

Image Segmentation using Multiresolution Texture Gradient and Watershed Algorithm

Roshni V.S

Centre for Development of Advanced Computing,
Thiruvananthapuram, India

Raju G

Kannur University
Kannur, India

ABSTRACT

The wavelet transform as an important multi resolution analysis tool has already been commonly applied to texture analysis and classification. Mathematical morphology is very attractive for automatic image segmentation because it efficiently deals with geometrical descriptions such as size, area, shape, or connectivity that can be considered as segmentation-oriented features. This paper presents an image-segmentation system based on some well-known strategies implemented in a different methodology. The segmentation process is divided into three basic steps, namely: texture gradient extraction, marker extraction, and boundary decision. Texture information and its gradient are extracted using the decimated form of a complex wavelet packet transform. A novel marker location algorithm is subsequently used to locate significant homogeneous textured or non textured regions. The goal of boundary decision is to precisely locate the boundary of regions detected by the marker extraction. This decision is based on a region-growing algorithm which is a modified flooding based watershed algorithm.

General Terms

Image Processing, Pattern Recognition, Image Segmentation

Index Terms

Texture analysis, image segmentation, wavelet transforms, texture gradient, watershed transform

1. INTRODUCTION

Texture provides vital information for many image segmentation tasks. The last three decades has witnessed extensive research on texture segmentation [1-5]. In the 1980s, most traditional approaches included gray level co-occurrence matrices (GLCM) [6], second-order statistic method[7.], Gauss–Markov random field [8], and local linear transform[9], where operations were restricted to the analysis of spatial relations between neighborhood pixels in a small image region. Hence, their performance is the best for the class of so called micro textures. Study on human vision system indicates that the spatial/frequency representation, which preserves both global and local information, is adequate for quasi-periodic signals. This observation has motivated researchers to develop the multi resolution texture analysis models, such as the wavelet transform [10-14] and the Gabor transform [15-19]. The commonly adopted multi resolution analysis approach is to transform a texture image into a local spatial/frequency representation by convolving this image with a bank of filters with some tuned parameters. The wavelet transform and the Gabor transform are the most popular multi resolution methods. When compared to the wavelet transform, the Gabor transform needs to select the filter parameters according to different texture. There is a compromise between redundancy and completeness in the design of the Gabor filters because of nonorthogonality. The effect of the Gabor transform is also limited to its filtering area.

Consequently, we chose the wavelet transform to obtain the spectral information of the texture image.

It has been shown that the texture features which can effectively define directional and spatial/ frequency characteristics of the patterns lead to good texture analysis [20]. The Dual Tree Complex Wavelet Transform (DTXWT) [21] is an over complete wavelet that provides both good shift invariance and directional selectivity over the discrete wavelet transform(DWT) and is computationally faster than the Gabor transform. However, it is not suitable for textured images where the dominant frequency channels are located in the middle frequency channels. Therefore an appropriate way to perform wavelet transform for textured image is to use the concept of tree-structured wavelet transform or wavelet packets. To exploit the advantages of these, we have taken in this work the combined transform ie dual tree complex wavelet packet transform.

The watershed transform based segmentation approach works on morphological principles [22-25]. If we regard a grayscale image as a topographic relief, the gray value at a given location represents the elevation at that point. If this relief is to be flooded, the water will fill up lower elevation points first, and then the water level will increase. When water coming for two different regions meets, a watershed is created. The different regions that were flooded are called catchment basins. If this process is applied to a gradient image (in which each pixel corresponds to the modulus of the gradient at a particular point), the watersheds correspond exactly to the crest lines of the gradient, which are associated with the edges of the image. Therefore, the catchment basins are the segmented objects in the image. The main disadvantage is that for most natural images the watershed transformation produces excessive over-segmentation caused by spurious gradients. Marker based segmentation is one of the best solutions to reduce oversegmentation.

We have used a novel marker based solution by flooding basins from certain sources only rather than the minima. To achieve better segmentation performance we have adopted a modified flooding algorithm incorporating a technique based on sequential repeated scan which is described in detail in this paper.

This paper is organized as follows. In section 2 we briefly review the dual tree complex wavelet transform, wavelet packet transform and texture gradient extraction methodology. Section 3 focuses on the novel algorithm for creation of marker image and the modified watershed algorithm. Experimental results are shown in section 4. Section 5 concludes with future works.

