

University of Warwick institutional repository: <http://go.warwick.ac.uk/wrap>

This paper is made available online in accordance with publisher policies. Please scroll down to view the document itself. Please refer to the repository record for this item and our policy information available from the repository home page for further information.

To see the final version of this paper please visit the publisher's website. Access to the published version may require a subscription.

Author(s): Turgay Celik and Tardi Tjahjadi

Article Title: Unsupervised colour image segmentation using dual-tree complex wavelet transform

Year of publication: 2010

Link to published article:

<http://dx.doi.org/10.1016/j.cviu.2010.03.002>

Publisher statement: "NOTICE: this is the author's version of a work that was accepted for publication in Computer Vision and Image Understanding. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Computer Vision and Image Understanding, VOL:114, ISSUE:7, July 2010, DOI: 10.1016/j.cviu.2010.03.002

Unsupervised Colour Image Segmentation using Dual-tree Complex Wavelet Transform

Turgay Celik and Tardi Tjahjadi,

Abstract

In this paper we present an effective unsupervised colour image segmentation algorithm which uses multiscale edge information and spatial colour content. The multiscale edge information is extracted using the dual-tree complex wavelet transform. Binary morphological operators are applied to the edge information to detect seed regions which are large enough to exclude boundary-only regions. The segmentation of homogeneous regions is obtained using region growing followed by region merging in the CIE $L^*a^*b^*$ colour space. We also present an edge-preserving smoothing filter as a pre-process for the algorithm. We compare our algorithm with state-of-the-art algorithms and show its superior performance.

Keywords: unsupervised colour image segmentation, multiscale edge detection, dual-tree complex wavelet transform, discrete wavelet transform, multiscale analysis

I. INTRODUCTION

Image segmentation is the process of subdividing an image into homogeneous regions. With colour images, the objects are segmented with respect to their colour and spatial features. Image segmentation techniques can be categorized into two groups [1]: soft segmentation and hard segmentations. In soft segmentation, each point in the feature space is associated with a label whose confidence value is computed using some function related to the distance of each converged cluster. For example, Tai *et al.* uses a Markov network to optimize a global objective function that combines the advantages of global colour statistics and local contextual image statistics [1]. The method is computationally complex. The method we proposed in this paper falls into latter category. There are four main approaches to hard image segmentation. The thresholding approach is based on the assumption that clusters in the histogram correspond to either background or objects of interest that can be extracted by separating these clusters [2]–[5]. Boundary-based methods assume that there is significant change in pixel properties, such as intensity, colour and texture between different regions [6], [7]. Region-based methods assume that neighbouring pixels within the same region have similar pixel properties [8], [9]. Hybrid methods tend to combine boundary detection and region growing to achieve better segmentation [10]–[18].

One of the milestones in the hybrid methods is the unsupervised algorithm called JSEG [10]. JSEG uses colour quantization and local windows to compute J -images (corresponding to texture segmentation) at different scales. The J -images are combined via multiscale region growing to obtain image segmentation. Since the computation of J -images requires non-linear numerical computations, the JSEG algorithm has computational complexity which increases with the number of scales used. One drawback of the algorithm is its sensitivity to noise. Since JSEG is based on the results of the colour quantization, another drawback is that its performance is essentially limited by the colour quantization. The values of J -images cannot differentiate the regions with similar distribution of textural patterns but different colour contrast results in another disadvantage of JSEG. A modified

version of JSEG proposed by Wang et al [11] uses an adaptive mean-shift clustering (AMS) for nonparametric clustering of the image data instead of the colour quantization algorithm used in JSEG. They use Gaussian mixture modelling (GMM) of the image data constructed with classifications obtained by AMS in the calculation of J -images. The segmentation results are better in terms of better boundary representation and object segmentation. Since the method is a mixture of JSEG, AMS and parameter estimation using GMM, it is computationally expensive.

The multiscale nature and robustness to noise of the wavelet transform makes it an attractive tool for colour image segmentation. Multiscale edge information in non-decimated wavelet transform domain has been used with a watershed transformation at each scale and a hierarchical region merging procedure to connect segmented regions at different scales [12]. Although the method achieves good performance against noise, the algorithm produces oversegmented results. In [13], multiscale image representations and watershed transforms are used. The scale-space is based on a vector-valued diffusion scheme, and colour gradients in the YUV colour space are computed at each scale. The dynamics of the contours in scale-space are used to validate detected contours. The segmentation results are visually pleasant but the performance of the algorithm against noise is not determined.

Recently, decimated wavelet and watershed transforms are used to increase the robustness of an unsupervised colour image segmentation algorithm against noise [14]. Using the horizontal and vertical subbands, a colour gradient magnitude image is computed at the lowest resolution to detect edges. An initial segmentation map is obtained by applying a watershed transform to the gradient magnitude image. The inverse wavelet transform is then used to project this initial segmentation to finer scales, until the full resolution image is achieved. Finally, a region merging based on the CIE $L^*a^*b^*$ colour distances is applied to obtain the final segmentation. Since the performance of the algorithm mainly depends on the correctness of the detected edges in the scale in which watershed algorithm is applied, further improvement can be achieved with a better edge detector. Also, its performance depends on the watershed algorithm which requires a threshold selection process.

Multiscale edge detection could also be performed in the image domain using wavelet transform to obtain the multiscale edge features which have good time-spatial characteristics. Mallat *et al.*'s approaches [19] classify singularity points as the local maxima of gradient modulus or the zero-crossings of wavelet coefficients as edges. Sun *et al.* [20] used Hidden Markov Model with wavelet transform to detect multiscale edges and fuse them to generate an edge map at the pixel resolution level for object recognition. Zhang and Bao [21] defined a scale product as the multiplication of two adjacent scales of wavelet coefficients to amplify edge structures while reducing noise, and using a single threshold classified the local maxima of the product as edges. DWT-based methods generally suffers from the shift-variance and lack of directionality required in multiscale analysis to detect multiscale edges. Wang and Jiao [22] utilised the better directionality and shift invariance of the complex wavelets, and the scale product similar to that in [21] to detect edges. But as with most of DWT-based methods, they used wavelet coefficients multiplication and thresholding to detect multiscale edges. There are two drawbacks with this approach. The first and the most important is that coefficients multiplication using intra-scales may obscure low coefficients which contains information associated with true edges. The second drawback is the dependency to thresholding. To overcome these drawbacks we propose a Bayesian approach to detect multiscale edges using complex wavelet coefficients.

Region based merging is generally the final process for most colour image segmentation algorithms. Region merging usually involves statistical parameters for inter and intra region correlations [23], [24] which are expensive to compute. Region adjacency graph (RAG) [25] based merging represents each region as a graph node and an edge exists between two adjacent nodes. A cost is assigned to each graph edge expressing the dissimilarity between two adjacent nodes. The most similar pairs of adjacent nodes have the edges with the minimum cost and are merged. The initialization of RAG, the calculation of edge costs and the

iteration involved in merging are computationally expensive. To reduce computational complexity Shih and Cheng [26] used two criteria for region merging: colour similarity and size of region. In the first pass, the criteria are applied sequentially as follows. Firstly, if the colour distance between two adjacent regions is less than a threshold, i.e., 0.1, the two regions are merged and the mean colour of the merged region is computed. This comparison is repeated for all regions including the newly merged regions until no more merging is possible. Secondly, if the number of pixels in a region is smaller than a threshold, i.e., $1/150$ of the image size, the region is merged to its neighbouring region with the smallest colour distance. This procedure is repeated until there are no regions with size smaller than the threshold. In the second pass, any two adjacent regions with size smaller than $1/10$ of the image size and the colour difference between them is greater than 0.2 are merged. This process is repeated until no more merging is possible. The algorithm is simple and provides good results.

We proposed an unsupervised colour image segmentation algorithm to address the aforementioned drawbacks, which uses the multiscale structure of the dual-tree complex wavelet transform (DT-CWT) [27] to detect representative edge information. The motivation for using DT-CWT is due to its ability to detect directional edges better than DWT, increased redundancy and the dual-tree implementation for real-time performance. The edge information is used to generate the representative seed region for each homogeneous region in the finest resolution of the image. We employ morphological binary operators rather than a watershed algorithm to detect seed regions in order to significantly reduce the computational complexity of the overall algorithm. Since edge information is used to find seed regions, only a few pixels are not assigned to any seed region. Thus, the pixel assignment performed by the region growing is computationally inexpensive. A region growing and region merging are then applied to obtain the overall segmentation. The region merging is performed in the uniform CIE $L^*a^*b^*$ colour space [28] to address oversegmentation. We also introduced a new edge-preserving smoothing filter which is used as a pre-process for the algorithm.

The paper is organized as follows. Section II presents the proposed algorithm. Section III presents the experimental results and discussions. Finally, Section IV concludes the paper.

II. PROPOSED ALGORITHM

Figure 1 shows the overview of the proposed algorithm. The first stage of the algorithm transforms the image from the RGB to the CIE $L^*a^*b^*$ colour space. The second stage smoothes the image using a non-linear pre-filtering operation (see Section II-A). The aims of the pre-filtering are two folds: to remove small variations in the input image while retaining a contrast between different regions; and to reduce the noise level.

A multiscale edge map is found using DT-CWT (see Section II-B and II-C) and combined with the result of a Canny edge detection to detect finer boundary edges. The resulting edge map is used to find seed regions that correspond to regions to be segmented (see Section II-D). Since the seed regions do not fully encompass their corresponding region, a region growing (see Section II-E) is applied. Since the region growing may produce oversegmentation when a sub-region is incorrectly segmented as more than two spatially neighbouring regions, a region merging (see Section II-E) is also applied. When every region has been segmented and represented with an average colour (pseudo-colour) vector in the CIE $L^*a^*b^*$ colour space, an inverse image transformation is performed to convert the segmented image to its RGB representation.

A. Pre-filtering image

A commonly used pre-filter is a Gaussian or weighted mean filter. However, the noise reduction is achieved at the expense of losing the details of object boundaries. A solution to this problem is to use an edge preserving smoothing technique (e.g., an

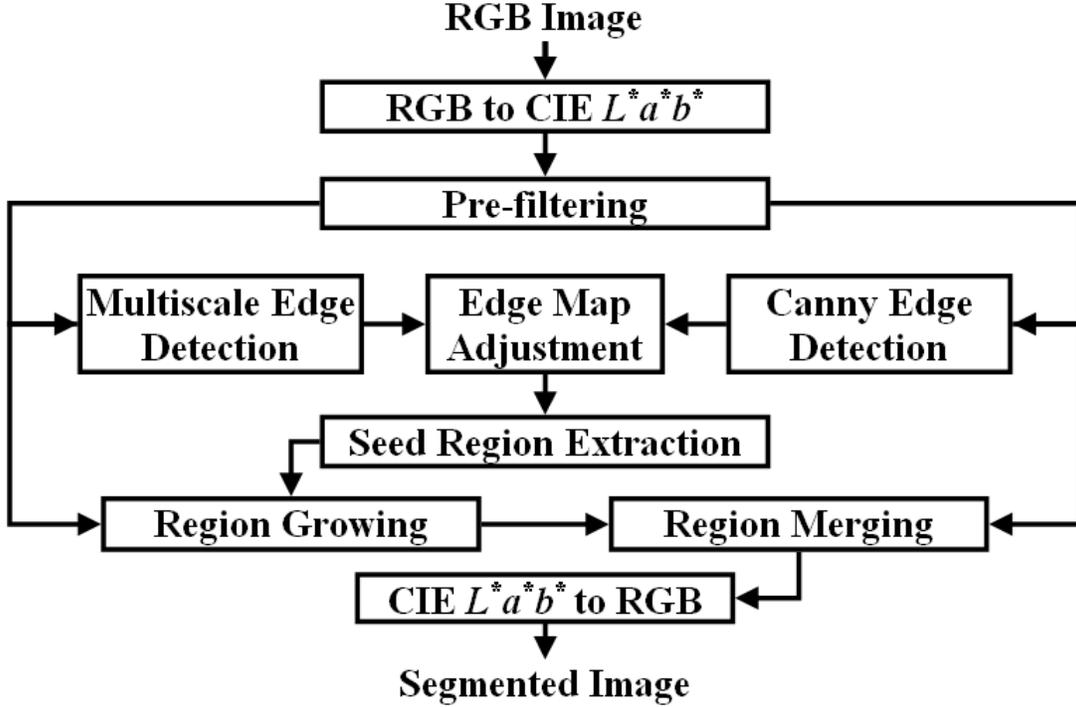


Fig. 1. Proposed colour image segmentation algorithm.

adaptive mean filter) where the amount of blurring of each pixel is determined from the local information in a specified $n \times n$ neighbourhood. Tomasi and Manduchi's bilateral filter [29] is the state of the art for edge-preserving smoothing. The filter combines grey levels or colours of pixels based on both their geometric closeness and their photometric similarity. The main drawback of this method is the need to tune parameters for domain filtering (σ_d) and range filtering (σ_r). It also necessary to specify the bilateral filter half-width (w). Different parameters generate different filtering results, and bilateral filtering provides excellent results with a good choice of parameters but requires high computation for each pixel.

