CHAPTER 5

COLOUR TEXTURE ANALYSIS

Paul F. Whelan and Ovidiu Ghita

Vision Systems Group, School of Electronic Engineering
Dublin City University, Dublin, Ireland
E-mail: paul.whelan@dcu.ie & ghitao@eeng.dcu.ie

This chapter presents a novel and generic framework for image segmentation using a compound image descriptor that encompasses both colour and texture information in an adaptive fashion. The developed image segmentation method extracts the texture information using low-level image descriptors (such as the Local Binary Patterns (LBP)) and colour information by using colour space partitioning. The main advantage of this approach is the analysis of the textured images at a micro-level using the local distribution of the LBP values, and in the colour domain by analysing the local colour distribution obtained after colour segmentation. The use of the colour and texture information separately has proven to be inappropriate for natural images as they are generally heterogeneous with respect to colour and texture characteristics. Thus, the main problem is to use the colour and texture information in a joint descriptor that can adapt to the local properties of the image under analysis. We will review existing approaches to colour and texture analysis as well as illustrating how our approach can be successfully applied to a range of applications including the segmentation of natural images, medical imaging and product inspection.

1. Introduction

Image segmentation is one of the most important tasks in image analysis and computer vision[1,2,3,4]. The aim of image segmentation algorithms is to partition the input image into a number of disjoint regions with similar

properties. Texture and colour are two such image properties that have received significant interest from research community[1,3,5,6], with prior research generally focusing on examining colour and texture features as separate entities rather than a unified image descriptor. This is motivated by the fact that although innately related, the inclusion of colour and texture features in a coherent image segmentation framework has proven to be more difficult that initially anticipated.

1.2 Texture Analysis

Texture is an important property of digital images, although image texture does not have a formal definition it can be regarded as a function of the variation of pixel intensities which form repeated patterns[6,7]. This fundamental image property has been the subject of significant research and is generally divided into four major categories: statistical, model-based, signal processing and structural[2,5,6,8], with specific focus on statistical and signal processing (e.g. multi-channel Gabor filtering) methods. One key conclusion from previous research[5,6] is the fact that the filtering-based approaches can adapt better than statistical methods to local disturbances in texture and illumination.

Statistical measures analyse the spatial distribution of the pixels using features extracted from first and second–order histograms[6,8]. Two of the most investigated statistical methods are the gray-level differences[9] and co-occurrence matrices[7]. These methods performed well when applied to synthetic images but their performance is relatively modest when applied to natural images unless these images are defined by uniform textures. It is useful to note that these methods appear to be used more often for texture classification rather than texture-based segmentation. Generally these techniques are considered as the base of evaluation for more sophisticated texture analysis techniques and since their introduction these methods have been further advanced. Some notable statistical techniques include the work of Kovalev and Petrou[10], Elfadel and Picard[11] and Varma and Zisserman[12].

Signal processing methods have been investigated more recently. With these techniques the image is typically filtered with a bank of filters of differing scales and orientations in order to capture the frequency

changes[13,14,15,16,17,18]. Early signal processing methods attempted to analyse the image texture in the Fourier domain, but these approaches were clearly outperformed by techniques that analyse the texture using multi-channel narrow band Gabor filters. This approach was firstly introduced by Bovik et al[13] when they used quadrature Gabor filters to segment images defined by oriented textures. They conclude that in order to segment an image the spectral difference sampled by narrow-band filters is sufficient to discriminate between adjacent textured image regions. This approach was further advanced by Randen and Husoy[18] while noting that image filtering with a bank of Gabor filters or filters derived from a wavelet transform[19,20] is computationally demanding. In their paper they propose the methodology to compute optimized filters for texture discrimination and examine the performance of these filters with respect to algorithmic complexity/feature separation on a number of test images. They conclude that the complicated design required in calculating the optimized filters is justified since the overall filter-based segmentation scheme will require a smaller number of filters than the standard implementation that uses Gabor filters. A range of signal processing based texture segmentation techniques have been proposed, for more details the reader can consult the reviews by Tuceryan and Jain[6], Materka and Strzelecki[8] and Chellappa et al[5].

1.2 Colour Analysis

Colour is another important characteristic of digital images which has naturally received interest from the research community. This is motivated by advances in imaging and processing technologies and the proliferation of colour cameras. Colour has been used in the development of algorithms that have been applied to many applications including object recognition[21,22], skin detection[23], image retrieval[24,25,26] and product inspection[27,28]. Many of the existing colour segmentation techniques are based on simple colour partitioning (clustering) techniques and their performance is appropriate only if the colour information is locally homogenous.

Colour segmentation algorithms can be divided into three categories, namely, pixel-based colour segmentation techniques, area based

segmentation techniques and physics based segmentation techniques[3,29,30]. The pixel-based colour segmentation techniques are constructed on the assumption that colour is a constant property in the image to be analysed and the segmentation task can be viewed as the process of grouping the pixels in different clusters that satisfy a colour uniformity criteria. According to Skarbek and Koschan[30] the pixel based colour segmentation techniques can be further divided into two main categories: histogram-thresholding segmentation and colour clustering techniques. The histogram-based segmentation techniques attempt to identify the peaks in the colour histogram[21,27,31,32,33] and in general provide a coarse segmentation that is usually the input for more sophisticated techniques. Clustering techniques have been widely applied in practice to perform image segmentation[34]. Common clustering-based algorithms include K-means[35,36,37], fuzzy C-means[35,38], mean shift[39] and Expectation-Maximization[40,41]. In their standard form the performance of these algorithms have been shown to be limited since the clustering process does not take into consideration the spatial relationship between the pixels in the image. To address this limitation Pappas[37] has generalized the standard K-means algorithm to enforce the spatial coherence during the cluster assignment process. This algorithm was initially applied to greyscale images and was later generalized by Chen et al[42]. Area based segmentation techniques are defined by the region growing and split and merge colour segmentation schemes[30,43,44,45,46]. As indicated in the review by Lucchese and Mitra[3] the common characteristic of these methods is the fact that they start with an inhomogeneous partition of the image and they agglomerate the initial partitions into disjoint image regions with uniform characteristics until a homogeneity criteria is upheld. Area-based approaches are the most investigated segmentation schemes, due in part to the fact that the main advantage of these techniques over pixel-based methods is that the spatial coherence between adjacent pixels is enforced during the segmentation process. In this regard, notable contributions are represented by the work of Panjwani and Healey[47], Tremeau and Borel[46], Celenk[34], Cheng and Sun[44], Deng and Manjunath[48], Shafarenko et al[32] and Moghaddamzadeh and Bourbakis[45]. For a complete evaluation of these colour segmentation techniques refer to the reviews by Skarbek and

Koschan[30], Lucchese and Mitra[3] and Cheng et al[29]. The third category of colour segmentation approaches is represented by the physics-based segmentation techniques and their aim is to alleviate the problems generated by uneven illumination, highlights and shadows which generally lead to over-segmentation[49,50,51]. Typically these methods require a significant amount of a-priori knowledge about the illumination model and the reflecting properties of the objects that define the scene. These algorithms are not generic and their application is restricted to scenes defined by a small number of objects with known shapes and reflecting properties.

1.3 Colour-Texture Analysis

The colour segmentation techniques mentioned previously are generally application driven, whereas more sophisticated algorithms attempt to analyze the local homogeneity using complex image descriptors that include the colour and texture information. The use of colour and texture information collectively has strong links with the human perception and the development of an advanced unified colour-texture descriptor may provide improved discrimination over viewing texture and colour features independently. Although the motivation to use colour and texture information jointly in the segmentation process is clear, how best to combine these features in a colour-texture mathematical descriptor is still an open issue. To address this problem a number of researchers augmented the textural features with statistical chrominance features[25,52,53]. Although simple, this approach produced far superior results than texture only algorithms and in addition the extra computational cost required by the calculation of colour features is negligible when compared with the computational overhead associated with the extraction of textural features. In this regard, Mirmehdi and Petrou[54] proposed a colour-texture segmentation approach where the image segmentation is defined as a probabilistic process embedded in a multiresolution approach. In other words, they blurred the image to be analysed at different scale levels using multiband smoothing algorithms and they isolated the core colour clusters using the K-means algorithm, which in turn guided the segmentation process from blurred to focused

