Image Segmentation Using a Texture Gradient Based Watershed Transform

Paul R. Hill, C. Nishan Canagarajah, and David R. Bull

***Abstract*—The segmentation of images into meaningful and homogenous regions is a key method for image analysis within applications such as content based retrieval. The watershed transform is a well established tool for the segmentation of images. However, watershed segmentation is often not effective for textured image regions that are perceptually homogeneous. In order to properly segment such regions the concept of the "texture gradient" is now introduced. Texture information and its gradient are extracted using a novel nondecimated form of a complex wavelet transform. A novel marker location algorithm is subsequently used to locate significant homogeneous textured or non textured regions. A marker driven watershed transform is then used to properly segment the identified regions. The combined algorithm produces effective texture and intensity based segmentation for the application to content based image retrieval.**

***Index Terms*—Image edge analysis, image segmentation, image texture analysis, wavelet transforms.**

I. Introduction

THE watershed transform is a well established tool for the segmentation of images. Instead of using the image directly, the transform uses a gradient image extracted from the original image. The initial stage of any watershed segmentation method is therefore to produce a gradient image from the actual image. Some element of smoothing is always necessary within gradient extraction schemes (e.g., [1]) in order to emphasize the significant gradient within the image and reduce the gradient caused by noise or other minor structures.

Vital information characterising texture can be lost in this smoothing operation. In order to improve the generalization of watershed techniques and apply them properly to images containing significant amounts of texture content, the texture content information should be preserved within the algorithm.

Texture boundaries have been used for the effective partitioning of natural images using the edge flow technique [2]. However, this technique does not use a measure of texture gradient but compares the texture content at each pixel to its neighbors in order to "flow" its texture content in the maximum gradient direction. Where "texture flows" meet, boundaries are constructed. Although often effective, this method makes no use of the watershed technique. By using the watershed transform with a texture gradient we make use of the well understood theoretical basis and the large body of work associated with the watershed transform.

However, over-segmentation, a major problem with the watershed transform, will not be solved by the use of the texture gradient. There are many solutions to the problem of over-segmentation discussed below. We develop a novel marker based solution (basins are flooded from selected sources rather than minima). This method lends itself well to the intended application of image region characterization for content based retrieval. This is because the resulting boundaries will still be centered on key gradient maxima and the regions can be made to be over a minimum segmentation size.

II. Image Segmentation

The aim of computer based image segmentation is to isolate/distinguish/subdivide a digital image into its constituent parts or objects. Image segmentation is often used as an initial transformation for general image analysis and understanding. Therefore image segmentation has a large number of applications within many disparate fields. Examples of image segmentation applications include remote sensing, medical image analysis and diagnosis, computer vision and the image segmentation necessary to enable region indexing in a Content Based Retrieval (CBR) application. Image segmentation to enable region indexing within a CBR system is the intended application and therefore the focus of this work.

The ultimate aim of an automatic image segmentation system is to mimic the human visual system in order to provide a meaningful image subdivision. This is often unfeasible, not only because of the technical difficulties involved but also because scenes and objects themselves can be hierarchically segmented with no indication of what level of hierarchy is meaningful to an intended user. For example, a picture of a crowd can be segmented into either groups of people, individuals or elements of each individual e.g., torso, limbs etc. This generally means there is rarely any "correct" spatial segmentation for natural images.

Another difficulty of mimicking the human approach to segmentation is that a human can draw on an unconstrained and simply enormous set of object models with which to match scene elements to. Whilst some model matching segmentation algorithms have been successfully implemented (e.g., Eigenface matching in the Photobook application developed by Pentland *et al.* [3]) these lead to the problem that the system can only recognize objects it is looking for. This creates problems of generalization within unconstrained domains such as natural images.

The more usual methods of image segmentation rely on more primitive visual aspects of images such edges, color, texture,

Manuscript received May 6, 2002; revised October 10, 2002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nicolas Merlet.

The authors are with the University of Bristol, Bristol BS5 6JF, U.K. (e-mail: paul.hill@bristol.ac.uk).

Digital Object Identifier 10.1109/TIP.2003.819311

1057-7149/03$17.00 © 2003 IEEE

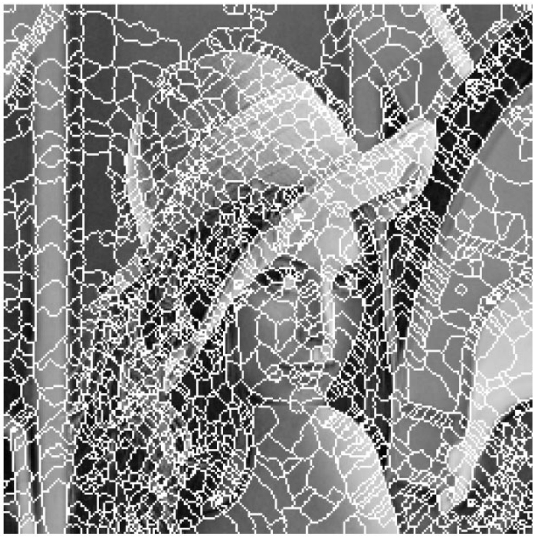


Fig. 1. Result of a simple watershed transform of the lena test image.

motion (for video) or various combinations. There are two generally used methods for primitive image segmentation: discontinuity based methods and similarity based methods. The interface between two different contiguous homogenous regions is usually marked by a discontinuity in gray-level, color, or texture. A segmentation can therefore be based on the detection of such discontinuities. Similarity based methods of segmentation are based on similarity comparisons between pixels and regions. Rudimentary similarity based methods include simple thresholding and region growing. More sophisticated similarity based methods include the so called split and merge and clustering techniques in order to partition pixels within a defined feature space. The watershed transformation is a technique for segmenting digital images that uses a type of region growing method based on an image gradient. It thus effectively combines elements from both the discontinuity and similarity methods described above.

A. Watershed Transformation: A Review

Fig. 1 shows an example of a rudimentary watershed segmentation of the Lena test image. This image clearly shows the basic results of the watershed transformation, i.e., many closed and connected regions whose boundaries follow the contours of the image and where the union of all the regions forms the entire image.

The watershed method was originally developed by Lantuéjoul [4] and is extensively described together with its many applications by Beucher and Meyer [5]. Since its original development with grey-scale images [4] it has been extended to a computationally efficient form (using FIFO queues) [6] and applied to color images [7]. The main advantages of the watershed method over other previously developed segmentation methods are [5]:

- The resulting boundaries form closed and connected regions. Traditional edge based techniques most often form disconnected boundaries that need post-processing to produce closed regions.
- The boundaries of the resulting regions always correspond to contours which appear in the image as obvious contours of objects. This is in contrast to split and merge methods where the first splitting is often a simple regular sectioning of the image leading sometimes to unstable results.
- The union of all the regions form the entire image region.

The main disadvantage is that for most natural images the watershed transformation produces excessive over-segmentation (see Fig. 1). This is unacceptable for a segmentation method intended to produce local characterization for CBR. Therefore some form of pre or post processing is required in order to produce a segmentation that better reflects the arrangement of objects within the image. Such pre or post-processing methods include: region merging [8], hierarchical watersheds [8], marker based watershed segmentation [9] and adaptive smoothing [10] morphological filtering [11].

All these processes have advantages and disadvantages and should be selected according to the intended application. However none are ideally suited to the application to content based retrieval as discussed below.

III. Texture Gradient

A. Texture Characterization

In order to produce a texture gradient we first need to characterize the texture content of the image at each pixel. A number of methods have been proposed to do this. One of the most popular techniques is the use of a set of scaled and orientated complex Gabor filters (e.g., [12]). By suitable spanning of the frequency space, each pixel can be characterized in texture content. However, when considering the differences in texture within an image (e.g., the texture gradient) this often produces suboptimal characterization for segmentation. To produce an optimal system, the Gabor filters need to be tuned to the texture content of the image. Different schemes for adaptive Gabor filtering have been implemented [13], [14]. These and other schemes use arbitrary techniques that are entirely separate from the texture feature extraction process whilst also being excessively computationally complex.

In order to integrate an adaptive scheme with the texture feature extraction process we have developed the Non-Decimated Complex Wavelet Packet Transform (NDXWPT). The magnitude of the coefficients of each complex subband can be used to characterize the texture content. This is because the basis functions from each subband (very closely) resemble Gabor filters, i.e., they are scale and directionally selective whilst being frequency and spatially localized.

Each pixel can therefore be assigned a feature vector according to the magnitudes of the NDXWPT coefficients. A pixel at spatial position (x, y) has one feature for each NDXWPT subband coefficient magnitude at that position: defined as $T_i(x, y)$, where i is the subband number. A feature vector $\boldsymbol{T}(x, y)$ is therefore associated with each pixel characterising the texture content at that position.

IV. Non-Decimated Complex Wavelet Transform

In order to develop the nondecimated complex wavelet packet transform (NDXWPT) transform, it is first instructive to develop a nonadaptive version using the NDXWT transform.

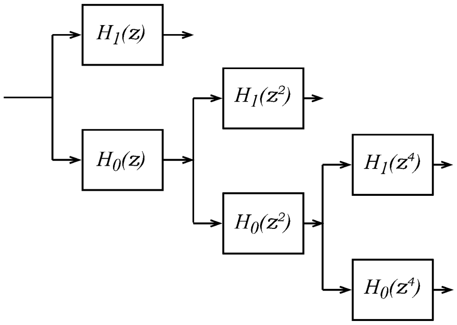


Fig. 2. Non-decimated wavelet transform.

The structure of the nondecimated complex wavelet packet transform is based upon the Non-Decimated Wavelet Transform (NDWT) (defined in [15]) combined with the Complex Wavelet Transform (XWT) (defined in [16]).

A. Non-Decimated Wavelet Transform (NDWT)

The nondecimated wavelet transform uses the same structure as the DWT but without any subsampling. This leads to a transform that has every subband having exactly the same number of coefficients as the number of samples in the original signal. This of course leads to an overcomplete representation. However, this transform has the advantage of being completely shift invariant, an aspect which has been exploited by applications such as image fusion [15], where it was described as the Shift Invariant Discrete Wavelet Transform (SIDWT). The structure of the NDWT is shown in Fig. 2. This clearly shows that in order to retain the same effective filtering without the downsampling, the filters at each stage must be upsampled with the right number of zeros. The filters in Fig. 2 are best represented in the z-transform domain with the filters $H_0(z)$ and $H_1(z)$ representing the low and high pass filters respectively. The filters used at level i are $H_0(z^{2^{i-1}})$ and $H_1(z^{2^{i-1}})$. The $z^{2^{i-1}}$ terms represent the original filters upsampled with 2^{i-1} zeros in-between each original filter tap.

