

Image Segmentation with Texture Gradient and Spectral Clustering

Indu V Nair.

Department of Electronics and Communication
Federal Institute of Science and Technology
Ernakulum, India

Kumari Roshni V.S.

Associate Director
Centre for Development of Advanced Computing
Trivandrum, India

ABSTRACT

For some applications the whole image cannot be processed directly because it is inefficient and impractical. Segmentation results in a set of images that cover the entire image. This work proposes a two stage segmentation method, which effectively process both the textured and non-textured regions. Dual Tree Complex Wavelet Transform, an extension of discrete wavelet transform, extracts texture feature from the image and orientation median filtering reduces the double edge effect at the texture edges. Watershed transform of Gaussian gradient of combined texture and non-texture feature give the first stage segmentation. The initial segmentation into super-pixels reduces computational burden and the second stage uses spectral clustering technique to cluster these primitive regions.

General Terms

Feature, segmentation, interpolation, clustering, algorithm.

Keywords

Dual Tree Complex Wavelet Transform, texture, watershed, super pixels, spectral clustering, GLCM.

1. INTRODUCTION

Image segmentation subdivides images into meaningful regions. This is often used as an initial stage for various image analysis techniques. Therefore image segmentation has many applications such as in computer vision, medical image analysis and diagnosis, remote sensing etc. There are mainly two kinds of images which catch the attention of researchers, namely textured and nontextured images. Texture analysis became most popular because in real life, objects and most of the images are textured in nature. Various statistical and structural approaches are available for texture analysis.

Watershed transform [11] is a powerful segmentation tool, which uses image gradient as input. Even though each regional minimum is small and insignificant, they form their own catchment basins leads to over segmentation of image. So a novel spectral clustering technique [18] is used as the second stage, which clusters the primitive region thus avoiding an excessive amount of segmentation. However, over segmentation is better than under segmentation since the former has more chances to match than later.

This paper is structured as follows. In Section II, texture watershed transform is briefly described. This section includes DTCWT for texture extraction, orientation median filtering for noise suppression, Gaussian gradient extraction and finally the watershed segmentation. GLCM, a statistical texture extraction method is given in Section III. Section IV explains

spectral clustering technique, used as second stage of segmentation. Section V gives an idea about experiments and results and relationship of this work to existing one is given in Section VI. Finally Section VII briefly explains the conclusion of this work and future scope.

2. TEXTURE WATERSHED ALGORITHM

Texture watershed segmentation is a method that uses texture gradient information of input image for segmentation. Texture watershed algorithm extracts both the texture gradient and intensity gradient separately and integrates them to apply watershed segmentation.

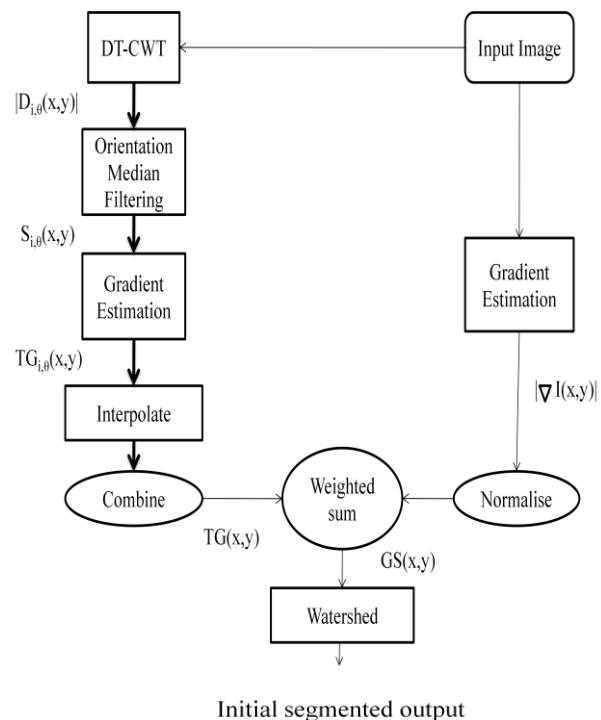


Fig 1: Block diagram of texture watershed algorithm

Fig. 1. shows the texture watershed algorithm. Thick connecting arrows denote multiple images such as wavelet subbands. DTCWT extracted texture features undergo post processing by orientation median filtering. Texture gradient obtained from post processed information is interpolated to the size of original image. This texture gradient is integrated with intensity gradient to obtain a total gradient. The watershed transform on this total gradient gives initial segmentation.