images. The experimental results indicate that their algorithm is able to produce accurate image segmentation even in cases when it has been applied to images with poorly defined regions. A related approach is proposed by Hoang et al[55] where they applied a bank of Gabor filters on each channel of an image represented in the wavelength-Fourier space. Since the resulting data has large dimensionality (each pixel is represented by a 60 dimensional feature vector) they applied Principal Component Analysis (PCA) to reduce the dimension of the feature space. The reduced feature space was clustered using a K-means algorithm, followed by the application of a cluster merging procedure. The main novelty of this algorithm is the application of the standard multiband filtering approach to colour images and the reported results indicate that the representation of colour-texture in the wavelength-Fourier space proved to be accurate in capturing texture statistics. Deng and Manjunath[48] proposed a different colour-texture segmentation method that is divided into two main computational stages. In the first stage the colours are quantized into a reduced number of classes while in the second stage a spatial segmentation is performed based on texture composition. They argue that decoupling the colour similarity from spatial distribution was beneficial since it is difficult to analyse the similarity of the colours and their distributions at the same time. Tan and Kittler[33] developed an image segmentation algorithm where the texture and colour information are used as separate attributes within the segmentation process. In their approach the texture information is extracted by the application of a local linear transform while the colour information is defined by the six colour features derived from the colour histogram. The use of colour and texture information as separate channels in the segmentation process proved to be opportune and this approach has been adopted by many researchers. Building on this, the paper by Pietikainen et al[31] evaluates the performance of a joint colour Local Binary Patterns (LBP) operator against the performance of the 3D histograms calculated in the Ohta colour space. They conclude that the colour information sampled by the proposed 3D histograms is more powerful then the texture information sampled by the joint LBP distribution. This approach has been further advanced by Liapis and Tziritas[24] where they developed a colour-texture approach used for image

retrieval. In their implementation they extracted the texture features using the Discrete Wavelet Frames analysis while the colour feature were extracted using 2D histograms calculated from chromaticity components of the images converted in the CIE Lab colour space.

In this chapter we detail the development of a novel colour texture segmentation technique (referred to as CTex) where the colour and texture information are combined adaptively in a composite image descriptor. In this regard the texture information is extracted using the LBP method and the colour information by using an Expectation-Maximization (EM) space partitioning technique. The colour and texture features are evaluated in a flexible split and merge framework were the contribution of colour and texture is adaptively modified based on the local colour uniformity. The resulting colour segmentation algorithm is modular (i.e. it can be used in conjunction with any texture and colour descriptors) and has been applied to a large number of colour images including synthetic, natural, medical and industrial images. The resulting image segmentation scheme is unsupervised and generic and the experimental data indicates that the developed algorithm is able to produce accurate segmentation.

2. Algorithm Overview

The main computational components of the image segmentation algorithm detailed in this chapter are illustrated in Fig. 1. The first step of the algorithm extracts the texture features using the Local Binary Patterns method as detailed by Ojala[56]. The colour feature extraction is performed in several steps. In order to improve the local colour uniformity and increase the robustness to changes in illumination the input colour image is subjected to anisotropic diffusion-based filtering. An additional step is represented by the extraction of the dominant colours that are used for initialization of the EM algorithm that is applied to perform the colour segmentation. From the LBP/C image and the colour segmented image, our algorithm calculates two types of local distributions, namely the colour and texture distributions that are used as input features in a highly adaptive split and merge architecture. The output of the split and merge algorithm has a blocky structure and to

improve the segmentation result obtained after merging the algorithm applies a pixelwise procedure that exchanges the pixels situated at the boundaries between various regions using the colour information computed by the EM algorithm.

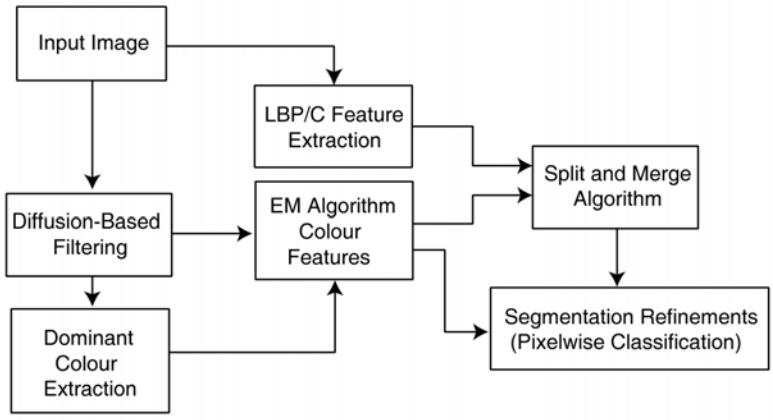


Fig. 1. Overview of the CTex colour-texture segmentation algorithm.

3. Extraction of Colour-Texture Features

As indicated in Section 1 there are a number of possible approaches for extracting texture features from a given input image, the most relevant approaches either calculate statistics from co-occurrence[7] matrices or attempt to analyze the interactions between spectral bands calculated using multi-channel filtering[13,15,17]. In general, texture is a local attribute in the image and ideally the texture features need to be calculated within a small image area. But in practice the texture features are typically calculated for relatively large image blocks in order to be statistically relevant. The Local Binary Patterns (LBP) concept developed by Ojala et al[56] attempts to decompose the texture into small texture units and the texture features are defined by the distribution (histogram) of the LBP values calculated for each pixel in the region under analysis. These LBP distributions are powerful texture descriptors since they can be used to discriminate textures in the input image irrespective of their size (the dissimilarity between two or more textures can be determined by using a

histogram intersection metric). An LBP texture unit is represented in a 3×3 neighbourhood which generates 2^8 possible standard texture units. In this regard, the LBP texture unit is obtained by applying a simple threshold operation with respect to the central pixel of the 3×3 neighbourhood.

$$T = t\big(s(g_0 - g_c),\ldots, s(g_{P-1} - g_c)\big)$$
$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{1}$$

where T is the texture unit, g_c is the grey value of the central pixel, g_P are the pixels adjacent to the central pixel in the 3×3 neighbourhood and s defines the threshold operation. For a 3×3 neighbourhood the value of P is 9. The LBP value for the tested pixel is calculated using the following relationship:

$$LBP = \sum_{i=1}^{P-1} s(g_i - g_c) * 2^i \tag{2}$$

where $s(g_i - g_c)$ is the value of the thresholding operation illustrated in equation (1). As the LBP values do not measure the greyscale variation, the LBP is generally used in conjunction with a contrast measure, referred to as LBP/C. For our implementation this contrast measure is the normalized difference between the grey levels of the pixels with a LBP value of 1 and the pixels with a grey level 0 contained in the texture unit. The distribution of the LBP/C of the image represents the texture spectrum. The LBP/C distribution can be defined as a 2D histogram of size $256 \times b$, where the b defines the number of bins required to sample the contrast measure (Fig. 2). In practice the contrast measure is sampled in 8 or 16 bins (experimentally it has been observed that best results are obtained when b=8).

As mentioned previously the LBP texture descriptor has good discriminative power (see Fig. 2 where the LBP distributions for different textures are illustrated) but the main problem associated with LBP/C texture descriptors is the fact that they are not invariant to

rotation and scale (see Fig. 3). However the sensitivity to texture rotation can be an advantageous property for some applications such as the inspection of wood parquetry, while for other applications such as the image retrieval it can be a considerable drawback. Ojala et al[57] have addressed in the development of a multiresolution rotationally invariant LBP descriptor.

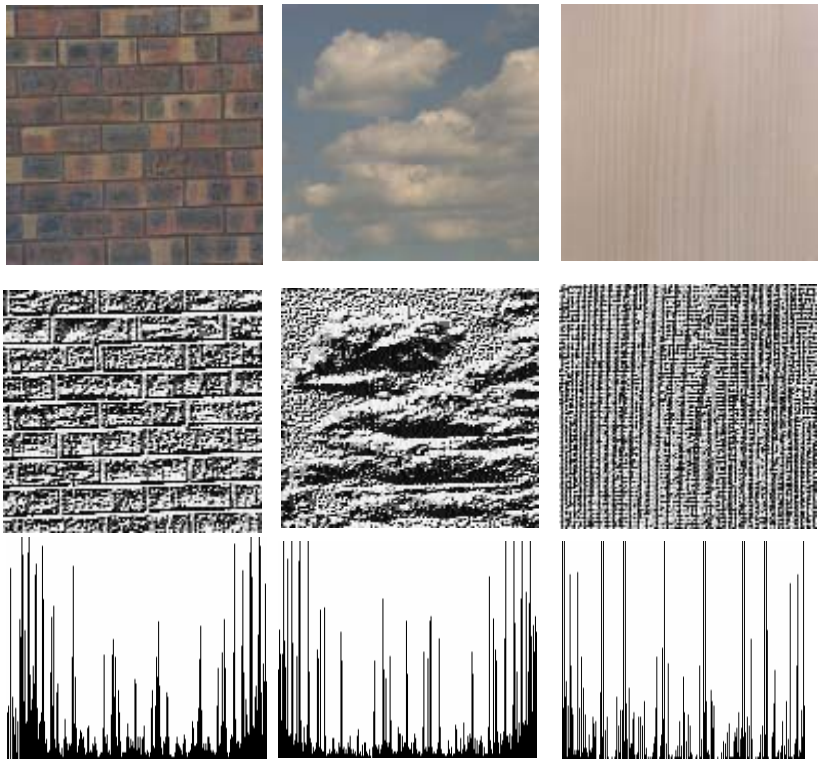
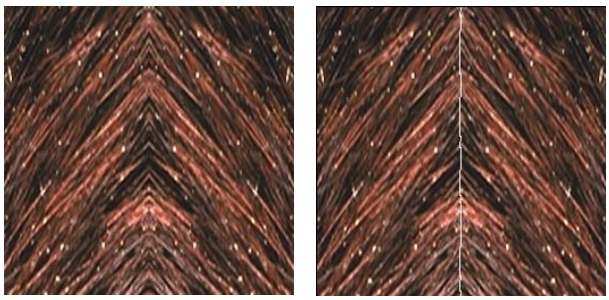


Fig. 2. The LBP distributions associated with different textures. First row – Original images (brick, clouds and wood from the VisTex database[66]). Second row – LBP images. Third row – LBP distributions (horizontal axis: LBP value, vertical axis: the number of elements in each bin).