B. Complex Wavelet Transform (XWT)

The XWT was developed by Magarey and Kingsbury [16] for motion estimation. It was soon surpassed by the dual tree wavelet transform (DT-CWT) as the XWT wasn't found to be invertible and to have inflexible frequency characteristics. However, unlike the DT-CWT, the XWT is ideally suited to a conversion to a nondecimated form.

1) Original Complex Filters: The original form of the XWT as used in [16] is based upon two complex valued FIR filters that approximate two Gabor filters:

$$h_o(n) \approx a_0 e^{(n+0.5)^2/2\sigma_0^2} e^{j\omega_0(n+0.5)} \qquad (1)$$

$$h_1(n) \approx a_1 e^{(n+0.5)^2/2\sigma_1^2} e^{j\omega_1(n+0.5)}$$

$$\text{for } n = -D, \ldots, D-1. \qquad (2)$$

This pair of complex high pass (h_1) and low pass (h_0) filters are used with a standard DWT structure [17] to form the XWT in one dimension. The bandpass and lowpass coefficients at level m can be considered to be the downsampled output of equivalent wavelet $\psi^{(m)}$ and scaling $\phi^{(m)}$ filters, respectively

$$\textbf{bandpass } y^{(m)}(n) = \sum_k x(k)\psi^{(m)}(2^m n - k) \qquad (3)$$

$$\textbf{lowpass } \hat{y}^{(m)}(n) = \sum_k x(k)\phi^{(m)}(2^m n - k). \qquad (4)$$

The key property of this transform is that (under certain conditions) given the Gabor like FIR filters h_0 and h_1 the equivalent wavelet $\psi^{(m)}$ and scaling $\phi^{(m)}$ filters are also approximately Gabor like

$$\psi^{(m)}(n) \approx a_m e^{(n-n_m)^2/2\sigma_m^2} e^{j\omega_m(n-n_m)} \qquad (5)$$

$$\phi^{(m)}(n) \approx \hat{a}_m e^{(n-n_m)^2/2\hat{\sigma}_m^2} e^{j\hat{\omega}_m(n-n_m)}$$

$$\text{for } n = -(2^m-1)D, \ldots, (2^m-1)(D-1). \qquad (6)$$

For the parameters within (1) and (2) set to $[\omega_0 = \pi/6,\ \omega_1 = 0.76\pi,\ \sigma_0 = 0.97, \sigma_1 = 1.07,\ a_0 = 0.47,\ a_1 = 0.43j,\ D = 2]$ the high and low pass filters can be closely approximated by the rationally valued filters

$$h_0 = \frac{1-j, 4-j, 4+j, 1+j}{10} \qquad (7)$$

$$h_1 = \frac{-1-2j, 5+2j, -5+2j, 1-2j}{14}. \qquad (8)$$

Of key importance is the ratio of the lowpass $\psi^{(m)}$ and bandpass $\phi^{(m)}$ filter's center frequencies at each level m. This ratio can be defined as:

$$R_m = \frac{\omega_m}{\hat{\omega}_m} \qquad (9)$$

where ω_m and $\hat{\omega}_m$ are the center frequencies of $\psi^{(m)}$ and $\phi^{(m)}$, respectively. Simply using the filter in (7) and (8) was found to give wide variations of $w_m/\hat{w}_m$ from level to level (especially in the first few levels). A constant ratio of R_m over all levels is desirable as this will consistently tile the frequency space. In order to make this ratio constant for all levels a prefilter f is used to convolve the signal at the first level in addition to h_0 and h_1 filters (see [16] for justification). f is defined as

$$f = \frac{-j, 5, j}{5}. \qquad (10)$$

This new structure using the prefilter is shown in Fig. 3.

Fig. 4(a) and (b) show the impulse responses of the resulting scaling ψ and wavelet ϕ filters at level 4 respectively. The Gabor filter nature of these effective filters is clear from these figures. Fig. 4(c) shows the frequency response of the first four level wavelet functions of the XWT together with the fourth level scaling function. This plot shows the expected Gaussian shape to the frequency response of the Gabor like effective filters in octave frequencies. It also shows how the negative frequencies are effectively negated for all effective filters.

2) New Complex Filters: The use of the even length complex filters above will place the subband coefficients halfway between the original signal samples. When the subsequently presented nondecimated form (see Section IV-C) is used, this means that there will not be a direct one-to-one mapping relation between each subband coefficient and each image pixel. In

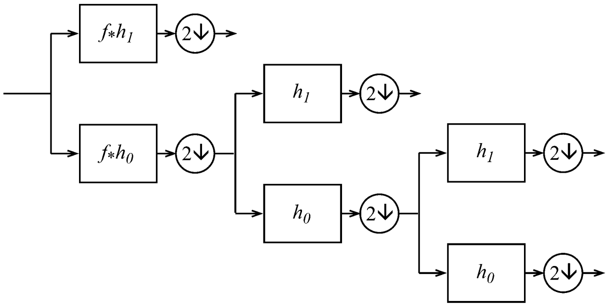


Fig. 3. Wavelet decomposition with prefilter f.

order to overcome this problem, the filters are modified to have five instead of four taps. This is effected by initially removing the 0.5 offset within equations 1 and 2, i.e.,

$$h_o(n) \approx a_0 e^{(n)^2/2\sigma_0^2} e^{j\omega_0(n)} \tag{11}$$

$$h_1(n) \approx a_1 e^{(n)^2/2\sigma_1^2} e^{j\omega_1(n)}$$

$$\text{for } n = -D, \ldots, D-1. \tag{12}$$

For the parameters within (11) and (12) set to $[\omega_0 = \pi/6,\ \omega_1 = 0.82\pi,\ \sigma_0 = 0.97,\ \sigma_1 = 1.07,\ a_0 = 0.47,\ a_1 = 0.43j,\ D = 2]$ the high and low pass filters can be closely approximated by the odd length, rationally valued filters:

$$h_0 = \frac{1-4i, 19-11i, 36, 19+11i, 1+4i}{76} \tag{13}$$

$$h_1 = \frac{-4+1i, 9-14i, 26i, -9-14i, 4+1i}{60}. \tag{14}$$

These filters produce the frequency responses shown in Fig. 5(c). Fig. 5(a) and (b) also show the effective fourth level complex wavelet and scaling functions produced from these new filters. Through the comparison of Figs. 4 and 5 it is clear that these filters produce wavelet characteristics that are virtually identical to the original filters.

These new filters also share or mirror the properties of the original filters:

Symmetry: Each filter is linear phase. A linear phase filter response is often required for applications such as texture analysis where phase distortions of signal frequency components are not desirable.

Regularity: The regularity of the wavelets are determined by the number of zeros of $H_0(\omega)$ at $\omega = \pi$. In common with the original filters, $H_0(\pi) = 0$. This means that the spatial filters ψ are sampled versions of a continously differentiable wavelet.

DC gain: The DC gain of h_0 is unity. This ensures that the passband gain of the decomposition is consistently unity over the entire decomposition.

Length: Each filter is of length 5. Although longer than the original filters these are the minimum odd length filters that are practical. This minimizes the computational effort for the transform whilst placing the output coefficients coincident with the input samples. The use of rationally valued filters also increases the potential implementation speed of the transform.

3) Perfect Scaling for the New Complex Filters: As shown in Section IV-B-I the ratio of the center frequencies R_m is kept constant over all the levels by using the pre-filter f. This filter was found by solving an Eigen-value problem for the filters 7 and 8 given in [16]. Instead of solving the same Eigen-value problem for the new filters 13 and 14, the same prefilter (10) is used with the new filters. This ensures perfectly scaled wavelets (see below) with the minimum sized pre-filter. R_m for the old filters is 0.35. R_m for the new filters is 0.36. The maximum variation of R_m over all levels for the old filters is 3.2%. The maximum variation of R_m over all levels for the new filters is 3.8% [These compare to the maximum variation of R_m for the old filters without the prefilter of 39%]. This shows that the use of the pre-filter f (10) for the new filters keeps the wavelets perfectly scaled to virtually the same tolerance as the old perfectly scaled filters whilst having the smallest complexity pre-filter. Additionally, the positioning of the wavelet and scaling function center frequencies of the new filters gives almost ideal tiling of the 1-D and 2-D frequency space over all levels. Also R_m being approximately equal to 0.36 over all levels gives the 2-D orientation subband angle of 19°. This is identical to the 2-D orientation subband angle of the old filters as given in [16].

C. Conversion From the XWT to the NDXWT

The implementation of the XWT is identical to the implementation of the DWT with complex valued filters. The conversion of the XWT to its nondecimated form (NDXWT) is therefore identical to the conversion of the DWT to its nondecimated form (NDWT), i.e., all subsampling is removed and all filters past the first level are upsampled as shown in Fig. 2.

D. Extension of the NDXWT to Two Dimensions

The extension of the NDXWT to two dimensions follows easily from the extension of the DWT to two dimensions [17]. The vertical and horizontal complex and imaginary parts of the filtering process are treated separately. The NDXWT therefore also results in six differently oriented complex valued subbands at each scale. The complex basis functions of these six subbands at the fourth level of decomposition are shown in Fig. 6. This shows how these basis functions are equivalent to similarly scaled and orientated two dimensional Gabor functions. A normal wavelet decomposition for two levels using the above complex filters will result in a tiling of the frequency space as shown Fig. 7. This shows an analysis of the frequency content in octave bands that can be continued to further levels. Octave band analysis is the usual scale resolution used with Gabor filtering [18]. Six orientations at each level is also a reasonable orientation selectivity for a similar Gabor analysis. The NDXWT therefore mirrors a common spread of Gabor analysis filters.

The nondecimation of the NDXWT results in all the complex subbands being the same size as the original image. This of course leads to extreme over-completeness. However the advantage of this method for generating the texture gradient needed for segmentation is that there is a one-to-one mapping of the filter results in each subband with the original pixels.

