Texture Segmentation Using Different Orientations of GLCM Features

Andrik Rampun
Department of Computer Science
Aberystwyth University,
SY23 3DB Wales, UK
yar@aber.co.uk

Harry Strange
Department of Computer Science
Aberystwyth University,
SY23 3DB Wales, UK
hgs08@aber.co.uk

Reyer Zwiggelaar
Department of Computer Science
Aberystwyth University,
SY23 3DB Wales, UK
rrz@aber.co.uk

ABSTRACT

This paper describes the development of a new texture based segmentation algorithm which uses a set of features extracted from Grey-Level Co-occurrence Matrices. The proposed method segments different textures based on noise reduced features which are effective texture descriptor. Each of the features is processed including normalisation and noise removal. Principal Component Analysis is used to reduce the dimensionality of the resulting feature space. Gaussian Mixture Modelling is used for the subsequent segmentation and false positive regions are removed using morphology. The evaluation includes a wide range of textures (more than 80 Brodatz textures) and in comparison (both qualitative and quantitative) with state of the art techniques very good segmentation results have been obtained.

Categories and Subject Descriptors

H.4 [**Texture Segmentation**]: Artificial Intelligence; D.2.8 [**Texture Analysis**]: Machine Learning

General Terms

Computer Vision

Keywords

Grey Level Co-occurrence Matrix, Gaussian Mixture Modeling , Data Normalisation/Smoothing, Texture Segmentation

1. INTRODUCTION

Robust texture based segmentation can be difficult, especially for complex textures. Nevertheless, it is an important technique in computer vision due to its capability in a range of different applications such as biomedical image analysis [9, 16], industrial inspection [28, 27] and satellite image analysis [7, 37]. Due to the possible complexity and diversity of textures, it is impossible to provide a universal definition of textures [22]. Despite this lack of a universally agreed definition, there is significant variation in intensity between nearby pixels; that is, at the limit of resolution, there is non-homogeneity. Secondly, texture is homogeneous at some spatial scale [20]. Taylor et al. [35] defined texture as a self-similarity pattern which is represented by similar signal statistics, whereas Cross and Jain [8] suggested that texture is a stochastic, possibly periodic and two-dimensional image field. Textures can be divided into three categories according to their properties [35], namely homogenous, weakly-homogeneous and inhomogeneous. Homogenous texture has visible repetitive structures and weakly-homogeneous texture involves local spatial variation in textural elements which leads to more or less repetitiveness. On the other hand, inhomogeneous texture has no repetitive structures and self similarity is absent. Figure 1 shows examples of textures taken from the Brodatz album [5].

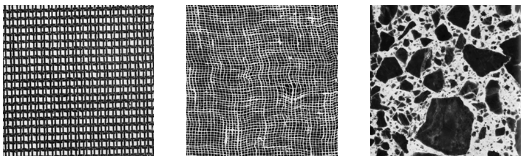

Figure 1: Examples of three different textures: from left to right homogenous, weakly-homogeneous and inhomogeneous, respectively.

The basic idea of texture segmentation is the grouping or clustering of pixels based on their properties such as contrast, homogeneity, entropy, energy, and/or maximum probability. Pixels that represent the same texture should be grouped together so that further analysis can be done on the various texture regions [10]. However, this is not always the case because there are many situations where two or more different textures have very similar properties leading to incorrect segmentation results. Another challenge of texture segmentation is to identify the boundaries between regions when the regions have similar textures. Texture segmentation can also be quite difficult when there is noise affecting the differentiation between textures. This research aims to develop a robust texture segmentation algorithm able to cope with these problems. The algorithm starts by extracting 16 different features based on Grey-Level Co-occurrence Matrices (GLCM) to capture important information from the textures. Each of the features is then processed including normalisation as well as noise removal. Principal Component Analysis (PCA) is employed to reduce the dimensionality. Then Gaussian Mixture Modelling is used in the

MIRAGE '13, June 06 - 07 2013, Berlin, Germany
Copyright 2013 ACM 978-1-4503-2023-8/13/06 ...$15.00.

segmentation and false positive regions are removed using morphology.

