ISSN:0975-9646

Smita Patil et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7353-7358

Color Image Segmentation using Median Cut and Contourlet Transform : A Parallel Segmentation Approach

Mrs. Smita Patil[#1], Prof. Aparna Junnarkar[*2]

[#]Computer Engg. Department, Savitribai Phule Pune University
Pune, Maharashtra, India

[*]PES Modern College of Engineering
Pune, Maharashtra, India

***Abstract*— This paper presents a parallel implementation of color image segmentation algorithm using multiresolution technique. The idea is to achieve the complete and significant objects in the image using contourlet transform based image segmentation and to explore current multi-core architectures present in commercial processors in order to speed up the segmentation process for large size images. The algorithm attempts to use contourlet transform for texture feature extraction which can represent boundaries more accurately than that of discrete wavelet transform, so the contourlet transform is applied in proposed algorithm. This segmentation algorithm contains three stages. Initially the color quantization step is used to obtain a coarse image representation followed by parallel feature extraction using contourlet transform algorithm to exploit the multicore architecture. This task is executed sequentially by each assigned thread. Finally results returned by all threads are merged to achieve final segmentation output. Thus, color image segmentation can be obtained with reduced computational cost for large size images by proposed method. The algorithm is able to work with finite number of threads, so as to take full advantage of the upcoming processors having any number of cores.**

Keywords*— Image segmentation, Color quantization, Contourlet transform, Region merging, Multicore processor

I. INTRODUCTION

Image segmentation is an essential process for many applications in various domains such as remote sensing, security, medical imaging and many more. In image analysis process, segmentation is widely addressed issue as the first step before applying to images higher-level operations such as recognition, semantic interpretation and representation. Although it has been widely studied in many decades, in many cases, it still remains an open problem particularly in the unsupervised case since no prior information is given and the process is totally blind. Another problem with previous segmentation algorithm is the high computational cost for large size images [1].

The aim of this paper is to develop a parallel implementation for the image segmentation algorithm proposed by Baatz and Schape [2] and to use multiresolution transform to achieve good segmentation results with color texture based segmentation. The idea is to utilize the parallel processing capability present in most modern processors, specifically the multiple computing cores in one processor to speed up segmentation process without loss of segmentation quality. Therefore, the proposed solution does not require special hardware and can run on low cost machines that are commercially available.

In proposed method, color image segmentation is done using contourlet transform which represents boundaries more accurately. The proposed algorithm consists of three major steps. Initially color quantization is used for coarse image representation, followed by parallel texture information extraction by contourlet transform method which represents boundaries more accurately and then region combination is performed to obtain final segmentation output as shown in Figure 1.

Algorithm uses *Java JVM* threads in order to exploit the parallel processing capability of current processors with multiple cores. The experiments are carried out on the Berkeley benchmark dataset, for parallelism, JVM multicore programming is done and the evaluations are provided.

The rest of the paper is organized as follows. Section 2 describes the literature review. In Sect.3, gives description on the Median Cut algorithm for color image quantization,, details of texture extraction method based on contourlet transform and the parallel segmentation process. Experimental results are presented in Sect. 4. Conclusion is given in Sect. 5.

II. LITERATURE REVIEW

The aim of image segmentation is the domain-independent partition of the image into a set of regions which are visually distinct and uniform with respect to some property, such as grey level, texture or color.

A review of available color image segmentation techniques is provided in [3-4]. However, color segmentation approaches are based on monochrome segmentation approaches operating in different color spaces. The properties of several color representations, the segmentation methods and color spaces (RGB normalized RGB space HIS, CIE Luv, YIQ, YUV and their properties) are discussed in [5] and segmentation approaches are categorized into four classes namely, pixel based segmentation, area based segmentation, edge based segmentation and physics based segmentation. The complexities encountered in segmenting color images with complex texture are analyzed in [6]. Only two color spaces, RGB and HSI are discussed Texture is considered to be the major problem for all segmentation techniques, so much more focus is on texture analysis than on color representation and the problems of feature extraction in images with textural variations are discussed particularly.

An image texture can be regarded as local spatial variations in pixel intensities and orientation. It is widely recognized that a visual texture, which humans can easily perceive, is very difficult to describe [7]. There is no precise, general definition of texture exists in the computer vision literature. There are infinite numbers of possible textures and it is difficult to define nature of texture itself. However, measure of performance of algorithm can be determined on particular set of images or a specific application. However, texture characteristics are important in identifying objects and scenes in computer vision, and texture segmentation in the image-processing field, is also a difficult problem because natural textures are complex and diverse. There is wide variety of different approaches for texture representation, from the extraction of basic or complex features to the construction of a proper image model.