1) Comparison to Gabor Filters: A simple, equivalent and commonly used method of obtaining a pixel by pixel texture analysis of an image is Gabor filtering within the frequency domain [18]. The image is transformed into the frequency domain using a 2-D FFT. The product of the chosen Gabor filters defined

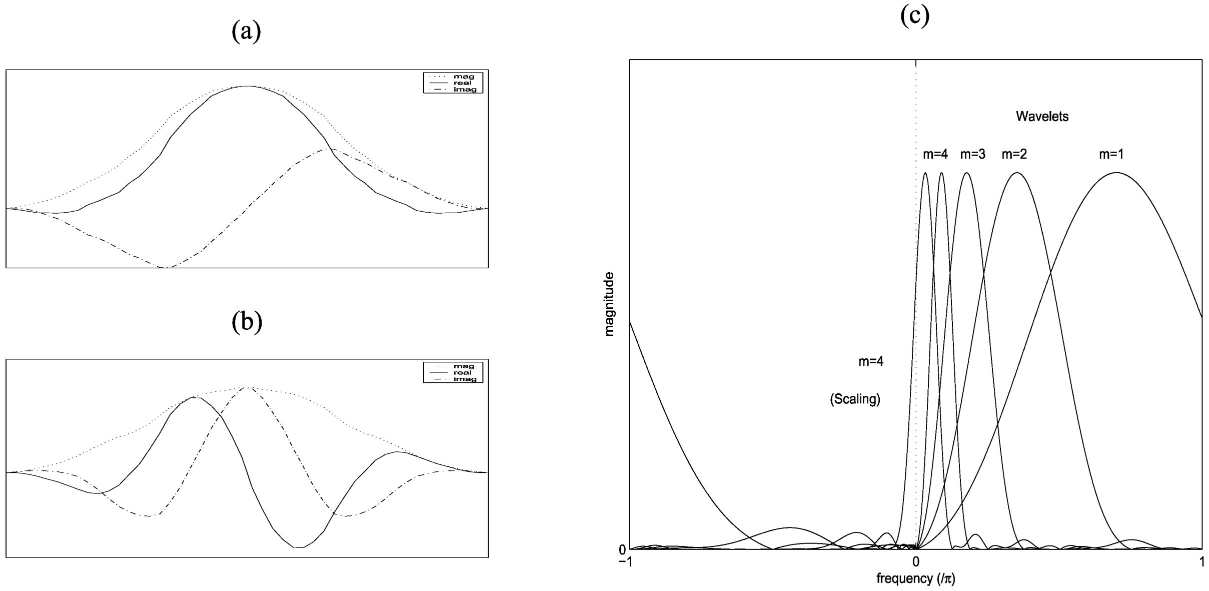


Fig. 4. Spatial and frequency representations of wavelets created from (7) and (8). (a) XWT equivalent scaling filter(level 4). (b) XWT equivalent wavelet filter (level 4). (c) Magnitude frequency responses of XWT equivalent wavelet filters ($m = 1$, 2, 3, 4) and scaling filter ($m = 4$).

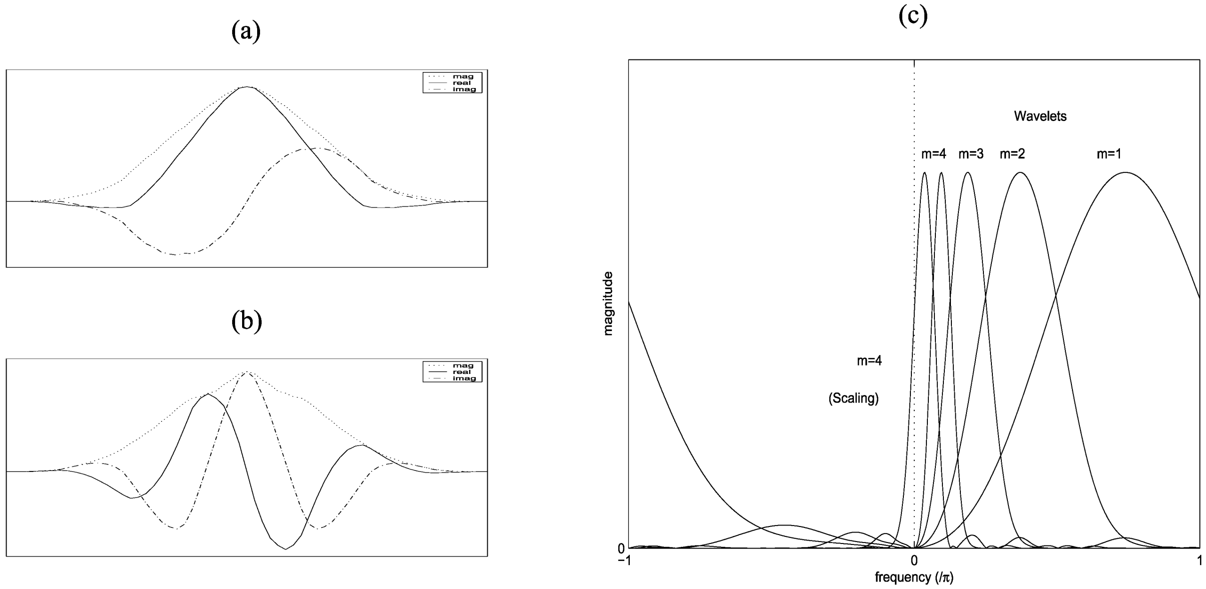


Fig. 5. Spatial and frequency representations of wavelets created from (13) and (14). (a) XWT equivalent scaling filter(level 4). (b) XWT equivalent wavelet filter (level 4). (c) Magnitude frequency responses of XWT equivalent wavelet filters ($m = 1$, 2, 3, 4) and scaling filter ($m = 4$).

in the frequency domain and this FFT image are then inverted to give complex filtered images. The magnitude of the complex filtered images are then used to represent the texture content at each pixel. This is depicted in Fig. 8. Figs. 9 and 10 show the magnitude of a similar Gabor and 2-D NDXWT filtered image respectively together with the filters' frequency responses. This shows the equivalence of the methods.

The first advantage of using a spatial filtering method such as the NDXWT is that the basis functions are spatially limited. The method of filtering in the frequency domain used by most Gabor techniques will be equivalent to using unlimited sized basis functions. Spatial localization of basis functions is important for texture segmentation where the texture characterization should be as localized as possible.

The second advantage of using a spatial filtering method such as the NDXWT is that there is no windowing (necessary with frequency domain filtering techniques). This effect can be seen when comparing the filtered image in figures. The Gabor filtered image has significant edge artefacts.

Finally, and most importantly, a complete NDXWT decomposition will be considerably less computationally complex compared to an equivalent Gabor analysis. Table I shows the number of floating pointing operations required for a three level, six orientation NDXWT on a 256 × 256 image. This table also shows how many floating point operations are required for an equivalent analysis using Gabor filters. The operations necessary for the NDXWT transform are the simple filter operations at each NDXWT level and the final conversion to the

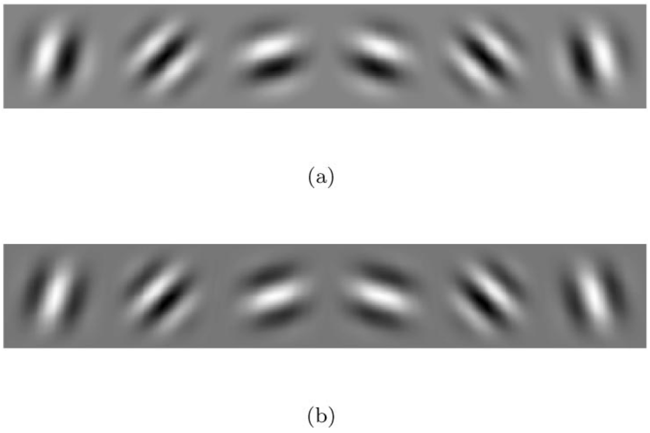


Fig. 6. Grey-scale $(\mathrm{grey} = 0)$ plot of the impulse response for the 6 wavelet filters at level 4 of the 2-D NDXWT. (a) real part. (b) imaginary part.

final complex subbands given in Section VII. The operations necessary for an equivalent Gabor analysis are: An initial forward FFT, the multiplication of each frequency defined Gabor filter with the FFT image and an inverse FFT for each Gabor filter. Not only is it clear that a NDXWT decomposition is more efficient than equivalent Gabor analysis, these figures are based on treating the rationally valued filters as floating point numbers. They could therefore be implemented to further reduce the complexity of the transform.

E. Gradient Extraction From the NDXWT Transform

Fig. 11 shows the magnitude of a single, orientated, second scale subband from a NDXWT complex wavelet decomposition of the Lena test image, Fig. 12(f). This image shows how the texture content is highlighted by wavelet subbands (see the feather region on Lena's hat). It is therefore reasonable to characterize the texture content at each spatial position (x, y) using the feature vector $\boldsymbol{T}(x, y)$ (where i is the NDXWT subband number and the vector $T_i(x, y)$ values are defined as the NDXWT subband coefficient magnitude at that position). This is made possible because the nondecimated wavelet subbands are all the same size as the image.

A simple approach to obtaining the texture gradient of an image would then be to calculate the gradient of each subband magnitude and sum them. This would work for purely textured images. However, all texture extraction methods will give high energy values over simple intensity boundaries found in nontextured image regions (see the edge of the top of the hat in Fig. 11). The gradient of the subband magnitudes will give a double edge at such intensity boundaries. The gradient of each subband should therefore aim at step detection rather than edge detection. A simple method to perform this is a $x - y$ separable median filtering on the magnitude image followed by gradient extraction. This has the effect of removing the edges and preserving the steps. The texture content can then be represented by the median filtered versions of the subband magnitudes $\boldsymbol{MT}(x, y)$. This can be represented by:

$$MT_i(x, y) = \mathrm{MedianFilter}\ (T_i(x, y)) \text{ for } 1 < i \leq n \quad (15)$$

where n is the number of subbands. The median filtering should reflect the size of the edges in the complex subbands produced by simple intensity boundaries in order to negate them, i.e., octave scale median filtering. The length of the separable median filtering is therefore defined as $2^{(l+2)}$ where l is the decomposition level of the given subband. A morphological thinning operation would also be a possible solution to this problem but the median filtering was found to be an initially simple solution.

In order to calculate the gradient of the texture content one needs to consider the gradient within the multidimensional feature space. The simplest way to do this is to sum the gradients obtained for each of the individual features. Defining $TG(x, y)$ to be the magnitude of the texture gradient we have:

$$TG(x, y) = \sum_{i=1}^{n} \frac{|\boldsymbol{\nabla(MT_i(x, y))}|}{l_2(MT_i)} \quad (16)$$

where n is the number of subbands and ∇ is approximated using a Gaussian derivative gradient extraction technique [1] [with the scale parameter σ set to 2.0]. $l_2(MT_i)$ is the l_2 norm energy of the median filtered subband i and is included to normalize the effect of each subband on the gradient.

