-likeness: The decomposition based on line-likeness requires two dictionaries corresponding to textures with very similar, and very different direction in every local region, respectively. We propose 2 transformations based on image gradients as dictionaries $T_{s,4}$ and $T_{w,4}$ to represent these two types of textures:

$$T_{s,4}(I(x,y)) = \begin{cases} I(x,y) & \text{if } d(x,y) < \delta_{simi} \\ I(x,y)\cdot(1-d(x,y)) & \text{otherwise} \end{cases}, \quad d(x,y) = \mathrm{std}(g(\xi)),\ \xi \in N \tag{16}$$

$$T_{w,4}(I(x,y)) = \begin{cases} I(x,y) & \text{if } d(x,y) > \delta_{simi} \\ I(x,y)\cdot d(x,y) & \text{otherwise} \end{cases}, \quad d(x,y) = \mathrm{std}(g(\xi)),\ \xi \in N \tag{17}$$

where $d(x,y)$ measures the standard deviation ($\mathrm{std}(\cdot)$) of the gradient magnitudes $g(\xi)$ in the neighbourhood N. Since we need to separate the texture into components having similar but not identical direction, the pixel gradients $g(\xi)$ are quantized into 4 directions 0, $\frac{\pi}{2}$, $\frac{\pi}{4}$ and $\frac{3\pi}{4}$ representing horizontal, vertical and diagonal directions respectively. The threshold δ_{simi} comprises both parameter sets $\mu_{s,4} = [\delta_{simi}]$ and $\mu_{w,4} = [\delta_{simi}]$ for the line-like and non-line-like dictionaries respectively.

By applying the transformation $T_{s,4}$ to the image, the line-like regions where the standard deviation d is lower than the threshold δ_{simi} are preserved while the non-linelike regions where the standard deviation of gradients are larger than the threshold are removed. Conversely, the transformation $T_{w,4}$ preserves regions where the standard deviation of gradients is large.

A decomposition of our example image by line-likeness is shown in Fig. 2(c1), 2(d2) and 2(d3).

5) Analysis of the insensitivity: Above, we discussed how the proposed transformations highlight the components corresponding to the certain texture characteristics. In this section, we apply the transformations to the components with the opposite texture characteristics, that is, we apply the dictionaries for the weak component of a texture characteristic to the strong component, and vice versa. As shown in Fig. 3, this yields results either nearly to entirely untextured (Fig. 3(2a), (2b), (2c), (2e), (2f), (2g) and (2h)) or only having slight changes from the original component (Fig. 3(2d)). It proves that the proposed transformations cannot represent the other texture characteristics, therefore they can be used as dictionaries discriminating different textures.

6) Summary: All the filters or transformations used as the dictionaries according to different characteristics are summarized in Table I. The value of ϕ used in Algorithm 2 was 0.05 for all texture characteristics. With the proposed dictionaries, the image is decomposed into pairs of components according to different characteristics. In the next section we describe the manipulation of these components so that the individual texture in the output image will be more distinct from each other.

IV. Texture manipulation

In Section III, the input image is decomposed into $s_{s,i}$ and $s_{w,i}$ according to 4 specific texture characteristics, where $i = 1,2,3,4$ represent coarseness, contrast, directionality and line-likeness respectively. By modifying each of the individual texture components described in Section II-B and recombining them, the textures can be manipulated to be more different with respect to the specific characteristics represented by the modified components. In this section, we describe the various manipulations that are applied to transform the components.

A. Manipulations for the coarseness-decomposed components

To enhance the coarse component, $s_{s,1}$, we found that the NL-means filter works well because weak edges are further suppressed, enhancing texture coarseness.

For the enhancement of the fine component, $s_{w,1}$, we need to increase the number of edges because the coarseness is defined as the number of edges in a neighbourhood. The sticks filter [21] with stick length 5 was applied to transform the component because of its success in line and boundary detection. Most edges, even weak ones, can be detected and enhanced by sticks filter. Therefore, the fineness of the fine-texture component will be increased.

Fig. 4(a1) and 4(a2) show an example of coarse component enhancement; Fig. 4(b1) and 4(b2) demonstrate fine component enhancement.

B. Manipulations for the contrast-decomposed components

To enhance the high-contrast component $s_{s,2}$, we propose to use Laplacian filtering and median filtering together to increase the contrast as follows:

$$s'_{s,2} = s_{s,2} + \mathrm{medfilt}(|\nabla^2(s_{s,2})|) \tag{18}$$

where $|\nabla^2(\cdot)|$ is a 5×5 Laplacian filter, and $\mathrm{medfilt}(\cdot)$ is a 3×3 median filter. The magnitude of the Laplacian is high in areas where intensity change is strongly non-linear. These magnitudes are then tempered by the median filter and added back into the original texture increasing the contrast of already high-contrast neighbourhoods.

For the low-contrast component $s_{w,2}$, we need to decrease local intensity diversity for the already low-contrast regions. We propose a piece-wise power-law transformation with two thresholds T_1 and T_2 as:

$$s'_{w,2}(i) = \begin{cases} (\frac{s_{w,2}(i)}{T_1})^{\gamma_1}\cdot T_1, & \text{if } s_{w,2}(i) < T_1 \\ (\frac{s_{w,2}(i)-T_1}{T_2-T_1})^{\gamma_1}\cdot(T_2-T_1)+T_1, & \text{if } T_1 \le s_{w,2}(i) < T_2 \\ (\frac{s_{w,2}(i)-T_2}{1-T_2})^{\gamma_2}\cdot(1-T_2)+T_2, & \text{if } s_{w,2}(i) \ge T_2 \end{cases}, \tag{19}$$

where $\gamma_1 > 1$ and $0 < \gamma_2 < 1$, $T_1 = 0.35$, and $T_2 = 0.85$. This compresses the intensities of the darkest and brightest pixels, which reduces overall contrast. Examples of high- and low-contrast component enhancement are shown in Fig. 4(c1) and 4(c2), and Fig. 4(d1) and 4(d2), respectively.