(a) (b)

Fig. 3. Segmentation of a test image that demonstrates the LBP/C texture descriptors sensitivity to texture rotation. (a) Original image defined by two regions with similar texture and different orientations (from the VisTex database[66]). (b) Colour-texture segmentation result.

3.1. Diffusion-Based Filtering

In order to improve the local colour homogeneity and eliminate the spurious regions caused by image noise we have applied an anisotropic diffusion-based filtering to smooth the input image (as originally developed by Perona and Malik[58]). Standard smoothing techniques based on local averaging or Gaussian weighted spatial operators[59] reduce the level of noise but this is obtained at the expense of poor feature preservation (i.e. suppression of narrow details in the image). To avoid this undesired effect in our implementation we have developed a filtering strategy based on anisotropic diffusion where smoothing is performed at intra regions and suppressed at region boundaries[41,58,60]. This non-linear smoothing procedure can be defined in terms of the derivative of the flux function:

$$u_t = div(D(|\nabla u|)\nabla u) \tag{3}$$

where u is the input data, D represents the diffusion function and t indicates the iteration step. The smoothing strategy described in equation (3) can be implemented using an iterative discrete formulation as follows:

$$I_{x,y}^{t+1} = I_{x,y}^{t} + \lambda \sum_{j=1}^{4} [D(\nabla_j I)\nabla_j I] \tag{4}$$

$$D(\nabla I) = e^{-\left(\frac{\nabla I}{k}\right)^2} \in (0,1] \tag{5}$$

where $\nabla_j I$ is the gradient operator defined in a 4-connected neighbourhood, λ is the contrast operator that is set in the range $0<\lambda<0.16$ and k is the diffusion parameter that controls the smoothing level. It should be noted that in cases where the gradient has high values, $D(\nabla I)\rightarrow 0$ and the smoothing process is halted.

3.2. Expectation-Maximization (EM) Algorithm

The EM algorithm is the key component of the colour feature extraction. The EM algorithm is implemented using an iterative framework that attempts to calculate the maximum likelihood between the input data and a number of Gaussian distributions (Gaussian Mixture Models - GMM)[40,41]. The main advantage of this probabilistic strategy over rigid clustering algorithms such as K-means is its ability to better handle the uncertainties during the mixture assignment process. Assuming that we try to approximate the data using M mixtures, the mixture density estimator can be calculated using the following expression:

$$p(x \mid \Phi) = \sum_{i=1}^{M} \alpha_i \, p_i(x \mid \Phi_i) \tag{6}$$

where x=[x_1, …, x_k] is a k–dimensional vector, α_i is the mixing parameter for each GMM and $\Phi_i=\{\sigma_i, m_i\}$. The values σ_i, m_i are the standard deviation and the mean of the mixture. The function p_i is the Gaussian distribution and is defined as follows:

$$p_i(x \mid \Phi_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{\|x-m_i\|^2}{2\sigma_i^2}}, \quad \sum_{i=0}^{M} \alpha_i = 1 \tag{7}$$

The algorithm consists of two steps, the expectation and maximization step. The expectation step (E-step) is represented by the expected log-likelihood function for the complete data as follows:

$$Q(\Phi, \Phi(t)) = E[\log p(X, Y \mid \Phi) \mid X, \Phi(t)] \tag{8}$$

where Φ(t) are the current parameters and Φ are the new parameters that optimize the increase of Q. The M-step is applied to maximize the result obtained from the E-step.

$$\begin{gathered} \Phi(t+1) = \arg\max_{\Phi} Q(\Phi \mid \Phi(t)) \text{ and} \\ Q(\Phi(t+1), \Phi(t)) \geq Q(\Phi, \Phi(t)) \end{gathered} \tag{9}$$

The E and M steps are applied iteratively until the increase of the log-likelihood function is smaller than a threshold value. The updates for GMMs can be calculated as follows:

$$\alpha_i(t+1) = \frac{\sum_{j=1}^{N} p(i \mid x_j, \Phi(t))}{N} \tag{10}$$

$$m_i(t+1) = \frac{\sum_{j=1}^{N} x_j \, p(i \mid x_j, \Phi(t))}{\sum_{j=1}^{N} p(i \mid x_j, \Phi(t))} \tag{11}$$

$$\sigma_i(t+1) = \frac{\sum_{j=1}^{N} p(i \mid x_j, \Phi(t)) \left\| x_j - m_i(t+1) \right\|^2}{\sum_{j=1}^{N} p(i \mid x_j, \Phi(t))} \tag{12}$$

where $p(i \mid x_j, \Phi) = \dfrac{\alpha_i p_i(x_j \mid \Phi_i)}{\sum_{k=1}^{M} \alpha_k p_k(x_j \mid \Phi_K)}$.

The EM algorithm is a powerful space partitioning technique but its main weakness is its sensitivity to the starting condition (i.e. the initialization of the mixtures Φ_i). The most common procedure to initialize the algorithm consists of a process that selects the means of the mixture by picking randomly data points from input image. This initialization procedure is far from optimal and may force the algorithm to converge to local minima. Another disadvantage of the random initialization procedure is the fact that the space partitioning algorithm may produce different results when executed with the same input data. To alleviate this problem a large number of algorithms have been developed to address the initialization of space partitioning techniques[41,61,62].

3.3. EM Initialization using Colour Quantization

The solution we have adopted to initialize the parameters for mixtures $\Phi_i=\{\sigma_i, m_i\}$, $i = 1...M$ with the dominant colours from the input image, consists of extracting the peaks from the colour histogram calculated after the application of colour quantization. For this implementation we applied linear colour quantization[63,64] by linearly re-sampling the number of colours on each colour axis. The dominant colours contained in the image represented in the colour space C are extracted by selecting the peaks from the colour histogram as follows:

$$P_j = \arg\max_{C}(ColorHistogram) \text{ , j = 1,…, M} \tag{13}$$

Experimentally it has been observed that the EM initialization is optimal when the quantization levels are set to low values between 2 to 8 colours for each component (i.e. the quantized colour image will have 8×8×8 colours - 3 bits per each colour axis - if the quantization level is set to 8). This is motivated by the fact that for low quantization levels the colour

histogram is densely populated and the peaks in the histogram are statistically relevant. The efficiency of this quantization procedure is illustrated in Fig. 4 where we illustrate the differences between initializing the EM algorithm using the more traditional random procedure and our approach (see Ilea and Whelan[41] for more details).