For an unsupervised filtering which does not require any parameter selection, we propose an edge preserving smoothing filter (EPSF) which employs adaptive filter coefficients for each pixel value. The coefficients are

$$c_i = \exp(-d_i) \quad (1)$$

where $d_i, i = 1 \dots 24$ are the colour distances between the centre pixel and its neighbouring pixels, normalized to the range of $[0, 1]$, i.e.,

$$d_i = \sqrt{\frac{(L_c^* - L_i^*)^2 + (a_c^* - a_i^*)^2 + (b_c^* - b_i^*)^2}{3}} \quad (2)$$

where (L_c^*, a_c^*, b_c^*) and (L_i^*, a_i^*, b_i^*) are respectively the CIE $L^*a^*b^*$ colour values of the centre pixel and its neighbouring pixels. For a neighbouring pixel, the corresponding coefficient is high if its colour value is close to the value of the centre pixel, and vice versa.

A dynamic 5×5 mask centred at spatial location (x, y) is defined as

$$T(x, y) = \frac{1}{\sum_{i=1}^{24} c_i} \begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 \\ c_6 & c_7 & c_8 & c_9 & c_{10} \\ c_{11} & c_{12} & 0 & c_{13} & c_{14} \\ c_{15} & c_{16} & c_{17} & c_{18} & c_{19} \\ c_{20} & c_{21} & c_{22} & c_{23} & c_{24} \end{bmatrix}. \quad (3)$$

The filtering of the image is achieved by convolving the mask $T(x, y)$ with each of the CIE $L^*a^*b^*$ components separately. As the colour distance gets larger, coefficients due to small colour distances from the centre pixel increase their relative difference

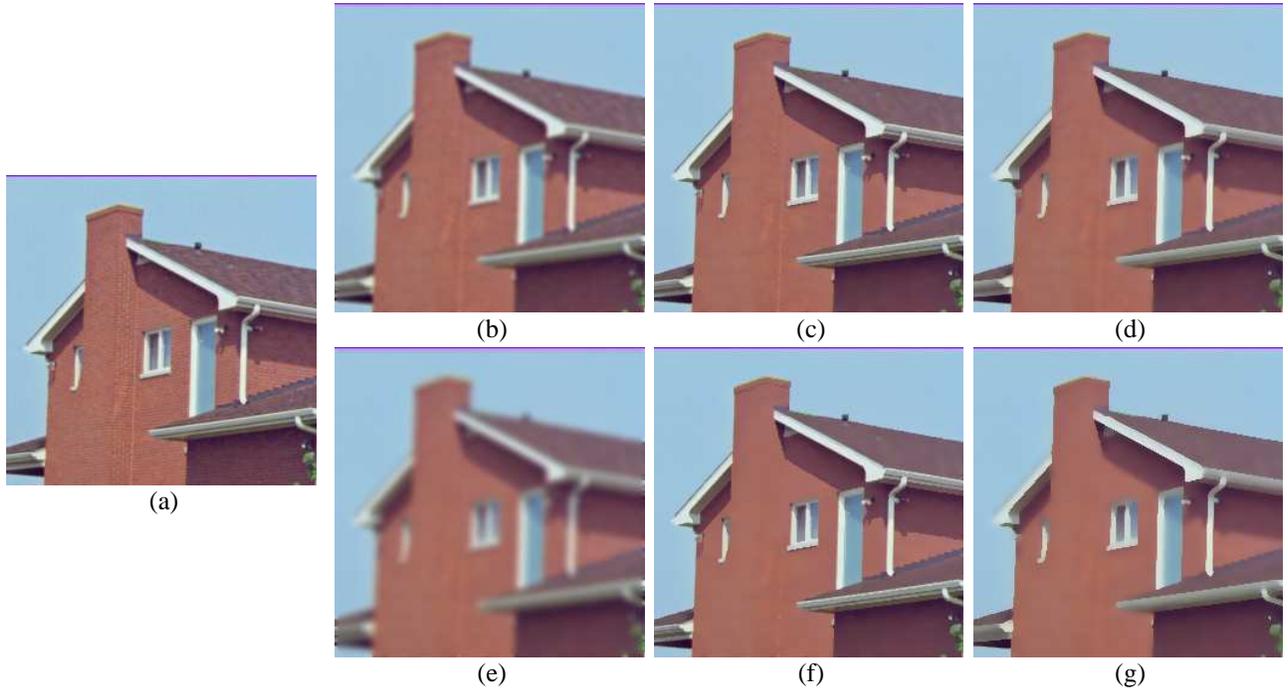


Fig. 2. EPSF filtering of House image: (a) Original image; (b) After Eq. (4) is applied once; (c) After EPSF is applied once; (d) After bilateral filtering [29] is applied once; (e) After Eq. (4) is applied 5 times; (f) After EPSF is applied 5 times; and (g) after bilateral filtering [29] is applied 5 times.

value from coefficients with large colour distance, thus reducing any blurring. The centre pixel of the mask is set to 0 in order to remove impulsive noise. Note that when EPSF is applied to object edge pixels, the colour distances d_i are large for the neighbouring pixels that do not belong to the object and hence the colours of these neighbouring pixels have a small effect on the final colour of the centre pixel. Thus, the proposed filter is considered as an edge preserving filter. Note also that the size of EPSF may be extended to any $n \times n$ size by following a similar methodology. However, as the filter size is increased the blurring is also increased.

Figure 2 shows the effects on an image when EPSF is applied once, iteratively, and with constant EPSF coefficients, e.g.,

$$T(x, y) = \frac{1}{44} \begin{bmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 4 & 2 & 1 \\ 2 & 4 & 0 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix}. \quad (4)$$

It also shows the application of the bilateral filter in [29] with parameters $w = 2, \sigma_d = 3, \sigma_r = 0.1$. Note that $w = 2$ is selected in order to use the same local spatial support, and other parameters are tuned to give a good performance. As is seen mainly from the pixel profiles around object boundaries, noise is reduced almost without blurring by using both bilateral filtering and EPSF. However as the number of iterations increases EPSF retains higher contrast than bilateral filtering at region boundaries. More iterations will remove more noise but at the expense of losing details around object boundaries. We found that 5 iterations are sufficient to reduce noise without blurring.

Figure 2 shows that even when an image is not noisy, EPSF retains the contrast between regions but removes some small details within a region, e.g., the shape of the individual bricks within a wall region are no longer recognizable. However, it is expected that if the image is noisy, then EPSF should remove noise while retaining clear region boundaries. Figure 3 shows the effects of applying EPSF with constant coefficients, EPSF with dynamic coefficients and bilateral filtering on a noisy House image using 5 iterations. Figure 3 shows that EPSF and bilateral filtering are good at removing noise and retaining high contrast between regions, but EPSF does not require any parameters to tune as in the case with the bilateral filter. In the

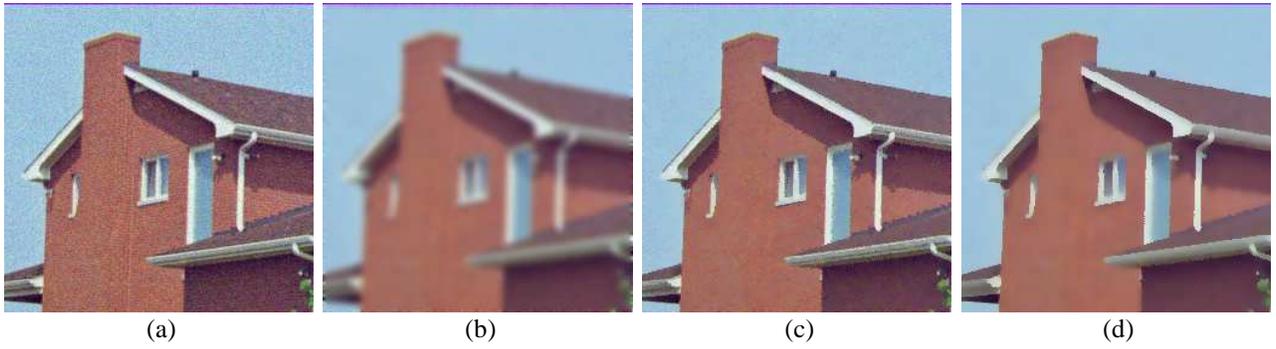


Fig. 3. EPSF filtering of a noisy image: (a) Original image with added Gaussian noise (SNR = 26 dB); (b) After (Eq. (4)) is applied 5 times; (f) After EPSF is applied 5 times; and (g) after bilateral filtering [29] is applied 5 times.

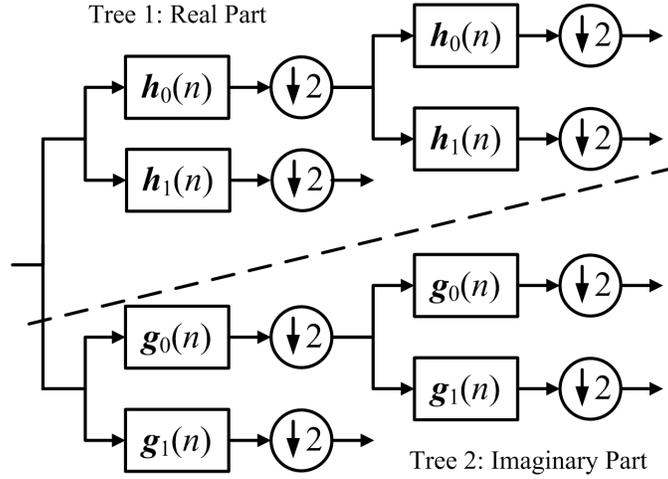


Fig. 4. Two-level 1-D dual-tree complex wavelet transform (DT-CWT).

absence of noise, EPSF retains a higher contrast between regions.

B. Dual-tree complex wavelet transform

The ordinary DWT is shift variant because of the decimation operation exploited in the transform. As a result, a small shift in the input signal can cause a very different set of wavelet coefficients produced at the output. For that, Kingsbury [27] introduced a new kind of wavelet transform, called the dual-tree complex wavelet transform (DT-CWT), which exhibits approximate shift invariant property and improves directional resolution when compared with that of the DWT. At each scale, the DT-CWT produces six directional subbands, oriented at $\pm 15^\circ$, $\pm 45^\circ$, and $\pm 75^\circ$, while the DWT produces only three directional subbands, oriented at 0° , 45° , and 90° .