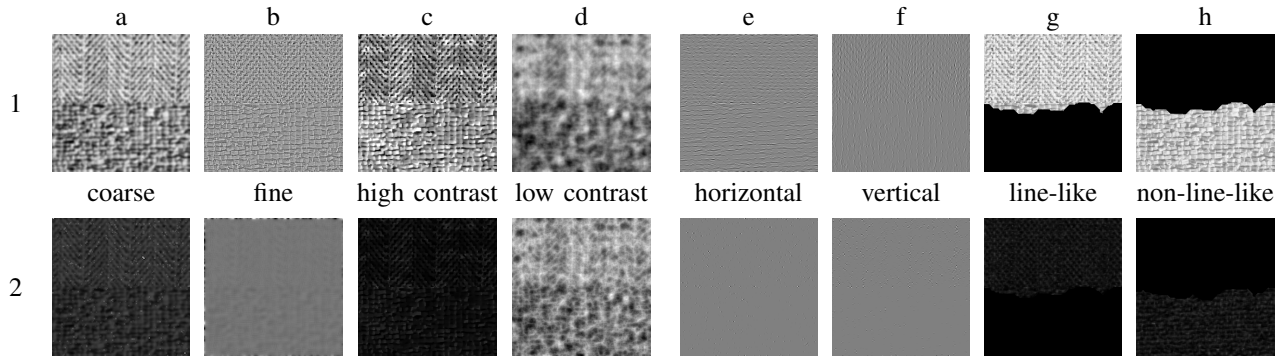


Fig. 3. Example of transforming each component in Fig. 2 by dictionaries corresponding its opposite characteristic. Top: image texture components resulting from MCA decomposition; Bottom: components transformed by the dictionary chosen for the opposite characteristic (e.g. coarse image is transformed by the fine dictionary). Columns: image components according to different characteristics. (a) and (b) are the coarseness components, (c) and (d) are the contrast components, (e) and (f) are the directionality components, (g) and (h) are the line-likeness components.

	Filters or transformations used as dictionaries for the proposed characteristics-based MCA with the selection of parameters			
	Strong		Weak	
Characteristic	Dictionary ($T_{s,i}$)	Init. Parameters $\mu_{s,i}$	Dictionary ($T_{w,i}$)	Init. Parameters $\mu_{w,i}$
Coarseness	Bilateral filter	range parameter $\sigma_r = 10$, spatial parameter $\sigma_d = 5$	wavelet thresholding	$\delta_h = 0.95\|h(x,y)\|$, $\delta_v = 0.95\|v(x,y)\|$, $\delta_d = 0.95\|d(x,y)\|$
Contrast	Anisotropic Shrink	$k = 0.95$	Anisotropic Diffusion	$k = 0.95$
Directionality	SWT thresholding	$\theta = 0.95 \max(\|\omega\|)$	SWT thresholding	$\theta = 0.95 \max(\|\omega\|)$,
Line-likeness	Gradient-similarity-based filtering	$\delta_{simi} = 0.95$,	Gradient-similarity-based filtering	$\delta_{simi} = 0.95$,

TABLE I
FILTERS OR TRANSFORMATIONS USED AS DICTIONARIES FOR THE PROPOSED CHARACTERISTICS-BASED MCA WITH THE SELECTION OF THE INITIAL PARAMETERS. THE FILTERS AND TRANSFORMATIONS ARE PROPOSED IN SEC. III.

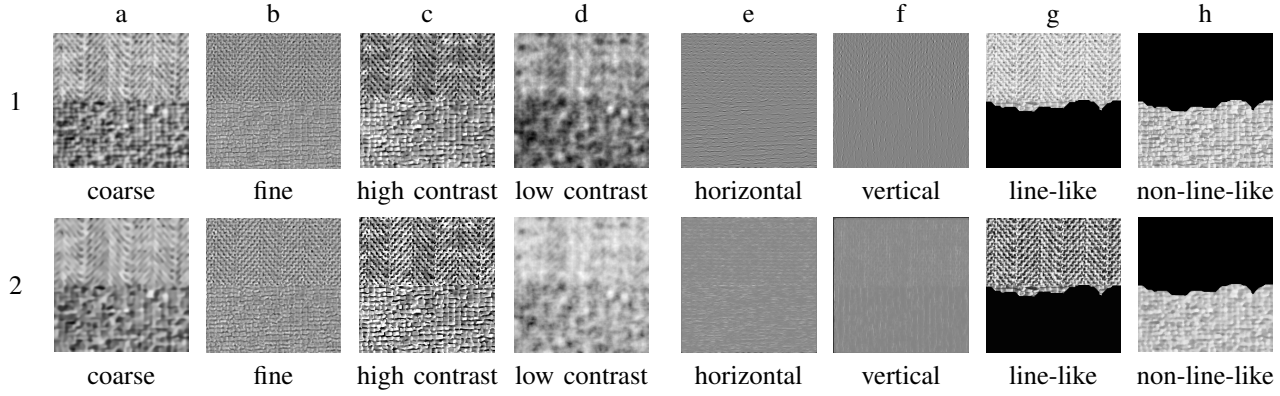


Fig. 4. Example manipulation of each component. Rows: image components before and after manipulation. Columns: image components according to different characteristics. (a) and (b) are according to coarseness, (c) and (d) are according to contrast, (e) and (f) are according to directionality, (g) and (h) are according to line-likeness. Parameters used for enhancement: $T_1 = 0.35$, $T_2 = 0.85$, $\gamma_1 = 1.5$, $\gamma_2 = 0.5$.