4. Image Segmentation Algorithm

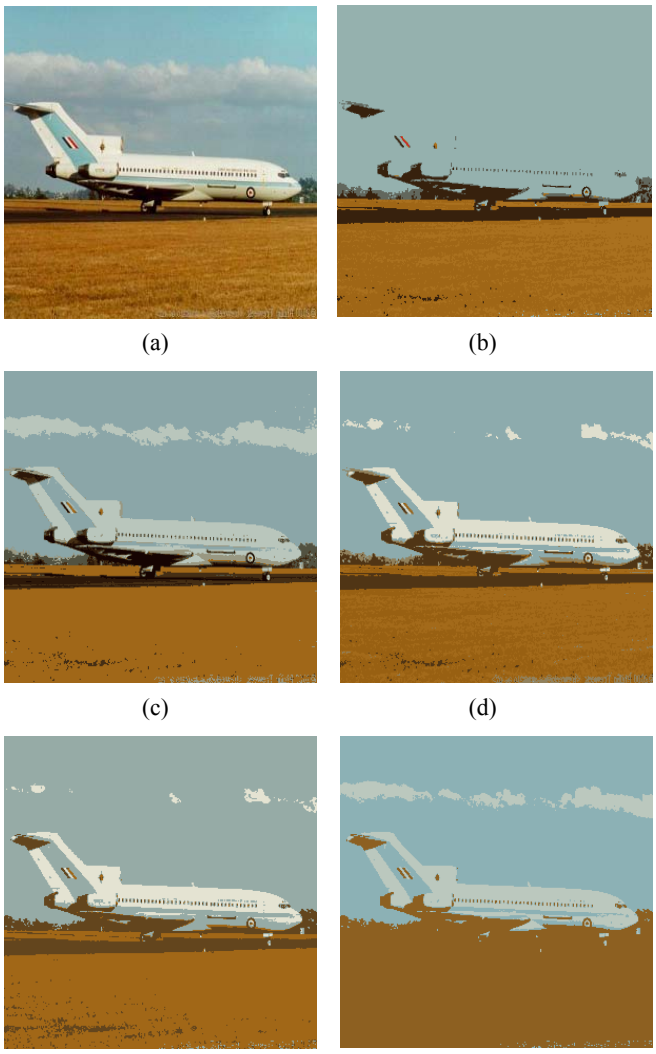
The image segmentation method used in our implementation is based on a split and merge algorithm[65] that adaptively evaluates the colour and texture information. The first step of the algorithm recursively splits the image hierarchically into four sub-blocks using the texture information extracted using the Local Binary Patterns/Contrast (LBP/C) method[53,56,57]. The splitting decision evaluates the uniformity factor of the region under analysis that is sampled using the Kolmogorov-Smirnov Metric (MKS). The Kolmogorov-Smirnov metric is a non-parametric test that is employed to evaluate the similarity between two distributions as follows:

$$MKS(s,m) = \sum_{i=0}^{n} \left| \frac{H_s(i)}{n_s} - \frac{H_m(i)}{n_m} \right| \tag{14}$$

where n represents the number of bins in the sample and model distributions (H_s and H_m), n_s and n_m are the number of elements in the sample and model distributions. We have adopted the MKS similarity measure in preference to other statistical metrics (such as the G-statistic or χ^2 test) as the MKS measure is normalized and its result is bounded.

To evaluate the texture uniformity within the region in question, the pairwise similarity values of the four sub-blocks are calculated and the ratio between the highest and lowest similarity values are compared with a threshold value (split threshold).

$$U = \frac{MKS_{\max}}{MKS_{\min}} \tag{15}$$



(a) (b)

(c) (d)

(e) (f)

Fig. 4. EM colour segmentation. (a) Original image[71]. (b) Colour segmentation using random initialization (best result). (c-f) Colour segmentation using colour quantization. (c) Quantization level 4. (d) Quantization level 8. (e) Quantization level 16. (f) Quantization level 64.

The region is split if the ratio U is higher than the split threshold. The split process continues until the uniformity level imposed by the split threshold (S_{th}) is upheld or the block size is smaller than a predefined size value (for this implementation the smallest block size has been set to 16×16 or 32×32 based on the size of the input image). During the splitting process two distributions are computed for each region resulting after the split process, the LBP/C distribution that defines the texture and the distribution of the colour labels computed using the colour segmentation algorithm previously outlined. The processing steps required by the split phase of the algorithm are illustrated in Fig. 5.

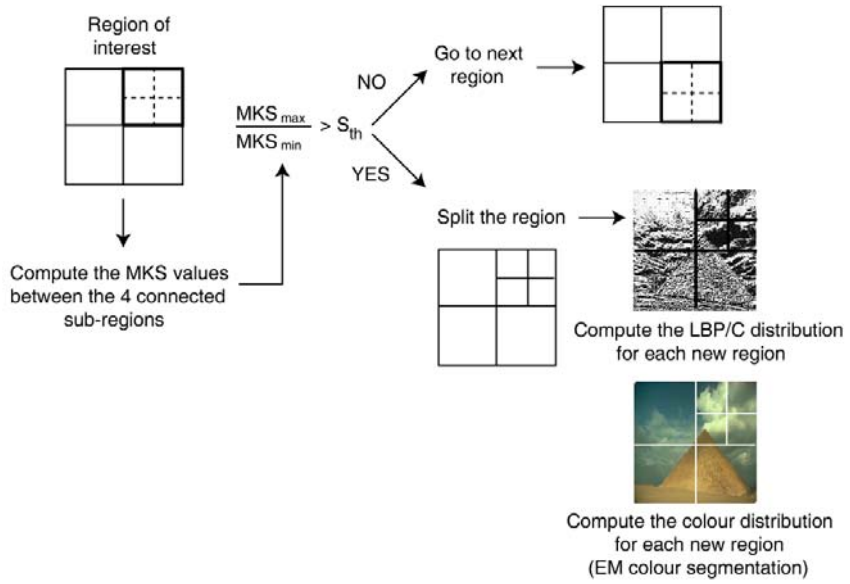


Fig. 5. The split phase of the CTex image segmentation algorithm.

The second step of the image segmentation algorithm applies an agglomerative merging procedure on the image resulting after splitting in order to join the adjacent regions that have similar colour-texture characteristics. This procedure calculates the merging importance (MI) between all adjacent regions resulting from the split process and the

adjacent regions with the smallest MI value are merged. Since the MI values sample the colour-texture characteristics for each region, for this implementation we developed a novel merging scheme[41] that is able to locally adapt to the image content (texture and colour information) by evaluating the uniformity of the colour distribution. In this regard, if the colour distribution is homogenous (i.e. it is defined by one dominant colour) the weights w_1 and w_2 in equation (16) are adjusted to give the colour distribution more importance. Conversely, if the colour distribution is heterogeneous the texture will have more importance. The calculation of the weights employed to compute the MI values for merging process (see equation 16) is illustrated in equations (17 and 18).

$$MI(r_1, r_2) = w_1 * MKS(TD_1, TD_2) + w_2 * MKS(CD_1, CD_2) \tag{16}$$

where r_1, r_2 represent the adjacent regions under evaluation, w_1 and w_2 are the weights for texture and colour distributions respectively, MKS defines the Kolmogorov-Smirnov Metric, TD_i is the texture distribution for region i and CD_i is the colour distribution for region i. The weights w_1 and w_2 are calculated as follows:

$$K_i = \frac{\arg\max_C(CD_i)}{N_i} \quad , K_i \in (0,1] \text{ and } i = 1,2 \tag{17}$$

where $\arg\max_C(CD_i)$ is the bin with the maximum number of elements in the distribution CD_i, N_i is the total number of elements in the distribution CD_i and C is the colour space.

$$w_2 = \frac{\sum_{i=1}^{2} K_i}{2} \quad \text{and} \quad w_1 = 1 - w_2 \tag{18}$$

where w_1 and w_2 are the texture and colour weights employed in equation (16). The merging process is iteratively applied until the minimum value for MI is higher than a pre-defined merge threshold (i.e. $MI_{min} > M_{th}$), see Fig. 6.

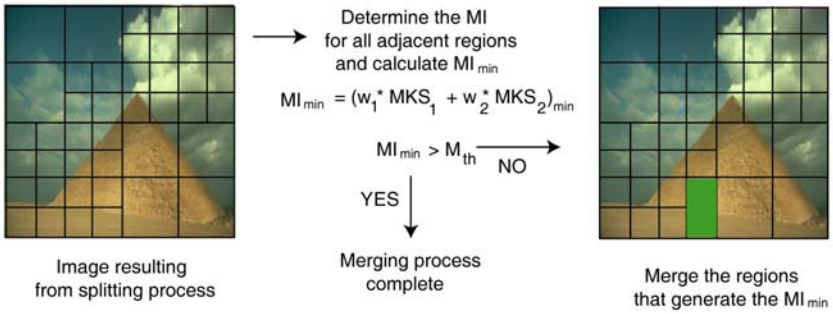


Fig. 6. The merge phase of the image segmentation algorithm (the adjacent regions with the smallest MI value are merged and are highlighted in the right hand side image).

The resulting image after the application of the merging process has a blocky structure since the regions resulting from the splitting process are rectangular. To compensate for this issue the last step of the algorithm applies a pixelwise procedure that exchanges the pixels situated at the boundaries between adjacent regions using the colour information computed from the colour segmentation algorithm previously outlined. This procedure calculates for each pixel situated on the border the colour distribution within an 11×11 window and the algorithm evaluates the MKS value between this distribution and the distributions of the regions which are 4-connected with the pixel under evaluation. The pixel is re-labelled (i.e. assigned to a different region) if the smallest MKS value is obtained between the distribution of the pixel and the distribution of the region that has a different label than the pixel under evaluation. This procedure is repeated iteratively until the minimum MKS value obtained for border pixels is higher than the merge threshold (M_{th}) to assure that the border refinement procedure does not move into regions defined by different colour characteristics. We have evaluated the pixelwise procedure for different window sizes and this experimentation indicates that window sizes of 11×11 and 15×15 provided optimal performance. For small window sizes the colour distribution became sparse and the borders between the image regions are irregular. Typical results achieved

after the application of the pixelwise procedure are illustrated in Figs. 7 and 8. Figure 8d illustrates the limitation of the LBP/C texture operator when dealing with randomly oriented textures (see the segmentation around the border between the rock and the sky).