The DT-CWT also yields perfect reconstruction by using two parallel decimated filter-bank trees with real-valued coefficients generated at each tree. The one-dimensional (1-D) DT-CWT decomposes the input signal $f(x)$ by expressing it in terms of a complex shifted and dilated mother wavelet $\psi(x)$ and scaling function $\phi(x)$; that is,

$$f(x) = \sum_{l \in Z} s_{j_0, l} \phi_{j_0, l}(x) + \sum_{j \geq j_0} \sum_{l \in Z} c_{j, l} \psi_{j, l}(x), \quad (5)$$

where Z is the set of natural numbers, j and l refer to the index of shifts and dilations respectively, $s_{j_0, l}$ is the scaling coefficient and $c_{j, l}$ is the complex wavelet coefficient with $\phi_{j_0, l}(x) = \phi_{j_0, l}^r(x) + \sqrt{-1} \phi_{j_0, l}^i(x)$ and $\psi_{j, l}(x) = \psi_{j, l}^r(x) + \sqrt{-1} \psi_{j, l}^i(x)$, where the superscripts r and i denote the real part and the imaginary part, respectively. In the 1-D DT-CWT case, the set

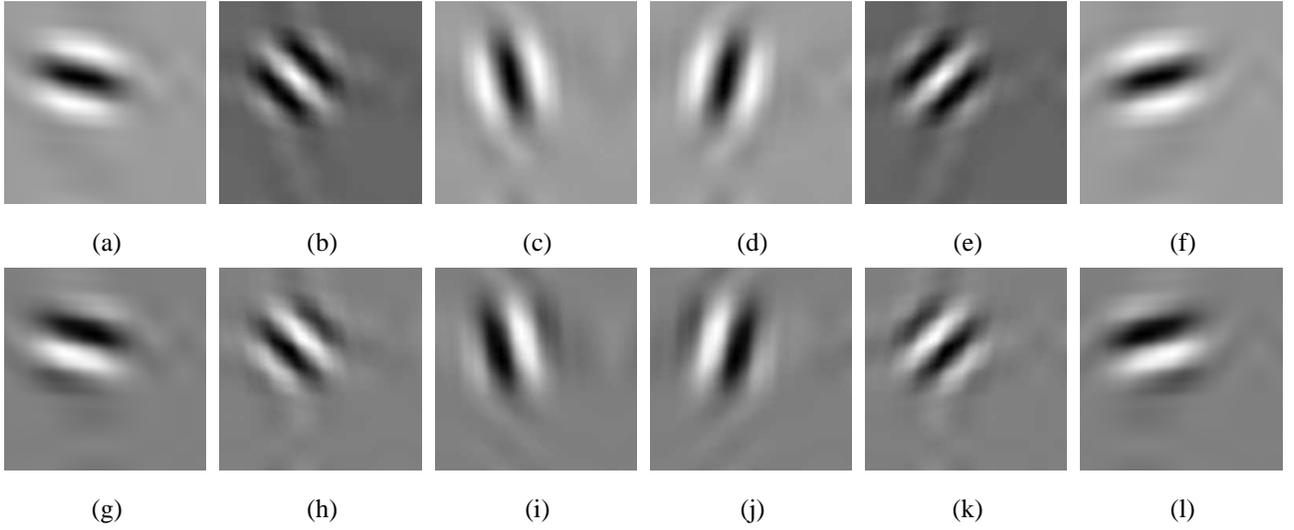


Fig. 5. The real (R) and imaginary (I) parts of the impulse responses of the 2-D DT-CWT filters under 6 directional subbands: (a) R_{-15° ; (b) R_{-45° ; (c) R_{-75° ; (d) R_{+75° ; (e) R_{+45° ; (f) R_{+15° ; (g) I_{-15° ; (h) I_{-45° ; (i) I_{-75° ; (j) I_{+75° ; (k) I_{+45° ; (l) I_{+15° .

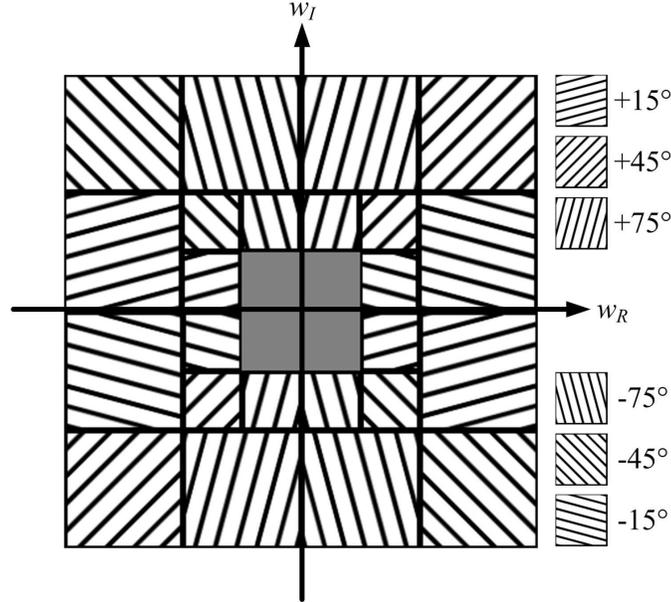


Fig. 6. Frequency-domain partition resulted from a two-level 2-D DT-CWT decomposition, where w_R and w_I are the real axis and the imaginary axis of the complex frequency domain, respectively.

$\{\phi_{j_0,l}^r, \phi_{j_0,l}^i, \psi_{j_0,l}^r, \psi_{j_0,l}^i\}$ forms a tight wavelet frame with double redundancy. The real and imaginary parts of the 1-D DT-CWT are computed using separate filter banks with filters h_0 and h_1 for the real part, and g_0 and g_1 for the imaginary part, as illustrated in Figure 4 [27]. The outputs from the two trees in Figure 4 are interpreted as the real and the imaginary parts of the complex coefficients.

Similar to the 1-D DT-CWT, the two-dimensional (2-D) DT-CWT decomposes a 2-D image $f(x, y)$ through a series of dilations and translations of a complex scaling function and six complex wavelet functions $\psi_{j,l}^\theta$, i.e.,

$$f(x, y) = \sum_{l \in \mathbb{Z}^2} s_{j_0,l} \phi_{j_0,l}(x, y) + \sum_{\theta \in \Theta} \sum_{j \geq j_0} \sum_{l \in \mathbb{Z}^2} c_{j,l}^\theta \psi_{j,l}^\theta(x, y). \quad (6)$$

where $\theta \in \{\pm 15^\circ, \pm 45^\circ, \pm 75^\circ\}$ provides the directionality of the complex wavelet function. That is, the decomposition of

$f(x, y)$ by exploiting the DT-CWT produces one complex-valued low-pass subband and six complex-valued high-pass subbands at each level of decomposition, where each high-pass subband corresponds to one unique direction θ . The impulse responses of the six complex wavelets of the 2-D complex wavelet transform are illustrated in Figure 5. The frequency-domain partition resulted from a two-level 2-D DT-CWT decomposition is graphically shown in Figure 6.

C. Generating multiscale edge map using DT-CWT

Edges between regions are expected to be present even when the image is processed via different scales and different orientations. Interscale features of DT-CWT with 6 different orientations may be used to extract edges with different orientations in different scales. The coefficients in the subbands may be modelled using Gaussian distributions or a mixture of Gaussians. The latter requires the weights and parameters of each distribution and thus requires high computational time for the corresponding model. However, if the norm of each complex coefficient in a subband is used, which is a form of a nonlinear transformation of the coefficients, the overall distribution for the coefficients within a subband may be modelled with the Rayleigh distribution [30]. Scharcanski *et al.* only used LH and HL subbands to estimate gradient magnitude and direction in each scale [30]. The estimated gradient information is enhanced using scale and space consistency. Unlike their method, we model the magnitude of each complex subband coefficient in DT-CWT as Rayleigh distribution. Since at each scale, DT-CWT has 6 subbands, we have 6 different sources of information whereas the DWT based method in [30] provides only a single source of information.

Unlike the method in [30] we define the test statistics $G_{R_\theta C_\theta} = \sqrt{R_\theta^2 + C_\theta^2}$, $\theta = \pm 15^\circ, \pm 45^\circ, \pm 75^\circ$. R_θ and C_θ are the real and imaginary coefficients with the same directionality in each subband respectively. The distribution of the test statistics follows the Rayleigh distribution. The distribution of $G_{R_\theta C_\theta}$ consists of edges and non-edges. Non-edge information represents homogeneous texture information and the edge is the passage between regions. Thus, $G_{R_\theta C_\theta}$ consists of two classes: c_e and c_{ne} , where the first represents the edges and the second represents the non-edges which may be considered as homogeneous regions, i.e., non-edge regions. The distribution of c_e is approximated by

$$p_{R_\theta C_\theta}(r/c_e) = \frac{r}{\sigma_e^2} e^{-r^2/(2\sigma_e^2)} \quad (7)$$

where $r \in G_{R_\theta C_\theta}$ and σ_e^2 is the parameter for its Rayleigh distribution. Similarly the distribution of c_{ne} is

$$p_{R_\theta C_\theta}(r/c_{ne}) = \frac{r}{\sigma_{ne}^2} e^{-r^2/(2\sigma_{ne}^2)} \quad (8)$$

where σ_{ne}^2 is the parameter for its Rayleigh distribution. The overall distribution of $G_{R_\theta C_\theta}$ is approximated by a linear combination of distributions of the two classes, i.e.,

$$p_{R_\theta C_\theta}(r) = \pi_e p_{R_\theta C_\theta}(r/c_e) + \pi_{ne} p_{R_\theta C_\theta}(r/c_{ne}) \quad (9)$$

where π_e and π_{ne} are the weights of the corresponding classes and $\pi_e + \pi_{ne} = 1$.

In order to estimate the distributions for c_e and c_{ne} the corresponding parameters π_e , σ_e^2 and σ_{ne}^2 are estimated by maximizing the likelihood function

$$\ln(L) = \sum_{all(x,y)} \ln(p_{R_\theta C_\theta}(G_{R_\theta C_\theta})). \quad (10)$$

Using the optimization in Eq. (10) the following iterative equations are derived for estimating π_e , σ_e^2 and σ_{ne}^2 :

$$\tau_n^i(t) = \frac{\pi_i^t \frac{r_n}{\sigma_i^2} e^{-r_n^2/(2\sigma_i^2)}}{\sum_j \pi_j^t \frac{r_n}{\sigma_j^2} e^{-r_n^2/(2\sigma_j^2)}} \quad (11)$$

$$\sigma_i(t+1) = \frac{\sum_n \tau_n^i(t) r_n^2}{2 \sum_n \tau_n^i(t)} \quad (12)$$

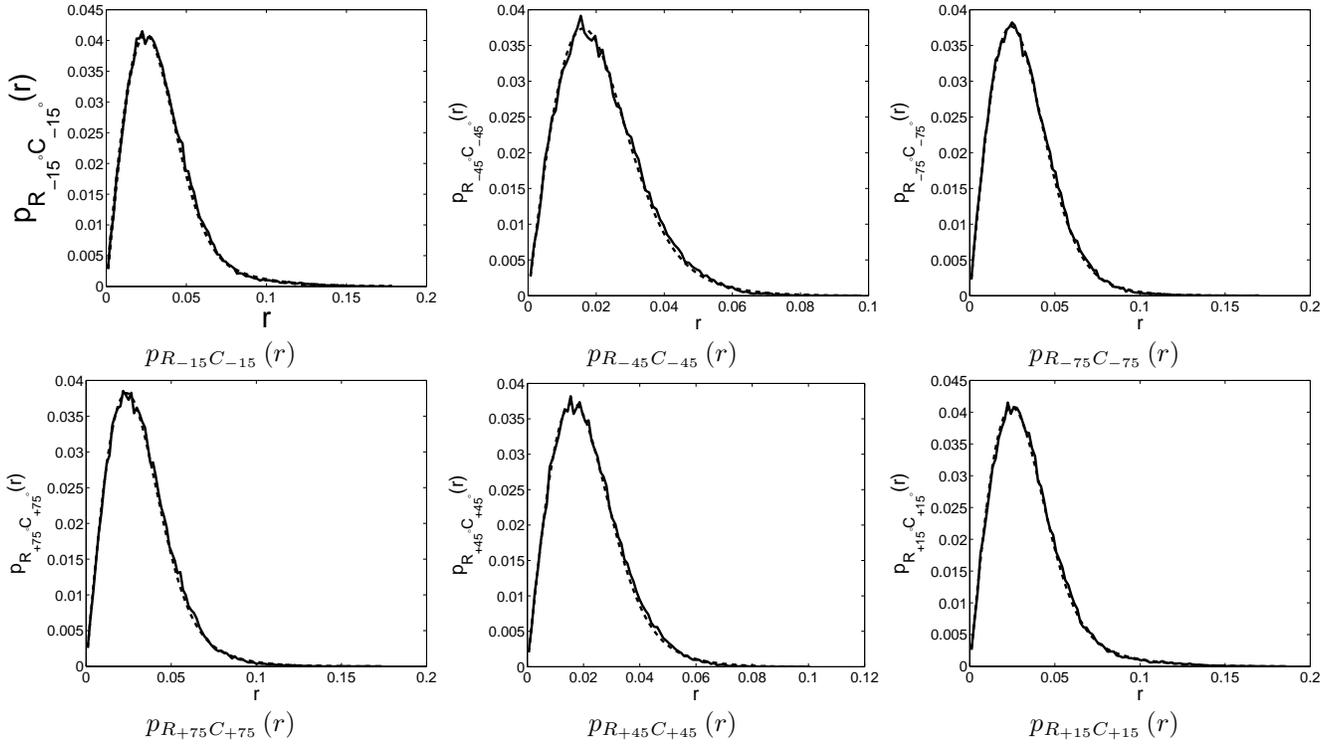


Fig. 7. Actual (solid line) data and estimated (dashed line) $p_{R_{\theta}C_{\theta}}(r)$.