2.1 Extraction of texture feature

Texture analysis plays an important role in the analysis of different types of images. Gabor filters [3], pyramid-structured wavelet transform [5], and tree structured wavelet transform [4] etc are used as feature extraction tools. The lack of shift invariance and poor directional selectivity are two main disadvantages of real DWT [6] whereas DTCWT gives better directional selectivity and good shift invariance. The dual-tree CWT [7] consists of two parallel wavelet filter bank trees that contain carefully designed filters.

This work used complex filters of length-10 [8]. Table.1 contains coefficients of the filters in level 1 and table.2 contains those of the remaining levels.

Table 1. First level DTCWT coefficients

Tree a		Tree b	
H0a	H1a	H0b	H1b
0	0	0.01122679	0
-0.08838834	-0.01122679	0.01122679	0
0.08838834	0.01122679	-0.08838834	-0.08838834
0.69587998	0.08838834	0.08838834	-0.08838834
0.69587998	0.08838834	0.69587998	0.69587998
0.08838834	-0.69587998	0.69587998	-0.69587998
-0.08838834	0.69587998	0.08838834	0.08838834
0.01122679	-0.08838834	-0.08838834	0.08838834
0.01122679	-0.08838834	0	0.01122679
0	0	0	-0.01122679

Table 2. Remaining levels DTCWT coefficients

Tree a		Tree b	
H00a	H01a	H00b	H01b
0.03516384	0	0	-0.03516384
0	0	0	0
-0.08832942	-0.11430184	-0.11430184	0.08832942
0.23389032	0	0	0.23389032
0.76027237	0.58751830	0.58751830	-0.76027237
0.58751830	-0.76027237	0.76027237	0.58751830
0	0.23389032	0.23389032	0
-0.11430184	0.08832942	-0.08832942	-0.11430184
0	0	0	0
0	-0.03516384	0.03516384	0

The DTCWT decomposes an image into six oriented ($\pm 15^\circ, \pm 45^\circ, \pm 75^\circ$) sub-images at each level. The magnitude of complex coefficients at level i , orientation θ denoted by $|D_{i,\theta}(x,y)|$. Fig. 4(a) shows the -75 degree oriented subband of input test image given in Fig. 2 at first level decomposition.

2.2 Orientation median filtering

When the gradient is extracted directly from wavelet subbands double edges are formed in the gradient magnitude. If the watershed transform is directly applied on this gradient, it results in a false narrow region along the boundary (Fig.4(b)). So as a solution, a separable orientation median filtering is applied to the wavelet subband before gradient extraction.

Median filter is commonly used as an edge smoothing filter of an image. A 2D median filter uses a rectangular or a square $M \times M$ window. The intensity of each pixel in the image is replaced by the median of the intensities of the points in the $M \times M$ window. Separable median filter [9] or ‘median of median’ is obtained from the successive applications of one-dimensional median filter first along the rows and then along the columns. Block diagram is given in Fig. 3. Since the wavelet subbands have different orientations, orientation adaptation is applied to separable median filter.

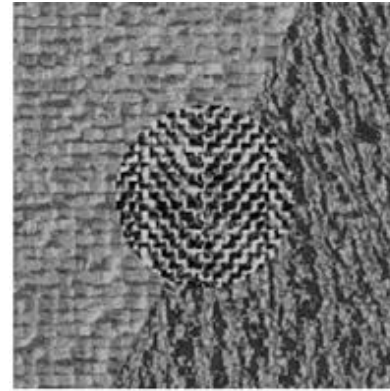


Fig 2: Test image1

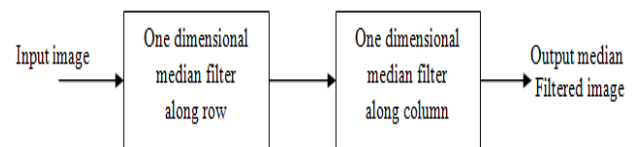


Fig 3: Block diagram of a separable median filter

$$S_{i,\theta}(x,y) = \text{MedFilt}\left(\theta + \frac{\pi}{2}\right)\left(\text{MedFilt}(\theta)(D_{i,\theta}(x,y))\right) \quad (1)$$

Where $D(x,y)$ is the wavelet subband coefficient at level i and orientation θ . $S(x,y)$ is the median filtered output, Fig. 4(c). First apply median filter along the line normal to the subband orientation and then along the line parallel to that orientation. Though the output of nonseparable filter is not identical with that of separable filter, performance of both in noise smoothing is very close. Separable filtering is more computationally efficient than the nonseparable case.