In this paper, we propose a texture based segmentation method by doing additional processing to each of the extracted features before clustering. Since individual features extracted from GLCM are less effective in discriminating different textures, we process each of the features before they are used. The proposed method can discriminate well between different textures by taking the mean value of two different orientations for each feature. The originality of this paper lies in the use of a simple and effective smoothing method before extracted features from GLCM can be used for clustering. This method is different from the existing methods, which use GLCM features in the sense that:

1. We introduced a new method to process each of the extracted features from GLCM. Most existing methods do not implement a futher processing on each of the extracted features and have poor results when dealing with noisy, complex textures and unclear boundaries.
2. We maximise the information captured by extracting each feature at two different orientations.
3. We only use features extracted from GLCM without combining them with other features from Local Binary Pattern (LBP), Gabor Filters, etc.
4. We do not use any similarity measures but only relying on the noise reduced features.

2. PROPOSED METHODOLOGY

The overview of the proposed methodology is illustrated in Figure 2. The various steps are described in subsequent subsections.

In order to make sure that the segmentation results are less influenced by intensity aspects we normalise the images using:

$$J(p,q) = \frac{I(p,q) - \mu}{\sigma} \quad (1)$$

where *I(p,q)* represents a pixel greylevel value at position *(p,q)*, μ and σ are the average image intensity and standard deviation, respectively.

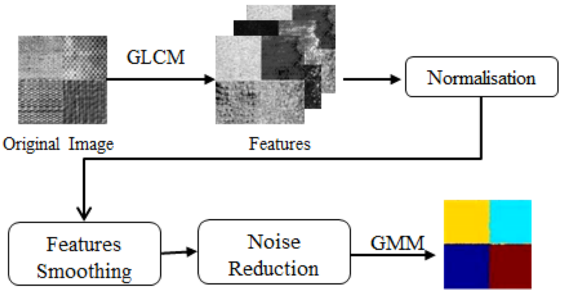

Figure 2: Overview methodology of the proposed algorithm. The methodology consists of six main steps: feature extraction, normalisation, feature smoothing, noise removal, clustering and segmentation.

2.1 Grey Level Co-occurrence Matrix (GLCM)

The algorithm uses GLCM due to its simplicity and large range of potential features. Haralick [17] suggested the use of GLCM for the definition of textural features. Moreover, Soh and Tsatsolis [32] proposed ten features that can be extracted from GLCM and three features were suggested by Clausi [7]. The GLCM is defined as the joint probability of occurrence of two grey level values at a given offset both in the terms of distance and orientation [3, 23]. This means, the values of co-occurrence matrix elements present relative frequencies with two neighbouring pixels separated by distance $\vec{d}$ and with grey level values i and j. Mathematically, a GLCM can be defined over an $m \times n$ image (I), parameterized by an offset (Δx,Δy), as:

$$C_{\Delta x, \Delta\, y}(i,j) = \sum_{p=1}^{m}\sum_{q=1}^{n} \begin{cases} 1, & \text{if } I(p,q) = i \text{ and} \\ & I = (p + \Delta x,\, q + \Delta y) = j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where i and j are the grey level values in the image, m and n are the dimension of the image, and (Δx,Δy) is the co-occurrence matrix's directional offset.