C. Manipulations for the directionality-decomposed components

Intuitively, directionality is enhanced by making the horizontal component more horizontal, and the vertical component more vertical. The SWT was used because it can represent textures of different directions in different sub-bands [28]. Since the wavelet coefficients in one sub-band represent intensity variation in a specific direction, they are independent of the coefficients in other sub-bands. The horizontal morphological component $s_{s,3}$ was manipulated as:

$$s_{s,3}^{'} = \text{iswt}\left(\omega_{s,3,h}^{'}, \omega_{s,3,v}^{'}, \omega_{s,3,d}^{'}\right), \tag{20}$$

where $\text{iswt}(\cdot)$ is the inverse SWT transform, $\omega_{s,3,h}$, $\omega_{s,3,v}$ and $\omega_{s,3,d}$ are the horizontal, vertical and diagonal SWT coefficients of the horizontal morphological component $s_{s,3}$:

$$[\omega_{s,3,h}, \omega_{s,3,v}, \omega_{s,3,d}] = \text{swt}\left(s_{s,3}\right), \tag{21}$$

where $\text{swt}(\cdot)$ denotes the forward SWT transform, and the coefficients are manipulated as:

$$\begin{aligned} \omega_{s,3,h}^{'} &= a \cdot \omega_{s,3,h} \\ \omega_{s,3,v}^{'} &= 0 \\ \omega_{s,3,d}^{'} &= \omega_{s,3,d} \end{aligned}$$

where a is the amplifying coefficient which was set to 1.5 in experiments. For the vertical morphological component $\alpha_{w,3}$, we amplified the vertical wavelet coefficients $\omega_{w,3,v}$ and set the horizontal wavelet coefficients $\omega_{w,3,h}$ to zero. Fig. 4(e1) and 4(e2) show an example of horizontal directionality enhancement; Fig. 4(f1) and 4(f2) show an example of vertical directionality enhancement.

D. Manipulations for the line-likeness-decomposed components

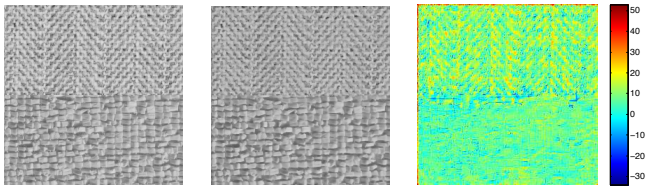
We make use of adaptive histogram equalization [22] to enhance the line-like component because it increases the intensity contrast, making the line or boundary between different primitives more obvious, while decreasing the intensity contrast between two texture elements with very similar intensities, and removing very weak edges. In the experiment, we set the filtering window size as 8×8 and the contrast enhancement limit as 0.01, showing in Table II. As a result, the line-likeness of the line-like component will be increased as shown in Fig. 4(g1) and 4(g2).

For the non-line-like component, $s_{w,4}$, we apply the power-law transform with $\gamma > 1$:

$$s^{'}_{w,4} = s^{\gamma}_{w,4}, \tag{22}$$

where $s_{w,4}$ is the non-line-like component. In our experiments we used $\gamma = 1.5$. Therefore, boundaries and the local contrast are suppressed and the line-likeness of the non-line-like component will be further decreased, as shown in Fig. 4(h1) and 4(h2).

Table II summarizes the manipulation methods for each component and the selection of the parameters. After recombination of these manipulated components $s^{'}_{s,i}$ and $s^{'}_{w,i}$ as in Eq. 6, the textures in the resulting image are more different with respect to the chosen texture characteristics. However, as shown in Fig. 5, textures are not degraded but are instead enhanced to accentuate their own properties.



(a) input image (b) manipulated image (c) "difference" image

Fig. 5. Manipulated image after recombining the manipulated components in Fig. 4. (a) is the input image as shown in Fig. 4(c1); (b) is the manipulated image after decomposition, manipulation and re-combination; (c) shows the intensity differences between the input and manipulated image which was converted to a signed colour-mapped image to highlight the location and magnitude of intensity changes from (a) to (b).

V. Experiments and Analysis

A. Experimental images and comparator methods

The proposed method is tested and evaluated on synthetic textural images and real-world images containing different textures. The synthetic images are synthesized by combining

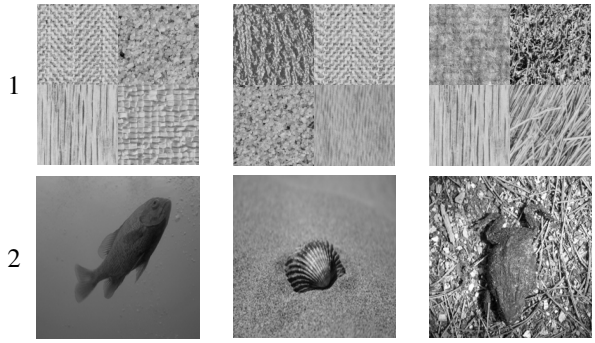


Fig. 6. Samples of the textural images used in our experiments. Row 1 are synthetic texture by combining 4 textures from Brodatz texture database and SIPI texture database. Row 2 are real world images from Alpert et al.'s segmentation evaluation database.

textures from the Brodatz texture database [6] and the SIPI texture database [2]. The real-world images and their ground truth are from the Berkeley segmentation dataset [1]. Fig. 6 shows several sample images.

The performance of the proposed method is measured by comparing the results from different texture-based segmentation methods applied to the textures modified by different methods. The following texture-enhanced methods are used as comparators:

1) VISUShrink [9] (VISU), which enhances texture by removing noise via shrinking wavelet coefficients in high-frequency sub-bands not exceeding certain thresholds;
2) the traditional "cartoon+texture" MCA-based filtering [5] (MCA-CT), which separates the image into 3 parts: cartoon, texture, and noise components, then recombines only the cartoon and texture parts, which eliminates the detected noise and enhances texture;
3) unsharp masking [23] (UM) which enhances texture by emphasizing its high frequency contents;
4) shock filtering [31] (SHK), which smooths along the coherent texture flow orientations and reduces diffusivity at non-coherent structures, which enhances texture details; and
5) the coherence-enhancing diffusion filtering [30] (CDF),which preserves strong discontinuities at edges while removing artifacts from smooth regions, so that image textures are enhanced.