Fig 4 (a): Wavelet subband (-75°)



Fig 4 (b): Gradient of (a)



Fig 4 (c): Median filtered subband



Fig 4 (d): Gradient of (c)

2.3 Texture Gradient Computation

Image gradient is a directional change in the intensity or colour in an image, which is used to extract information from images. The gradient of a two-variable function at each image point is a 2D vector with the components given by the derivatives in the horizontal and vertical directions. Here texture image is smoothened by median filtering; this removes double edge in the gradient image. The gradient operator approximation used is the commonly used Gaussian derivative function [10].

The 2D Gaussian Kernel is given as,

$$G(x, y, \sigma) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2)$$

Where x and y are horizontal and vertical directions and σ is the standard deviation. 2D Gaussian derivative function along x and y directions are given in (3) and (4):

$$\frac{\partial G}{\partial x}(x, y, \sigma) = -\frac{x}{\sigma^2} G(x, y, \sigma) \quad (3)$$

$$\frac{\partial G}{\partial y}(x, y, \sigma) = -\frac{y}{\sigma^2} G(x, y, \sigma) \quad (4)$$

The gradient magnitude $TG_{i,\theta}(x, y)$ of each subband is, therefore given by,

$$TG_{i,\theta}(x, y) = \sqrt{(S_{i,\theta}(x, y) * G'_x)^2 + (S_{i,\theta}(x, y) * G'_y)^2} \quad (5)$$

Where $S_{i,\theta}(x, y)$ is the median filtered output. G'_x and G'_y are the Gaussian partial derivative filters in the x and y

directions and $*$ denotes convolution. The texture gradient of the median filtered image is shown in Fig. 4 (d).

2.4 Interpolation

A pixel at spatial position (x, y) has one feature for each texture gradient complex wavelet sub-band defined as $TG_i(x, y)$. Assuming a square image of dimension $N \times N$, each subband at level l of the wavelet decomposition has the dimension of $N/2^l$. However for all i , the dimension of $TG_i(x, y)$ has to be same as the image since there is the same number of features for all pixels. Therefore $TG_i(x, y)$ is assigned with the value of the spatially related subband coefficient magnitude,

$$T_i(x, y) = \left| TG_i \left(\left\lfloor \frac{x}{2^l} \right\rfloor, \left\lfloor \frac{y}{2^l} \right\rfloor \right) \right| \quad (6)$$

where l is the decomposition level of the transform and $\lfloor \cdot \rfloor$ is a function that returns the integer part of the fraction.

Many approaches are possible for the fusion of gradient information from different subbands, into a single texture gradient function, $TG(x, y)$. They can be simply combined by taking the weighted sum of magnitude of each subband.

$$TG(x, y) = \sum T_i(x, y) \quad (7)$$

2.5 Intensity gradient computation

Intensity gradient of the input image, $| \nabla I(x, y) |$, shown in Fig.5(a) is also computed by using Gaussian derivative operator as for the texture gradient.

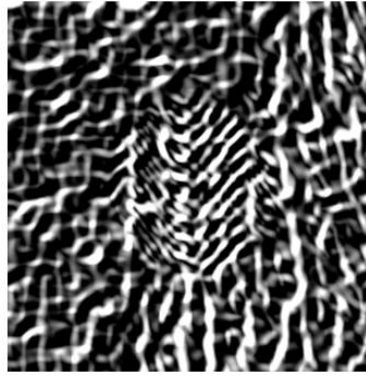


Fig 5 (a): Intensity gradient

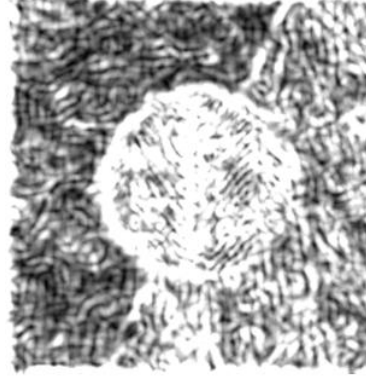


Fig 5 (b): Total gradient

$$\nabla I(x, y) = \sqrt{(I(x, y) * G'_x)^2 + (I(x, y) * G'_y)^2} \quad (8)$$

Where $I(x, y)$ is the input image. G'_x and G'_y are the Gaussian partial derivative filters in the x and y directions and $*$ denotes convolution.