2.2 Feature Extraction

Feature extraction is an important task to capture information representing all textures in an image. This process can be done using several techniques such as Discrete Wavelet Transformation (DWT) [14], Gabor Filters [19, 15, 4, 38], GLCM [17, 32, 7], and Local Binary Patterns (LBP) [24]. Numerous investigations have been carried out regarding the set of features to be considered for use in the algorithm. We selected the following features: a) eight Haralick's statistical features [17] were selected, namely Contrast, Correlation, Sum Average, Sum Variance, Sum Entropy, Difference Variance, Difference Entropy and Energy, b) the features suggested by Soh and Tsatsoulis [32] were thoroughly investigated and six of them were selected: Cluster Prominence, Cluster Shade, Dissimilarity, Entropy, Homogeneity and Maximum Probability, and c) the last two features chosen are Inverse difference and Inverse difference moment as suggested by Clausi [7]. However, to maximise the information captured from the textures, each feature is considered at two different orientations. In the proposed method, we used 0° and 45° with a distance (Δx,Δy) = 1. We did not add 90° and 135° due to the increased computational load. In addition, although the segmentation results are slightly better using four different orientations, the final segmentation results are similar after morphology is performed. Furthermore, the distance (Δx,Δy) = 1 is choosen to avoid loss of information as reported in [11]. On the other hand, we used a small window size of 5×5 throughout the process (other window sizes such as 3×3 and 7×7 are also possible). In total, 32 features are extracted from the original image.

2.3 Feature Normalisation and Smoothing

After the extraction of the features, each of them was normalised between 0 and 1. This normalisation of each of the features resulted in consistency in the dynamic range for the data [34], for which we used:

$$Normalised(e(p,q)) = \frac{e(p,q) - E_{min}}{E_{max} - E_{min}} \tag{3}$$

where $e(p,q)$ is a feature value and E_{min} and E_{max} are the minimum and maximum values of $e(p,q)$, respectively.

Once all features were normalised, we smoothed each of the normalised features. To achieve this, we did run a 5×5 window over the combination of two orientations for the normalised feature images. Other window sizes were investigated, but 5×5 tended to give the best segmentation results. Our smoothing method can be expressed as:

$$f_r(p,q) = \frac{\sqrt{2f_1^2} + \sqrt{2f_2^2}}{2} \tag{4}$$

where f_r is the resulting feature, f_1^2 is the feature at the first orientation and f_2^2 is the feature at the second orientation. f_1^2 and f_2^2 are the matrix product of each feature ROI by itself which can be described as

$$f_\ell^2(n,m) = \sum_k f_\ell(n,k) f_\ell(k,m) \tag{5}$$

where $\ell \in \{1,2\}$. Figure 3 shows an example of f_r before and after using equation (4).

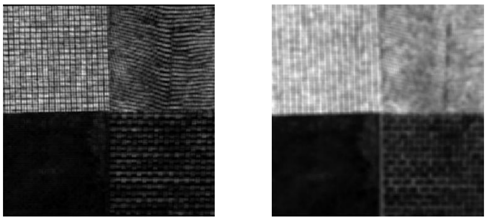

Figure 3: Original feature extracted from GLCM (left) and processed feature using equation (4) (right).

2.4 Noise Reduction

According to [2, 25, 26], noise reduction is a process to minimise noise in an image and attempts to recover an underlying perfect image from a degraded copy. In addition, Garcia [13] states that noise reduction is used to reduce experimental noise or small-scale information while keeping the most important aspects of a dataset. The proposed algorithm used two-stage noise reduction based on the discrete cosine transform (DCT) followed by replacement of each pixel by the average of the neighbouring pixel values. Both processes used a 5×5 window. DCT encoding is illustrated in Figure 4. Each feature is treated as two dimensional data. The two-dimensional DCT is defined as:

$$Y(u,v) = \sqrt{\frac{2}{n}}\sqrt{\frac{2}{m}} \sum_{p=1}^{n}\sum_{q=1}^{m} \Lambda(p)\Lambda(q)$$

$$\cos\left(\left[\tfrac{\pi.u}{2.n}(2p-1)\right]\left[\tfrac{\pi.v}{2.m}(2q-1)\right]\right) f_r(p,q) \tag{6}$$

where

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} \text{ for } \xi = 1, q = n, m \\ 1 \text{ otherwise} \end{cases} \tag{7}$$

n and m are the dimensions of the ROI and $f_r(p,q)$ represents the feature at position (p,q).