$$\pi_i^{t+1} = \frac{1}{N} \sum_{n=1}^N \tau_n^i(t), \quad i = c_e, c_{ne} \quad (13)$$

where t and $t + 1$ are respectively the current and next iteration, and N is the total number of coefficients in $G_{R_{\theta}C_{\theta}}$. The correct estimation of these parameters is crucial for the success of the edge detection. Figure 7 shows the actual distribution of $p_{R_{\theta}C_{\theta}}(r)$ and its estimation using π_e , σ_e^2 and σ_{ne}^2 for the 6 directional subbands for the House image in the second scale of DT-CWT domain. Note there is a close agreement between the actual and estimated distributions. Once the parameters have been accurately estimated, the Bayes theorem is used to estimate the conditional distributions for c_e and c_{ne} as follows:

$$p_{R_{\theta}C_{\theta}}(c_e/r) = \frac{\pi_e p_{R_{\theta}C_{\theta}}(r/c_e)}{\sum_j \pi_j p_{R_{\theta}C_{\theta}}(r/c_j)} \quad (14)$$

$$p_{R_{\theta}C_{\theta}}(c_{ne}/r) = \frac{\pi_{ne} p_{R_{\theta}C_{\theta}}(r/c_{ne})}{\sum_j \pi_j p_{R_{\theta}C_{\theta}}(r/c_j)}. \quad (15)$$

Using $p_{R_{\theta}C_{\theta}}(c_e/r)$ the edge map for the corresponding scale is computed by using a combination of all edge maps in all subbands, i.e., the overall edge map in a given scale is defined as $p_s(c_e/r)$ and is equivalent to the maximum of all edge maps in all subbands, i.e.,

$$p_s(c_e/r) = \max \left\{ p_{R_{\theta}C_{\theta}}(c_e/r)_{\theta \in \{\pm 15, \pm 45, \pm 75\}} \right\} \quad (16)$$

where s corresponds to the scale. Note that $\max\{\}$ may generate false alarms for a given scale s , but the false alarms are reduced by the multiscale nature of DT-CWT.

Figure 8 demonstrates the extraction of the overall edge map for an image using Eq. (16). It is clear that different subbands detect different types of edges, i.e., different directionalities in DT-CWT produce different edges. The combination of edge maps, $p_s(c_e/r)$, produces an edge map for a corresponding scale s , which provides the contours for region boundaries. Figure 8 also shows that the edges detected in subbands provide reasonable edge maps and the model proposed in Eq. (16) generates overall edges for the corresponding scale. Assuming that if a pixel is detected as an edge pixel then that pixel remains so

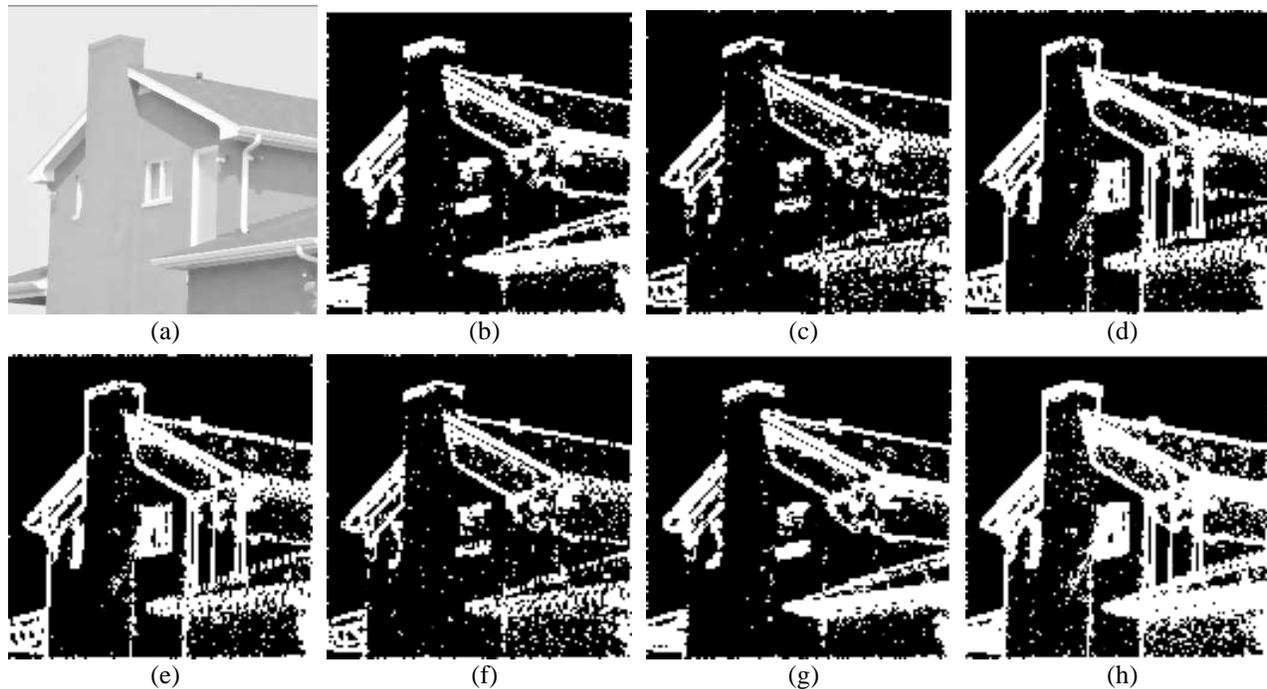


Fig. 8. Edge map for scale $s = 1$ using the maximum of subband edge maps: (a) L^* component of image in Figure 2(f); (b)-(g) Edge maps in 6 directional subbands; and (h) Overall edge map using Eq. (16).

through the consecutive scales, i.e., interscale continuum of the pixel is guaranteed if the pixel is an edge pixel, we propose

$$p_s(c_e/r) = f(p_s(c_e/r), p_{s+1}(c_e/r)) \quad (17)$$

where $f(\cdot)$ is any function which may be used for enhancing the edge map using scales of s and $s + 1$. In [30] it is proposed that $f(a, b) = 2/(\frac{1}{a} + \frac{1}{b})$, where it is assumed that if a pixel has a high edge dependency than the edge dependency should also be high at the corresponding spatial location in the next scale $s + 1$. For example, if $a=1, b=1$ (i.e., the corresponding pixel is an edge pixel at both scales s and $s + 1$) then the likelihood of the corresponding updated edge pixel at scale s becomes $f(a, b) = 1$. If $a=1, b=0.5$ (i.e., the corresponding pixel is an edge pixel at scale s and the likelihood that it is defined as an edge pixel at $s + 1$ is half) then the corresponding $f(a, b) = 0.67$ is smoother for a corresponding $p_s(c_e/r)$. This means that if a pixel is defined as an edge pixel at a finer scale (i.e., image resolution) and the likelihood that it is an edge pixel reduces to half of the maximum value for the next resolution, then the likelihood at a finer resolution retains this interscale dependency. Each edge map is then updated using the iterative formula

$$p_s^u(c_e/r) = f(p_{s+1}^u(c_e/r), p_s(c_e/r)) \quad (18)$$

$s = S - 1, \dots, 1$, where $p_s^u(c_e/r)$ is the updated edge map and S is the number of scales used. Figure 9 demonstrates the update of an edge map Figure 9(b) (generated using only Eq. (16)), which consists of high detailed edges and numerous false detections. The false detections are reduced in Figure 9(c) by using Eq. (18), clearly showing that Eq. (18) brings the multiscale edge continuum to the current scale with nonlinear functionality. Although some details are lost, the edge information is enhanced and false detections are reduced. Since a colour image is used, the overall edge map is

$$p_{s,all}(c_e/r) = \max(p_{s,L^*}^u(c_e/r), p_{s,a^*}^u(c_e/r), p_{s,b^*}^u(c_e/r)) \quad (19)$$

i.e., the maximum of the edge map of each of the CIE $L^*a^*b^*$ components. Figure 10 shows the edge detection using $p_{s,all}(c_e/r)$ for the House image. The size of the edge map obtained using (19) is half the size of the original image if the

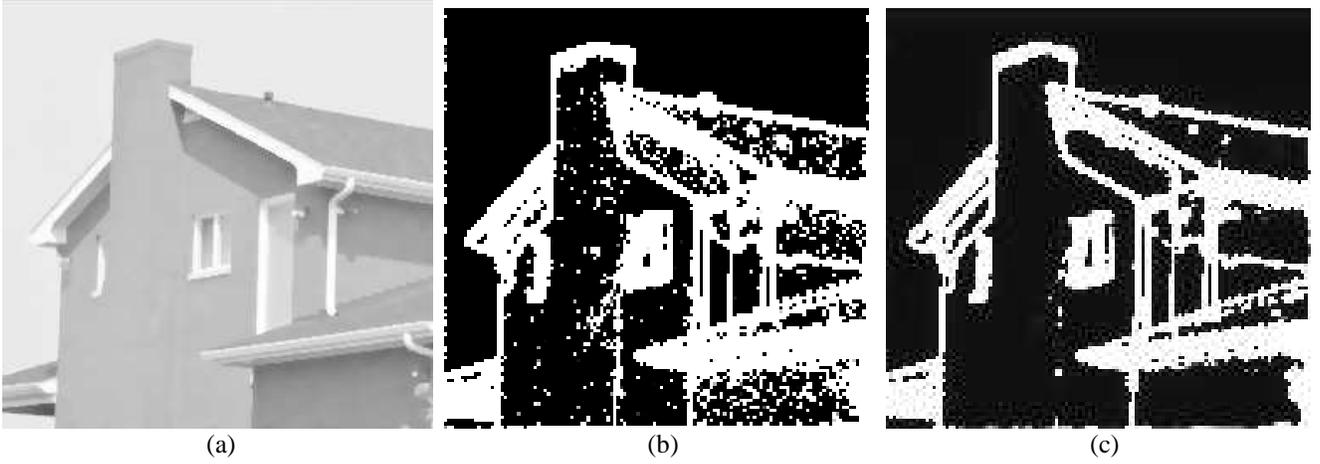


Fig. 9. Final edge map at $s = 1$ by using the iterative edge map update: (a) Original image; (b) Original edge map; and (c) Updated edge map.