2.6 Normalization and Gradient Combination

Normalization is performed [1] to reduce the noise in gradient images. Hence, this work first finds the median of both texture and intensity gradient. The intensity gradient is normalized by four times the median intensity gradient, w_I and the texture gradient is normalized by the median of the texture gradient, w_T . The four times in intensity gradient normalization is because there are sharp peaks, whereas the texture gradient is smoother. So the latter must be amplified in order to avoid suppression.

This work now combines the texture and intensity-gradient information to obtain a final gradient which is to be further segmented. The final single-valued gradient surface is computed as the combination of the texture gradient and intensity gradient.

$$GS(x, y) = \frac{\nabla I(x, y)}{w_I} + \frac{TG(x, y)}{w_T} \quad (9)$$

$GS(x, y)$ is the total gradient obtained after gradient combination, shown in Fig.5 (b) .

2.7 Watershed segmentation

The watershed transformation considers image gradient as a topographic surface. Pixels having highest gradient intensities correspond to watershed lines that form the region boundaries. Water placed on any pixel enclosed by a common watershed line flows downhill to a common local intensity minimum. Pixels enclosed by this common watershed line form a segment. Noise and other local irregularities in gradient image lead to over segmentation, as seen in Fig. 6 (a), which is the well known drawback of the watershed transform. The solution is to incorporate a preprocessing stage to limit the number of allowable regions in the image gradient. Watershed transform using immersion simulation is an efficient algorithm [12]. A concept of marker based watershed transform [13] can be used as a solution to this problem.

Here this work uses morphological H-minima transform [14] to modify the gradient surface by suppressing the gradient minima. The local gray level minima with dynamic lower than

a parameter h are removed. This is done by filling the valleys in the gray value relief, until the local minimum is increased by h . The H-minima transform and subsequent watershed transformation gives the modified segmented output as in Fig.6 (b). The output is not yet free from over segmentation. So a second stage clustering technique is used to cluster the over segmented regions.

3. FEATURE EXTRACTION USING GLCM

Since the over segmented result of watershed segmentation is irregular, it is impossible to directly extract the texture feature from these regions. Hence the extraction of the textural features [15] of original image is done by using gray level co-occurrence matrices (GLCM), which can construct features from different directions. It is a traditional statistical method for texture analysis and the matrix considers the relationship between the values of a reference and neighbor pixel. GLCM gives the value of how often combination of reference pixel and its neighbor occur in an image. Wavelet transform is also an effective tool for feature extraction.

This work used a 7×7 window with one pixel offset (a reference pixel and its immediate neighbor) to create a texture image.

Steps in creating a symmetrical normalized GLCM:

1. Create a framework matrix.
2. Decide offset value between the reference and neighbour pixel.
3. Count the occurrences and create the GLC matrix.
4. Find the transpose and add to the matrix to make it symmetrical.
5. Normalize the matrix by dividing each element with the total combinations of pixels, to turn into probabilities.

Most of the texture calculations are weighted averages of these probabilities. Texture calculation results in a single value, which represents the entire window. Then put that value in the place of centre pixel of the window and repeat this process by moving the window throughout the image. The unoccupied edge pixels are filled with nearest texture value. Since all the gray level combinations do not occur in the image, usage of all the 256×256 cells creates zero values at many places. So scale the gray scale value to lower level, which reduces the number of zeros and thus improves the