Fig. 12(d)–(f) show the texture gradient image TG (using the NDXWT) of the images shown in Fig. 12(a)–(c) respectively. The TG gradient clearly highlights the edge of the texture regions in the artificial texture images together with the natural image (see edge of the feathers in the hat). Clearly this gradient is suited to the detection of texture boundaries. In order to preserve the ability of the system to detect intensity changes, this gradient is combined with a simple intensity gradient as follows:

$$G(x, y) = \frac{mix \times (|\boldsymbol{\nabla f}|)^{1.2}}{(F(|\boldsymbol{MT}(x, y)|))^3} + (TG(x, y))^3 \quad (17)$$

where $G(x, y)$ is the final combined gradient and *mix* is a suitably chosen constant for mixing the intensity and texture gradients. ∇f is just the gradient of the plain intensity image calculated using the Gaussian derivative technique defined in [1].

$F(|\boldsymbol{MT}(x, y)|)$ is defined as

$$F(|\boldsymbol{MT}(x, y)|) \\ = \begin{cases} mean(|\boldsymbol{MT}|) & \text{if } |\boldsymbol{MT}(x, y)| <= mean(|\boldsymbol{MT}|) \\ (|\boldsymbol{MT}(x, y)|) & \text{if } |\boldsymbol{MT}(x, y)| > mean(|\boldsymbol{MT}|) \end{cases} \quad (18)$$

The parameters within the definition of G were obtained heuristically and are justified as follows:

The power 1.2 of $(|\boldsymbol{\nabla f}|)^{1.2}$: This is included to increase the dynamic range of the gradient image emphasising the larger gradient values.

The power of 3 of $(TG(x, y))^3$: This is included to emphasize the larger texture gradient values. This is necessary because the newly defined texture gradient has a smaller dynamic range.

The function F: The $(F(|\boldsymbol{MT}(x, y)|))^3$ factor is included to reduce the effect of the spurious gradients within highly textured regions. The F function is included so the

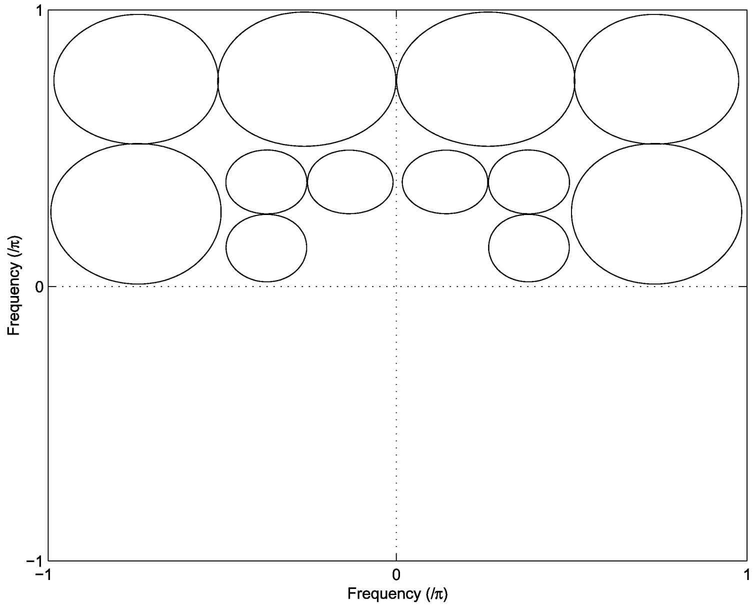


Fig. 7. Tiling of the frequency space for the first two levels of NDXWT decomposition.

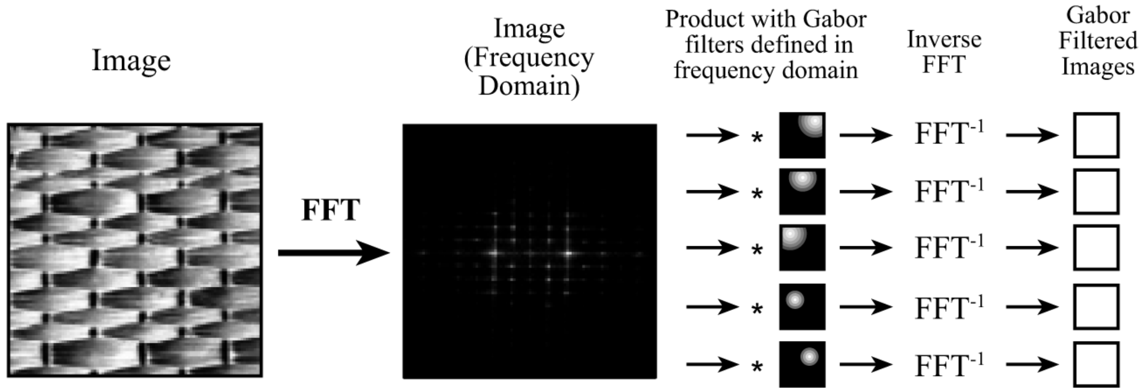


Fig. 8. Gabor filtering for texture analysis.

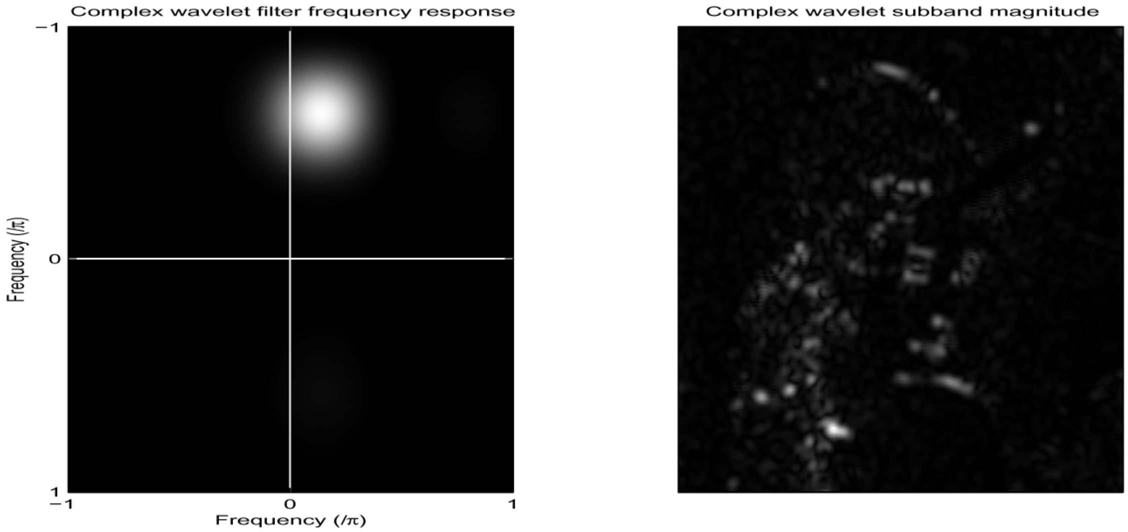


Fig. 9. Non-decimated complex wavelet subband and frequency response.

$(F(|\boldsymbol{MT}(x,y)|))^3$ factor does not relatively elevate the importance of the regions where $|\boldsymbol{MT}(x,y)|)$ is small.

Fig. 13 shows the gradient G for the Lena image. It clearly contains both texture and intensity boundary gradients.

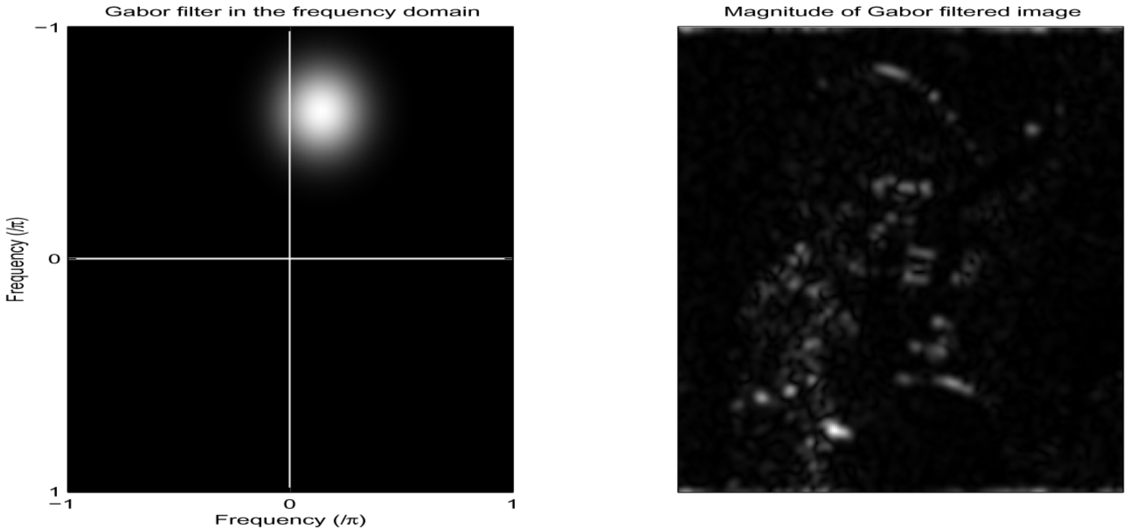


Fig. 10. Gabor filter in the frequency domain together with Gabor filtered image.

TABLE I
COMPARATIVE COMPUTATIONAL COMPLEXITY OF NDXWT AND GABOR TEXTURE ANALYSIS FILTER TECHNIQUES

Analysis Technique	**Number of Floating Point Operations (flops)**
NDXWT 20 subbands (3 levels decomposition)	57815424
Gabor Analysis (20 filters)	135601960

V. THE NON-DECIMATED COMPLEX WAVELET PACKET TRANSFORM (NDXWPT)

A. The Discrete Wavelet Packet Transform (DWPT)

The choice of the wavelet scale parameter within a DWT is 2^i, where $i \in \{0, 1, 2 \ \ldots.\}$. In many applications, such as compression, this is a sensible choice. However, many analysis applications (such as texture classification and segmentation) require a finer frequency analysis. One approach to achieve this is to use the Continuous Wavelet Transform (CWT) to obtain better frequency resolution. The CWT provides an almost unlimited flexibility but comes at the price of greatly increased complexity. An alternative method to achieve better frequency localization whilst retaining the structure of a discrete decomposition is the so called wavelet packet transform.