2. TEXTURE GRADIENT

The watershed technique is a region-growing algorithm that analyzes an image as a topographic surface. It detects the regional minima of the gradients in the gray level image and grows these minima according to the gradient values. Traditionally the gradient image is derived from the change in gray level intensity under the perception that region boundaries are likely where the intensity gradient is large. To use texture boundaries for segmentation of textured images, texture gradient introduced by Hill et al [26-27] is used in our work. Compared to [26], the texture gradient computation method is implemented as follows:

The decimated dual tree complex wavelet packet transform (DDTXWPT) representation is used for simplicity, reduced redundancy and reduced computational complexity. Separable, orientation-adaptive weighted median filtering is used for post processing of texture features.

2.1. The Dual Tree Complex Wavelet Packet Transform: A Review

Complex wavelet transform introduced by Kingsbury has been found to solve the problems of DWT namely lack of shift variance and redundancy. But since the filters work with complex coefficients, achieving perfect reconstruction beyond the first level of decomposition was a major issue. For many applications it is necessary that the transform be perfectly invertible. Hence the enhanced version of this was introduced as dual tree complex wavelet transform (DTXWT) which was able to retain the attractive properties of complex wavelets as shift invariance, good directional selectivity, limited redundancy and efficient N- order computation. The DTXWT comprises of two parallel wavelet filter bank trees that contain carefully designed filters of different delays that minimize the aliasing effects due to down sampling. However it decomposes a signal into a set of frequency channels that have narrower bandwidths in the lower frequency region. The transform is suitable for signals consisting primarily of smooth components so that their information is concentrated in the low frequency regions. However, it may not be suitable for texture signals whose dominant frequency channels are located in the middle frequency region. Thus, an appropriate way to perform the wavelet transform for textures is to detect the significant frequency channels and then to decompose them further. The ideal method to achieve good frequency localization whilst retaining the structure of a discrete decomposition is the so called wavelet packet transform decomposition (DWPT). But the real wavelet packet transform is not shift invariant, which produces artifacts. Also, it is not rotation invariant because of its separability. Hence the integration of these or the dual tree complex wavelet packet transform has been used in our work [28].

The wavelet packets introduced by Coifman *et al.* [29] represents the generalization of the method of multi-resolution decomposition. In pyramid structured wavelet transform, the wavelet decomposition is recursively applied to the low frequency sub bands to generate the next level hierarchy. The key difference between the traditional pyramid algorithm and the wavelet packet algorithm is that the recursive decomposition is no longer applied to the low frequency sub-bands. Instead, it is applied to any of the frequency bands based on some criterion, leading to quad tree structure decomposition. The concept of wavelet packet bases has

been generalized to obtain multiresolution decomposition of an image. A given function, say Θ_0 , can be used to generate a library of wavelet packet basis functions $\{\Theta_q\}_{q \in N}$ as follows:

$$\begin{aligned}\Theta_{2q}(x) &= \sqrt{2} \sum_k h(k) \Theta_q(2x - k), \\ \Theta_{2q+1}(x) &= \sqrt{2} \sum_k g(k) \Theta_q(2x - k),\end{aligned}\tag{1}$$

where the function Θ_0 and Θ_1 can be identified with the scaling function ϕ and the mother wavelet ψ , respectively. Equation (6) uniquely defines a library of wavelet packet bases as a set of orthonormal basis functions of the form $\Theta_q(2^l x - k)$. Each element in the library is determined by a subset of indices l , k , and q , which corresponds to the scaling, dilation, and oscillation parameters, respectively. A set of 2-D wavelet packet basis functions can be obtained from the tensor product of two separable 1-D wavelet basis functions in the horizontal and vertical directions.

Full wavelet packet decomposition at every scale will produce a large number of coefficients. Therefore, only the dominant frequency channels based on Shannon's entropy criterion are used. The best-basis wavelet packet tree is computed as follows

(i) Decompose a given image into four sub-images by convolution and decimation with a pair of QMF's, as shown in figure 2. The given image can be viewed as parent node and sub-images as the children nodes of a tree.

(ii) Compute the Shannon's entropy (\mathcal{E}^i) of the parent and children of this tree using equation (8).

(iii) If the sum of the entropy of four children nodes is higher than the entropy of parent node, then decomposition for this parent node is aborted.

(iv) If the sum of entropy of children nodes is lower than entropy of parent node, then above decomposition is further applied to each of the children nodes.

2.2 Texture Gradient Derivation

The aim of our work is to compute the texture gradient from these subband images which can be used for classification and segmentation. The integration of the gradient images formed from the subbands is the simplest option. But a simple summing of the gradient images will not produce the desired texture gradient. The solution is to smoothen the texture subband magnitudes before the application of the gradient operator. In this case, the "noise" in question is any wavelet response with a small spatial extent—indicating a local edge rather than an extended area of texture. Median filtering is well known as a nonlinear edge-preserving smoothing or noise removal technique.