The comparator methods were selected to include at least one representative of each of the major types of texture enhancement methods discussed in Section I. To further measure the performance of the manipulation of the texture components proposed in this paper, we also use the MCA with proposed dictionaries but no manipulation of the components prior to recombination (MCA-NM) as another comparator to demonstrate the the manipulations are central to the success our method. Image segmentation tests were carried out as follows:

1) the test images were enhanced as described in Sections II, III, and IV, as well as with each of the comparator methods listed above;
2) Tamura features, consisting of the four specific char-

Characteristic	Filters or transformations used to manipulate components and the selection of parameters			
	Strong		Weak	
Coarseness	NL-means filter	similarity window: 7×7, filtering parameter: $h = 0.1$	Stick filter	stick length: $l = 5$
Contrast	Laplacian Filter Median Filter	filtering window: 5×5, filtering window: 3×3	Piece-wise power-law transformation	$\gamma_1 = 1.5$, $\gamma_2 = 0.5$, $T_1 = 0.35$, $T_2 = 0.85$
Directionality	SWT enhancing	amp. coefficient: $a = 1.5$,	SWT enhancing	amp. coefficient: $a = 1.5$
Line-likeness	Adaptive histogram equalization	filtering window: 8×8, contrast enhancement limit: 0.01	power-law transformation	$\gamma = 1.5$

TABLE II
FILTERS OR TRANSFORMATIONS USED TO MANIPULATE COMPONENTS AND THE SELECTION OF PARAMETERS. THE MANIPULATIONS ARE PROPOSED IN SEC. IV.

acteristics: coarseness, contrast, directionality and line-likeness, were extracted from the 15×15 neighbourhood of every pixel of each original test image, and each enhanced test image from each enhancement method to create feature maps;

3) the feature maps were segmented using various segmentation methods: (1) mean-shift [10], (2) k-means clustering [15], (3) Gaussian mixture model [3] (GMM), and (4) adaptive thresholding [19];
4) the segmentation accuracy of each segmentation of each original and each enhanced test image was computed by comparing the segmented result with the ground truth using the metric:

$$c = \frac{|\{S(x,y) \mid S(x,y) = G(x,y)\}|}{N}, \qquad (23)$$

where S denotes the segmented label image, G represents the ground truth label image, and N is the total number of the pixels in the image;
5) the segmentation accuracy of the test images before and after enhancement were compared to evaluate the effect of different texture enhancement methods on segmentation accuracy;

B. Results for segmenting synthetic images

Fig. 7 shows the results of segmentation by various segmentation algorithms of an example synthetic image after texture enhancement by VISUShrink (VISU), unsharp masking (UM), shock filtering (SHK), coherence-enhancing diffusion (CDF), "cartoon+texture" MCA filtering (MCA-CT), the "texture characteristic" MCA filtering without manipulation (MCA-NM) and the proposed method (proposed). The VISU and MCA-CT filter preserve the significant textures well but degrade the weak textures because processes them as noise. The shock filter can enhance the texture edges well but it breaks some smooth regions. The coherence-enhancing diffusion changes the shapes of the textures a lot by merging the small smooth regions close to each other. The unsharp masking can enhance the texture, especially the local contrast of the texture very well, but it creates unwanted edge effects at the same time. Moreover, since all these methods process the textures as a single "texture" component, all the different textures in the image are enhanced to the same extent. Differently, the proposed method can enhance different textures to different extents with respect to their own properties because it separates the textures into components representing different visual characteristics and modifies these components in different ways. As a result, the proposed texture enhancement method results in higher segmentation accuracy than other enhancement methods for every segmentation algorithm. Table III shows the average performance of our method and comparator texture enhancement methods in combination with various segmentation algorithms over 150 synthetic images.

C. Results for segmenting real-world images

Fig. 8 shows the transformation and segmentation of an example real-world image after texture enhancement by the same methods as those in Sec. V-B. Use of the proposed method prior to segmentation results in higher segmentation accuracy than other enhancement methods for every segmentation algorithm. Table. IV shows the average performance of our method and comparator texture enhancement methods in combination with various segmentation algorithms over 50 images.

D. Discussion

The reason for the superior performance of our method is because for the given textures in an image, the clusters of local texture characteristic vectors for different textures are moved in a mutually divergent manner – a property not possessed by the comparator methods. We demonstrate this property using the four-texture image shown in Fig. 9.

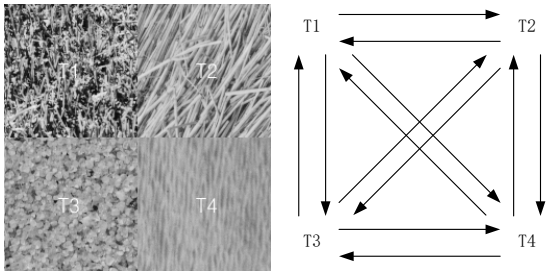


Fig. 9. An image with 4 textures. $T1$, $T2$, $T3$, $T4$ are 4 different textures, resulting in 12 ordered-pair-wise Mahalanobis distances. Textures become mutually more different due to a particular enhancement method if the 12 Mahalanobis distances are increased after processing.