An example is the use of statistical features [7,8], for example Haralick proposed use of statistical features in the form of co-occurrence matrices [8]. The co-occurrence matrix is defined as the joint probability density of two pixels in different positions by choosing a distance and the orientation. A more complex feature extraction approach can make use of the geometrical features such as fractal dimension [9-12]. In these situations, the choice of fractal geometry is motivated by the observation that the fractal dimension is relatively insensitive to image scaling and shows a strong correlation with human judgment of surface roughness. But the problem is estimating the fractal dimension of real-world data is very sensitive to numerical or experimental noise, and particularly sensitive to limitations on the amount of data, therefore it is less effective for texture analysis. Another approach to texture representation includes texture models, such as Markov random fields (MRF) models [13-16] that are graphical models in which a set of random variables have a Markov property described by an undirected graph. Currently, texture representation via feature extraction related literature based on methods based on signal processing with Gabor and wavelet transforms. The Gabor filter is a linear filter used for edge detection. There is similarity between frequency and orientation representations of Gabor filters and that of the human visual system, and have been proved to be appropriate for texture representation and discrimination [17]. But, the major drawback of Gabor filtering is the excessive computational effort due to the large number of filters. Gabor filters and wavelet-based techniques on the other hand compute the textural characteristic by first transforming the image into the frequency domain and then dividing the domain into several frequency subbands. The distribution of energy in each of these subbands is used as the basis for distinguishing different textures. The difference between the two techniques lies on the way the frequency domain is divided, as well as on the types of the filter used. Wavelet-based approaches [18-20] gain more attention in texture segmentation because of their multiresolution property, which leads to multiresolution segmentation. As the extra information is made available through different resolutions, multiresolution segmentation is advantageous, as it performs the segmentation algorithm over a range of spatial scales of the input image. Wavelet-based methods can represent functions that have discontinuities and sharp peaks; also, they can deconstruct and reconstruct signals and images accurately; in addition, they have good spatial and frequency localization. So, they attracted a great deal of attention. A newly proposed method to representing texture features is the contourlet transform [23], which is a two-dimensional transform developed for digital images and is built with Laplacian pyramidal filter and directional filter banks. It is a multiresolution and multidirectional transform with several good properties. It is compact multiscale image representation with high degree of directionality and is relevant to texture analysis and feature extraction in natural images.

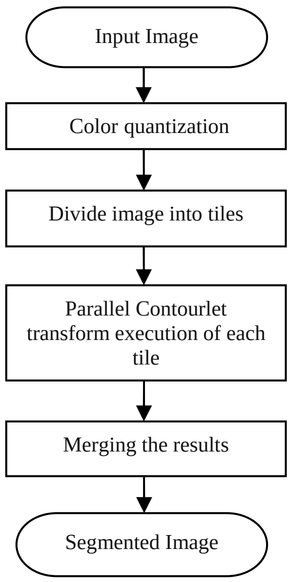


Fig. 1 Proposed Segmentation Algorithm

III. PROPOSED SEGMTENTATION METHOD

A. Coarse Image Representation

In this step Median Cut Algorithm is used for converting a true color image to an image using 256 colors. The idea behind median cut algorithms is to have each entry in the color map represent the same number of pixels in the original image. The algorithm divides the color space based on the distribution of the original colors. Initially color histogram of original image is obtained. Then, the algorithm proceeds as follows [21]:

1. Search the smallest box which contains all the colors in the image.
2. Sort the enclosed colors along the longest axis of the box.
3. Divide the box into 2 regions at median of the sorted list.

4. Steps 1, 2 and 3 are repeated until the original color space has been divided into 256 regions.

To divide the color space following method is used. The color palette for the output image contains the representative colors of the final.

Take the average of the colors in each box to form the representative colors, and then the appropriate color map index assigned to each color in that box. Finally, inverse mapping from the original colors to colors in the output palette is determined, by which each original color is replaced by a color in the output palette.

Because this algorithm uses image information in dividing the color, it consistently gives good results while having memory and time complexity no worse than popularity algorithms [22].