The median filtering should reflect the size of the edges in the complex subbands produced by simple intensity boundaries. For this, we used a separable direction oriented weighted median filter. As indicated earlier, the aim is to preserve the step and provide smoothing. One method that tends to preserve 2-D step discontinuities well is to filter a 2-D signal along the horizontal direction with a 1-D median filter and then filter the result along the vertical direction with another 1-D median filter or separable median filtering, and is often used in 2-D median filtering applications. The first median filter neighborhood extends in a line normal to the subband orientation. By choosing the

direction orthogonal to that of the artifact, the extent of the median filter may be minimized. This is desirable, as it avoids over smoothing and retains as much of the fine structure as possible, while at the same time minimizing computation. A second pass is then made with a neighborhood at right angles to the first (i.e., parallel to the subband orientation) to reduce noise. The replacement is computed using the weighted median filter supported with the directional information. The smoothing performance obtained by the separable oriented weighted median filter is indicated in Fig 1 (b).

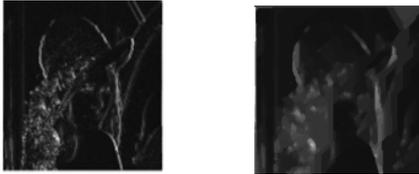


Figure 1 (a) DTXWPT subband (b) Directional Weighted Median filtered output

The filtered texture subbands are all up-sampled using simple linear interpolation, to give a set of images of identical dimensions. The filtered texture subband images are now suitable for gradient extraction. The gradient operator approximation used is the commonly used Gaussian derivative function. To obtain the combined gradient within the multidimensional feature space, we sum the gradients obtained for each of the individual subbands.

$$TG(x,y) = \sum_{i=1}^n \frac{TG_i(x,y)}{l_2(TG_i)} \quad (2)$$

where $TG(x,y)$ is the overall texture gradient, $TG_i(x,y)$ is that of each median filtered subband normalized by its l_2 norm energy.

In smooth regions, texture gradient may not give satisfactory results as intensity gradient dominates texture gradient. So intensity gradient modulated by a measure of texture activity also has to be taken into account. The aim is to suppress the intensity gradient in textured areas but leave it unmodified in smooth regions. When the texture gradient is then added, the combined result will be dominated by intensity gradient in smooth regions and texture gradient in textured regions, as required. The intensity gradient of the input image is obtained using the Gaussian derivative function and normalized by its l_2 norm energy. Scaling as required for increasing the dynamic range of the values to suit the texture gradient values is done before summing the intensity gradient and texture gradient. The results obtained are presented in section 4.

3. MARKER SELECTION AND WATERSHED ALGORITHM

Watershed algorithm is a mathematical morphology tool for image segmentation. The concept of watershed is based on visualizing an image in three dimensions, two spatial coordinates versus gray level. From, such a “topographic” interpretation we have to find the divide lines or watershed lines, which divide this topographic surface into catchment basins or watersheds. Ideally, there would be one (perceptual) region of the image for a single corresponding local minimum in the gradient function. However, factors like noise and other small structures in the image function can cause fluctuations in the gradient surface, resulting in the presence of many local minima. The solution often proposed is to use a marker-based watershed,

transposing the problem to that of marker selection. The impact of noise in the gradient image has been smoothed to a large extent by the methodology in section 2. For reducing the effect of over segmentation due to the presence of small structures, we have developed a novel marker selection algorithm wherein size plays a major role in deciding the regional minima.

The aim of marker identification is to identify regions that are homogeneous in terms of texture, color and intensity and of a significant size. To meet these criteria a minimum region size, moving threshold and region growing method was adopted as shown in Algorithm 3.1. This is similar to the strategy adopted in [26] but we have chosen the median for threshold computation instead of mean as median is not affected by outliers when compared to mean. This algorithm calculates the median and standard deviation of the gradient image (TG). Then several thresholded binary images are produced at reasonably spaced thresholds using the median and standard deviation of TG. 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.

Algorithm 3.1:

```

%% msize = minimum marker size
%% G = input gradient image
std = standard deviation of G
median = median of G
thresh[11] = {-0.5,-0.4,-0.3,-0.2,-0.1,0.0,0.1,0.2,0.3,0.4,0.5}
for j = 1:11
do { threshLvl = median + thresh[j] * std
threshImage = getThreshImage(threshLvl,G)
markerImage[j] = getConnectedRegionsLessThan(msize)
regionNumber[j] = noofRegions(markerImage[j])
indexLow = minValue(regionNumber)
return (markerImage(indexLow))

```

The key parameter in deciding the regions obtained through segmentation is the minimum acceptable marker area which varies across images. We present the observations in section 4.