The texture characteristics of coarseness, contrast, directionality, and line-likeness were computed for each texture

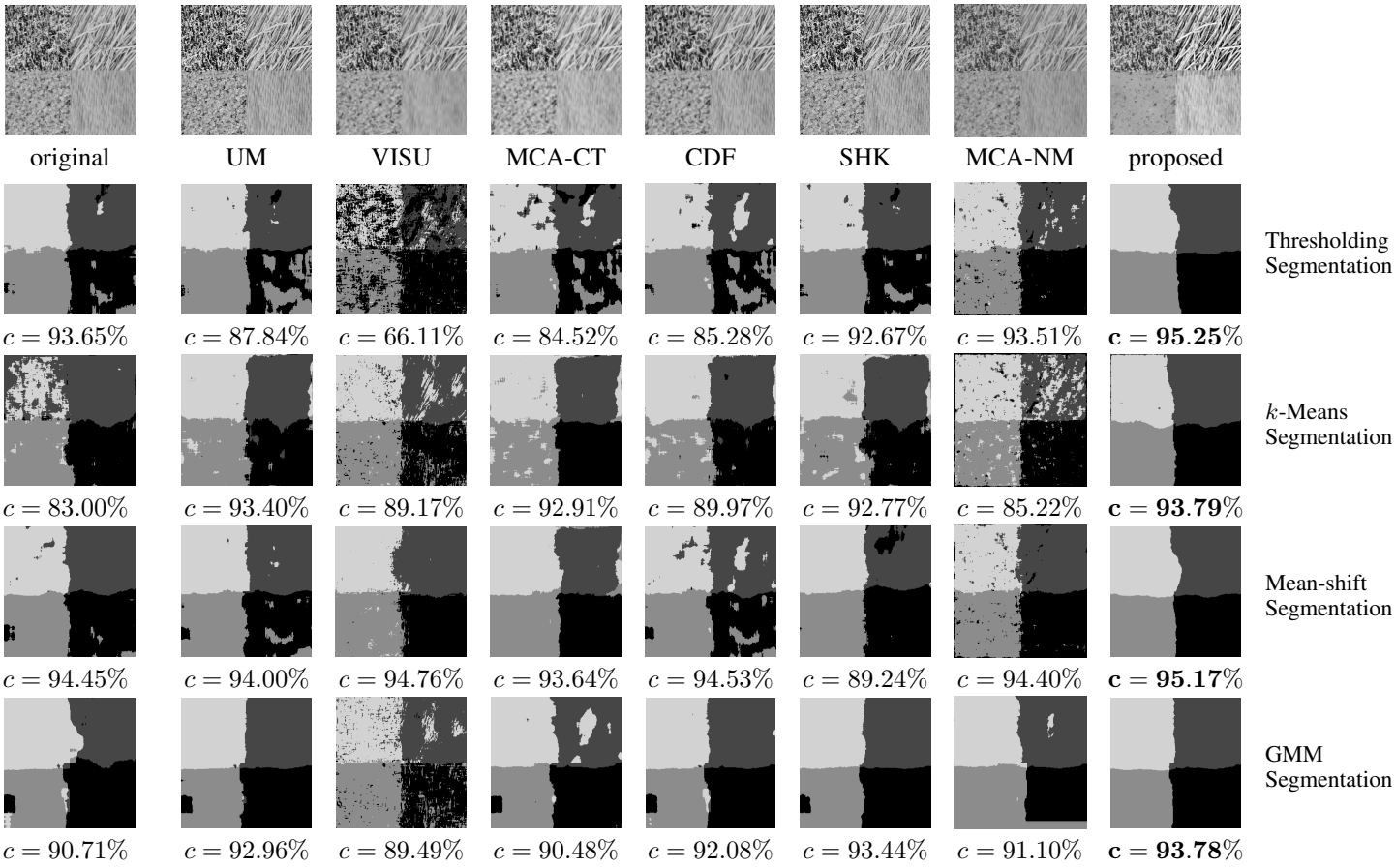


Fig. 7. Segmentation results for different enhanced images by different segmentation methods. Rows: image segmented by different segmentation methods. Columns: images enhanced by different enhancment methods: without manipulation (original), unsharp masking (UM), VISUShrink (VISU), MCA-based filter (MCA-CT), coherence-enhancing diffusion (CDF), shock filter (SHK), the "texture characteristic" MCA filtering without manipulation (MCA-NM) and the proposed method. The segmentation accuracy c is calculated by Eq. 23. The best result in each row is given in boldface.

	original	UM	VISU	MCA-CT	CDF	SHK	MCA-NM	proposed	MCA-CCD	MCA-CCL	MCA-CDL	MCA-CONDL
Thresholding	78.76%	85.75%	78.65%	85.35%	88.12%	88.36%	91.16%	**93.24%**	86.85%	85.43%	85.39%	83.66%
K-means	85.52%	92.90%	80.76%	90.14%	87.54%	91.46%	90.35%	**95.46%**	82.18%	82.98%	81.50%	82.53%
Mean-shift	91.54%	94.65%	94.68%	92.06%	90.78%	92.38%	95.77%	**96.80%**	90.25%	89.46%	87.70%	87.15%
GMM	88.78%	91.86%	89.48%	91.25%	89.68%	91.76%	92.43%	**94.75%**	89.98%	85.05%	86.22%	88.83%

TABLE III

AVERAGE SEGMENTING ACCURACY OVER 150 SYNTHETIC TEXTURE IMAGES FOR VARIOUS COMBINATIONS OF TEXTURE ENHANCEMENT METHODS AND SEGMENTATION ALGORITHMS. OUR PROPOSED METHOD EXHIBITS THE BEST AVERAGE PERFORMANCE FOR ALL SEGMENTATION ALGORITHMS.

	original	UM	VISU	MCA-CT	CDF	SHK	MCA-NM	proposed	MCA-CCD	MCA-CCL	MCA-CDL	MCA-CONDL
Thresholding	94.76%	82.37%	94.15%	94.85%	82.65%	82.97%	92.64%	**95.15%**	93.86%	93.20%	91.18%	90.55%
K-means	91.48%	83.45%	92.24%	94.52%	82.34%	83.75%	93.27%	**94.85%**	85.54%	80.28%	80.11%	82.45%
Mean-shift	84.64%	84.91%	91.75%	91.65%	84.43%	84.81%	91.80%	**92.77%**	92.42%	89.71%	90.07%	88.60%
GMM	94.56%	83.44%	90.48%	91.44%	93.51%	91.77%	94.38%	**95.55%**	93.95%	92.24%	92.66%	93.89%

TABLE IV

AVERAGE SEGMENTING ACCURACY OVER 50 REAL WORLD IMAGES FOR VARIOUS COMBINATIONS OF TEXTURE ENHANCEMENT METHODS AND SEGMENTATION ALGORITHMS. OUR PROPOSED METHOD EXHIBITED THE BEST AVERAGE PERFORMANCE FOR ALL SEGMENTATION ALGORITHMS.