Wavelet packets are functions that are defined continuously but can be discretely implemented similarly to the wavelet and scaling functions within a DWT. For a given wavelet, an extensive library of wavelet packet bases can be generated for the analysis of many frequency ranges. A signal can then be adaptively decomposed using a selected set of these bases according to a selection condition, based on for example, entropy (see Section V-B).

The generalized decomposition step within a discrete wavelet transform is the splitting of the approximation subband at one level into a further approximation subband and a detail subband at the next level using low and high pass filters respectively. This is then iterated from level to level, further decomposing only the approximation subband at each level, generating a progressively coarser approximation to the signal together with the detail information lost going from level to level. The full discrete wavelet packet decomposition iteratively decomposes both the approximation and detail subbands at each level. This results in a larger binary tree decomposition that generates two branches from each node as shown in Fig. 14.

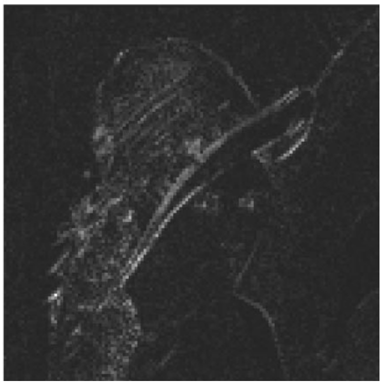
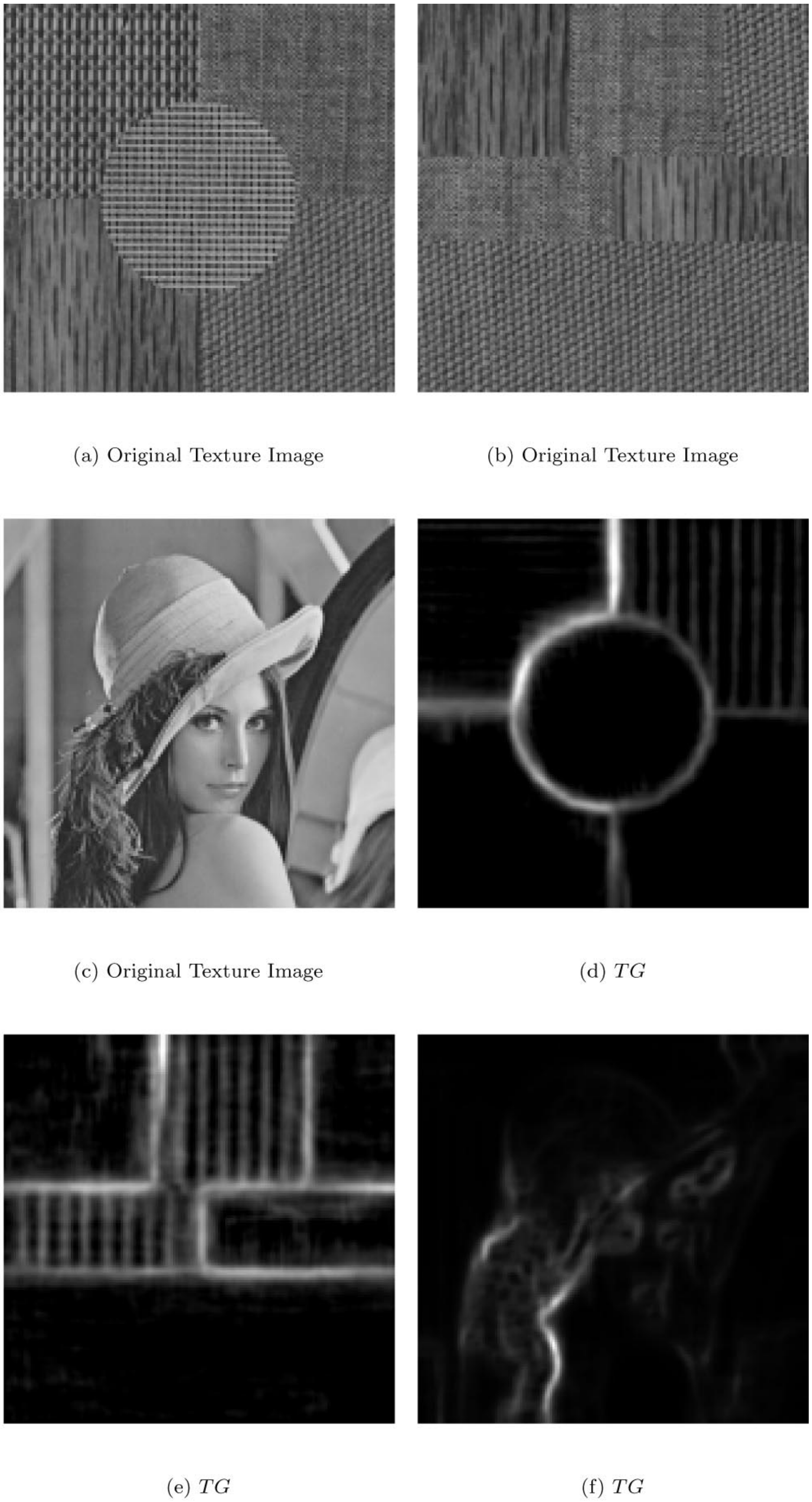


Fig. 11. Complex wavelet subband.

The normal DWT tree and the full DWPT binary tree are just two extremes of the many possible decomposition trees. Specific trees tailored to signal content can be created by only splitting each subband according to a defined splitting condition (see Section V-B).

Because of the potentially complex structure of the decomposition tree, it is important to properly label the produced subbands and equivalent filters. It is useful at this point to consider


(a) Original Texture Image

(b) Original Texture Image

(c) Original Texture Image

(d) TG

(e) TG

(f) TG

Fig. 12. Texture gradient TG for various images.

the effective filters at each node in the tree. I choose to label them $H_j^{(i)}, \ j = 0, \ldots, 2^i - 1$. In other words, $H_j^{(i)}$ is the jth equivalent filter through one of possible paths of length i. The ordering of the filters can be arbitrarily labeled but I choose the one corresponding to a full tree with a lowpass in the lower branch of each fork, and start numbering from the bottom [19].

In order to examine this labeling structure, consider all the possible effective filters at the first and second level of decomposition. It is convenient to do this in the z-transform domain.

The two effective filters at the first level of decomposition are:

$$H_0^{(1)}(z) = H_0(z), \quad H_1^{(1)}(z) = H_1(z). \tag{19}$$

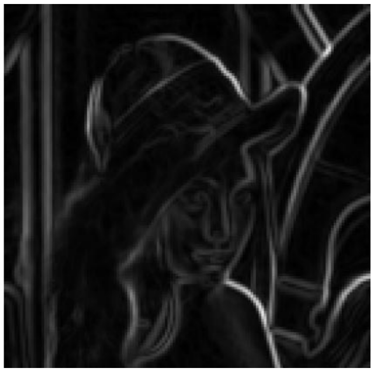


Fig. 13. G (from (18)) of the lena test image.

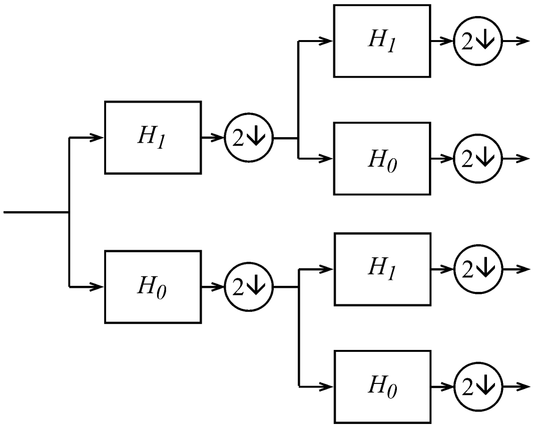


Fig. 14. Full wavelet packet binary tree for two levels.

Where H_0 and H_1 represent the FIR filters defined by the choice of discrete wavelet. The four effective filters at the second level of decomposition are:

$$\begin{aligned} H_0^{(2)}(z) =& H_0(z)H_0(z^2), \\ H_1^{(2)}(z) =& H_0(z)H_1(z^2) \qquad (20) \\ H_2^{(2)}(z) =& H_1(z)H_0(z^2), \\ H_3^{(2)}(z) =& H_1(z)H_1(z^2). \qquad (21) \end{aligned}$$

Where the z^2 terms represent the upsampling of zeros within the second filter.

The set of all possible trees up to a defined level forms the set from which the signal adapted ideal tree can be chosen from.

1) Continuous Wavelet Packet Functions: Although not used within the subsequent application it is instructive to see how the discrete wavelet packet transform is related to the continuous wavelet packet functions. The continuous wavelet ψ and scaling functions ϕ can be related to the discrete implementation filters (h_0 and h_1) using the two, two scale relations (assuming orthogonality).

$$\phi(t) = \sqrt{2}\sum_n h_0[n]\phi(2t-n) \qquad (22)$$

$$\psi(t) = \sqrt{2}\sum_n h_1[n]\phi(2t-n) \qquad (23)$$

Similarly, ϕ satisfies a two scale equation between any two iterated scale changes. If $h_j^{(i)}(z)$ is the impulse response of the iterated filter $H_j^{(i)}(z)$, the iterated two scale equation becomes:

$$\phi(t) = 2^{i/2}\sum_{k=0}^{L^{(i)}-1} h_0^{(i)}[k]\phi(2^i t-k) \qquad (24)$$

Similar limiting continuous "wavelets" can be defined for other iterated filters such as those in (20) and (21). For example, the four filters (whose impulse responses are $h_0^{(2)}, h_1^{(2)}, h_2^{(2)}$ and $h_3^{(2)}$) in (20) and (21) produce the four continuous wavelet packet functions ϕ and $\psi_i, (i=1,2,3)$ defined by:

$$\phi(t) = 2\sum_k h_0^2[k]\phi(4t-k) \qquad (25)$$

$$\psi_i(t) = 2\sum_k h_i^2[k]\phi(4t-k) \qquad (26)$$

B. Best Basis Selection

In order to tailor a wavelet packet decomposition to the signal content each subband $x_j^{(i)}$ (the output from the effective filter $H_j^{(i)}$ can be split only if a given condition is met. Two typical conditions are:

$$\text{condition1}: E(x_j^{(i)}) < E\left(x_{2j}^{(i+1)}\right) + E\left(x_{2j+1}^{(i+1)}\right) \qquad (27)$$

$$\text{condition2}: F\left(x_j^{(i)}\right) > T \qquad (28)$$

where E is a functional measure of information content within a subband (i.e., entropy), F is a suitably defined functional measure of general subband content (such as energy or variance) and T is a scalar threshold. If the condition is satisfied then subband $x_j^{(i)}$ is split into the subbands $x_{2j}^{(i+1)}$ and $x_{2j+1}^{(i+1)}$.