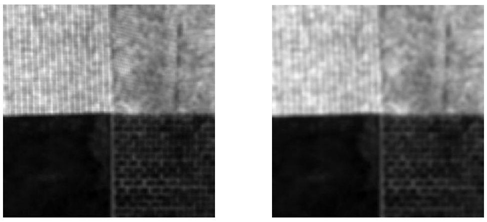

Figure 4: An example of processed feature, f_r encoded by DCT. Note that f_r is smoother after DCT encoding. The difference can be seen on the top left and bottom right regions (before and after DCT is performed).

Garcia [13] introduced a fast robust version of a discretized smoothing spline which allows robust smoothing of equally spaced data in one and higher dimensions. In fact, the method [13] has the ability to deal with weighted, missing and outlien values by using an iterative procedure. Therefore the method [13] is employed in the proposed algorithm for feature smoothing. After each of the feature is smoothed, the average of the neighbouring pixel values replaces each pixel. A small window (3×3, 5×5 and 7×7) is suggested in [36] to avoid the boundary between textures to be smeared due to the averaging. In our proposed method we used a window size of 5×5. The final noise reduction step is to perform Principal Component Analysis (PCA) which allows us to analyse the correlation matrix and all possible Principal Components (PCs), thus reduce dimensionality and so remove redundant feature aspects [18]. The PCA process is summarised as: (a) Organize data as an $a \times b$ matrix, where a is the number of measurement types and b is the number of samples (b) Subtract off the mean for each measurement type (c) Calculate the Singular Value Decomposition (SVD) or the eigenvectors of the covariance [31].

2.5 Image Segmentation

Image segmentation is performed using GMM to cluster the data in a low dimensional data space. A three-dimensional data space is chosen due to high percentage accuracy of clustering ($> 80\%$) reported in [12]. A GMM is a parametric probability density function represented as a weighted sum of Gaussian components [29]. It clusters a set of data by selecting the component that satisfies the Maximum a-Posterior (MAP) parameter estimation and uses an iterative algorithm that converges to a local optimum. GMM is choosen as recomended by [29] for classes that have different sizes and correlation within them as well as its ability to form smooth approximations to arbitrarily shaped densities. Figure 5 shows an example of data distribution in three dimensional data space.

The final step in the segmentation process is the removal of misclassified regions. For each texture class we remove

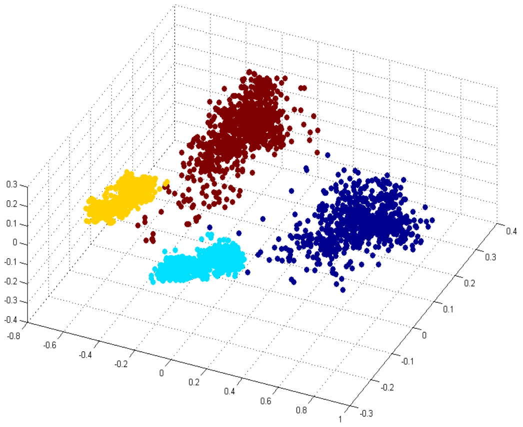

Figure 5: Example of data distribution in three dimensional data space. This is for the example texture image shown in Figures 3 and 4 which contains four different textures.

any islands within the segmented region by performing hole filling on a per-class basis. This is performed using an iterative approach where, for each texture class in turn, the pixels corresponding to that texture class are set as foreground and the remaining pixels as background. Hole filling is performed on each of these images using morphological reconstruction [33]. Figure 6 shows an example of segmented image before and after morphology.

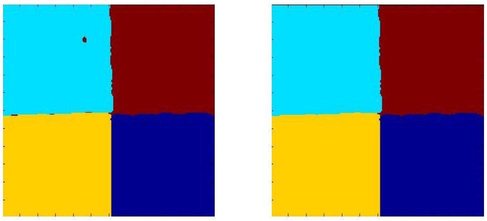

Figure 6: Small islands removal.

3. EXPRIMENTAL RESULTS

This section presents both qualitatively and quantitatively results. In total more than 80 different images with size 256 × 256 pixels were tested from the Brodatz album [5]. Each image contains 2 to 5 different textures. The proposed method was evaluated on various different images and divided into several categories:

1. Same texture at different orientations.
2. Same texture at different scales.
3. Different textures with similar intensity values.
4. Similar textures.
5. Different region shapes.