B. Contourlet transform based feature extraction

As texture is an effective classification for image segmentation and is often used in combination with color information to achieve better results, to better analyze texture, Contourlet transform based texture- feature analysis method is used, which is directional multiresolution transform. The primary goal of the contourlet approach [24], was to gain a sparse expansion for typical images that are piecewise smooth away from *smooth contours*. The 2-D wavelets basis functions as shown in Figure 3(a), miss directionality and are only good at capturing *point* discontinuities, but unable to capture the geometrical smoothness of the contours. Normally, a separable 2-D wavelet transform provides:

- Multiresolution, which is the ability to represent the transform with varying resolution from coarse to fine
- Localization, which is the ability of the basis elements to be localized in both the spacial and frequency domains
- Critical sampling, which is the ability for the basis elements to have small redundancy.

However, it is not capable of providing:

- Directionality, which is having basis elements defined in various directions
- Anisotropy, which is having basis elements defined in various aspect ratios and shapes.

The contourlet transform is related to capture curves instead of points, and provides directionality and anisotropy. The resulting transform has the multiscale and time-frequency-localization properties of wavelets, along with that it supports a high degree of directionality and anisotropy. Specifically, contourlet transform consists of basis functions that are oriented at any power of two's number of directions with flexible aspect ratios, with some examples shown in Figure 2(b). With a set of basis functions, contourlets can provide a smooth contour with fewer coefficients compared with wavelets, as illustrated in Figure 2(c). contourlets that match with *both* location and direction of image contours produce significant coefficients. Likewise, the contourlet transform effectively exploits the fact image edges are localized in both location and direction.

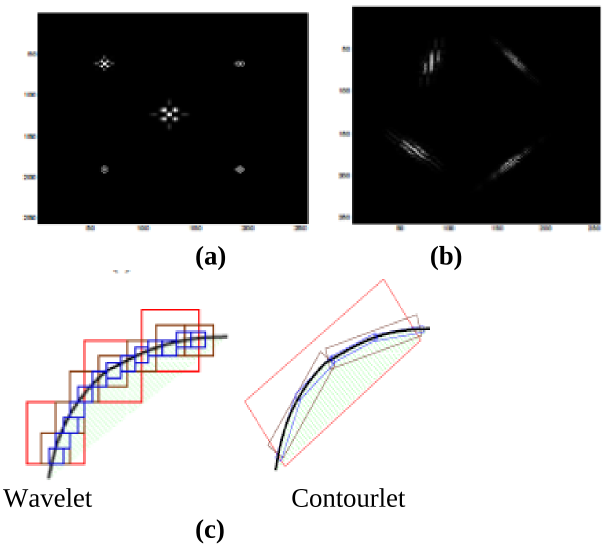


Fig.2. Contourlet and wavelet representations for images.
(a) Examples of five 2-D wavelet basis images.
(b)Examples of four contourlet basis images.
(c) Illustration showing how wavelets having square supports that can only capture point discontinuities, whereas contourlets having elongated supports that can capture linear segments of contours, and thus can effectively represent a smooth contour with fewer coefficients [25].

It is proposed by Minh Do and Martin Vetterli and provides sparse representation of images at both spatial and directional resolutions [24-25]. In addition, a multiresolution and directional decomposition by allowing different number of directions at each scale with flexible aspect ratio is supported. The double filter bank structure consists of the Laplacian pyramid catching the point discontinuities followed by a directional filter bank to link point discontinuities into linear structure. The contourlet transform supports the anisotropy scaling relation for curves by doubling the number of directions at every finer scale of the pyramid [26]. Since the contourlet transform has desirable properties for segmentation, it is used to extract boundaries of object of interest. In a 3 level decomposition, 8 subbands are generated. The large number of coefficients generated from this need not be involved in the classification step to reduce the computation time.

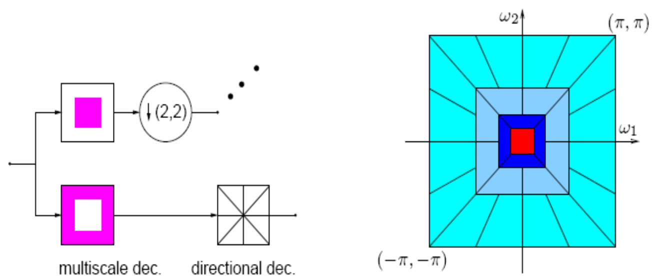


Figure 3: Pyramidal Directional Filter bank for contourlet transform (a)Block diagram, multiscale decomposition (b) resulting frequency division, number of directions increased with frequency [27].