Various entropy measures E have been defined such as:

Shannon Entropy: This is the most commonly used measure and is defined as $E = -\sum_k s_k^2 \log(s_k^2)$

Log Energy Entropy: $E = \sum_k \log(s_k^2)$

Norm Entropy: $E = \sum_k |(s_k)|^p$ where $1 \leq p$

where s_k is the coefficient sample within the given subband.

Within compression and denoising applications the entropy splitting conditions are usually used (i.e., condition 1). This is because these applications require the optimal decomposition based on the maximum information content.

Condition 2 is used within the interference suppression application given in [20]. The function F within this application is just the variance of the subband. This type of conditional test can tailor the tree to match the desired properties of the transform.

VI. The Structure of the NDXWPT Transform

The structure of the NDXWPT is a simple extension of the transform of the NDXWT to a packet transform, i.e., the complex filters are upsampled instead of the signal being downsampled with all complex subbands being further decomposed at each level. This results in a more versatile analysis resolution of the frequency space whilst retaining the property of the XWT of negating negative frequencies. Care, however, must be taken to

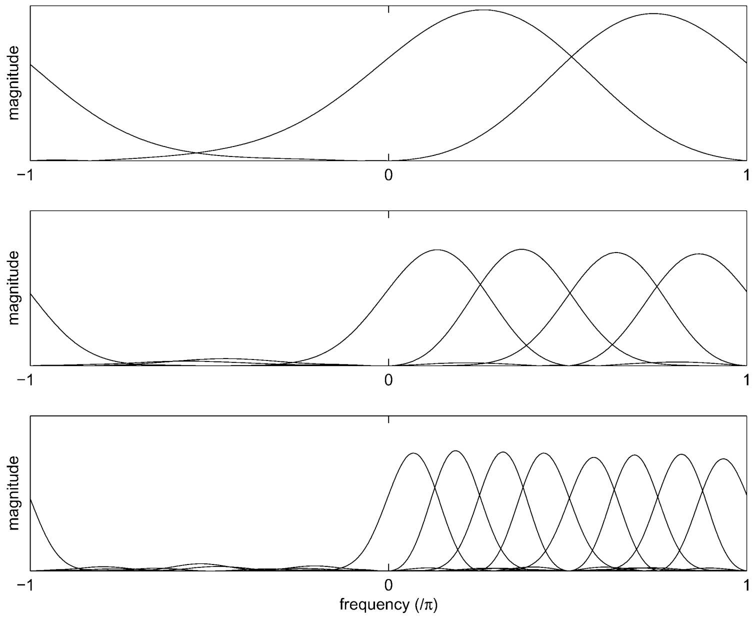


Fig. 15. Frequency coverage of the first three levels of a NDXWPT decomposition.

effect the correct conjugation of wavelet packet filters in order to consistently negate the correct frequency sign.

Fig. 15 shows the frequency response of the filters of the first three levels of a full NDXWPT decomposition using the complex filters shown in (13) and (14). It is clear that the NDXWPT is able to effectively tile one half of the frequency space to a required resolution.

VII. Extension of the NDXWPT to Two Dimensions

The NDXWPT transform can easily be applied in two dimensions in a similar way to the NDXWT, i.e., the NDXWPT is applied separately in the vertical and horizontal directions resulting in differently orientated and scaled complex subbands. As with the one dimensional case, care must be taken to effect the correct conjugation of the wavelet packet filters in order to consistently negate the same frequency sign.

VIII. Best Basis NDXWPT Decomposition for Adaptive Texture Content Analysis

The effective filters and the frequency tiling shown in Section VII depicts a two dimensional full wavelet packet decomposition. In order to tailor the transform to the image a best basis decomposition should be used similarly to that described in the one dimensional case given in Section V-B. The simple best basis selection rule was chosen to be condition 2 in (28), i.e., $F\left(x_j^{(i)}\right) < T$ where function F is the l_2 norm energy of the complex subband magnitude and T is a scalar threshold. The 2-D packet transform therefore follows the same structure as the 1-D packet transform except the functional measure F is used with the 2-D complex subband magnitude image. In addition, the 2-D complex magnitude is split into four further complex subbands at the next level of decomposition if the condition is not met. In order to reduce the extent of the decomposition to a manageable number of levels, I set a limiting level of the decomposition, i.e., the remaining subbands at that set level are then included in the final decomposition. Figs. 18 and 19 illustrates the frequency tiling of a best basis NDXWPT decomposition with the limiting decomposition level set to level 3 and the scalar threshold T set to 10.0. Each circle on these figures shows the frequency coverage of each effective filter. The larger circles obviously represent the subbands which were insignificant (in terms of the splitting condition) at the initial decomposition levels. As most natural images contain more low frequency information it was found that by increasing the scalar threshold T the best basis decomposition will tend to a standard wavelet decomposition (i.e., the NDXWT). Conversely by decreasing the scalar threshold T the best basis decomposition will tend to the full tree decomposition as depicted in Section V-B. The threshold T is set to 10.0

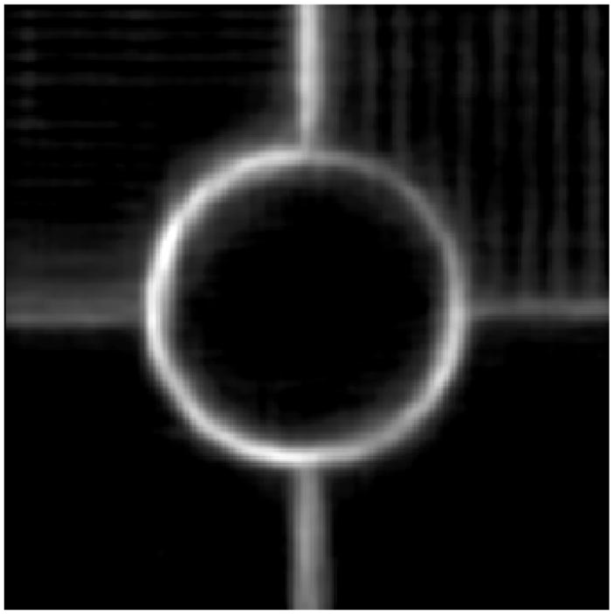
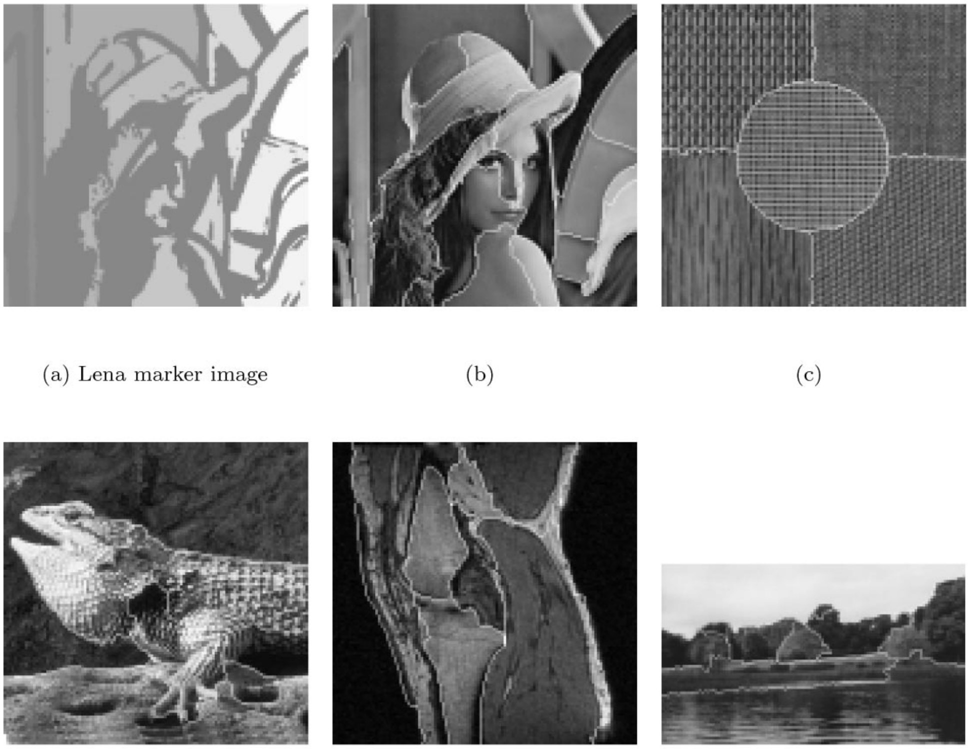


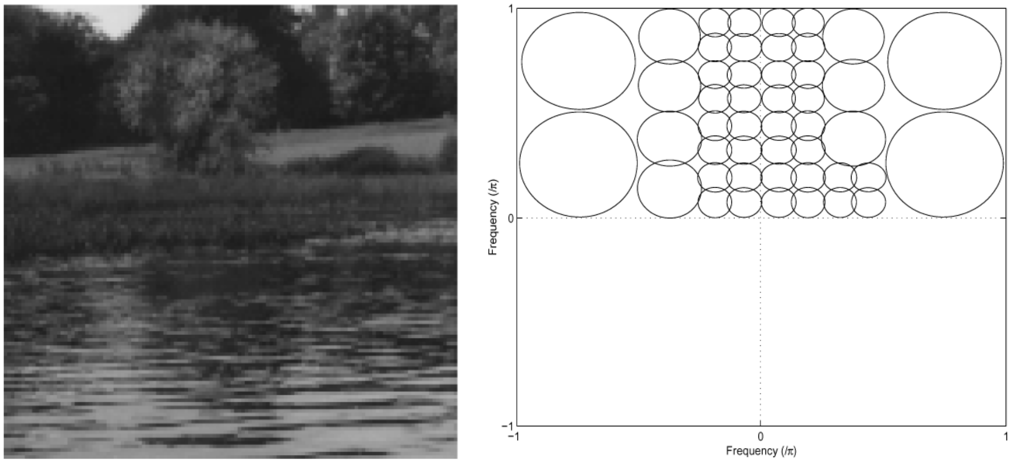
Fig. 16. Texture gradient TG for Fig. 12(a) using the NDXWPT.