To implement the watershed algorithm, we have adopted a different strategy of using appropriate arrays to store the labels and a repeated sequential scan based method for labeling as indicated in algorithm 3.2.

Algorithm 3.2:

```

%% Initialization:
Input image: The marker image, f
Output image: s
%% arrays v and s initialized to zero
Initialize arrays v and s to zero.
vmax, smax, fmax = Infinity.
Currentlabel = 0
min = 0, smin = 0
Scan the image from top left and bottom right.
Step1, step2, step3 are done sequentially till no change is made in v and s.

```

Step1

```

%% edges of non local minima plateaus
Scan the image from top left to bottom right and execute the following

```

```

if v(i,j) ≠ 1.
{
In each 3*3 window, check all the neighbors of f(i,j).
If any of the neighbors of f(i,j) is less than f(i,j),
v(i,j) is set to 1.
}

Step 2
%%%% the inside of non-local minima plateaus%%%%
Scan the image from top right to bottom left and execute the
following
if v(i,j) ≠ 1.
{min= vmax
In each 3*3 window, check all the neighbors of f(i,j).
{if any neighbor pixel of f(i,j) and f(i,j) are equal and
corresponding neighbor pixel of v(i,j) >0
{ {If any neighbor pixel of v(i,j) < min
min= the neighbor pixel of v(i,j)
}
}
If min ≠ vmax and v(i,j) ≠ min + 1
{v(i,j) = min + 1 }
}
}

Step3
%%%%: local minima plateaus %%%
Scan the image from top right to bottom left and execute the
following :
if v(i,j) = 0.
{ min = smax
If any neighbor pixel of f(i,j) and f(i,j) are equal and
neighbor pixel of s(i,j) >0
{ If neighbor pixel of s(i,j) < min
{ min = neighbor pixel of s(i,j)
[p,q] = pixel position of the above neighboring pixel of s(i,j)
}
}
If s(i,j) = 0
{If min ≠ smax
{ s(i,j) = s(p,q)
Currentlabel = Currentlabel + 1
else If
{min ≠ smax and s(p,q) < s(i,j)
s(i,j) = s(p,q)}
}
}
}
else
If v(i,j) = 1
%%%% edges of non- local minima plateaus %%%
{min = fmax
smin = smax
If any neighbor pixel of f(i,j) < f(i,j)
{(If any neighbor pixel of f(i,j) < min or any neighbor
pixel of f(i,j) = min) and (any neighbor pixel of s(i,j) > 0 and any
neighbor pixels of s(i,j) < smin )
{ min = neighbor pixel of f(i,j)
smin = neighbor pixel of s(i,j)
[p,q] = pixel position of neighbor pixel of s(i,j) }
}
}
}
else
%%%%comment: the inside of non- local minima
plateaus%%
{min = vmax
smin = smax
If any neighbor pixel of f(i,j) and f(i,j) are equal and any
neighbor pixel of v(i,j) >0
{ If any neighbor pixel of v(i,j) < min or any neighbor
pixel of v(i,j) = min) and (any neighbor pixel of s(i,j) > 0 and

```

```

any neighbor pixel of s(i,j) < smin )
{ min = neighbor pixel of v(i,j)
smin = neighbor pixel of s(i,j)
[p,q] = pixel position of the above neighbor pixel of
s(i,j)
}
}
}
}
If smin ≠ smax and s(p,q) ≠ s(i,j)
{ s(i,j) = s(p,q) }

```

In the above algorithm, two arrays v and s are used. s is an array to store labels, while v is an array to store distance from its lower pixels or plateaus. v is used instead of queue to divide non-minima plateaus evenly. The input is the marker image obtained from algorithm 3.1. Sequential scanning of pixels from top-left to bottom-right is done for each step until no values in s and v are modified in the current scanning. We briefly describe the steps involved in the segmentation process as given below:

Step 1: Finding the edges of non-local minima plateau.

Consider each pixel in f . For a 3×3 neighborhood, if any of its neighbors is less than the current pixel, the current pixel position in v is set to 1, if it is not equal to 1. If the pixel is on a plateau no lower neighbors are found.

Step 2: Finding the inside of non-local minima plateau

Consider each pixel in f ; For a 3×3 neighborhood, if any of its neighbors is equal to current pixel, current pixel position in v is 0, neighbor pixel position in v is greater than 0, min is made equal to the value of the neighbor pixel in v . Increment min by 1 and assign to current pixel in v .