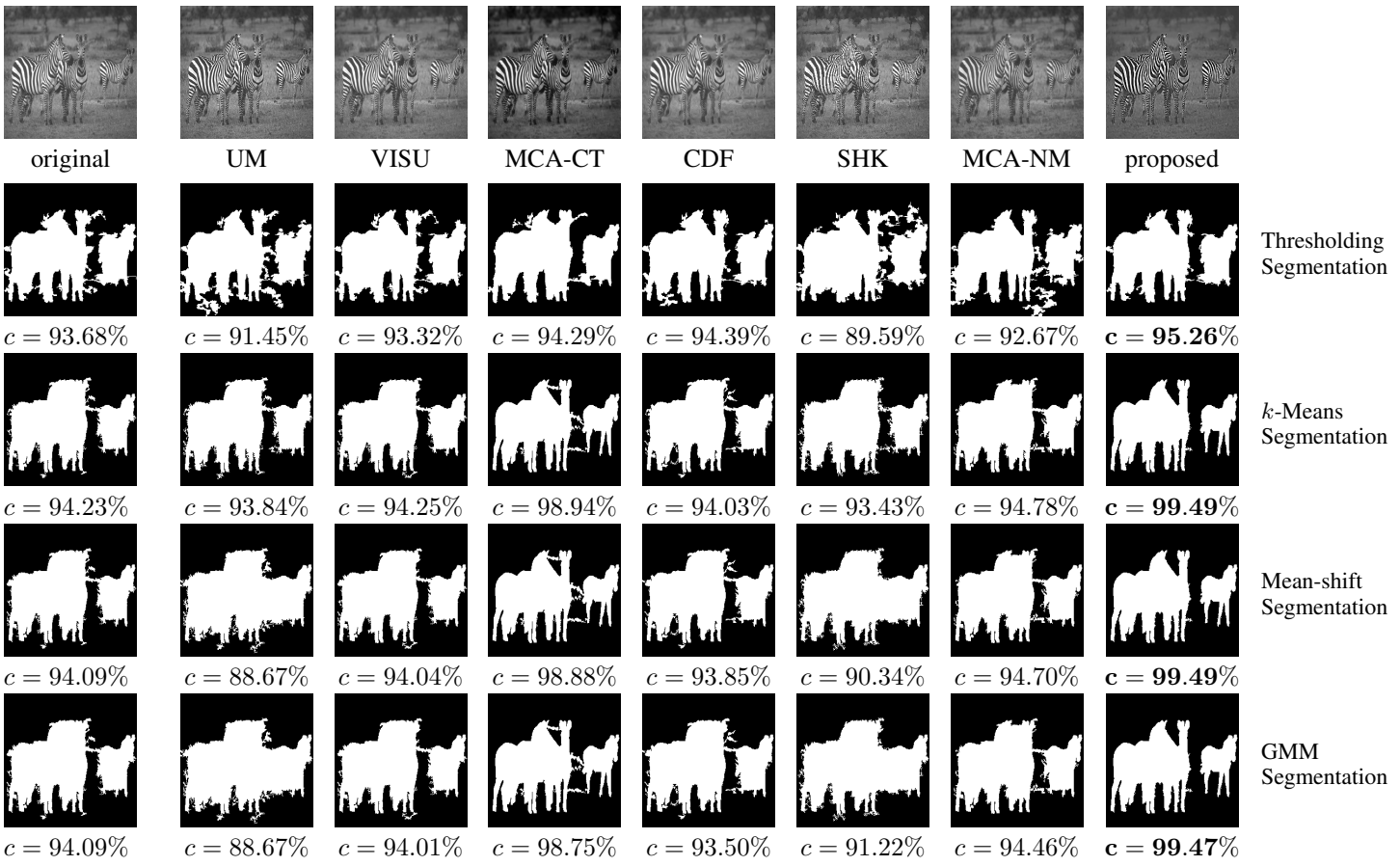


Fig. 8. Segmentation results of different enhanced real-world images by different segmentation methods. Rows: image segmented by different segmentation methods. Columns: image enhanced by different enhancement methods: without manipulation (original), unsharp masking (UM), VISUShrink (VISU), MCA-based filter (MCA-CT), coherence-enhancing diffusion (CDF), shock filter (SHK), the "texture characteristic" MCA filtering without manipulation (MCA-NM) and the proposed method. The segmentation accuracy c is calculated by Eq. 23. The best result in each row is given in boldface.