We use percentage (%) to illustrate the ratio of pixels which are classified correctly and incorrectly. The classification is based on the ground truth as defined by the datasets. Correct classified pixels are those which fall in the same region as defined in the ground truth. In other words, they are the pixels which are successfully categorised by the algorithm in comparison to the ground truth. On the other hand, misclassified pixels are those which fall in different regions compared to the ground truth. This means, smaller percentage of misclassified pixels indicates a better segmentation result.

Firstly we present the segmentation results using an image containing the same textures at different orientations as show in the first and second row of Figure 7. Note that the first original image is a synthetic texture followed by a real texture. We rotated the selected images by $0°$, $-45°$, $90°$ and $45°$ respectively and good segmentation results were obtained. There are misclassified pixels in the $-45°$ rotation and $45°$ rotation regions which mainly effect the boundaries. Nevertheless, this experiment suggests that the algorithm works correctly at varing orientations. Secondly, we present the segmentation result on the same texture at different scales. We resized each of the textures by 100%, 200% and 300% of the original texture.The result shows a good segmentation was obtained although there are misclassified pixels along the boundary. However, this suggests that the proposed method is capabale to discriminate the same texture at different scales. In the fourth row, we present the segmentation result using different textures with similar intensity values. Each of the selected textures has an average intensity value of 119, 121, 130 and 131 respectively. The result indicates that the proposed method can discriminate different textures without being influenced by intensity. Similarly, the fifth row shows the segmentation result with intensity values of 158, 144, 147 and 147 respectively. A good result was obtained both on the regions and boundary with precentage error of $\approx 0.73\%$.

On the other hand, the last two rows show segmentation results of an image containing four similar textures. The upper part of the image contains two similar textures with different intensities which is similar to the lower part of the image. Segmentation results show that a good texture classification rate was obtained with $\approx 0.90\%$ and $\approx 1.63\%$ misclassified pixels, respectively. This shows that the proposed method can differentiate different textures although in small variations. In Figure 8, we present the segmentation results on different region shapes. It shows that although most of the regions are segmented successfully there are some regions over segmented becaused the pixels along the boundaries were misclassified. This can be seen clearly in the first row of Figure 8 which indicates that the proposed method is less effective in segmenting regions with different shapes. However, in homogeneous textures the method shows good segmentation results as shown in the last two rows of Figure 8.