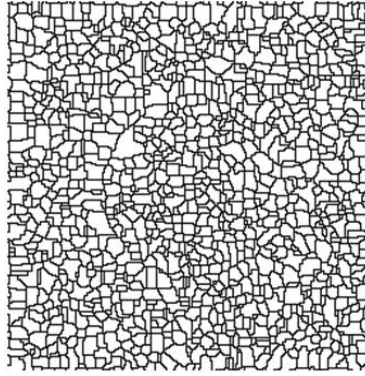


Fig 6 (a): Watershed transform

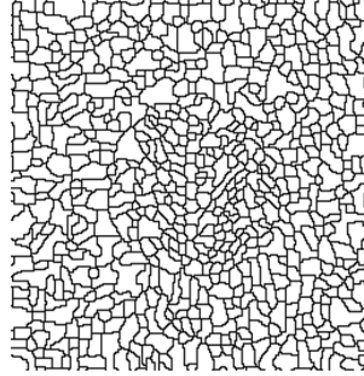


Fig 6 (b): Modified watershed

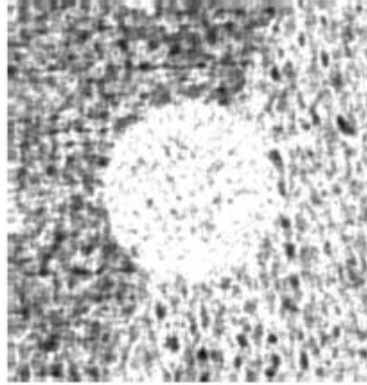


Fig 6 (c): GLCM variance



Fig 6 (d): Spectral clustered output

statistical value. The size of the GLCM is determined by the number of gray level used. GLCM gives various groups of texture measures [16], Fig.6(c) shows variance and Fig.7(c) shows homogeneity.

4. SPECTRAL CLUSTERING

Spectral clustering has emerged recently as a popular clustering method that uses eigenvectors of a matrix derived from the data. Several algorithms have been proposed using the eigenvectors in slightly different ways. The set of small regions produced by over segmentation is referred to as “superpixels”. The pre-segmentation into superpixels reduces the computational burden, because the number of super pixels n is much less than total number of pixels in the image. So it is possible to assign nodes to each region than to each pixel.

A set of data points x_1, x_2, \dots, x_n can be represented as a similarity graph, $G(V, A)$ whose nodes $V = \{v_1, v_2, \dots, v_n\}$ corresponds to the data points and edges defined through the $n \times n$ adjacency matrix A , which gives the similarity between each pairs of sample points v_i and v_j . Currently, spectral clustering algorithm can be divided into three types. Recursive spectral algorithms, such as SM algorithm [18], use the information in a single eigenvector [17]. These algorithms first divide data into two and then recursively generate more number of partitions. Multi-way spectral algorithms, such as NJW [20] algorithm, use multiple eigenvectors to directly get multiple partitions of data. Non-spectral algorithms are simple

clustering algorithm that clusters the data quickly. The paper [22], explains multi-way spectral clustering can perform better and is applied here.

This work used GLCM for texture feature extraction and then used the mean value of each region as its feature. Given a set of block features $\{x_1, x_2, \dots, x_n\}$ that is to be partitioned into k clusters. Here the value of kernel parameter σ is set as 0.3. The algorithm [21] is described in detail as follows.

Step1: Compute affinity matrix $A \in R^{n \times n}$. Affinity gives the similarity information between data points. For $i \neq j$,

$$A_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (10)$$

$A_{ij} = 0$ otherwise.

Step 2: Compute the diagonal matrix D . Its diagonal element (i, i) is the sum of each row in A .

$$D(i, i) = \sum_j A_{ij} \quad (11)$$

Then construct Laplacian matrix, L .

$$L = D - A \quad (12)$$

Step 3: Compute the eigen-values and eigenvectors of L .

Step 4: Choose the eigenvectors corresponding to the k biggest eigen-values and get the matrix U of size $n \times k$.

Step 6: Make each row of U as a point in R^k and then use k -means (KM) to cluster them into k clusters.

Step 7: When the i -th row of U is assigned to the j -th cluster, the original data point x_i is then assigned to the j -th cluster.

The output in Fig.6 (d), shows that the over segmented regions are clustered into k clusters which provide more meaningful segmentation.

5. EXPERIMENTS AND DISCUSSION

In this section two more images, shown in Fig. 7(a) and 8(a), of size 256×256 are selected to verify this proposed method and the final segmented results obtained are given in Fig. 7(d) and Fig.8(d). In texture extraction by using DTCWT, length 10 filters have been used. In [7] Type C filter shows better performance. The value of standard deviation σ in gradient detection, is set as 3. The watershed and modified watershed transform of test image1 gives 1496 and 708 regions respectively. These regions are meaningfully grouped into 3 clusters using spectral clustering as seen in Fig.6 (d). Table.3 shows the comparison among different methods for different input images.