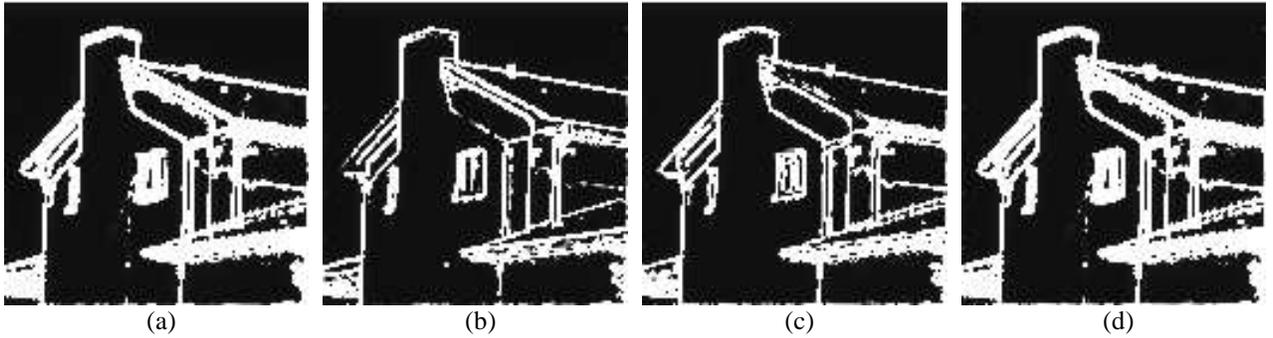


Fig. 10. The union of edge maps from all colour channels: (a) L^* channel; (b) a^* channel; (c) b^* channel; and (d) Overall edge map using Eq. (19).

edge map is constructed with $s = 1$, i.e., at the finest resolution of DT-CWT. We use bilinear interpolation to resize the edge map to the original image size. Due to the blocking effect in interscale edge detection and bilinear interpolation, the enlarged edge map produces thick edges. Thus, the Canny edge detector is applied to all colour channels to thin the edges.

The overall edge map detected by Canny edge detector is given by

$$e_{L^*a^*b^*} = \Phi(L^*) \cup \Phi(a^*) \cup \Phi(b^*) \quad (20)$$

where $\Phi(\cdot)$ refers to the Canny edge detector. Since the Canny edge detector produces binary results, $e_{L^*a^*b^*}$ is a binary map where 0 represents no edge and 1 represents an edge. The overall edge map in the image domain is simply the multiplication of $e_{L^*a^*b^*}$ with the upscaled version of $p_{1,all}(c_e/r)$, i.e.,

$$p_{L^*a^*b^*}(c_e/r) = e_{L^*a^*b^*} \times \Theta(p_{1,all}(c_e/r)) \quad (21)$$

where $\Theta(\cdot)$ refers to the upscaling function using bilinear interpolation.

The edge map obtained using Eq. (21) represents a value in the range of $[0, 1]$, where a value close to 1 means stronger edges, and vice versa. In order to classify a pixel into one of the two classes (c_e, c_{ne}) , i.e., to generate the overall binary edge map BEM , Otsu's automatic thresholding method [4] is used since the proposed algorithm is unsupervised. The Otsu's method is quite successful in thresholding a given data into two classes. The threshold is automatically determined and is set to the value which maximizes the discrimination criterion (σ_B^2/σ_w^2) , where σ_B^2 is the between-class variance and σ_w^2 is the within-class variance. In Figure 11(a) the upper left window is detected as part of the upper left roof frame, but after applying Eq. (21) it becomes detached with a different boundary. The thinning effect of Eq. (21) is also noticeable. In generating the

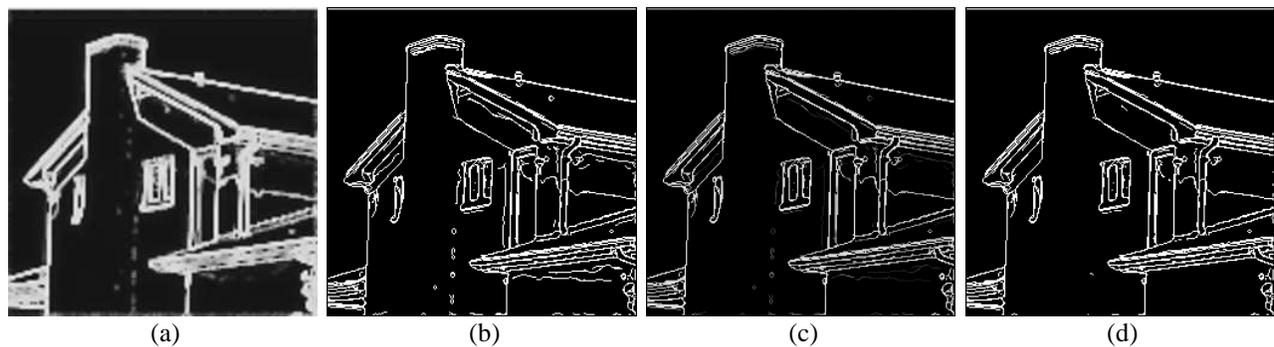


Fig. 11. Using Eq. (21) for the image in Figure 2(f): (a) $\Theta(p_{1,all}(c_e/r))$; (b) $e_{L^*a^*b^*}$; (c) $p_{L^*a^*b^*}(c_e/r)$; and (d) BEM .

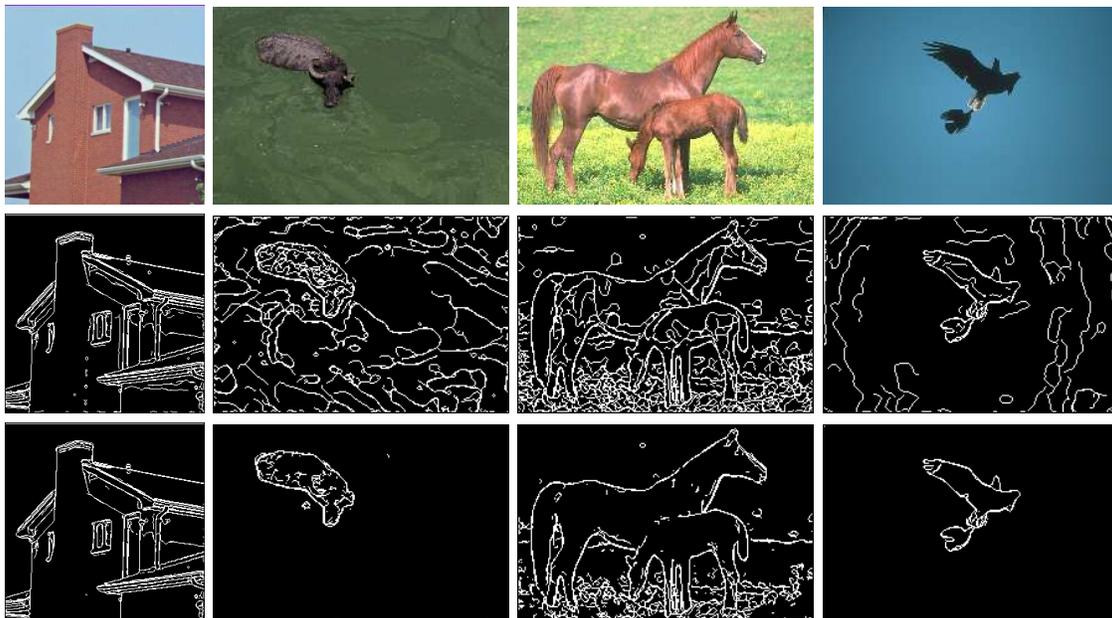


Fig. 12. Sample results for Canny edge detector and BEM : (row 1) Original images; (row 2) using only Canny edge detector; and (row 3) BEM .

BEM (Figure 11(d)) the Canny edge detector is applied with its upper and lower thresholds set to 0. However, we note that since different types of images have different types of edges which can only be detected by using different sets of thresholds, it is not possible to use a single set of thresholds which is effective for all images.

Figure 12 shows some edge detection results using the Canny edge detector $e_{L^*a^*b^*}$ with manually selected optimum thresholds for different types of images. Note that, the two thresholds for the Canny edge detector also used in generating BEM are always set to 0. Figure 12 shows that unlike BEM (generated by the proposed multiscale edge detector), using only the Canny edge detector produces false detection for the highly textured areas.

D. Detecting seed points to represent colour image regions

The homogeneous colour image regions are represented by seed points and each cluster of connected seed points corresponds to a region of colour or texture dissimilar to other neighbouring regions. The regions are segmented into disjoint areas using the multiscale edge information given by Eq. (21) and their BEM . Thus, the region edge information is used to identify seed points. Note there are no edges within a homogeneous region. Seed points are detected using the following rules: if a pixel is not an edge pixel in BEM , and there are no edge pixels within its 3×3 neighbourhood then the pixel is classified as a seed

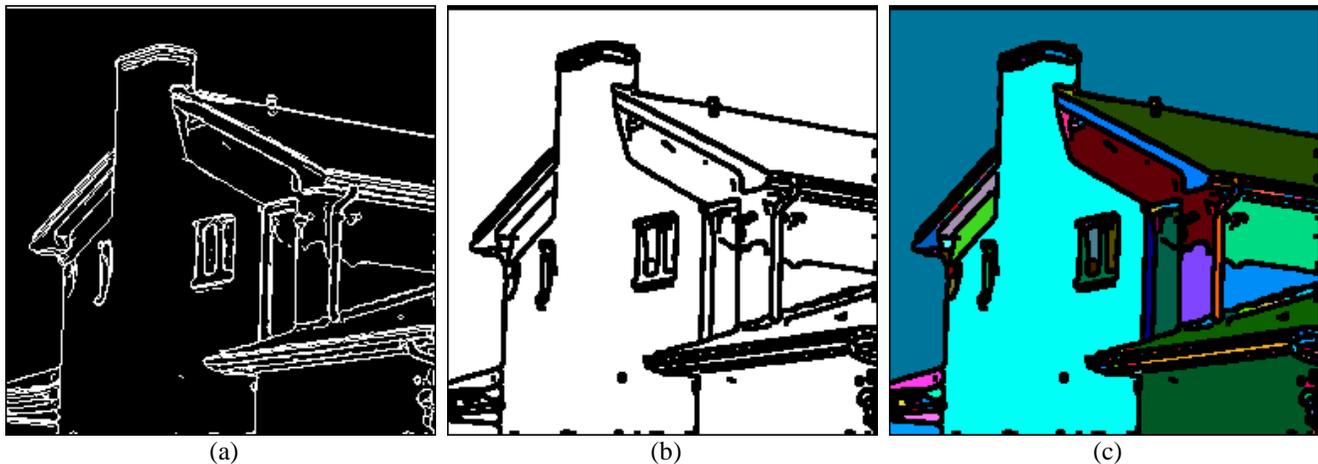


Fig. 13. Seed extraction: (a) BEM in Figure 11(d); (b) SM ; and (c) Coloured S_i 's.

pixel, i.e.,

$$SM(x, y) = \begin{cases} 1 & \text{if } (TEM(x, y) \equiv 0) \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

where $TEM(x, y) = \sum_{u=-1}^1 \sum_{v=-1}^1 BEM(x+u, y+v)$, and SM is a seed map with seed points labelled as 1 (white) and non-seed points labelled as 0 (black), and (x, y) is a spatial pixel location.

It is clear from Figure 13(b) that each region in House image is represented by a group of seed points. In order to group all the seed points detected within each region to obtain a single representative seed region S_i , an algorithm based on the connected component labelling algorithm [31] is used to generate a label matrix LM . The algorithm is a two-pass process. At the start of the process all labels in LM and the current label i are initialised to 0. In the first pass, SM is examined from its upper-left to its lower-right corner. When the current pixel at (x, y) is found to be ON (i.e., $SM(x, y) = 1$), the neighbouring pixels to its left $((x-1, y))$, top $((x, y-1))$, top-left $((x-1, y-1))$ and top-right $((x+1, y-1))$ are examined. The following three situations can occur: none of these neighbours is ON, and a new label (equals to the value of current label i incremented by 1) is assigned to the current pixel in LM ; one of the neighbours is ON, and the current label is set to the label of that neighbour and assigned to the current pixel in LM ; and more than one neighbours are ON, the current pixel in LM is set to one of the labels, and all labels of these neighbours are put in an equivalence class to indicate that these labels correspond to the same region on the basis of being connected. After the first pass is completed, all labels in each equivalence class are merged to a single label. After this process, each region S_i is represented by a label i . Figure 13(c) shows the 144 seed regions (S_i) obtained by applying the labelling algorithm on SM .