Step 3: Finding local minima plateau

In this step, labels are given to pixels on local minima plateaus and those labels are delivered to its neighbors. A new label is assigned to each pixel whose $v[i,j]$ is zero if all its neighbor pixels with same grey level have no labels. This may cause -assignment of new labels to pixels on non-minima plateaus (if $v[i,j]$ is not calculated yet), and -assignment of different labels to pixels on same local minima plateau. However, those labels are overwritten by correct labels in the subsequent scans.

4. RESULTS

The Brodatz image database with 116 images that have significant amount of texture information were chosen for performance evaluations. We used 40 textures for our experiments which are shown in Fig. 2. Every original image is of size 512×512 pixels with 256 gray levels. Each texture image class was broken down into 100 overlapping images of dimension 128×128 sub-samples for segmentation purpose.

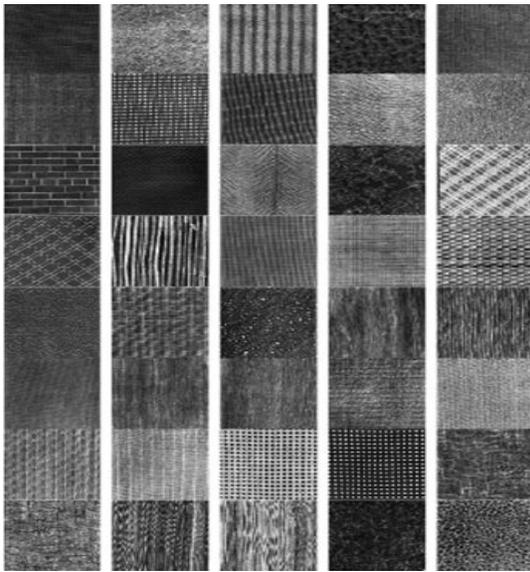


Figure 2

We present the results of our experiments for the various steps of texture characterization, post processing, gradient extraction, marker extraction and segmentation. Both qualitative and quantitative measures are adopted for comparison of the performance obtained for the various test images taken for our experiments.

Texture characterization: In the experiments, for most of the images, the feature of DTXWPT is extracted from the selected frequency regions in the third scale, three middle and high frequency regions in the second scale and three middle and high frequency regions in the first scale. For a few images the selection was limited to the third scale and first scale while for two images it was entirely from the third scale. Although extensive simulations were done using the above images, only four sample images (D1 – D4) (Fig 3) are chosen for demonstration.

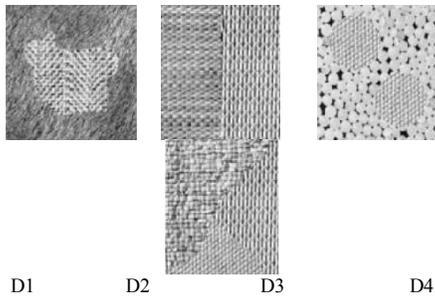


Figure 3 Sample images D1 – D4

The texture feature extraction results for these images are presented via table 1. The number of subbands for dual tree complex wavelet packet transforms is of the order of multiples of 8. Hence for decompositions beyond second level, the number of subbands expands to 512. But since selective decomposition is applied, the final chosen list for the above images has come down to the presented figures.

Table 1

Image	No. of level of decomposition	No. of selected subbands			
		Level 1	Level 2	Level 3	Total
D1	3	4	24	96	124
D2	2	12	16	-	28
D3	3	8	16	64	88
D4	2	6	40	-	46

We present the results of post processing the complex packet wavelet sub bands for texture gradient extraction quantitatively by the measurement of PSNR, the peak-signal-to-noise ratio.

The PSNR value in each case is calculated by

$$PSNR(db) = 10 \log_{10} \left(\frac{\sum_i \sum_j MAX^2}{\sum_i \sum_j \|V(i,j) - \omega(i,j)\|^2} \right) \quad (3)$$

where $v(i, j)$ and $\omega(i, j)$ are the *gradient vectors* at pixel (i, j) in the original(complex subband) and final texture gradient images obtained after applying the post processing techniques respectively and MAX denotes the maximum possible gradient magnitude in the image. It may be observed that the separable directional weighted median filter outperforms the ordinary median filtering technique.