(a) Lena marker image (b) (c)

(d) (e) (f)

Fig. 17. Marker and segmentation results.

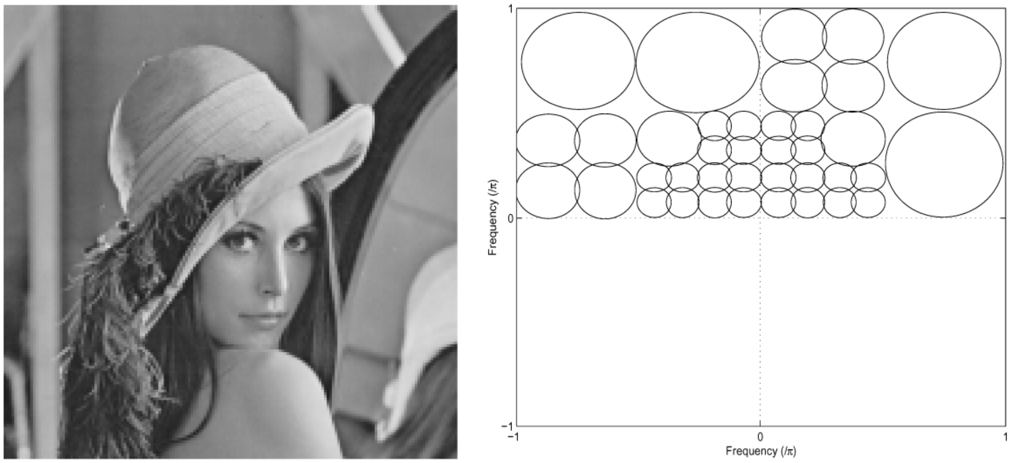


(a) Original Image (b) Frequency Coverage

Fig. 18. Tiling of the frequency plane for a NDXWPT decomposition.

within the decompositions shown in Figs. 18 and 19 in order to demonstrate how a compromise between these extremes whilst showing how the best basis selection method clearly tailors the decomposition to the image. The tiling of the frequency plane of the Autumn image shown in Fig. 18 shows how the best basis method has selected an increased frequency resolution in the vertical direction. This is because of the predominance of the vertical frequency content evident within the Autumn image caused by the waves on the lake. The tiling of the frequency plane of the Lena image shown in Fig. 19 shows how the best basis method has selected an increased frequency resolution in different frequency elements but generally biased to the lowpass subbands. This is because the Lena image is typical of natural images in that the majority of the image content is concentrated in the low-frequency elements.

Setting the threshold T to 10.0 gave a clear demonstration how the best basis selection will tailor itself to the image as shown in Figs. 18 and 19. However it was found that adopting



(a) Original Image

(b) Frequency Coverage

Fig. 19. Tiling of the frequency plane for a NDXWPT decomposition.

a threshold value of $T = 30.0$ produced an effective best basis selection whilst having a manageable number of subbands for analysis.

Many different functional measures F may be used for the splitting condition such as entropy, energy or variance. These may provide the best basis on which to split each subband for different applications. However it was decided that a simple energy measure would provide the most general best basis selection as it would concentrate the analysis on the actual frequency content.

A. Texture Gradient TG Using the Best Basis NDXWPT Decomposition

The redefinition of the texture gradient using the NDXWT transform as given in Section IV-E is now used with the NDXWPT transform. All the equations can be reused using the subbands from the NDXWPT in the same way as the subbands are used with the NDXWT. The median filtering scale (i.e., the length of the separable median filter in (15)) is set similarly to Section IV-E according to the level of the decomposition. Fig. 16 shows the improved texture gradient (TG) using the NDXWPT transform for Fig. 12(a). It is clearly similar to the texture gradient of the artificial texture image shown in Fig. 12(a) using the NDXWT transform shown in Fig. 12(d). However, the use of the NDXWPT transform will ensure that the texture gradient will be tailored to the image content.

B. Final Gradient Image $G(x, y)$ Using the Best Basis NDXWPT Decomposition

A small number of alterations of the final image gradient definition (18) are required to be effective with the best basis NDXWPT decomposition. Firstly, the number of subbands may vary from decomposition to decomposition, therefore the number of subbands *(nos)* must be used to normalize their total effect. Secondly, the definition of $|\boldsymbol{MT}(x, y)|$ (used in (18)) is kept to calculate the texture content using only the equivalent subbands of a NDXWT transform. The inclusion of wavelet packet subbands was found to confuse the ability of this measure to reduce the intensity gradient image $(|\boldsymbol{\nabla f}|)$ in areas of high texture content. $G(x, y)$ was therefore redefined as

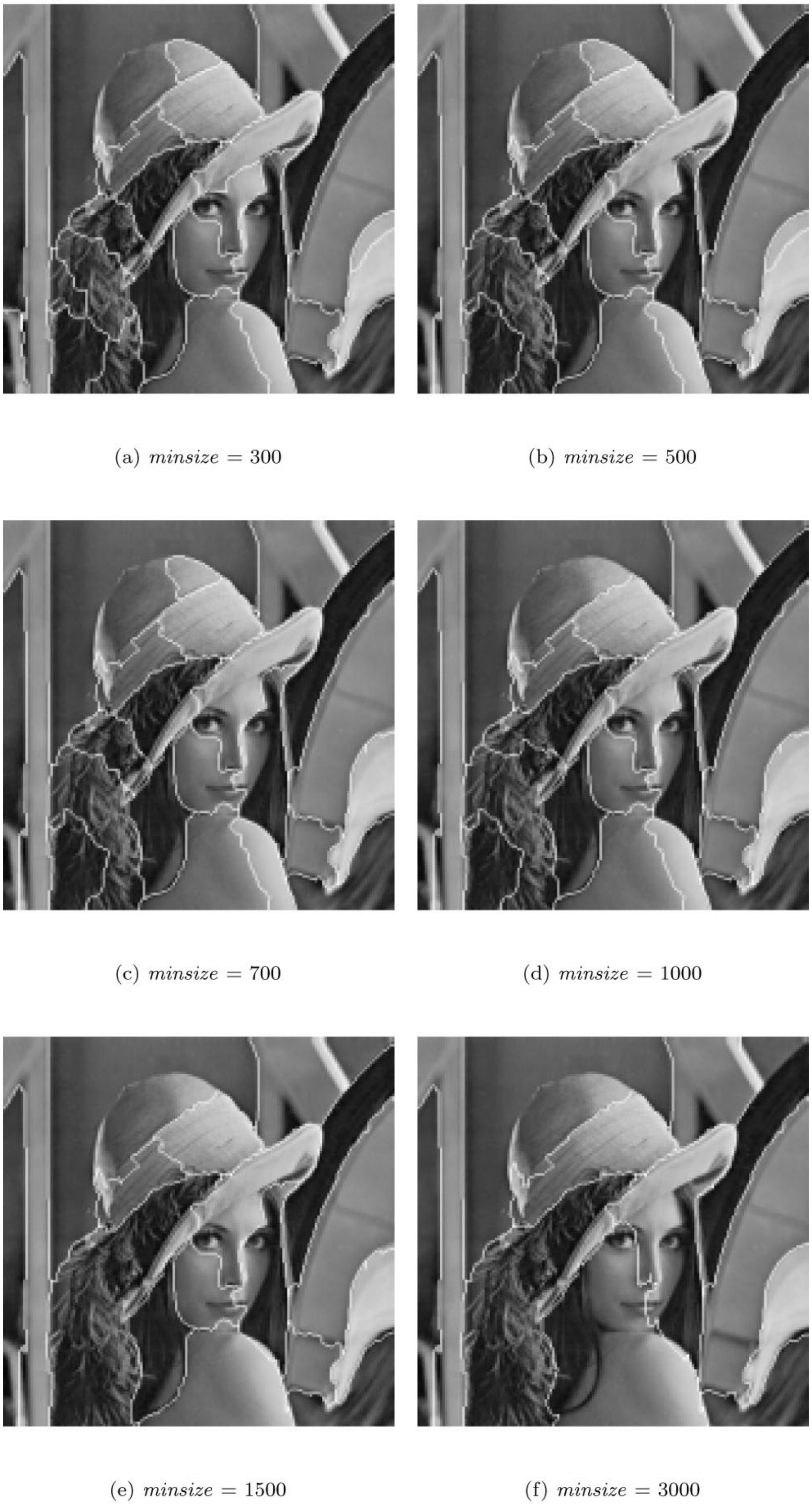
$$G(x,y) = \frac{mix \times (|\boldsymbol{\nabla f}|)^{1.2}}{(F(|\boldsymbol{MT_{xwt}}(x,y)|))^3} + \left(\frac{TG(x,y)}{nos}\right)^3 \quad (29)$$

where *nos* is the number of complex subbands produced by the NDXWPT decomposition and $\boldsymbol{MT_{xwt}}$ is defined to only include those subbands found within a NDXWT decomposition.

IX. Marker Selection

The problem of over-segmentation of the watershed method was dealt with through the flooding from selected sources (i.e., marker driven segmentation). The other methods were not chosen as they did not apply easily to texture gradients [11] or they tend to produce small residual regions (hierarchical watersheds [8]) and therefore were not suited to an application to region characterization.

Most of the marker selection methods suggested by Beucher [5] are application dependant. The aim of marker identification within a content based retrieval application is to pinpoint regions that are homogeneous in terms of texture, color and intensity and of a significant size. To meet these criteria a minimum region, moving threshold and region growing method was adopted as shown in Algorithm IX. 1. This algorithm calculates the mean and standard deviation of the gradient image (G). Then several thresholded binary images are produced at reasonably spaced thresholds using the mean and standard deviation of G. For each binary thresholded image, the number of closed and connected regions greater than the given minimum size is calculated. The threshold with the maximum number of connected regions is used as the output marker image. This is a similar method to that developed by Deng and Manjunath [21] although this not applied to marker selection.



(a) $minsize = 300$ (b) $minsize = 500$

(c) $minsize = 700$ (d) $minsize = 1000$

(e) $minsize = 1500$ (f) $minsize = 3000$

Fig. 20. Variation of the size parameter *minsize* for lena image.

Consider the gradient shown in Fig. 13. If we raise and lower the threshold line by the values given in Algorithm IX.1 and keep only the contiguous areas over size 300 pixels we obtain a marker image for the maximum number of contiguous regions shown in Fig. 17(a). The different greyscale values within this image showing the different labeled makers. This marker image is then used to produce the segmentation shown in Fig. 17(b).