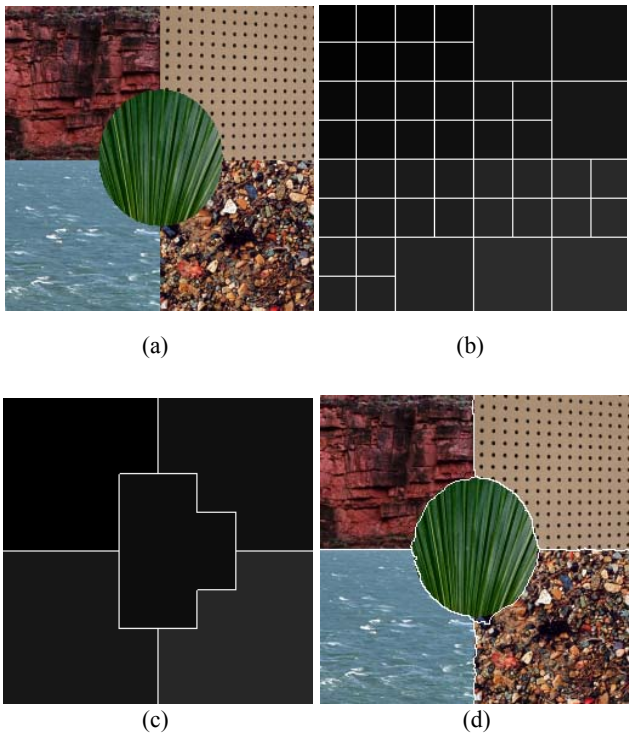
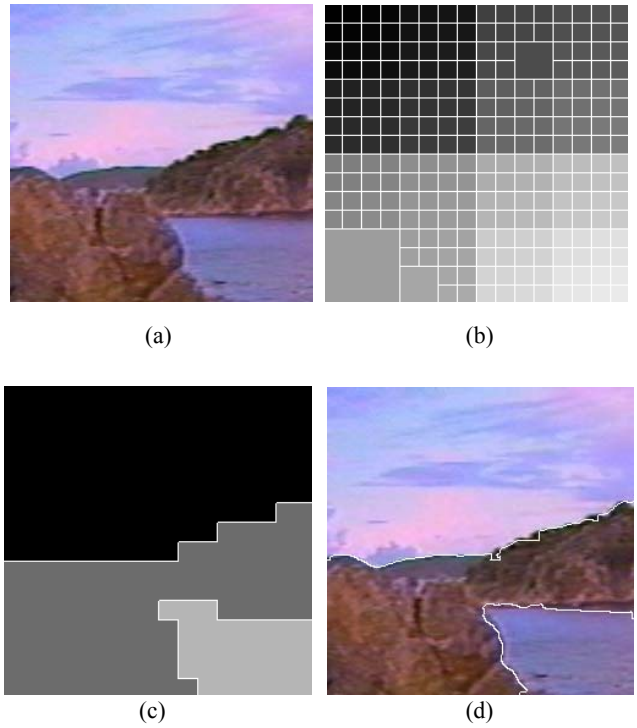


Fig. 7. Image segmentation process. (a) Original image. (b) Image resulting from splitting (block size 32×32). (c) Image resulting from merging. (d) Final segmentation after the application of the pixelwise procedure.

5. Experimental Results

The experiments were conducted on synthetic colour mosaic images (using textures from VisTex database[66]), natural and medical images. In order to examine the contribution of the colour and texture information in the segmentation process the split and merge parameters were set to the values that return the minimum segmentation error. The other key

parameter is the diffusion parameter k and its influence on the performance of the algorithm will be examined in detail.



(a) (b)

(c) (d)

Fig. 8. Image segmentation process. (a) Original image[56]. (b) Image resulting from splitting (block size 16×16). (c) Image resulting from merging. (d) Final segmentation after the application of the pixelwise procedure.

5.1 Segmentation of Synthetic Images

As the ground truth data associated with natural images is difficult to extract and is influenced by the subjectivity of the human operator, the efficiency of this algorithm is evaluated on mosaic images created using various VisTex reference textures. In our tests we have used 15 images where the VisTex textures were arranged in different patterns and a number of representative images are illustrated in Fig. 9. Since the split

and merge algorithm would be favoured if we perform the analysis on test images with rectangular regions, in our experiments we have also included images with complex structures where the borders between different regions are irregular.

Fig. 9. Some of the VisTex images used in our experiments. (From top to bottom and left to right: Image 3, Image 9, Image 10, Image 11, Image 13 and Image 5)

An important issue for our research is to evaluate the influence of the colour and texture information in the segmentation process. In this regard, we have examined the performance of the algorithm in cases where texture alone, colour alone and colour-texture information is used in the segmentation process.

The experimental results are illustrated in Table 1 and it can be observed that texture and colour alone results are generally inferior to results obtained when texture and colour local distributions are used in the segmentation process. The balance between the texture and colour is performed by the weights w_1 and w_2 in equation (16) and to obtain the texture and colour alone segmentations these parameters were overridden with manual settings (i.e. w_1=1, w_2=0 for texture alone segmentation and w_1=0, w_2=1 for colour alone segmentation). When the colour and texture distributions were used in a compound image descriptor these parameters

were computed automatically using the expressions illustrated in equations (17) and (18). For all experiments the initial number of mixtures (GMMs) are set to 10 (M =10). The inclusion of colour and texture in a compound image descriptor proved to improve the overall segmentation results. The contribution of colour to the segmentation process will be more evident when the algorithm is applied to natural images where the textures are more heterogeneous than those in the test images defined by VisTex textures.

Table 1. Performance of our CTex colour-texture segmentation algorithm when applied to VisTex mosaic images (% error given).

Image Index	Texture-only (%)	Colour-only (%)	Colour-Texture (%)
Image 1	0.33	2.49	0.45
Image 2	0.98	2.08	1.77
Image 3	5.47	2.10	0.88
Image 4	4.31	4.94	1.81
Image 5	8.77	4.30	3.17
Image 6	4.70	5.11	4.06
Image 7	33.55	2.52	6.57
Image 8	1.82	5.25	2.07
Image 9	18.47	1.15	0.50
Image 10	3.63	2.07	1.89
Image 11	33.73	2.52	1.81
Image 12	5.25	2.77	2.29
Image 13	40.56	4.39	3.87
Image 14	3.18	0.58	0.75
Image 15	2.60	1.94	1.94
Overall:	11.15	2.97	2.25

The segmentation results reported in Table 1 were obtained in the condition that the split and merge parameters are set to arbitrary values to obtain best results. From these parameters the split threshold has a lesser importance since the result from the split phase does not need to be optimized. In our experiments we have used large values for this parameter that assure almost a uniform splitting of the input image and as

a result the split threshold has a marginal influence on the performance of the algorithm. The merge threshold has a strong impact on the final results and experimentally it has been determined that this threshold parameter should be set to values in the range (0.6-1.0) depending on the complexity of the input image (the merge threshold should be set to lower values when the input image is heterogeneous (complex) with respect to colour and texture information). The optimal value for this parameter can be determined by using the algorithm in a supervised scheme by indicating the final number of regions that should result from the merging stage. A typical example that illustrates the influence of the merge threshold on the final segmented result is illustrated in Fig. 10.

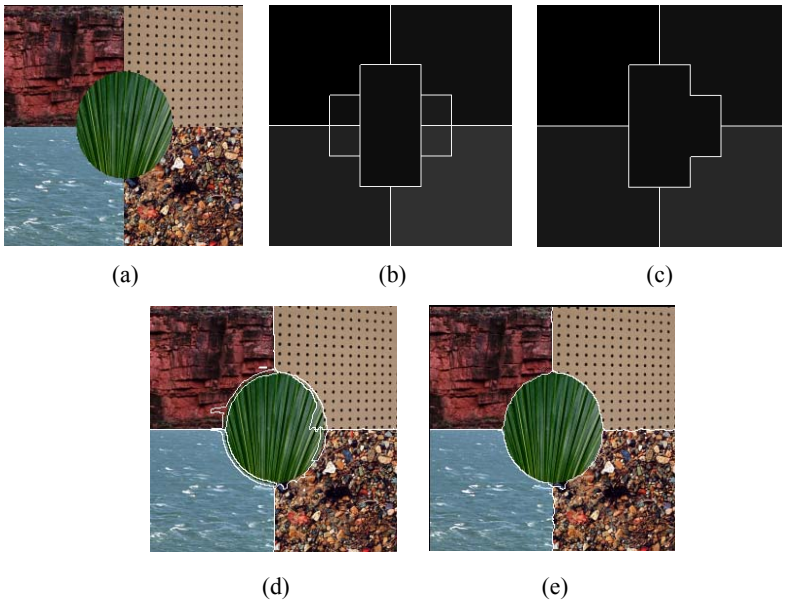


Fig. 10. Example outlining the influence of the merge threshold. (a) Original image. (b) The image resulting from the merge stage (M_{th}=0.8). (c) The image resulting from the merge stage (M_{th}=1.0). (d) The final segmentation result after pixelwise classification (M_{th}=0.8). (e) The final segmentation result after pixelwise classification (M_{th}=1.0).