C. Parallel Implementation

The parallel implementation of the segmentation algorithm uses the Java JVM for parallelization and divides computing task in different threads. The basic idea is splitting the image into regions that will denote tiles. Each tile is processed by different threads, which perform contourlet transform based coefficient calculation, using the sequential algorithm, with some synchronization actions. While doing the assigned task, one thread can perform its task more quickly than others. After performing assigned task each thread returns the result which is collectively passed to next region merging step. The reproducibility of the segmentation result, however, is an important goal. At the end of each step of segmentation (after all segments have been visited), the all segments will be merged sequentially. As the task of each thread will be independent, there is no need for critical sections in the code.

The division of image in tiles, and consequently the division of work in threads, can impact the final result of segmentation. To achieve better performance in a given architecture with multiples cores, the ideal is that the number of threads is always equal to the number of processor cores available. In this algorithm implementation, the number of threads is kept equal to number of available cores that will run on the processor. This guarantees the same tile division and the reproducibility of segmentation results for different architectures.

1) Task Distribution: The first step of the parallel algorithm consists of determining the number of tasks to be generated. Depending on available number of cores image can be divided into parts. For example for dual core system, image is divided into two parts as shown in figure 4. If only a thread exists, the computation is sequential. For more than one threads, the image can be divided into distinct areas. Each thread is responsible for processing the pixels included in its areas using selected core.

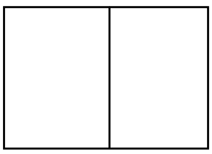


Figure 4. Dividing image into parts

2) Feature extraction: After the initial division of the image, each image segment will be given to available cores for applying contourlet transform based feature extraction. Each thread executes the contourlet transform algorithm inside its own tile. For a pixel x(i, j), the contourlet transform will be implemented on a 16×16 square centred at the pixel x by a thread. In contourlet transform, the 16 × 16 matrix is decomposed into 2 levels four coefficient matrices The image is first decomposed into two pyramidal levels, which are then decomposed into four and eight directional subbands. Then, we use “Log energy” to compute each coefficient matrix’s energy. For example, the 4 × 8 size matrix cD is arranged into a vector S with 32 elements from the first row to the last row.

Then,

$$E(cD) = E(S) = \sum_i \log({s_i}^2) \quad \text{(i= 1, 2,…,31)} \qquad (1)$$

Compute each region’s energy; then find neighbours and count the distance; finally update and sort labels in order to the next iteration. This procedure is repeated until the entire list of segments in each thread is covered. After all the threads finish their computation, coefficient computation of entire image is completed, and the algorithm starts a new step, with another merge stage. In merging process number of regions is passed as input parameter, and final segmentation output is generated. Number of clusters, generate different orders of visitation for the segments, which affects the final result of segmentation. This parallel approach, however, faces one major problem, the reproducibility of the final result.

IV. Experimental Evaluation

In this section, the results obtained with the sequential and parallel implementation of segmentation are presented. The algorithm is tested on number of images. The following sections describe the environment used in the experiments, the test images, and the segmentation results, along with the evaluation of the performance obtained with the parallelization.

A. Test Environment

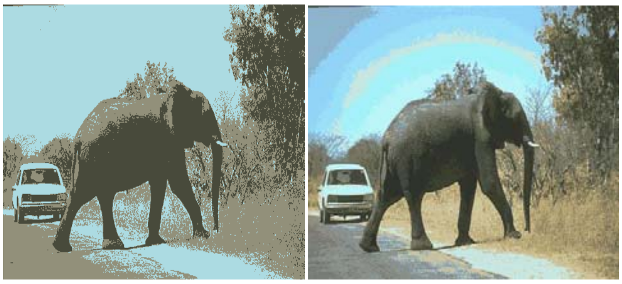
The experiments are performed on Pentium® Dual Core 2.30 GHz, 4 GB of RAM. The experiments are carried out on the Berkeley benchmark dataset [28], for which JVM multicore programming is done and the evaluations are provided.

The images with different sizes and features are used. Figure 5 and 6 shows original image, result of color quantization and result of sequential and parallel segmentation respectively. Testing is carried out on several color images with different sizes. All images are used to evaluate the performance gains and also used to compare the result generated from parallel segmentation to the result from sequential segmentation.

B. Results of Segmentation

Figure 5 (b) and (c) shows the result of the coarse image representation by K-Means color reduction algorithm and by Median cut algorithm respectively. It is noted that result of quantization and segmentation vary as we change number of clusters for K-Means. But for given input image they remain same in case of median cut. Thus, cluster value in K-Means e segmentation result. Results of sequential and parallel segmentation can be seen in figure 6. It is noted in the results the similarity between the results of parallel and sequential segmentation. It is important to notice that for the same number of threads, all the segmentation results are exactly the same no matter how many times the parallel algorithm runs.