Table 3. Comparison among different methods

Input Image	Watershed	Modified Watershed	Spectral Clustering
Test image1	1496	708	31
Image <i>Lena</i>	794	179	37
Test Image2	1496	666	42

6. RELATIONSHIP TO EXISTING WORK

In [23] the features of complex wavelet coefficients are modeled by using Hidden Markov Trees from the intensity component of image in colour space. The texture watershed segmentation is described in [24], used Gabor filters for texture extraction. Also a marker selection approach has been used for gradient modification. The papers [18], [19] explains various methods in spectral clustering, a weighted mean cut approach has been used in [1].



Fig 7 (a): Input image *Lena*



Fig 7 (b): Watershed

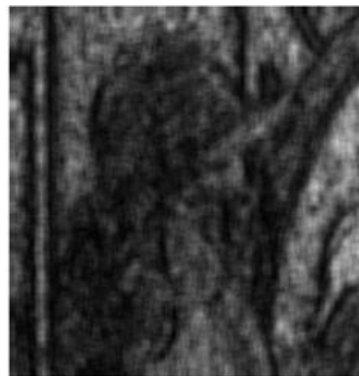


Fig 7 (c): GLCM Homogeneity



Fig 7 (d): Spectral clustered output

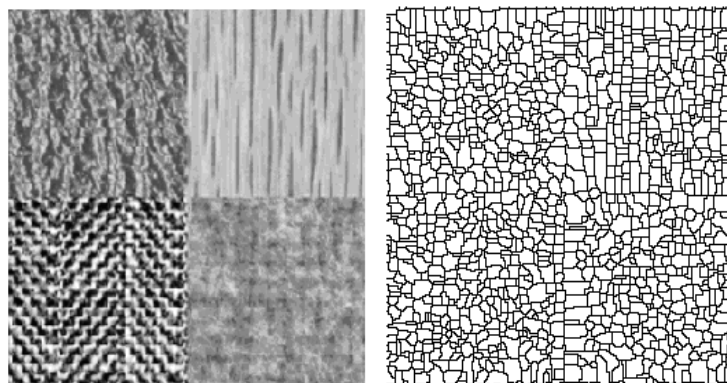


Fig 8 (a): Test image2

Fig 8 (b): Watershed

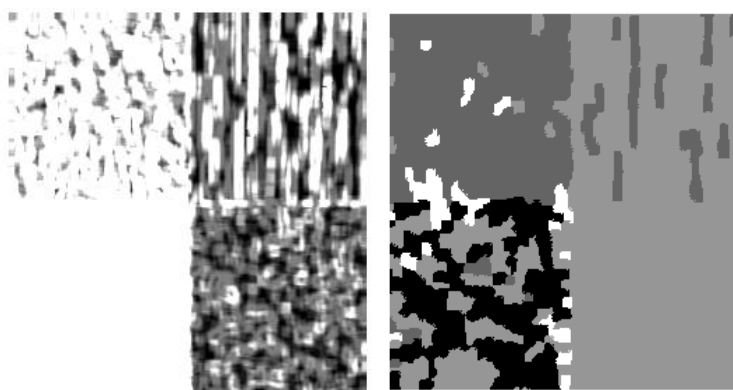


Fig 8 (c): GLCM variance

Fig 8 (d): Spectral clustered output

7. CONCLUSION

Texture image segmentation requires proper texture gradient extraction methods. DTCWT is an effective tool for texture extraction for gradient detection. Two stages were used in this work for image segmentation. Watershed is used to perform pre-segmentation and a recently developed spectral clustering to perform the final segmentation. Spectral clustering technique used in second stage, clusters the over segmented output of watershed transform. GLCM was used for feature extraction of input image for second stage.

Though this work mainly used the texture feature, other image features such as colour, intensity etc can be used for segmentation. Also normalized spectral clustering would give better performance than un-normalized.