It can be observed that even for non-optimal settings for the merge threshold the algorithm achieves accurate segmentation. The effect of the

sub-optimal setting for the merge threshold will generate extra regions in the image resulting from the merge stage and since these regions do not exhibit strong colour-texture characteristics they will have a thin long structure around the adjacent regions in the final segmentation results. These regions can be easily identified and re-assigned to the bordering regions with similar colour-texture characteristics. When the segmentation algorithm was tested on synthetic mosaic images the experimental data indicates that the algorithm has a good stability with respect to the diffusion parameter k and the benefit of using pronounced smoothing becomes evident when the image segmentation scheme is applied to noisy and low-resolution images. The influence of this parameter will be examined when we discuss the performance of the colour-texture segmentation scheme on natural and medical images.

5.2 Segmentation of Natural Images

The second set of experiments was dedicated to the examination of the performance of the CTex algorithm when applied to natural images. We applied the algorithm on a range of natural images and images with low signal to noise ratio (Figs. 11 to 13).

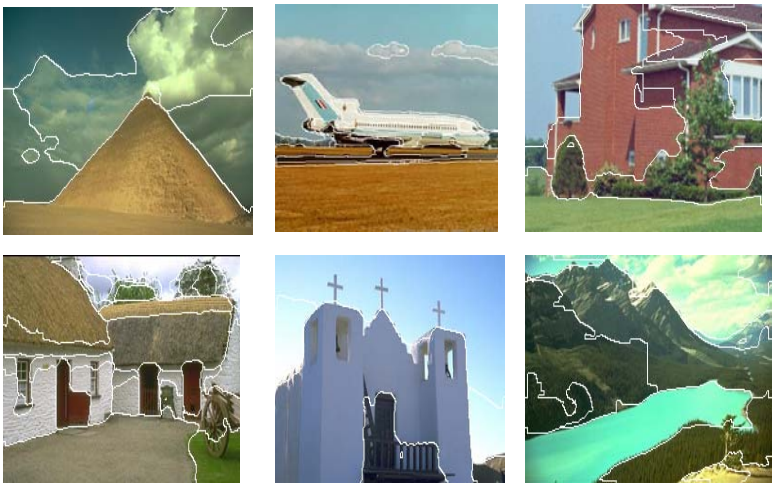


Fig. 11. Segmentation results when the algorithm has been applied to natural images (Berkley[67] and Caltech[71] databases).

The segmentation results obtained from natural images are consistent with the results reported in Table 1 where the most accurate segmentation is obtained when the colour and texture are used in a joint image descriptor. This can be observed in Fig. 12 where are illustrated the segmentation results obtained for cases where texture and colour information are used alone and when colour and texture distributions are used as joint features in the image segmentation process.

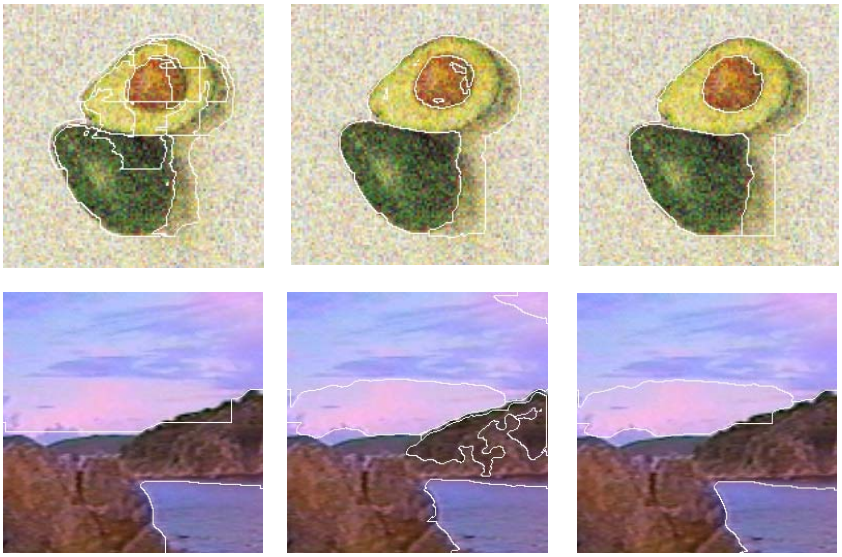


Fig. 12. Segmentation results. (First column) Texture only segmentation. (Second column) Colour only segmentation. (Third column) Colour-texture segmentation.

The diffusion filtering parameter k was also examined. The diffusion filtering scheme was applied to reduce the image noise, thus improving local colour homogeneity. Clearly this helps the image segmentation, especially when applied to images with uneven illumination and image noise. The level of smoothing in equation (5) is controlled by the parameter k (smoothing is more pronounced for high values of k). In order to assess the influence of this parameter we have applied the colour-texture segmentation scheme to low resolution and noisy images. The effect of the diffusion filtering on the colour-segmented result is

illustrated in Fig. 13 where the original image "rock in the sea" is corrupted with Gaussian noise (standard deviation = 20 grayscale levels for each component of the RGB colour space).

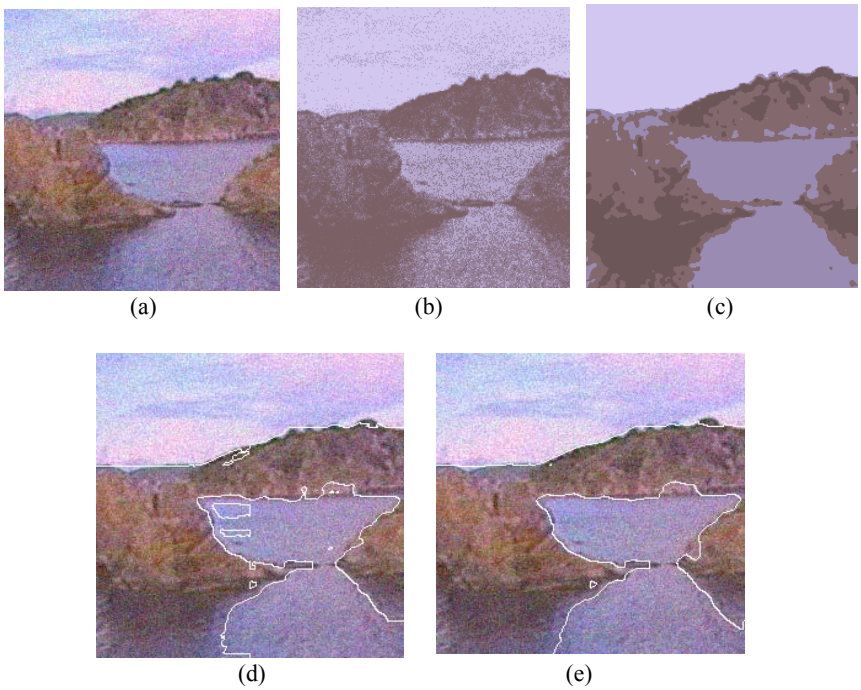


Fig. 13. Effect of the diffusion filtering on the segmentation results. (a) Noisy image corrupted with Gaussian noise (Oulu database[56,47,72,73]). (b) Image resulting from EM algorithm – no filtering. (c) Image resulting from EM algorithm – diffusion filtering k = 30. (d) Colour-texture segmentation – no filtering. (e) Colour-texture segmentation – diffusion filtering k = 30.

One particular advantage of our colour-texture segmentation technique is the fact that it is unsupervised and it can be easily applied to practical applications including the segmentation of medical images and product inspection. To complete our discussion on colour texture we will detail two case studies, namely the identification of skin cancer lesions[36] and the detection of visual faults on the surface of painted slates[28].

5.3 Segmentation of Medical Images

Skin cancer is one of the most common types of cancer and it is typically caused by excessive exposure to the sun radiation[68], but it can be cured in more than 90% of the cases if it is detected in its early stages. Current clinical practice involves a range of simple measurements performed on the lesion border (e.g. **A**symmetry, **B**order irregularity, **C**olour variation and lesion **D**iameter (also known as the ABCD parameters)). The evaluation of these parameters is carried out by manually annotating the melanoma images. This is not only time consuming but it is subjective and often non reproducible process. Thus an important aim is the development of an automated technique that is able to robustly and reliably segment skin cancer lesions in medical images[36,68,70]. The segmentation of skin cancer images it is a difficult task due to the colour variation associated within both the skin lesion and healthy tissue. In order to determine the accuracy of the developed algorithm the ground truth was constructed by manually tracing the skin cancer lesion outline and comparing it with the results returned by our colour-texture image segmentation algorithm (see Fig. 14). Additional details and experimental results are provided in Ilea and Whelan[36].