Table 2 : PSNR values of the complex subband of image D1 and the texture gradient images

Original	MF	WMF	DWMF
12.12	8.14	11.02	13.38
13.46	10.12	12.46	14.12
14.24	11.34	13.02	14.98
15.78	12.42	14.14	16.68
16.44	12.18	15.12	18.02
17.98	14.34	17.98	20.12
19.22	16.76	19.24	22.88
22.86	18.24	21.14	24.34

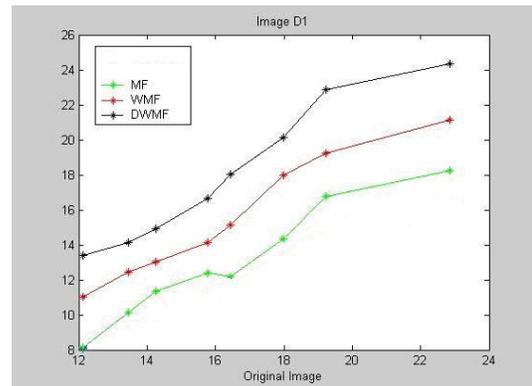


Figure 4 Graphic plot of table 2

Fig 5 gives a qualitative evaluation of the various texture gradient images obtained by combining the normalized cumulative gradient image from the subbands and the normalized intensity gradient image. The normalized

cumulative gradient image was scaled to 3 times the original value to increase the dynamic range of the gradient image. The normalized intensity gradient image was also scaled to 2 times the original value to emphasize the larger gradient values.

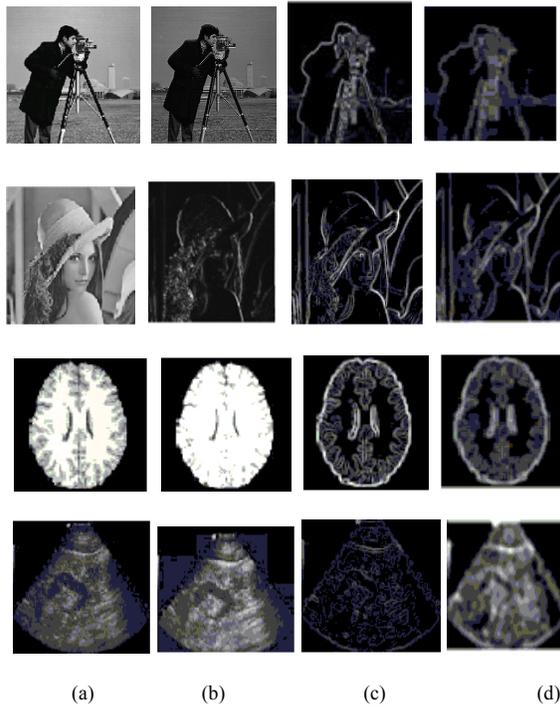


Figure 5 (a) original images of Cameraman, Lenna, MRI brain and ultrasound liver
 (b) DTXWPT subbands of the above
 (c) Texture gradient images of the above
 (d) Combined gradient images

The segmentation of the texture images shown in Fig. 6 shows how the variation of the *minsize* parameter changes the size of the segmented regions. By varying the threshold over the range of values mentioned in algorithm 3.1 and retaining only regions over a *minsize* of 1200 pixels we get a reasonable segmentation for the texture image in 6.1. Too small values of *minsize* may lead to oversegmentation and vice versa as illustrated by Fig 6. However the larger values may lead to separate regions being merged. This is a common problem for segmentation algorithms.

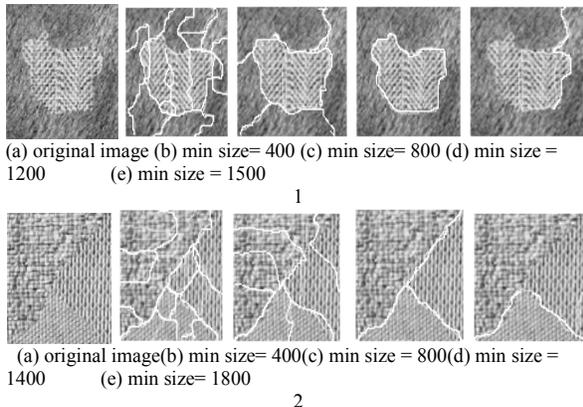


Figure 6.1 & 6.2 Original and segmented images

The first image we find the best segmentation obtained for the min size parameter value of 1200 while for the second case it

extends to 1400. Summarizing the results obtained for the test images chosen for our experiments, we find that the min size parameter varies from 1000 to 2000 for proper segmentation. In one case, as the no of regions in the image was around 9, perfect segmentation was obtained with a size of 800.