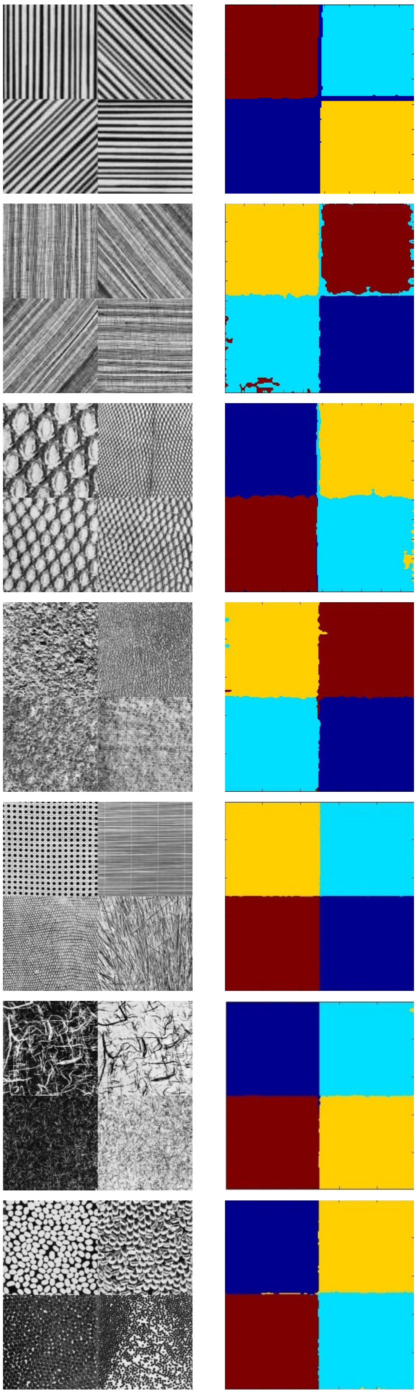

Figure 7: Segmentation results which cover category 1 to 4. The first two rows cover the segmentation results of the same texture at different orientations followed by the result of same texture at different scales which is covered in the third row. The fourth and fifth rows present the segmentation results of different textures with similar intensity values. The last two rows present the segmentation results of two images containing similar textures. The left column shows original images followed by segmentation results in the right column.

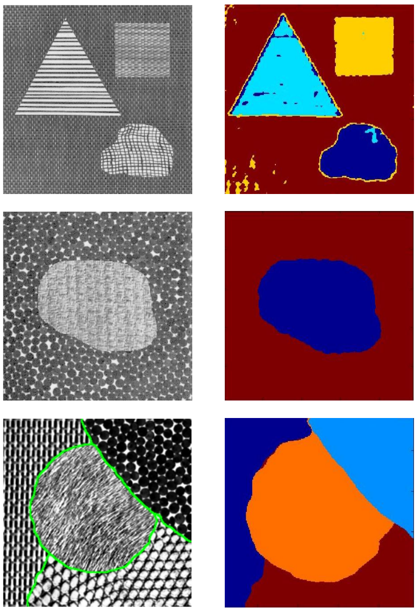

Figure 8: Segmentation results which cover category 5 which is different region shapes. The left column shows original images followed by segmentation results in the right column.

The next section compares the algorithm with two proposed methods in the literatures [10, 4]. Firstly, the results are compared with an algorithm proposed by [10] based on the two images shown in Figure 9. The first test image uses four similar textures and the second image contains four distinct textures. The quantitative comparison shows that the proposed method produces similar segmentation with percentage errors of $\approx 3.1\%$ and $\approx 3.5\%$, respectively compared to the results in [10], with especially better discrimination along the boundaries. Secondly, we compared our results in [4] qualitatively and the results are shown in Figure 10. The results show that the proposed method can discriminate different textures excellently with respect to the boundary among the regions compared to the results in [4]. In fact, all of the boundaries between regions are segmented incorrectly. The method in [4] is less effective clustering the pixels along the boundary as we can see their results showen in Figure 10. The proposed method produced good segmentation with percentage error of $\leq 3.8\%$.

Our last evaluation is another qualitative comparison with two methods which have been recently developed in the literatures [1, 39]. The input image contains five different textures with a size of 256×256 pixels. Based on the results shown in figure 11, our method performed visually better compared to the method of [1] and produced similar result with the method of [39]. This shows that the proposed method has similar performance in the terms of effectiveness in comparison to the recently developed methods. In terms of efficiency, the algorithm takes up to 180 seconds (150 second in average) to segment each image with the size of 256×256 pixels. The proposed algorithm is slightly slower compared to several methods in the literatures [6, 30, 21] with average time taken equal to 120 seconds.

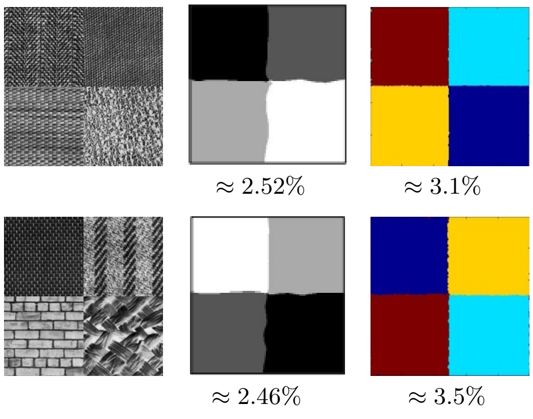

Figure 9: A Quantitative comparison. From left to right: original images, results of [10] and our results, respectively.