(a) (b) (c)

Fig 14. Segmentation of skin cancer lesion images (original images (b) & (c) courtesy of: © <Eric Ehrsam, MD >, Dermatlas; http://www.dermatlas.org).

5.4 Detection of Visual Faults on Painted Slates

Roof slates are cement composite rectangular slabs which are typically painted in dark colours with a high gloss finish. While their primary

function is to prevent water ingress to the buildings they have also a decorative role. Although slate manufacturing is a highly automated process, currently the slates are inspected manually as they emerge via a conveyor from the paint process line. Our aim was to develop an automated quality/process control system capable of grading the painted slates. The visual defects present on the surface of the slates can be roughly categorized into substrate and paint defects. Paint defects include no paint, insufficient paint, paint droplets, efflorescence, paint debris and orange peel. Substrate defects include template marks, incomplete slate formation, lumps, and depressions. The size of these defects ranges from $1mm^2$ to hundreds of mm^2 (see Fig. 15 for some representative defects).

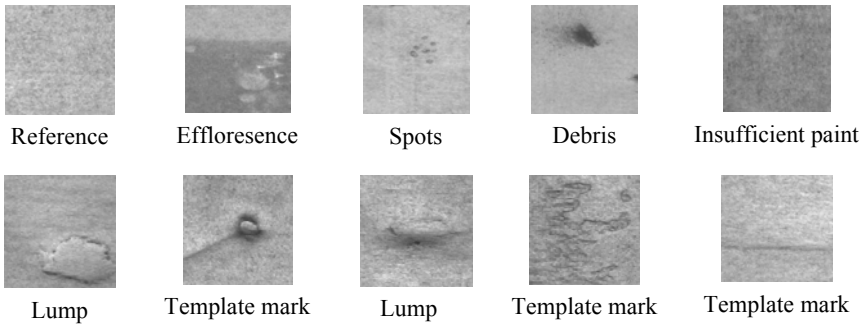


Fig 15. Typical paint and substrate defects found on the slate surface.

The colour-texture image segmentation algorithm detailed in this chapter is a key component of the developed slate inspection system (see Ghita et al. [28] for details). In this implementation for computational purposes the EM algorithm has been replaced with a standard K-means algorithm to extract the colour information. The inspection system has been tested on 235 slates (112 reference-defect free slates and 123 defective slates) where the classification of defective slates and defect-free slates was performed by an experienced operator based on a visual examination. A detailed performance characterization of the developed inspection system is depicted in Table 2. Fig. 16 illustrates the identification of visual defects (paint and substrate) on several representative defective slates.

Table 2. Performance of our colour-texture based slate inspection system.

Slate type	Quantity	Fail	Pass	Accuracy
Reference	112	2	110	98.21 %
Defective	123	123	0	100 %
Total	235			99.14 %

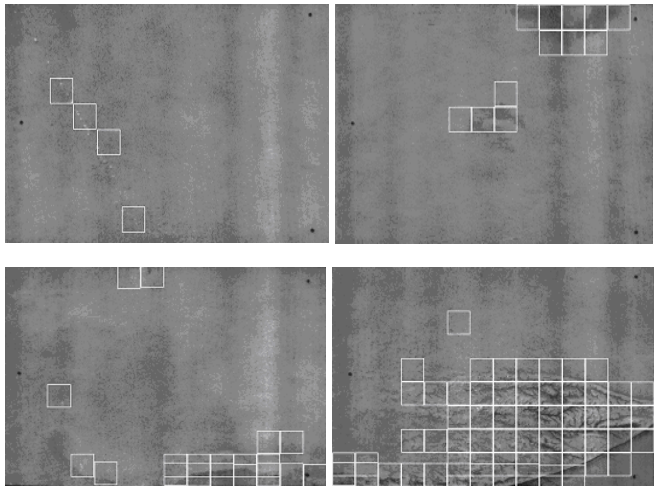


Fig 16. Identification of visual defects on painted slates.

6. Conclusions

In this chapter we have detailed the implementation of a new methodology for colour-texture segmentation. The main contribution of this work is the development of a novel image descriptor that encompasses the colour and texture information in an adaptive fashion. The developed image segmentation algorithm is modular and can be easily adapted to accommodate any texture and colour feature extraction techniques. The colour-texture segmentation scheme has been quantitatively evaluated on complex test images and the experimental results indicate that the adaptive inclusion of texture and colour produces superior results that in cases where the colour and texture information were used in separation. The CTex algorithm detailed in this chapter has been successfully applied to the segmentation of natural, medical and industrial images.

Acknowledgements

We would like to acknowledge the contribution of current and former members of the Vision Systems Group, namely Dana Elena Ilea for the development of the EM colour clustering algorithm and segmentation of medical images, Dr. Padmapryia Nammalwar for the development of the split and merge image segmentation framework and Tim Carew for his work in the development of the slate inspection system. This work has been supported in part by Science Foundation Ireland (SFI) and Enterprise Ireland (EI).

References

1. K.S. Fu and J.K. Mui, A survey on image segmentation, *Pattern Recognition*, 13, p. 3-16 (1981).
2. R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision*, Addison-Wesley Publishing Company (1993).
3. L. Lucchese and S.K. Mitra, Color image segmentation: A state-of-the-art survey, in Proc. of the *Indian National Science Academy*, vol. 67A, no. 2, p.207-221, New Delhi, India (2001).
4. Y.J. Zhang, A survey on evaluation methods for image segmentation, *Pattern Recognition*, 29(8), p. 1335-1346 (1996).
5. R. Chellappa, R.L. Kashyap and B.S. Manjunath, *Model based texture segmentation and classification*, in The Handbook of Pattern Recognition and Computer Vision, C.H. Chen, L.F. Pau and P.S.P Wang (Editors) World Scientific Publishing (1998).
6. M. Tuceryan and A.K. Jain, *Texture analysis*, in The Handbook of Pattern Recognition and Computer Vision, C.H. Chen, L.F. Pau and P.S.P Wang (eds.) World Scientific Publishing (1998).
7. R.M. Haralick, Statistical and structural approaches to texture, in *Proc of IEEE*, 67, p. 786-804 (1979).
8. A. Materka and M. Strzelecki, Texture analysis methods – A review, *Technical Report*, University of Lodz, Cost B11 Report (1998).
9. J.S. Wezska, C.R. Dyer, A. Rosenfeld, A comparative study of texture measures for terrain classification*, IEEE Transactions on Systems, Man and Cybernetics*, 6(4), p. 269-285 (1976).
10. V.A. Kovalev and M. Petrou, Multidimensional co-occurrence matrices for object recognition and matching. *CVGIP: Graphical Model and Image Processing*, 58(3), p. 187-197 (1996).

11. I.M. Elfadel and R.W. Picard, Gibbs random fields, cooccurrences and texture modeling, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1), p. 24-37 (1994).
12. M. Varma and A. Zisserman, Unifying statistical texture classification frameworks, *Image and Vision Computing*, 22, p. 1175-1183 (2004).
13. A.C. Bovik, M. Clark and W.S. Geisler, Multichannel texture analysis using localized spatial filters, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1), p. 55-73 (1990).
14. A.C. Bovik, Analysis of multichannel narrow band filters for image texture segmentation, *IEEE Transactions on Signal Processing*, 39, p. 2025-2043 (1991).
15. J.M. Coggins and A.K. Jain, A spatial filtering approach to texture analysis, *Pattern Recognition Letters*, 3, p. 195-203 (1985).
16. A.K. Jain and F. Farrokhnia, Unsupervised texture segmentation using Gabor filtering, *Pattern Recognition*, 33, p. 1167-1186 (1991).
17. T. Randen and J.H. Husoy, Filtering for texture classification: A comparative study, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4), p. 291-310 (1999).
18. T. Randen and J.H. Husoy, Texture segmentation using filters with optimized energy separation, *IEEE Transactions on Image Processing*, 8(4), p. 571-582 (1999).
19. C. Lu, P. Chung, and C. Chen, Unsupervised texture segmentation via wavelet transform, *Pattern Recognition*, 30(5), p. 729-742 (1997).
20. S. Mallat, Multifrequency channel decomposition of images and wavelet models, *IEEE Transactions on Acoustic, Speech and Signal Processing*, 37(12), p. 2091-2110 (1989).
21. B. Schiele and J.L. Crowley, Object recognition using multidimensional receptive field histograms, in Proc of the *4th European Conference on Computer Vision* (ECCV 96)*,* Cambridge, UK (1996).
22. M. Swain and D. Ballard, Color indexing, *International Journal of Computer Vision*, 7(1), p. 11-32 (1991).
23. M.J. Jones and J.M. Rehg, Statistical color models with application to skin detection, *International Journal of Computer Vision*, 46(1), p. 81-96 (2002).
24. S. Liapis and G. Tziritas, Colour and texture image retrieval using chromaticity histograms and wavelet frames, *IEEE Transactions on Multimedia*, 6(5), p. 676-686 (2004).
25. A. Mojsilovic, J. Hu and R.J. Safranek, Perceptually based color texture features and metrics for image database retrieval, in Proc. of the *IEEE International Conference on Image Processing* (ICIP'99), Kobe, Japan (1999).
26. C.H. Yao and S.Y. Chen, Retrieval of translated, rotated and scaled color textures, *Pattern Recognition*, 36, p. 913-929 (2002).