The database used for the quantitative evaluation of our algorithm includes 100 real images extracted from the Corel database for which manual segmentations provided by experts are available. To evaluate the segmentation algorithms, various metrics are available based on the availability of ground truth images. Segmentation evaluation metrics can be divided into boundary-based and region-based methods. For region-based evaluation, we investigate the widely used Global Consistency Error (GCE), as well as the Probabilistic Rand Index (PRI). The GCE measures the extent to which regions in one segmentation are subsets of regions in a second segmentation (i.e. the refinement). The PRI measures the consistency of labellings between a segmentation and its ground truth by the ratio of pairs of pixels having the same labels. The mean over all the human segmentations for an image takes their differences into account. As above, we plot the mean PRI (over all test images) against the number of regions to accentuate the relation between these two metrics.

Table 3

No: of regions	PRI	GCE
30	0.66	0.14
70	0.75	0.155
100	0.79	0.16
180	0.77	0.12
200	0.76	0.10
300	0.76	0.08
450	0.75	0.07

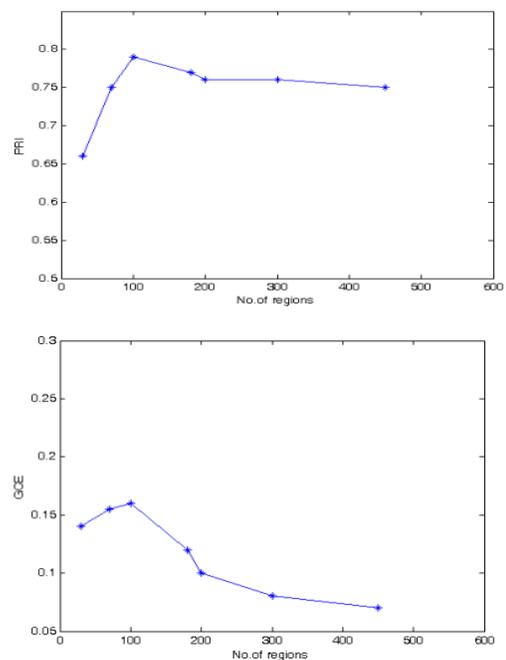


Figure 7 Plot of the PRI and GCE against the no of regions for the test images

A striking feature of the PRI and GCE graphs (Fig 7) is the difference in the general shapes of the curves as per the number of regions. In the PRI graph, it is found that the maximum PRI for segmentations into close to 100 regions, and drop for a small number of regions. The rapid decrease in the number of regions for low PRI is also visible. For the GCE plot, it is important to remember that one cannot examine trends in the GCE value as the number of regions varies, as this metric only makes sense when comparing segmentations into a similar number of regions. For this plot, the GCE decreases rapidly for regions beyond 100. The highest pixel classification rate was achieved for the minsize parameter of 1400 for most of the test images. The misclassified pixel rate evaluation measures the average overlapping pixels with highest overlapped region to its ground truth region without counting over-segmentation or under-segmentation. One important aspect in region segmentation evaluation is to examine how a region is segmented by a certain segmentation algorithm instead of measuring only misclassified pixels or correct regions. Oversegmentation rate indicates presence of regions falsely detected along the correct boundary or regions confused in homogeneous regions. The issue of over-segmentation has been handled by marker based segmentation to a large extent as evident in fig 7. The under-segmentation rate is found to be low generally for all the images.

5. CONCLUSION

This work focuses on the role of texture gradient in the segmentation of textured images. It has used the concept of region gradients to produce effective segmentation for natural and textured images. The first stage is the use of the combined complex wavelets and packet wavelets for feature extraction. The second stage proposes oriented weighted median filtering to correctly treat edge-responses in the texture features followed by computation of texture gradient. A novel marker selection algorithm and watershed segmentation algorithm has been proposed. It can be used for the segmentation of natural images based on texture and intensity boundaries. The 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 these algorithms with a usual image gradient will often lead to effective segmentation for non-texture images. However, the inclusion of a texture gradient based on the actual frequency content of the image will ensure that differently textured regions will be segmented effectively. The method combines the efficiency of morphological segmentation, moving threshold and region growing method. Qualitative and quantitative tests have confirmed the effectiveness of the method.