8. REFERENCES

- [1] Robert J. O'Callaghan, David R. Bull, "Combined Morphological-Spectral Unsupervised Image Segmentation", *IEEE Transactions on Image Processing*, Vol.14, 2005, pp.49-62.
- [2] P. Hill, C. Canagarajah, and D. Bull, "Image Segmentation Using a Texture Gradient-Based Watershed Transform," *IEEE Trans. Image Process.* vol. 12, no. 12, pp. 1618–1633, Dec. 2003.
- [3] Anil K. Jain, Farshid Farrokhnia "Unsupervised Texture Segmentation Using Gabor Filters", 1990. *IEEE Trans. on Image Processing*.
- [4] Chang T. and Kuo, C.C.J., 1993, "Texture Analysis and Classification with Tree-Structured Wavelet Transform," *IEEE Trans. on Image Processing*.
- [5] Manesh Kokare, B.N. Chatterji and P.K. Biswas, "Wavelet Transform Based Texture Features For Content Based Image Retrieval" ,Electronics and Electrical Communication Engineering Department, Indian Institute of Technology, Kharagpur.
- [6] Zhi Jin Wang, "Fast Image (Information) Retrieval (IR) Using Wavelet Coding" Spring 2006, Department of Electrical and Computer Engineering, San Diego State University.
- [7] N. Kingsbury, "Complex Wavelets for Shift Invariant Analysis and Filtering of Signals," *J. Appl. Comput. Harmonic Anal.*, vol. 10, no. 3, pp. 234–253, May 2001.
- [8] Musoko Victor, Procházka Ale, "Complex Wavelet Transform in Signal and Image Analysis", Institute of Chemical Technology, Department of Computing and Control Engineering ,Technicka.
- [9] Patrenahalli M. Narendra, "A Separable Median Filter for Image Noise Smoothing" *IEEE Transactions on Pattern analysis and machine intelligence*. Vol.Pami-3, No.1, January 1981.
- [10] Angeles Hernandez-Carrascal, Slawomir J. Nasuto, "Application of Gaussian Multi-scale Representation to Feature Tracking in Meteorological Satellite Imagery" ,

- 2010 EUMETSAT Meteorological Satellite Conference, Cordoba, Spain, 20-24 September 2010.
- [11] Jos B.T.M. Roerdink and Arnold Meijster “The Watershed Transform: Definitions, Algorithms and Parallelization Strategies”, *Fundamenta Informaticae* 41 (2001) IOS Press.
 - [12] L. Vincent and P. Soille, “Watersheds in Digital Spaces: an Efficient Algorithm Based on Immersion Simulations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.
 - [13] S. Beucher and F. Meyer, “The morphological approach to segmentation: The watershed transformation”. In *Mathematical Morphology in Image Processing* (E. R. Dougherty, Ed.), pp. 433_481, Dekker, New York, 1993.
 - [14] J. Ohser, K. Schlodtz “Image Processing and Analysis”, Clarendon press. Oxford 2006.
 - [15] Haralick, R.M. 1979. “Statistical and Structural Approaches to Texture” *Proceedings of the IEEE*, Vol. 67 pp. 786-804.
 - [16] Clausi, D. A. 2002. “An Analysis of Co-occurrence Texture Statistics as a Function of Grey-Level Quantization” *Canadian Journal of Remote Sensing* vol. 28 no. 1 pp. 45-62 .
 - [17] Y. Weiss, “Segmentation Using Eigenvectors: A Unifying View,” in *Proc. International Conference on Computer Vision*, vol. 2, 1999, pp. 975–982.
 - [18] J. Shi and J. Malik, “Normalized Cuts and Image Segmentation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
 - [19] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, “A min-max Cut Algorithm for Graph Partitioning and Data Clustering,” in *Proc. Int. Conf. Data Mining*, 2001, pp. 107–114.
 - [20] Ulrike von Luxburg, “A Tutorial on Spectral Clustering”, Max Planck Institute for Biological Cybernetics, Tübingen, Germany *Statistics and Computing*, 17 (4), 2007.
 - [21] F. Tung et al. “Enabling Scalable Spectral Clustering for Image Segmentation” , *Pattern Recognition* 43 (2010) 4069–4076 .
 - [22] Marina Meila, Jianbo Shi, “A Random Walks View of Spectral Segmentation”, In *Proceedings of International Conference on AI and Statistics*, 2001.
 - [23] N. K. C.W. Shaffrey and I. Jermyn, “Unsupervised Image Segmentation via Markov Trees and Complex Wavelets,” in *Proc. Int. Conf. Image Processing*, vol. 3, Sep. 2002, pp. 801–804.
 - [24] P. Hill, C. Canagarajah, and D. Bull, “Texture Gradient Based Watershed Segmentation,” in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, 2002, pp. 3381–3384.