27. C. Boukouvalas, J. Kittler, R. Marik and M. Petrou, Color grading of randomly textured ceramic tiles using color histograms, *IEEE Transactions on Industrial Electronics*, 46(1), p. 219-226 (1999).
28. O. Ghita, P.F. Whelan, T. Carew and P. Nammalwar, Quality grading of painted slates using texture analysis, *Computers in Industry*, 56(8-9), p. 802-815 (2005).
29. H.D. Cheng, X.H. Jiang, Y. Sun and J.L. Wang, Color image segmentation: Advances & prospects, *Pattern Recognition*, 34(12) p. 2259-2281, (2001).
30. W. Skarbek and A Koschan, Color image segmentation – A survey, *Technical Report*, University of Berlin (1994).
31. M. Pietikainen, T. Maenpaa and J. Viertola, Color texture classification with color histograms and local binary patterns, in Proc. of the *2nd International Workshop on Texture Analysis and Synthesis*, Copenhagen, Denmark, p. 109-112 (2002).
32. L. Shafarenko, M. Petrou and J. Kittler, Automatic watershed segmentation of randomly textured color images, *IEEE Transactions on Image Processing*, 6(11), p. 1530-1544 (1997).
33. T.S.C. Tan and J. Kittler, Colour texture analysis using colour histogram, *IEE Proceedings - Vision, Image, and Signal Processing*, 141(6), p. 403-412 (1994).
34. M. Celenk, A color clustering technique for image segmentation, *Graphical Models and Image Processing*, 52(3), p. 145-170 (1990).
35. R.O. Duda, P.E. Hart and D.E. Stork, *Pattern classification*, Wiley Interscience, 2nd Edition (2000).
36. D.E. Ilea and P.F. Whelan, Automatic segmentation of skin cancer images using adaptive color clustering", in Proc. of the *China-Ireland International Conference on Information and Communications Technologies* (CIICT 06), Hangzhou, China (2006).
37. T.N. Pappas, An adaptive clustering algorithm for image segmentation, *IEEE Transactions on Image Processing*, 14(4), p. 901-914 (1992).
38. R.L. Cannon, J.V. Dave and J.C. Bezdek, Efficient implementation of the fuzzy c-means clustering algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2), p. 249-255 (1996).
39. D. Comaniciu and P. Meer, Mean shift: A robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), p. 603-619 (2002).
40. J.A. Blimes, A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixed and Hidden Markov Models, *Technical Report*, University of California, Berkely, TR-97-021 (1998).
41. D.E. Ilea and P.F. Whelan, Color image segmentation using a self-initializing EM algorithm, in Proc. of the *International Conference on Visualization, Imaging and Image Processing* (VIIP 2006), Palma de Mallorca, Spain (2006).
42. J. Chen, T.N. Pappas, A. Mojsilovic, and B.E. Rogowitz, Image segmentation by spatially adaptive color and texture features, in *Proc.* of *International Conference on Image Processing* (ICIP 03), 3, Barcelona, Spain, p. 777-780 (2003).

43. J. Freixenet, X. Munoz, J. Marti and X. Llado, Color texture segmentation by region-boundary cooperation, in *European Conference on Computer Vision* (ECCV), Lecture Notes in Computer Science (LNCS 3022), Prague (2004).
44. H.D. Cheng and Y. Sun, A hierarchical approach to color image segmentation using homogeneity, *IEEE Transactions on Image Processing*, 9(12), p. 2071-2082 (2000).
45. A. Moghaddamzadeh and N. Bourbakis, A fuzzy region growing approach for segmentation of color images*, Pattern Recognition*, 30(6), p. 867-881 (1997).
46. A. Tremeau and N. Borel, A region growing and merging algorithm to color segmentation*, Pattern Recognition*, 30(7), p. 1191-1203 (1997).
47. D.K. Panjwani and G. Healey, Markov Random Field Models for unsupervised segmentation of textured color images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10), p. 939-954 (1995).
48. Y. Deng and B.S. Manjunath, Unsupervised segmentation of color-texture regions in images and video, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8), p. 800-810 (2001).
49. G. Healey, Using color for geometry-insensitive segmentation*, Optical Society of America*, 22(1), p. 920-937 (1989).
50. G. Healey, Segmenting images using normalized color, *IEEE Transactions on Systems, Man and Cybernetics*, 22(1), p. 64-73 (1992).
51. S.A. Shafer, Using color to separate reflection components, *Color Research and Applications*, 10(4), p. 210-218 (1985).
52. A. Drimbarean and P.F. Whelan, Experiments in colour texture analysis, *Pattern Recognition Letters*, 22, p. 1161-1167 (2001).
53. P. Nammalwar, O. Ghita and P.F. Whelan, Experimentation on the use of chromaticity features, Local Binary Pattern and Discrete Cosine Transform in colour texture analysis, in Proc. of the *13'th Scandinavian Conference on Image Analysis* (SCIA), Goteborg, Sweden, p. 186-192 (2003).
54. M. Mirmehdi and M. Petrou, Segmentation of color textures, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2), p. 142-159 (2000).
55. M.A. Hoang, J.M. Geusebroek and A.W. Smeulders, Color texture measurement and segmentation, *Signal Processing*, 85(2), p. 265-275 (2005).
56. T. Ojala and M. Pietikainen, Unsupervised texture segmentation using feature distributions, *Pattern Recognition*, 32(3), p. 477-486 (1999). See also University of Oulu Texture Database: http://www.outex.oulu.fi/temp/
57. T. Ojala, M. Pietikainen M and T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), p. 971-987 (2002).
58. P. Perona and J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), p. 629-639 (1990).
59. M. Sonka, V. Hlavac and R. Boyle, *Image processing, analysis and machine vision*, 2nd edition, PWS Boston (1998).

60. J. Weickert, *Anisotropic diffusion in image processing*, Teubner Verlag, Stuttgart (1998).
61. S. Khan and A. Ahmad, Cluster center initialization algorithm for K-means clustering, *Pattern Recognition Letters*, 25(11), p. 1293-1302 (2004).
62. J.M. Pena, J.A. Lozano and P. Larranaga, An empirical comparison of four initialization methods for the K-Means algorithm, *Pattern Recognition Letters*, 20(10), p. 1027-1040 (1999).
63. J. Puzicha, M. Held, J. Ketterer, J.M. Buhmann and D. Fellner, On spatial quantization of color images*, IEEE Transactions on Image Processing*, 9(4), p. 666-682 (2000).
64. X. Wu, Efficient statistical computations for optimal color quantization, Graphics Gems 2, Academic Press (1991).
65. P. Nammalwar, O. Ghita and P.F. Whelan, Integration of feature distributions for colour texture segmentation, in Proc. of the *17th International Conference on Pattern Recognition* (ICPR), Cambridge, UK, p. 716-719 (2004).
66. Vision Texture (VisTex) Database, Massachusetts Institute of Technology, Media Lab, http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html
67. D. Martin and C. Fowlkes and D. Tal and J. Malik, A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics, Proc. 8th *Int'l Conf. Computer Vision*, Vol. p.416-423 (2001). See also the Berkley Segmentation Dataset and Benchmark Database: www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/
68. NIH Consensus Conference. Diagnosis and treatment of early melanoma, *JAMA* 268, p. 1314-1319 (1992).
69. Dermatology Image Atlas: http://www.dermatlas.org
70. L. Xu, M. Jackowski, A. Goshtasby, D. Roseman, S. Bines, C. Yu, A. Dhawan and A. Huntley, Segmentation of skin cancer images, *Image and Vision Computing*, 17, p. 65-74 (1999).
71. Caltech Image Database. http://www.vision.caltech.edu/archive.html
72. P.P. Ohanian and& R.C. Dubes, Performance evaluation for four classes of textural features. *Pattern Recognition* 25:819-833 (1992)
73. A.K. Jain and K. Karu, Learning texture description masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18:195-205 (1996)