in Fig. 8 in a local window of size 15×15 about each pixel; this was done for both the original image, and the image after processing with our proposed MCA-based texture enhancement method. This resulted in four clusters of four-dimensional vectors for the original image ("before" clusters) and four clusters for the processed image ("after" clusters). We then computed the Mahalanobis distance between each "before" cluster mean and each other "before" cluster, resulting in twelve Mahalanobis distances $d^i_{\text{before}}, i = 1, \ldots, 12$. These 12 Mahalanobis distances are illustrated on the right side of Fig. 9. The same was done for the "after" clusters, resulting in twelve Mahalanobis distances $d^i_{\text{after}}, i = 1, \ldots, 12$, with the mean-cluster pairs indexed by i in the same order as for the "before" images. Finally, we computed the difference between the "before" and "after" Mahalanobis distances for each mean-cluster pairing:

$$D_i = d^i_{after} - d^i_{before}, \qquad i = 1, 2, \ldots, 12. \tag{24}$$

Mahalanobis distance [20] measures the distance between a point and a cluster of points normalized with respect to the spread of the distribution of points in the cluster (intuitively, distances along each dimensional axis can be thought of as being in units of *standard deviations*). If all of the "before" and "after" Mahalanobis distances between mean-cluster pairs increase after processing, it means that the clusters have mutually moved away from each other in the multidimensional space and become more separated, which means that the textures have become mutually more different with respect to their descriptions by the four texture characterstics. This is precisely the behaviour we observed with our method, while all of the comparator methods showed, at best, a mixture of clusters that move further apart and closer together.

Fig. 10 shows all twelve D_i (Mahalanobis distance differences) for the 4 textures in Fig. 9 after processing by our enhancement method and our comparator methods. The proposed method is the only one for which the Mahalanobis distances between clusters is consistently increased which explains its superior performance when used as a preprocessing step prior to segmentation. The comparator methods all exhibit pairs of clusters that become less separated, and even when clusters become more separated, the magnitude of the increased separation is generally less than that for our proposed method.

E. Necessity of the proposed texture characteristics

To determine if all the four characteristics are necessary for decomposition and manipulation of our proposed method, we tested using three of the four texture characteristics to decompose and manipulate the images, seg-

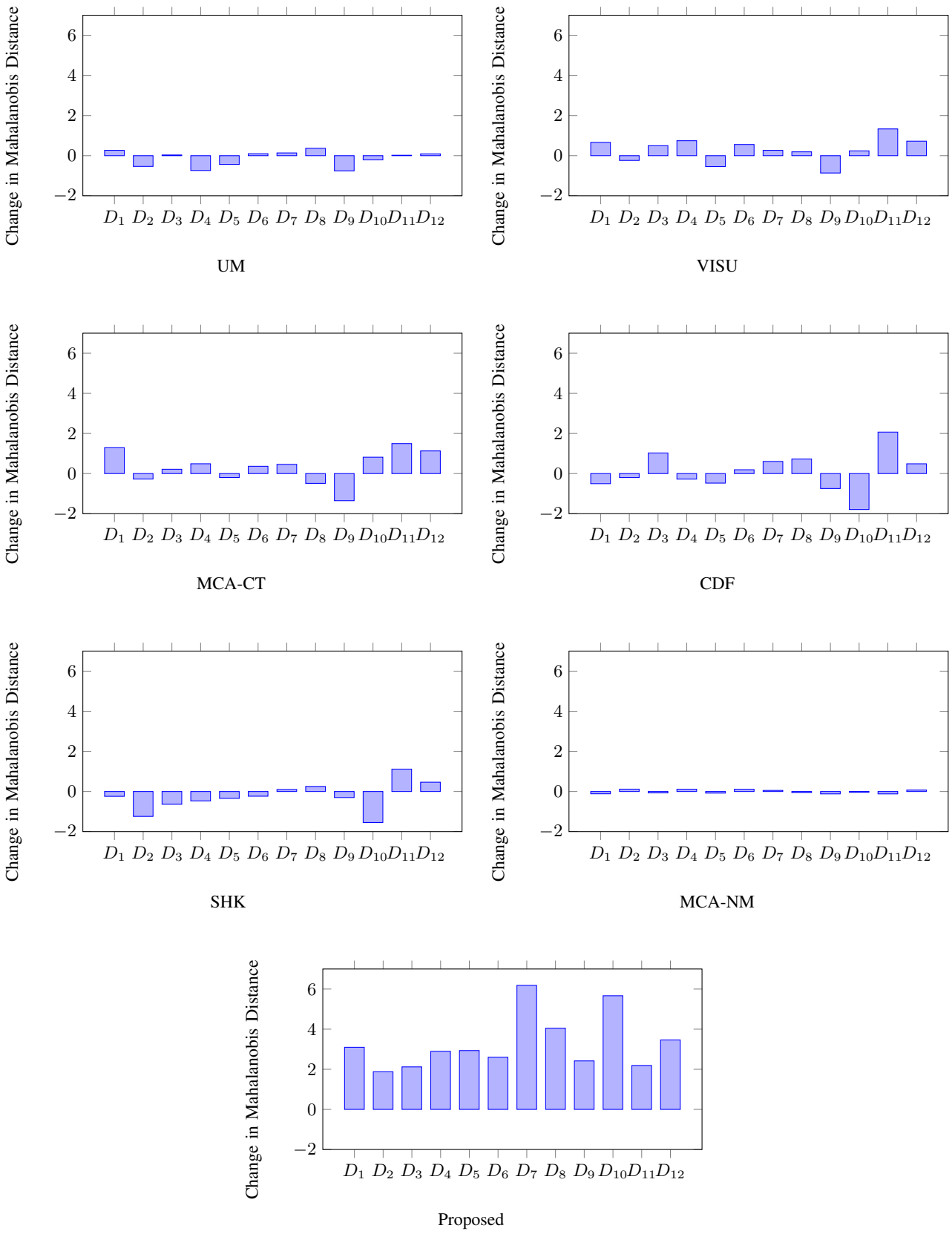


Fig. 10. The changes in the twelve Mahalanobis distances between clusters of local texture characteristic vectors in the image in Figure 9 after processing with unsharp masking (UM), VISUShrink (VISU), MCA-CT filtering (MCA-CT), coherence enhancing diffusion (CDF), shock filtering (SHK), 'texture characteristic" MCA filtering without manipulation (MCA-NM) and "texture characteristic" MCA filtering with proposed manipulation method (proposed). Only the proposed method increases the Mahalanobis distance between all clusters of textures descriptions which is the main mechanism behind the improved image segmentation results presented in Sections V-B and V-C. Each D_i is computed using Eq. 24.