The governing parameter of this method is therefore the scale factor, i.e., the minimum acceptable size of a marker areas (set to 300 pixels in this case).

```
Algorithm IX.1: MinsizeThreshold(minsz, $G$)
comment: minsz $=$ the minimum acceptable marker size
comment: $G =$ input gradient image
$std \leftarrow$ StandardDeviationOf $(G)$
$mean \leftarrow$ MeanOf $(G)$
$thresh s[12] = \{-0.9, -0.6, -0.55, -0.5, -0.4 -0.35, -0.3, -0.2, -0.1, -0.0, 0.1, 0.2\}$
for $i \leftarrow 1$ to 12
  do $thresholdLevel \leftarrow mean + threshs[i] \times std$
     $thresholdImage \leftarrow \mathrm{GTI}(thresholdLevel, G)$
     $markerImage[i] \leftarrow \mathrm{GCRLT}(minsz)$
     $regionNumber[i] \leftarrow \mathrm{NOR}(markerImage[i])$
$minIndex \leftarrow$ FindMinValue(regionNumbers)
return (markerImage(minIndex))
comment: $\mathrm{GTI}(\cdot) = \mathrm{GetThresholdImage}$
comment: $\mathrm{GCRLT}(\cdot) = \mathrm{GetConnectedRegionsLessThan}$
comment: $\mathrm{NOR}(\cdot) = \mathrm{Number\ of\ Regions}$
```

X. Effect of the Minimum Size Parameter Minsize

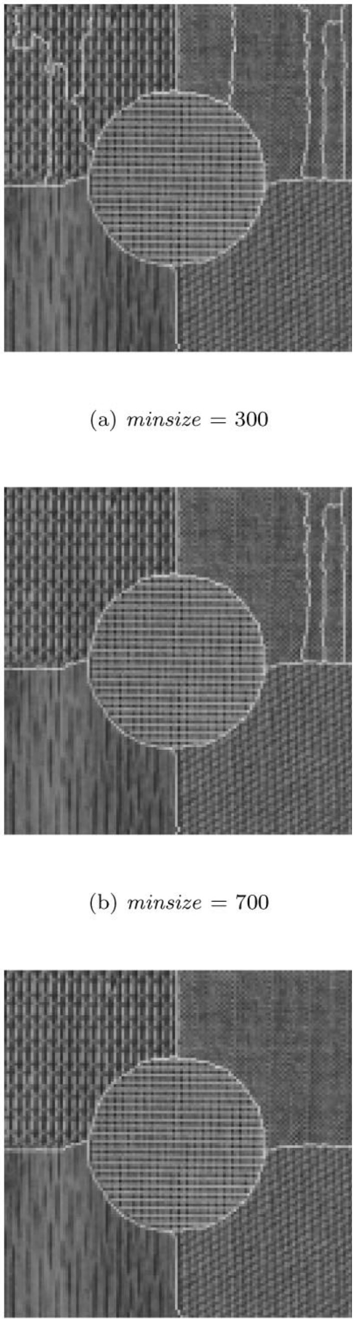
Figs. 20 and 21 show the segmentation of figures Fig. 12(a) and (c), respectively. The segmentation of the Lena image shown in Fig. 20 shows how the variation of the *minsize* parameter changes the size of the segmented regions. The segmentation of the artificial texture montage shown in Fig. 21 shows how the choice of the correct value of *minsize* will lead to effective segmentation (i.e., $minsize = 1500$). However, the segmentation of the image using small values of *minsize* show a segmentation of the herringbone pattern (top right) that reflects the change in orientation of the texture across the vertical strips.

The smaller the value of *minsize* may lead to oversegmentation. However the larger values may lead to separate regions being merged. This is a common problem for segmentation algorithms. However these images show how the presented algorithm is able to properly segment image regions according to intensity and texture changes.

XI. Conclusion

This work introduced the concept of texture gradient and has used it to produce an effective watershed segmentation technique for natural images based on intensity and texture boundaries. Additionally, a novel marker selection algorithm has been implemented to counteract the problem of over-segmentation whilst retaining key gradient boundaries whilst giving no small residual regions.

Using this marker selection scheme with a usual image gradient will often lead to effective segmentation for nontexture images. However, the inclusion of a texture gradient based on the actual frequency content of the image (using a complex wavelet packet transform) will ensure that differently textured regions will be segmented effectively.



(a) $minsize = 300$

(b) $minsize = 700$

(c) $minsize = 1500$

Fig. 21. Variations of size parameter *minsize* for image Fig. 12(a).

Traditional methods of marker extraction such as large scale low-pass filtering [5] or scale space morphological filtering [11] often move or remove salient, small scale gradient elements that can be vital for effective segmentation. Using markers extracted from the developed minimum region, moving threshold and region growing method, homogeneously textured regions can be identified. This marker extraction method uses the same gradient image as the subsequently implemented watershed transform. All small scale gradient features are therefore preserved often making a more effective segmentation.

Fig. 17(b)–(f) shows the segmentation results. This shows the method is able to give a good general segmentation of textured and natural images.

For an entirely automatic segmentation system, the current implementation gives good results compared to other comparable techniques [2], [18]. An automatic segmentation system is important for content based retrieval applications where human interaction is impossible or unfeasible (e.g., due to the number of image or video items). It is therefore the intention to include the developed automatic segmentation techniques within a subsequently developed content based retrieval application.

Although applied solely to greyscale images, the technique could be easily generalized to color images by averaging the resulting gradient images before using the marker extraction and watershed algorithms. Although not presently including any phase differential information in the texture gradient (as with the edge flow method), such information could be included in subsequent work as phase information is available within the complex wavelet transforms.

Acknowledgment

The authors would like to acknowledge the help of N. Kingsbury of the University of Cambridge for providing the Matlab code for the DT-CWT.

References

[1] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 269–285, Nov. 1986.

[2] W. Y. Ma and B. S. Manjunath, "Edge flow: a framework of boundary detection and image segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997, pp. 744–749.

[3] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Tools for content based manipulation of image databases," *Proc. SPIE*, vol. 2185, pp. 34–47, 1994.

[4] C. Lantuéjoul, "La Squelettisatoin et son Application aux Mesures Topologiques des Mosaiques Polycristalines," Ph.D. dissertation, School of Mines, Paris, France, 1978.

[5] S. Beucher and F. Meyer, "The morphological approach to segmentation: the watershed transformation," in *Mathematical Morphology and its Applications to Image Processing*, E.R. Dougherty, Ed. New York: Marcel Dekker, 1993, ch. 12, pp. 433–481.

[6] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 6, pp. 583–598, 1991.

[7] M. Petrou, L. Shafarenko, and J. Kittler, "Automatic watershed segmentation of randomly textured color images," *IEEE Trans. Image Processing*, vol. 6, pp. 1530–1544, Nov. 1997.

[8] S. Beucher, "Segmentation D'images et Morphologie Mathematique," Ph.D. dissertation, School of Mines, Paris, France, 1990.

[9] ——, "Segmentation tools in mathematical morphology," *Proc. SPIE*, vol. 1350, pp. 70–84, 1990.

[10] J. Chen, P. Saint-Marc, and G. Medioni, "Adaptive smoothing: A general tool for early vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 514–529, June 1991.

[11] P. Jackway, "Gradient watersheds in morphological scale-space," *IEEE Trans. Image Processing*, vol. 5, pp. 913–921, June 1996.

[12] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognit.*, vol. 23, pp. 1167–1186, 1991.

[13] W. E. Higgins, T. P. Weldon, and D. F. Dunn, "Gabor filter design for multiple texture segmentation," *Opt. Eng.*, vol. 35, no. 10, pp. 2852–2863, Oct. 1996.

[14] N. W. Campbell and B. T. Thomas, "Automatic selection fo Gabor filters for pixel classification," in *Proc. 6th Int. Conf. Image Processing and its Applications*, 1997, pp. 761–765.

[15] O. Rockinger, "Image sequence fusion using a shift invariant wavelet transform," *IEEE Trans. Image Processing*, vol. 3, pp. 288–291, 1997.

[16] J. F. A. Magarey and N. G. Kingsbury, "Motion estimation using a complex-valued wavelet transform," *IEEE Trans. Signal Processing*, vol. 64, Apr. 1998.

[17] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 674–693, July 1989.

[18] R. Porter, "Texture Classification and Segmentation," Ph.D. dissertation, Univ. Bristol, Bristol, U.K., 1997.

[19] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[20] J. C. Goshwami, *Fundamental of Wavelets: Theory, Algorithms, and Applications* New York, 1999.

[21] B. S. Manjunath and Y. Deng, "Color image segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, June 1999, pp. 446–451.

Paul R. Hill received the B.Sc. degree from the Open University, the M.Sc. degree from the University of Bristol, Bristol, U.K., and the Ph.D. also from the University of Bristol.

He is with Provision Communication Technologies working as a Senior Technical Engineer where he is currently developing optimized video codecs for wireless systems.

C. Nishan Canagarajah received the B.A. (Hons.) degree and the Ph.D. in DSP Techniques for Speech Enhancement, both from the University of Cambridge, Cambridge, U.K.

He is now a Reader in signal processing at the University of Bristol, Bristol, U.K. Prior to this, he was an RA and Lecturer at Bristol University investigating DSP aspects of mobile radio receivers. His research interests include image and video coding, multiview video compression and synthesis, information fusion, content-based retrieval, and the application of signal processing to audio and medical electronics. He is widely supported in these areas by industry, EU, and the EPSRC. He has published more than 200 papers and two books on mobile multimedia technology.

Dr. Canagarajah is a member of the EPSRC Peer Review College.

David R. Bull is Professor of signal processing and Head of the Electrical and Electronic Engineering Department at the University of Bristol, Bristol, U.K. He leads the signal processing activities within the Centre for Communications Research of which he is Deputy Director. He has worked widely in the fields of 1-D and 2-D signal processing and has published over 250 papers, articles, and books. His current research is focused on the problems of image and video communications for both low bit rate wireless, Internet, and broadcast applications. He is a member of the EPSRC Communications College and the Steering Group for the DTI/EPSRC LINK program in Broadcast Technology. He has served on the U.K. Foresight ITEC Panel and is a past Director of the VCE in digital broadcasting and multimedia technology. He is currently Chairman of a University spin-out company, ProVision Communication Technologies, Ltd., specializing in wireless multimedia communications. He is also currently involved in establishing a new DTI funded (£7.62 M) Centre for Communications Computing and Content that will be based at Bristol.