E. Region growing using seed regions and region merging

Each connected seed region S_i is grown and all pixels that are labelled as non-seed (e.g., the black pixels in Figure 13(b)) are assigned to the appropriate seed region. The region growing algorithm is as follows:

- 1) Create a link-list of neighbouring pixels for each seed region, and sort each link-list in ascending order with respect to the minimum colour distance of each pixel to its neighbouring S_i 's.
- 2) Remove the first element of each link-list which has the smallest distance to its nearest neighbouring seed region and assign the corresponding pixel to the nearest seed region. Update the mean of the corresponding seed region vector and

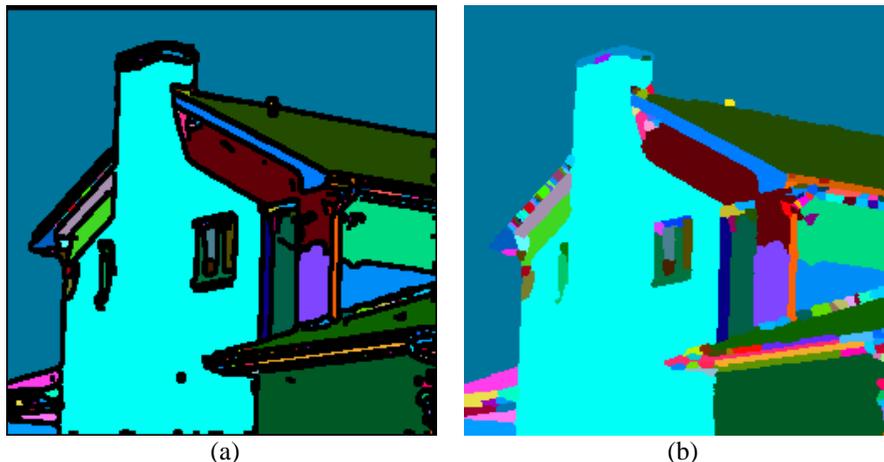


Fig. 14. Effect of region growing on seed regions: (a) Coloured seed regions; and (b) Result of region growing.

add the removed pixel's unlabelled neighbouring pixels to the link-list with respect to their distance to the nearest seed region.

3) Go to step 2 while there are elements in link-list to process.

Let $D(p_1, p_2, p_3) = \sqrt{p_1^2 + p_2^2 + p_3^2}$ and use $\Delta = \frac{D((L_r^* - L_{S_i}^*), (a_r^* - a_{S_i}^*), (b_r^* - b_{S_i}^*))}{D(L_r^*, a_r^*, b_r^*)}$ as the colour distance between pixel r with (L_r^*, a_r^*, b_r^*) and region S_i with region mean colour $(L_{S_i}^*, a_{S_i}^*, b_{S_i}^*)$. Other criteria may be used in region growing, e.g. the nearest pixels with the original colour value. But this results in an increase in memory requirement and computational complexity. Since the region growing is a process which needs high computation, its load should be kept as low as possible. Thus we applied the mean region colour as the only similarity criterion. After region growing is applied, each seed region S_i is grown to be R_i . The region growing process combats the enlargement of edges caused by the seed selection process. This is illustrated in Figure 14. Note that the black pixels in Figure 14(a) are merged with the coloured regions due to their similarity in colour. Figure 15 demonstrates the results of the region growing, where each region R_i is coloured with the average colour of the corresponding region. Note that there is an oversegmentation which is removed by region merging.

The region merging handles any oversegmentation by merging neighbouring regions that satisfy the merging conditions. A region should be merged to its neighbouring region with similar colour, i.e., the colour distance between them is smaller than a predefined threshold, and if its size is smaller than a predefined threshold. The size threshold becomes important if two neighbouring regions have similar colour but their sizes are not small, e.g., a sky region should not be merged with a sea region.

The merging algorithm in [26] is modified and used as follows. The $YCbCr$ colour space used in [26] is replaced with CIE $L^*a^*b^*$ since the latter provides better colour difference metrics. Two adjacent regions (R_i, R_j) with the smallest colour distance among all regions are examined first. In each iteration, if the distance is smaller than a threshold then the regions are merged. The mean colour of the merged region, and distances between the merged region and its neighbouring regions are re-calculated. This process is repeated until no two adjacent regions have colour distance smaller than the threshold. The colour distance [26] between two adjacent regions R_i and R_j is re-defined as

$$d(R_i, R_j) = \frac{D((L_{R_i}^* - L_{R_j}^*), (a_{R_i}^* - a_{R_j}^*), (b_{R_i}^* - b_{R_j}^*))}{\min(D(L_{R_i}^*, a_{R_i}^*, b_{R_i}^*), D(L_{R_j}^*, a_{R_j}^*, b_{R_j}^*))} \quad (23)$$

where $(L_{R_i}^*, a_{R_i}^*, b_{R_i}^*)$, and $(L_{R_j}^*, a_{R_j}^*, b_{R_j}^*)$ are the mean colour of region R_i and R_j , respectively. The threshold value for the colour similarity is set to 0.1. It is sometimes possible that two regions with large colour distance between them are merged

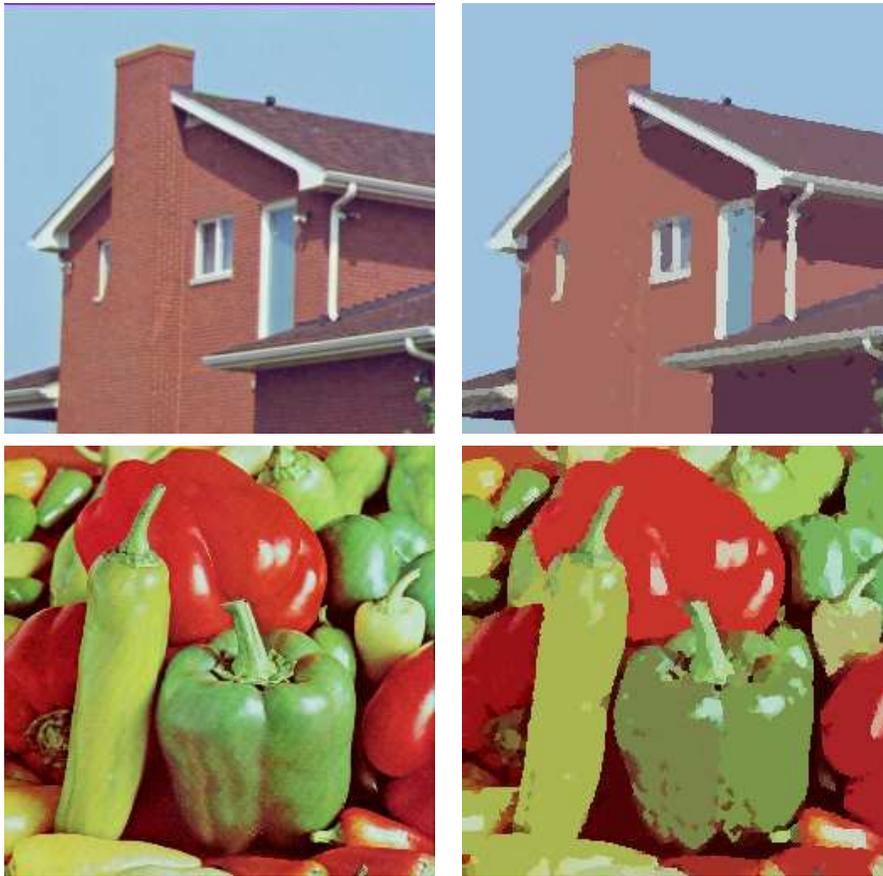


Fig. 15. Original and pseudo-colour representation of segmented images after region growing.

due to their small size. In order to remove this artifact, while merging with respect to region sizes, a second threshold is used and which is set to 0.15. This procedure is repeated until no regions have size smaller than the threshold. The size threshold used in this merging step is $1/150$ of the total number of pixels in an image.

Figure 16 demonstrates the region merging. Figure 16(d) shows the result of merging adjacent regions with colour distance smaller than 0.1, where similar regions but segmented differently in Figure 16(c) are merged together. Figure 16(e) shows further merging of regions with size smaller than $1/150$ of original image size to adjacent region with the smallest colour distance. The result consists of similar regions that should be merged together, i.e., these are results of oversegmentation. In order to achieve a better segmentation and to overcome oversegmentation, the same merging strategy is applied with different thresholds. In the second stage: the colour distance is set as 0.15 to prevent the algorithm from merging different regions whose sizes are suitable for merging; and the size threshold is set as $1/50$ of the total number of pixels in an image. Similar to the first stage, the size threshold is used in conjunction with the other condition of colour distance threshold of 0.225 to prevent merging of small sized regions that should not be merged. This is unlike the method in [26] which does not check for colour difference when merging regions with respect to size and thus may merge regions which are small in size but have very high contrast. Our method prevents adding very high contrast regions to the mean representation of big sized regions. The oversegmentation in Figure 16(e) is reduced in Figure 16(f) by the second stage.

The thresholds used in region merging affect the overall segmentation. If the colour distance threshold is too high then some different regions may be merged together; conversely there will be too many regions. Similarly for the size threshold: too high a threshold incorrectly merges different objects and too low results in oversegmentation. The selection of these thresholds is

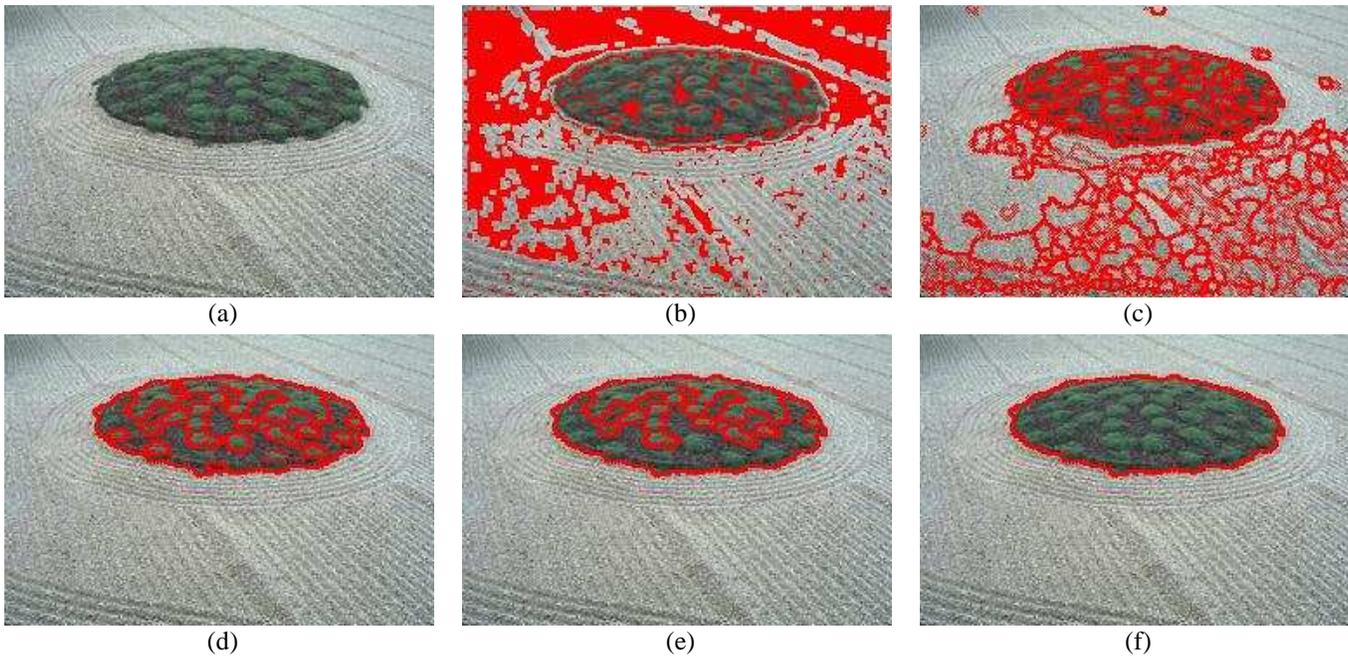


Fig. 16. Demonstration of region merging: (a) Original image; (b) Seed pixels are depicted in red; (c) Result of region growing; (d) Result of merging adjacent regions with colour distance smaller than 0.1; (e) The result of merging small regions with size smaller than 1/150 of the image; and (f) Final segmented image using further steps in region merging with different colour distance threshold of 0.15 and size threshold of 1/50 of the overall image size.

ad-hoc but better values are obtained if the content of the image is considered. For example, if the image contrast between regions is not high, than it would be better to set the colour distance thresholds as small as possible; and if there are small sized objects in the image, then the size threshold should be as high as possible to prevent losing objects.