6. REFERENCES

- [1] R.W. Connors and C.A. Harlow. 1980, "A theoretical comparison of texture algorithms," IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 2 .pp. 204-222
- [2] T.R. Reed, J.M.H. Du Buf,1993, "A review of recent texture segmentation, feature extraction techniques," CVGIP Image Understanding ,vol. 7 ,pp. 359-372.
- [3] N.R. Pal, S.K. Pal 1993, "A review on image segmentation techniques, Pattern Recognition ,"vol. 26, pp. 1277-1294,.
- [4] J. Zhang and T. Tan 2002, "Brief review of invariant texture analysis methods," Pattern Recognit., vol. 35, pp. 735– 747,.
- [5] C.-C. Chen and C.-C. Chen 1999, "Filtering methods for texture discrimination," Pattern Recognit. Lett., vol. 20, pp. 783–790,.
- [6] P.C. Chen, T. Pavlidis ,1979, "Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm," Comput. Graphics Image Processing, vol. 10, pp. 172-182.
- [7] P. C. Chen and T. Pavlidis 1983., "Segmentation by texture using correlation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 5, no. 1, pp. 64–69, Jan.
- [8] R. L. Kashyap and R. Chellappa 1983., "Estimation and choice of neighbors in spatial-interaction models of images," IEEE Trans. Inf. Theory, vol. 29, no. 1, pp. 60–72, Jan.
- [9] M. Unser 1986, "Local linear transforms for texture measurements," Signal Process., vol. 11, pp. 61–79,.
- [10] M.Unser 1995., "Texture classification and segmentation using wavelet frames," IEEE Trans. Image Process., vol. 4, no. 11, pp. 1549–1560,
- [11] T. Chang and C.-C. J. Kuo 1992., "A wavelet transform approach to texture analysis," in Proc. IEEE ICASSP, vol. 4, no. 23–26, pp.661–664.
- [12] T. Chang and C.-C. J. Kuo 1992., "Tree-structured wavelet transform for textured image segmentation," Proc. SPIE, vol. 1770, pp. 394–405,
- [13] T. Chang and C.-C. J. Kuo 1993., "Texture analysis and classification with tree-structured wavelet transform," IEEE Trans. Image Process., vol. 2, no. 4, pp. 429–441,
- [14] W. Y. Ma and B. S. Manjunath 1995, "A comparison of wavelet transform features for texture image annotation," Proc. IEEE Int. Conf. Image Processing, vol. 2, no. 23–26, pp. 256–259,.
- [15] W. Y. Ma and B. S. Manjunath 1996., "Texture features and learning similarity," Proc. IEEE CVPR, no. 18–20, 425–430,
- [16] B. S. Manjunath and W. Y. Ma ,1996., "Texture feature for browsing and retrieval of image data," IEEE Trans.Pattern Anal. Mach. Intell., vol. 18, no. 8, pp. 837–842
- [17] A. K. Jain and F. Farrokhnia 1991, "Unsupervised texture segmentation using Gabor filters," Pattern Recognition vol. 24, no. 12, pp. 1167-1186,.
- [18] W. E. Higgins, T. P. Weldon, and D. F. Dunn 1996., "Gabor filter design for multiple texture segmentation," Opt. Eng., vol. 35, no. 10, pp. 2852–2863,
- [19] N.W. Campbell and B. T. Thomas 1997,, "Automatic selection of Gabor filters for pixel classification," in Proc. 6th Int. Conf. Image Processing and its Applications, pp. 761–765.
- [20] N. Kingsbury 1998., "The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters," in IEEE Digital Signal Processing Workshop, vol. 86,
- [21] N. Kingsbury 2000, "The dual-tree complex wavelet transform with improved orthogonality and symmetry properties," in Proceedings IEEE Conference on Image Processing, Vancouver,
- [22] "Segmentation tools in mathematical morphology," Proc. SPIE, vol. 1350, pp. 70–84, 1990.

- [23] P. Jackway 1996., “Gradient watersheds in morphological scale- space,” *IEEE Trans. Image Processing*, vol. 5, pp. 913–921,
- [24] A.Bieniek, A.Moga ,2000., “An efficient watershed algorithm based on connected components”, *Pattern Recognition*, vol. 33 ,pp. 907–916
- [25] H. Sun, J. Yang, M. Ren 2005, “A fast watershed algorithm based on chain code and its application in image segmentation”, *Pattern Recognition Letters* 26 ,pp. 1266–1274,.
- [26] Paul R. Hill, C. Nishan Canagarajah, David R. Bull 2003 ,” Image segmentation using a texture gradient based watershed transform”, *IEEE Trans on Image Processing*, vol 12, No. 12,pp. 1618 – 1633,
- [27] Robert J. O’Callaghan and David R. Bull 2005., “Combined Morphological - Spectral Unsupervised Image segmentation,” *IEEE Transactions on Image Processing*, vol. 14, no. 1, pp. 49-62,
- [28] Roshni VS, Dr Raju G 2010, “Segmentation of image using texture gradient, marker and scan based watershed algorithm” , *The 2nd International Conference on Digital Image Processing*, Singapore,
- [29] R.R. Coifman and M.V. Wickerhauser 1992, “Entropybasedalgorithms for best basis selection,” *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 713-718.