menting the manipulated images with the four segmentation methods in Sec. V-A, and comparing the resulting segmentation accuracies with the proposed method. The texture manipulation methods based on three characteristics are: 1) "coarseness+contrast+directionality" MCA (MCA-CCD), 2) "coarseness+directionality+line-likeness" MCA (MCA-CDL), 3) "contrast+directionality+line-likeness" MCA (MCA-CONDL), and 4) "coarseness+contrast+line-likeness" MCA (MCA-CCL). Fig. S1 and Fig. S2 in the supplementary materials show the transformation and segmentation of example synthetic and real-world images after texture enhancement by the above four methods and the proposed method. Fig. S3 shows the changes in the twelve Mahalanobis distances between clusters of local texture characteristics vectors in the image in Fig. S1 after processing with all the comparator methods. The average performance of the comparators in segmenting synthetic and real world images are shown in the last 4 columns in Table III and Table IV respectively. Our proposed method which manipulates all the four characteristics results in consistently superior results compared to only using three characteristics. Moreover, using only three of the four texture characteristics with our method frequently results in worse average performance than the comparator methods. Therefore, the four proposed texture characteristics are necessary to guarantee a superior average performance over the comparator methods.

F. Running time

The texture transformation method was tested using a MATLAB R2013 implementation on a Intel Core 2 Duo(2.66 GHz) iMac with 8GB RAM. We ran the algorithm to process 200 images with the size of 256×256, including 150 synthetic images and 50 real-world images The average running time for these test images is 52.057 seconds per image.

VI. Conclusion

We novelly proposed to decompose the texture image using the MCA method according to different texture characteristics: coarseness, contrast, directionality and line-likeness. For every morphological component, we proposed transformations to enhance the characteristic captured by that component. The experimental results showed that the proposed texture enhancement method successfully enhanced the difference between textures with respect to the chosen texture characteristics while better preserving their visual appearance compared to other methods which led to improved texture-based segmentation results.

References

[1] The Berkeley Segmentation Dataset and Benchmark. http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images.html.

[2] The USC-SIPI Image Database. http://sipi.usc.edu/services/database/Database.html.

[3] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(7):719–725, Jul 2000.

[4] J. Bobin, Y. Moudden, J.-L. Starck, and M. Elad. Multichannel morphological component analysis. *Proceedings of Spars05*, pages 103–106, 2005.

[5] J. Bobin, J.-L. Starck, J. Fadili, Y. Moudden, and D. Donoho. Morphological Component Analysis: An Adaptive Thresholding Strategy. *Image Processing, IEEE Transactions on*, 16(11):2675–2681, Nov 2007.

[6] P. Brodatz. *A Photographic Album for Arts and Design*. Dover Publishing Co., Toronto, Canada, 1966.

[7] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65, 2005.

[8] A. Chambolle. Partial differential equations and image processing. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 1, pages 16–20, Nov 1994.

[9] H. A. Chipman, E. D. Kolaczyk, and R. E. McCulloch. Adaptive Bayesian wavelet shrinkage. *Journal of the American Statistical Association*, 92(440):1413–1421, 1997.

[10] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, May 2002.

[11] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[12] D. L. Donoho and J. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

[13] M. Fadili, J.-L. Starck, J. Bobin, and Y. Moudden. Image decomposition and separation using sparse representations: an overview. *Proceedings of the IEEE*, 98(6):983–994, 2010.

[14] R. Gribonval and M. Nielsen. Sparse representations in unions of bases. *Information Theory, IEEE Transactions on*, 49(12):3320–3325, 2003.

[15] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[16] L. Hong, Y. Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):777–789, 1998.

[17] T. Huang, G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(1):13–18, 1979.

[18] A. K. Jain. *Fundamentals of digital image processing*, volume 3. Prentice-Hall Englewood Cliffs, 1989.

[19] T. Kurita, N. Otsu, and N. Abdelmalek. Maximum likelihood thresholding based on population mixture models. *Pattern Recognition*, 25(10):1231–1240, 1992.

[20] Mahalanobis P. C. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[21] S. Pathak, D. Haynor, and Y. Kim. Edge-guided boundary delineation in prostate ultrasound images. *Medical Imaging, IEEE Transactions on*, 19(12):1211–1219, 2000.

[22] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355 – 368, 1987.

[23] A. Polesel, G. Ramponi, and V. J. Mathews. Image enhancement via adaptive unsharp masking. *IEEE transactions on image processing*, 9(3):505–510, 2000.

[24] J.-L. Starck, M. Elad, and D. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *Image Processing, IEEE Transactions on*, 14(10):1570–1582, Oct 2005.

[25] J.-L. Starck, Y. Moudden, J. Bobin, M. Elad, and D. L. Donoho. Morphological component analysis, 2005.

[26] H. Tamura, S. Mori, and T. Yamawaki. Textural Features Corresponding to Visual Perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460–473, 1978.

[27] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, 1998.

[28] X. H. Wang, R. Istepanian, and Y. H. Song. Microarray image enhancement by denoising using stationary wavelet transform. *NanoBioscience, IEEE Transactions on*, 2(4):184–189, 2003.

[29] J. Weickert. *Anisotropic diffusion in image processing*, volume 1. Teubner Stuttgart, 1998.

[30] J. Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2-3):111–127, 1999.

[31] J. Weickert. Coherence-enhancing shock filters. In B. Michaelis and G. Krell, editors, *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 1–8. Springer Berlin Heidelberg, 2003.

[32] M. Zibulevsky and B. A. Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural computation*, 13(4):863–882, 2001.