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

The performance of the proposed algorithm is evaluated using 300 natural images from the Berkeley dataset [32]. The performance is compared with that of JSEG [10] and Mean-shift [15]. The number of scales for both the proposed algorithm and JSEG is set as 3 and no other parameters are changed in the tests. The default optimum threshold values used for every image in [10], [15] are used for both JSEG and Mean-shift. For JSEG the colour quantization threshold is set to “automatic selection”, the region merge threshold is set to 0.4, and the number of scales is set to “automatic selection”. Mean-shift is a unified approach for colour image denoising and segmentation which uses a kernel in the joint spatial-range domain to filter image pixels in the CIE $L^*u^*v^*$ colour space. The filtered pixels are clustered to segment the objects. The algorithm requires a selection of spatial (h_s) and colour (h_r) bandwidths, and optionally a minimum area parameter M for region merging. The parameters $(h_s, h_r, M) = (7, 6.5, 20)$ are used for Mean-shift.

To compare the performance of the three algorithms quantitatively different measures are used. One measure is the peak signal to noise ratio (PSNR) between the original image and the pseudo-colour representation of segmented image, i.e.,

$$\text{PSNR} = \frac{1}{3} \sum_{i=1}^3 \left(10 \log_{10} \left(\frac{1}{\frac{1}{K} \sum_{x=1}^K (I_i(x) - \hat{I}_i(x))^2} \right) \right) \quad (24)$$

where I_i and \hat{I}_i are respectively the i th, $i \in \{R, G, B\}$, colour channel values of the original image and pseudo-colour representation of the segmented image, x is spatial location and K is number of pixels in the image. Both I_i and \hat{I}_i are normalized to 1. A high value of PSNR is highly likely to indicate that the original and pseudo-colour representation are quite similar. Incorrect segmentation where boundaries are not well defined is indicated by a low PSNR value.

Even though PSNR measures the similarity between two images, it does not consider the contextual information and is thus not a meaningful measure for the boundary of the overall segmented result. For this purpose, we use the methodology defined by Martin *et al.* [33] which considers human segmented images as ground truth boundaries. For a given ground truth boundary image and a segmentation result, the quantities Precision (P) and Recall (R) are computed to produce a precision-recall curve for the algorithm. Precision is the fraction of detections that are true positives rather than false positives, while Recall is the fraction of true positives that are detected rather than missed. Precision is a measure of how much noise is in the output of the detector and Recall is a measure of how much of the ground truth is detected. Even though these two quantities are good indicators for boundary representation, a single measure is required to show how good the segmented boundary is. Thus,

$$F = \frac{P \times R}{(\alpha \times R + (1 - \alpha) \times P)} \quad (25)$$

where α is a threshold and set to 0.5 as is used in [33]. The higher the value of F the better is the boundary representation.

Figure 17 clearly shows that the proposed algorithm (using default threshold values) is better in segmenting region boundaries when compared to the other algorithms (also using their default threshold values). The second row of Figure 17 shows a scene which consists of similar coloured objects where the contrast between object boundaries is not high, e.g., the contrast between some branches of tree and mountain. Even with this image condition, the proposed algorithm correctly detects most of the different objects in the scene, i.e., the branches, sea and mountain. Note that the oversegmentations are noticeable in the results by Mean-shift and there are missing boundaries in the results by JSEG. For example, in the first row JSEG segments the House image into 31 regions and fails to detect the upper left window, the left guttering which is extended toward a wall, and a small green plant near the left guttering. Furthermore the region boundaries are not sufficiently smooth. These missing regions are well detected by Mean-shift. However Mean-shift segments the image into 124 regions. This oversegmentation gives visibly pleasant result but poor regional representation. These drawbacks are overcome by the proposed algorithm which segments the House image into 32 regions, where regional information is retained with good boundary representation.

Table I shows the values of PSNR, number of segmented regions N_r , P , R and F for the images in Figure 17. In terms of PSNR, the performance of the proposed algorithm is second to Mean-shift. However, Mean-shift produces the most oversegmentation. Also, the number of segmented regions for JSEG and the proposed algorithm is about the same. This effect is reflected through the values of F for which our algorithm gives the highest (i.e., best) values. For instance Mean-shift segments Lena image (row 4 of Figure 17) into 354 regions and achieves the best PSNR value, but most of these regions are over-segmentation and should be merged. It is visually clear that Lena image does not have 352 regions, it thus means Mean-shift produces a large number of false positives. On the other hand, it also produces a large number of true positives. For this reason, the value of P , R and F are respectively 0.47, 0.98, and 0.64. On the other hand the values of F for the JSEG and the proposed algorithm are 0.69 and 0.75, respectively. This is mainly due to the low false positive rates of the two algorithms with respect to Mean-shift. Thus, the proposed algorithm achieves the best overall performance.

The performance of the proposed algorithm in the presence of different levels of noise is also evaluated and compared with JSEG and Mean-shift. The 512x512 House image (i.e., top-left image in Figure 17) contaminated with different levels of zero-mean Gaussian noise is used as the test image. The PSNR between the original House image and the segmentation of each noisy House image (for various noise levels) is calculated (for each of the three methods). The N_r , P , R , and F in each of the segmented images are also calculated. Table II shows that the values of PSNR, N_r , P , and F for Mean-shift deteriorate the most as the noise level is increased, thus indicating that the method is the least robust against noise. The change in values of PSNR for both JSEG and the proposed algorithm is similar as the noise level is increased. However, the proposed algorithm

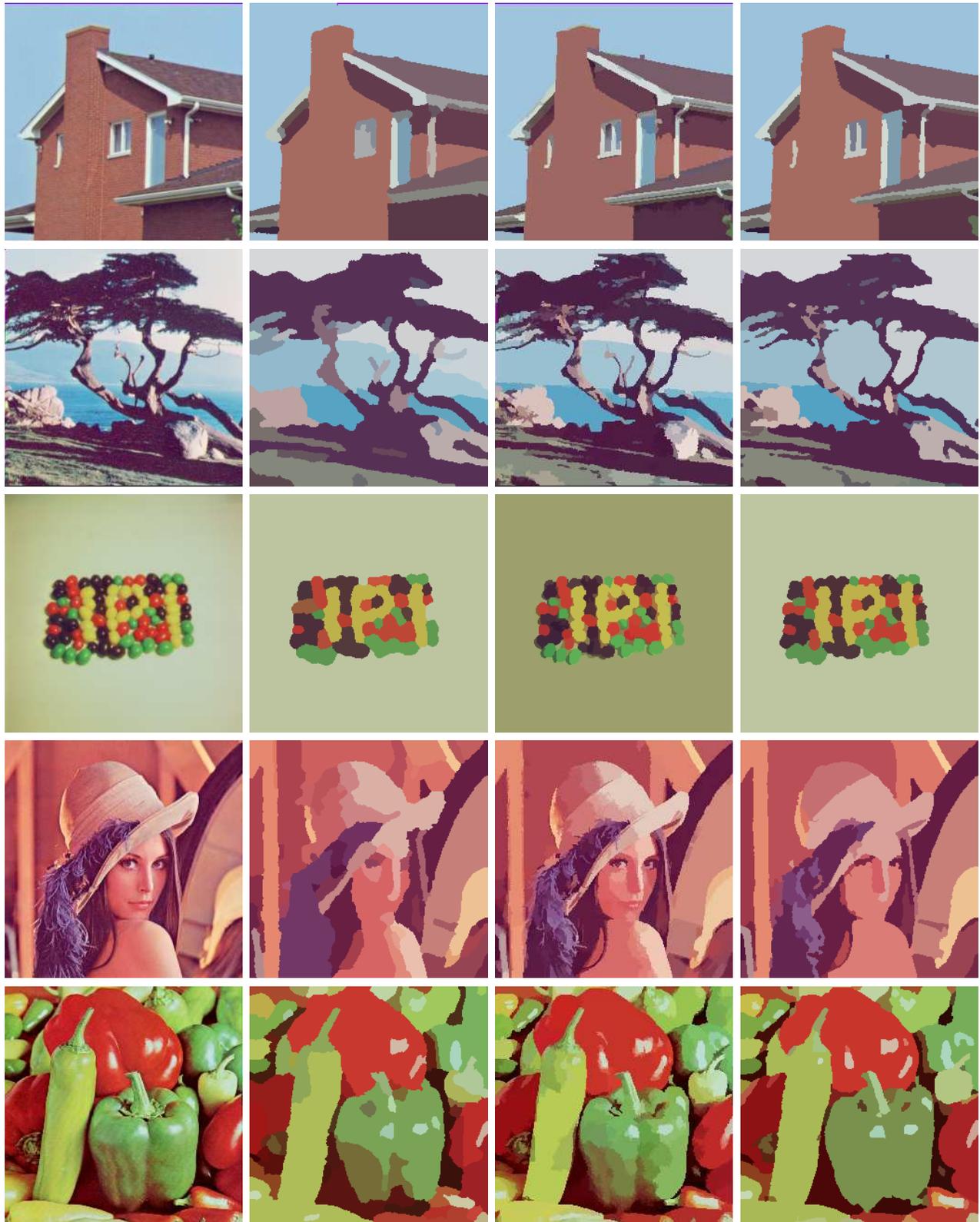


Fig. 17. A selection of segmentation results: (column 1) Original images; and (column 2 to 4) The pseudo-colour representation of segmented images respectively using JSEG, Mean-shift and the proposed algorithm.

TABLE I
PSNR's, N_r 's, P 's, R 's, AND F 'S FOR THE IMAGES IN FIGURE 17.

Row #	JSEG					Mean-shift					Proposed				
	PSNR	N_r	P	R	F	PSNR	N_r	P	R	F	PSNR	N_r	P	R	F
1	23.58	31	0.93	0.86	0.89	27.14	124	0.80	1.00	0.89	24.17	33	0.92	0.98	0.95
2	18.32	39	0.73	0.84	0.78	22.73	337	0.51	0.98	0.67	19.94	66	0.70	0.88	0.78
3	23.38	29	1.00	0.96	0.98	26.45	113	0.91	1.00	0.95	24.83	37	0.99	0.99	0.99
4	21.59	64	0.58	0.86	0.69	24.93	354	0.47	0.98	0.64	22.01	53	0.65	0.89	0.75
5	19.79	74	0.52	0.9	0.66	24.24	456	0.38	1.00	0.55	21.12	80	0.56	0.96	0.71

TABLE II
PSNR's, N_r 's, P 's, R 's, AND F 'S FOR DIFFERENT LEVELS OF NOISE INTRODUCED TO THE HOUSE IMAGE.

PSNR (dB)	JSEG					Mean-shift					Proposed				
	PSNR	N_r	P	R	F	PSNR	N_r	P	R	F	PSNR	N_r	P	R	F
16.38	23.01	31	0.86	0.90	0.88	24.70	1301	0.26	1.00	0.41	23.95	36	0.91	0.97	0.94
20.12	23.08	30	0.88	0.90	0.89	25.65	1128	0.27	1.00	0.43	23.86	34	0.92	0.97	0.94
21.34	23.35	30	0.88	0.89	0.88	26.77	677	0.36	1.00	0.53	24.06	34	0.92	0.97	0.94
26.04	23.50	29	0.90	0.88	0.89	27.38	259	0.61	1.00	0.76	24.09	34	0.92	0.98	0.95
30.00	23.48	29	0.90	0.88	0.89	27.45	209	0.66	1.00	0.80	24.16	34	0.91	0.98	0.94
36.00	23.62	30	0.90	0.89	0.89	27.31	138	0.76	1.00	0.86	24.16	34	0.91	0.97	0.94
39.95	23.61	30	0.92	0.86	0.89	27.23	124	0.78	1.00	0.88	24.25	33	0.91	0.98	0.94
∞	23.58	31	0.93	0.86	0.89	27.14	124	0.8	1.00	0.89	24.17	33	0.92	0.98	0.95

produces the most consistent N_r , thus the boundary representation of the algorithm is more robust with respect to noise. This is also reflected in values of F , which are plotted in Figure 18, which show that our algorithm performs best against noise. Although the F value is a suitable figure of merit to evaluate the quality of segmentation, it is also important to visually examine the effects of noise contamination on the segmentation to support the idea that F value gives a good indication of the performance of the segmentation algorithm. The sample results shown in Figure 19 support Figure 18 visually that the proposed algorithm almost performs well even for highly noisy images where the other algorithms fail, and its region boundaries are well represented when compared to those of the other algorithms.