(a)



(b) (c)

Figure 5 (a) Input Image, (b) shows result of quantization by K-Means with value 3 and (c) shows results of quantization by Median cut algorithm

(a) (b)

Figure 6 sequential and parallel segmentation (a) by using Median cut and countourlet transform and (b) by using K-Means with clusters value 3 and contourlet transform.

C. Performance Evaluation

The performance of the proposed algorithm has been evaluated for the accuracy of segmentation results and speed up in segmentation task. It is observed that segmentation results of K-Means and contourlet transform change as we change cluster value. By testing only, appropriate value of K is decided for input image, where as median cut color reduction. As shown figure 7, for tested input images accuracy of median cut and contourlet transform is better than K-means and contourlet transform for given cluster value. It is noted that there is increase in accuracy by 5 to 8 % in case of median cut and contoulet transform.

It is noticed that execution time of the segmentation is reduced with the increase in the number of cores.

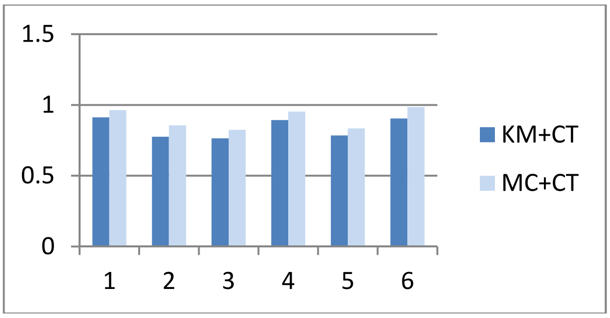


Figure 8 Graph of average accuracy of segmentation results

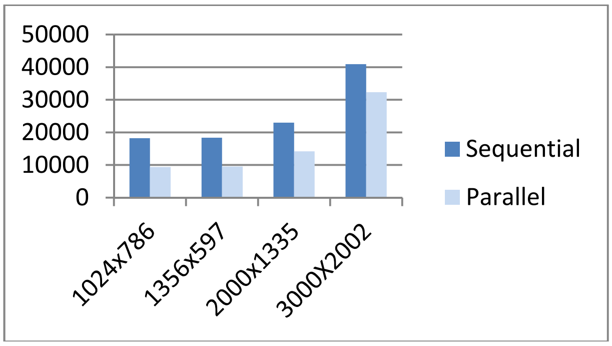


Figure 9 Graph showing excution time for sequential and parallel execution.

The results suggest an even greater reduction of segmentation time if more processors are used. Obviously, in the environment tested, does not compensate to run a number of threads greater than the number of processor cores. However, now days, many high-performance systems present two or more multicore processors sharing the same memory. The algorithm could benefit from this type of architecture, reducing even more the execution time.

The speedup obtained by the suggested parallel algorithm is measured as the ratio between the time of sequential execution and parallel execution time and shows the relative increase of the parallel performance.

It is observed that speedup of up to 1.5 is obtained. For very large images, reducing the segmentation time in 1.5 times is an important result, considering that the algorithm explores the full potential of the hardware present in most of the system. It can also be determined that the speedups obtained are not only affected by the image sizes, but also by the spectral and spatial attributes of image.

V. Conclusion

This work presents a parallel approach for color image segmentation based on multiresolution and multidirectional contourlet transformation and multicore architecture. The algorithm makes use of color texture information for image segmentation which uses color quantization for coarse image representation and contoulet energy function for segmentation. For coarse images representation, the algorithm uses Median cut algorithm replacing K-Means of earlier method. As per the experimental results proposed method gives results better than earlier method in terms of accuracy and there is no need to provide manual cluster value. It is observed that result for earlier method varying depending on number of clusters passed to K-means. This is not case for proposed method which does automatic.

Algorithm is tested on Berkeley segmentation database images and artificial images and results show that performance of proposed segmentation algorithm is 5 to 8% greater than K-means and contourlet transform method and peformance of parallel implementation is approx. 1.5 times faster than the sequential segmentation. This result is a very promising as it allows utilization of the processing power of current processors with multiple cores.

REFERENCES

[1] Wassenberg, J., Middelmann, W. and Sanders, P., "An Efficient Parallel Algorithm for Graph-Based Image Segmentation," CAIP '09: Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns, 2009.