The sample segmentation results using JSEG, Mean-shift, Jung's method [14]¹ and our proposed algorithm on the Berkeley data-set are shown in Figure 20. The corresponding F values are shown in Table III. It is very clear that in most of the results our algorithm performs better than JSEG and the Jung's method, and Mean-shift produces the most oversegmentation. The best boundary representation of the segmented images is achieved using the proposed algorithm.

In order to demonstrate that our algorithm could be used in optical character recognition algorithms we select an image from the internet and segmented it using 4 different segmentation algorithms as shown in Figure 21. The F values for 4 different segmentation algorithms are 0.68, 0.86, 0.83, and 0.97 for JSEG, Mean-shift, Jung's method [14], and the proposed method respectively. It is clear that both Mean-shift and the proposed algorithm correctly detect all the characters while the other algorithms miss some of them. Furthermore Mean-shift produces an oversegmentation which makes it unsuitable for the character recognition.

¹Thanks for Dr. Claudio Rosito Jung for providing us the segmented images.

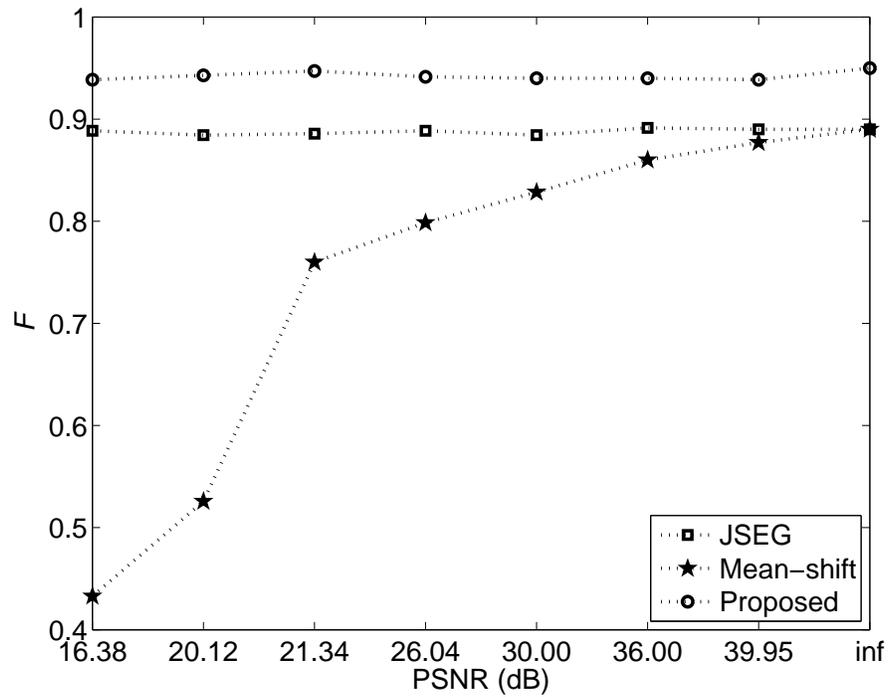


Fig. 18. F values of the algorithms versus different levels of Gaussian noise.



Fig. 19. Segmentation on noisy images: (column 1) Noisy images; and (column 2 to 4) The pseudo-colour representation of segmented images respectively using JSEG, Mean-shift and the proposed algorithm. The values of PSNR for the first and second row are respectively 26.04 dB and 16.38 dB.



Fig. 20. Sample segmentation results on the Berkeley data-set: (column 1) Original images; and (column 2 to 5) The pseudo-colour representation of segmented image respectively using JSEG, Mean-shift, the method proposed in [14] and the proposed algorithm.

TABLE III
 P 's, R 's, and F 's FOR THE SAMPLE SEGMENTATION RESULTS IN FIGURE 20.

Row #	JSEG			Mean-shift			JUNG [14]			Proposed		
	P	R	F	P	R	F	P	R	F	P	R	F
1	0.49	0.75	0.59	0.43	0.97	0.6	0.55	0.71	0.62	0.54	0.91	0.68
2	0.56	0.83	0.67	0.48	0.99	0.65	0.68	0.90	0.77	0.72	0.90	0.80
3	0.79	0.84	0.81	0.61	0.98	0.75	0.78	0.85	0.81	0.78	0.93	0.85
4	0.45	0.82	0.58	0.35	0.98	0.52	0.43	0.91	0.58	0.42	0.92	0.58
5	0.58	0.75	0.65	0.49	0.98	0.65	0.53	0.76	0.62	0.56	0.84	0.67
6	0.74	0.79	0.76	0.54	1.00	0.7	0.64	0.96	0.77	0.69	0.92	0.79



Fig. 21. Character segmentation on a simple image: (a) original image; (b) result using JSEG; (c) result using Mean-shift; (d) result using the method proposed in [14]; and (e) result using the proposed algorithm.

IV. CONCLUSION

This paper presents a novel algorithm for unsupervised colour image segmentation, which is based on a multiscale edge detection in DT-CWT domain and involves region growing and region merging. The proposed edge-preserving smoothing filter is good not only in removing noise, it also retains the contrast between regions. Using the Rayleigh distribution to model the complex subband coefficients provides a posteriori probability that a pixel is an edge pixel. This information is enhanced using the multiscale structure of DT-CWT. However the multiscale structure generates thick edges at the coarsest scale, thus requiring the Canny edge detector to thin the final edge map. The extracted edge map is used to find seed regions via the proposed morphological binary structures. These regions are further grown and merged to obtain the final segmented image. Thus, the performance of the edge detection affects the segmentation result. The region merging is also crucial for the overall algorithm performance: the better region merging criterion makes it possible to achieve better results.

Using the Berkeley data-set and the House image contaminated with various levels of Gaussian noise, we show that the performance of the proposed algorithm is better than JSEG, Mean-shift and the method proposed in [14]. The boundaries of objects segmented by the proposed algorithm are also the best. The algorithm only generates a segmentation error when an object edge cannot be detected due to very low contrast between different regions. The proposed algorithm can be used for both textured and non-textured objects.

ACKNOWLEDGMENTS

The authors would like to thank Warwick University Vice Chancellor Scholarship for providing the funds for this research.

REFERENCES

- [1] Y.W. Tai, J. Jia and C.K. Tang, "Soft Color Segmentation and Its Applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1520–1537, 2007.
- [2] H. Cheng, C. Chen, H. Chiu and H. Xu, "Fuzzy Homogeneity Approach to Multilevel Thresholding," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1084–1086, Jul 1998.
- [3] H.D. Cheng, X.H. Jiang and J. Wang, "Colour image segmentation based on homogram thresholding and region merging," *Pattern Recog.*, vol. 35, no. 2, pp. 373–393, Feb 2002.
- [4] N. Otsu, "A Threshold Selection Method from Grey-Level Histograms," *IEEE Trans. Sys. Man Cybernet.*, vol. 9, no. 1, pp. 62–66, Jan 1979.
- [5] P. Saha and J. Udupa, "Optimum image threshold via class uncertainty and region homogeneity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 5, pp. 689–706, Jul 2002.
- [6] J. Basak, B. Chanda and D. Dutta Majumder, "On edge and line linking with connectionist models," *IEEE Trans. Sys. Man Cybernet.*, vol. 24, no. 3, pp. 413–428, Mar 1994.
- [7] F.Y. Shih and S. Cheng, "Adaptive mathematical morphology for edge linking," *Information Sciences*, vol. 167, no. 1–4, pp. 9–21, Dec 2004.
- [8] S.A. Hojjatoleslami and J. Kittler, "Region growing: a new approach," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1079–1084, Jul 1998.
- [9] A. Tremeau and N. Borel, "A region growing and merging algorithm to colour segmentation," *Pattern Recog.*, vol. 30, no. 7, pp. 1191–1203, Jul 1997.
- [10] Y. Deng and B.S. Manjunath, "Unsupervised segmentation of colour-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug 2001.
- [11] Y. Wang, J. Yang and N. Peng, "Unsupervised color-texture segmentation based on soft criterion with adaptive mean-shift clustering," *Pattern Recogn. Lett.*, vol. 27, no. 5, pp. 386–392, Apr 2006.
- [12] P. Scheunders and J. Sijbers, "Multiscale watershed segmentation of multivalued images," *Proc. 16th Int. Conf. on Pattern Recognition*, vol. 2, pp. 855–858, 2002.
- [13] I. Vanhamel, I. Pratikakis, and H. Sahli, "Multiscale gradient watersheds of color images," *IEEE Trans. Image Process.*, vol. 12, no. 6, pp. 617–626, Jun 2003.
- [14] Claudio Rosito Jung, "Unsupervised multiscale segmentation of color images," *Pattern Recogn. Lett.*, vol. 28, no. 4, pp. 523–533, Mar 2007.
- [15] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 5, pp. 603–619, 2002.
- [16] J. Fan, D.K.Y. Yau, A.K. Elmagarmid and W.G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1454–1466, 2001.
- [17] E. Navon, O. Miller and A. Avertuch, "Color image segmentation based on adaptive local thresholds," *Image Vision Comput.*, vol. 23, pp. 69–85, 2005.
- [18] J. Fan, G. Zeng, M. Body and M.-S. Hacid, "Seeded region growing: an extensive and comparative study," *Pattern Recogn. Lett.*, vol. 26, no. 8, pp. 1139–1156, Jun 2005.
- [19] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 7, pp. 710–732, 1992.
- [20] J. Sun, D. Gu, Y. Chen and S. Zhang, "A multiscale edge detection algorithm based on wavelet domain vector hidden Markov tree model," *Pattern Recog.*, vol. 37, no. 7, pp. 1315–1324, Jul 2004.
- [21] L. Zhang and P. Bao, "Edge detection by scale multiplication in wavelet domain," *Pattern Recogn. Lett.*, vol. 23, no. 14, pp. 1771–1784, Dec 2002.
- [22] X.-L. Wang and L.-C. Jiao, "A SAR Image Despeckling Method Based on Dual Tree Complex Wavelet Transform," *Advances in Intelligent Computing*, Edited by D.S. Huang, X.-P. Zhang, G.-B. Huang, vol. 1, pp. 59–67, 2005.
- [23] S.-C. Zhu and A. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, pp. 884–900, 1996.

- [24] R. Nock and F. Nielsen, "Statistical Region Merging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1452–1458, Nov 2004.
- [25] K. Haris, S.N. Efstratiadis, N. Muglaviras and A.K. Katsaggelos, "Hybrid image segmentation using watershed and fast region merging," *IEE Trans. Image Process.*, vol. 7, pp. 1684–699, Dec 1998.
- [26] F.Y. Shih and S. Cheng, "Automatic seeded region growing for color image segmentation," *Image Vision Comput.*, vol. 23, no. 10, pp. 877–886, Sep 2005.
- [27] N. Kingsbury, "Image processing with complex wavelets," *Philos. Trans. Roy. Soc. Lon.*, vol. 357, pp. 2543–2560, 1999.
- [28] G. Wyszecki and W.S. Styles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 3rd ed., John Wiley and Sons, New York, 1982.
- [29] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. IEEE Int. Conf. on Comput. Vision*, pp. 839–846, 1998.
- [30] J. Scharcanski, C. Jung and R. Clarke, "Adaptive image denoising using scale and space consistency," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 1092–1101, 2002.
- [31] R. Gonzalez and R. Woods, *Digital Image Processing, 2nd Ed.* Upper Saddle River, New Jersey: Prentice-Hall, Inc., 2002.
- [32] D. Martin, C. Fowlkes, D. Tal and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," *Proc. IEEE Int. Conf. on Comput. Vision*, vol. 2, pp. 416–423, 2001.
- [33] D.R. Martin, C.C. Fowlkes and J. Malik, "Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, May 2004.