[2] Baatz, M. and Schäpe, A. “Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation,” XII Angewandte Geographische Informations verarbeitung, Wichmann-Verlag, Heidelberg, 2000.

[3] H. Cheng, X. Jiang, Y. Sun and J. Wang, “Color image segmentation: Advances & prospects,” Pattern Recognition, Vol. 34, No. 12, pp. 2259-2281, Dec. 2001.

[4] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 2nd ed., Beijing: Publishing House of Electronics Industry, 2007.

[5] W. Skarbek and A. Koschan.,“Colour Image Segmentation - A Survey,” Technical report. Tech. Univ. Of Berlin, Oct. 1991.

[6] E. M. Riseman and M. A. Arbib, “Computational Techniques in the Visual Segmentation of Static Scenes,” Computer Vision, Graphics and Image Processing, 6, 221-276, 1997.

[7] Tuceryan, M., Jain, A.K., Texture Analysis, the Handbook of Pattern Recognition and Computer Vision. 2nd edn. World Scientific Publishing Co, Singapore (1998)

[8] Haralick, R.M., “Statistical and structural approaches to texture,” Proc. IEEE **67**(5), 786–804 (1979)

[9] Chaudhuri, B.B., Sarkar, N., “Texture segmentation using fractal dimension,” IEEE Trans. Pattern Anal. Mach. Intell. 17(1), 72–77,1995.

[10] J. Gangepain and C. Roques-Carmes, “Fractal approach to two dimensional and three dimensional surface roughness,” Wear, vol. 109, pp. 119–126, 1986.

[11] A. Pentland, “Fractal-Based Description of Natural Scenes,” IEEE Trans. Pattern Analysis and Machine Intelligence, 6, 6, 661-674, 1984.

[12] B. Mandelbrot, “The Fractal Geometry of Nature,” W.H. Freeman, 1983.

[13] G. Cross and A. Jain, “Markov Random Field Texture Models”, IEEE Trans. Pattern Analysis and Machine Intelligence, 5, 1, 25-39, 1983.

[14] B. Manjunath and R. Chellappa, “Unsupervised Texture Segmentation Using Markov Random Fields,” IEEE Trans. Pattern Analysis and Machine Intelligence, 13, 5, 478-482, 1991.

[15] A. Speis and G. Healey, “An analytical and experimental study of the performance of markov random fields applied to textured images using small samples," IEEE Transactions on Image Processing, vol. 5, no. 3, pp. 447 45, 1996.

[16] Matreka, A., Strzelecki, M.,”Texture analysis methods: a review”. Eur Cooperation Sci Technol COST B11, 90---924 (1998)

[17] A. K. Jain and F. Farrokhnia, “Unsupervised texture segmentation using Gabor filters," IEEE International Conference on Systems, Man and Cybernetics, pp. 14 19, 1990.

[18] Mallat, S, “A theory for multiresolution signal decomposition: the wavelet representation,” IEEE Pattern Anal. Machine Intell. 11(7), 674–693 (1989)

[19] Unser, M. “Texture classification and segmentation using wavelet frames,” In: IEEE Trans. Image Process. 4(11), 1549–1560 (1995)

[20] Hsin, H.C, “Texture segmentation using modulated wavelet transform,” IEEE Trans. Image Process. 9(7), 1299–1302 (2000)

[21] Heckbert, P. "Color Image Quantization for Frame Buffer Display", Computer Graphics, Vol 16, no 3, pp. 297-303, 1982

[22] Li Kang, “Advance image processing homework 1: Color quantization : A Median Cut Approach”

[23] Ning-Yu An, Chi-Man Pun, “Color image segmentation using adaptive color quantization and multiresolution texture characterization,” Springer May 2012.

[24] Do, M.N.,Vetterli,M, “The contourlet transform an efficient directional multiresolution image representation,” IEEE Trans. Image Process. 14(2), 2091–2106 , 2005.

[25] Duncan D.-Y. Po and Minh N. Do, “Directional Multiscale Modeling of Images using the Contourlet Transform,” IEEE trans. processing,

[26] M. N. Do, “Directional Multiresolution Image Representation,” Ph.D. Thesis, EPFL, Lausanne, Switzerland, Dec. 2001.

[27] Minh N. Do, “Contourlets and sparse image expansions,” Proc. SPIE 5207,Wavelets: Applications in signal and Image Processing X,560,2003.

[28] Martin, D., Fowlkes, C., Tal, D., Malik, J, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” Proceeding of 8th International Conference on Computer Vision, vol. 2, pp.416–423, 2001