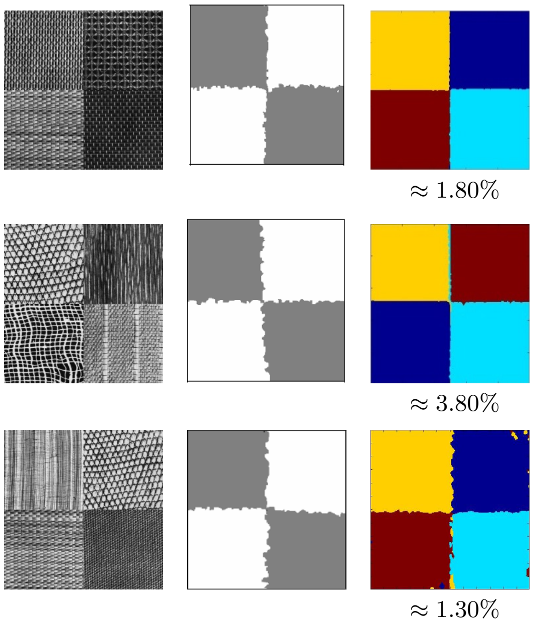

Figure 10: A Qualitative comparison. From left to right: original images, results of [4] and our results, respectively.

Nevertheless, considering the number of features extracted in our proposed algoritham and its accuracy the method is still acceptable. In short, the average results over 80 images indicate relatively small percentage error (≈ 3.57%). In fact, in respect to some of the segmentations, the proposed method can achieve accuracy of up to 99.27% (percentage error 0.73%). The proposed method produces similar results on average compared with the two cited alternative methods tested here. PCA is applied in the proposed method just before the clustering process to reduce the dimensionality. The smoother the features the better the segmentation result. Most homogenous and weekly-homogenous textures tend to produce smoother features (better segmentation results) compared to inhomogenous textures.

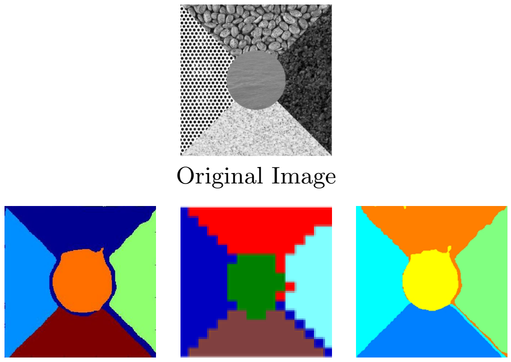

Figure 11: From the left segmentation result of our method, results of [1] and [39], respectively.

4. CONCLUSIONS

In summary, the proposed method uses 16 different features extracted based on GLCM to capture the textures information. However, to maximise the information captured from textures we extracted each feature at two different orientations. Linear normalisation is then applied by ranging the feature matrices values from 0 to 1. We then processed all the features based on equation (4). Before segmentation is performed, smoothing each of the feature matrices based on DWT is implemented. PCA is used to reduce dimensionality and the GMM algorithm is used to complete the segmentation process. The proposed method has been tested on a wide range of different textures. With a small percentage error of ≈ 3.57% the proposed method indicates a high percentage of accuracy both on the regions and boundaries. In fact, it can classify textures at varying orientations, scales and regions with similar textures. Moreover, the proposed method successfully discriminates textures which have similar intensity values without being influenced by intensity. False positive regions are removed using morphology. The next stage of research will test the new algorithm on (diffusion) Magnetic Resonance Imaging (MRI) images.

5. ACKNOWLEDGMENTS

The first author would like to thank for the awards given by Aberystwyth University under the Departmental Overseas Scholarship (DOS) and Doctoral Career Development Scholarships (DCDS).

6. REFERENCES

[1] A. Alrawi, M. S. Ali, and D. A. Ibrahim. Texture segmentation based on multifractal dimension. *International Journal on Soft Computing*, 3(1), 2012.

[2] H. C. Andrews. Gonochrome digital image enhancement. *Applied Optics*, 15(2):495–503, 1976.

[3] L. D. Bastos, P. Liatsis, and A. Conci. Automatic texture segmentation based on k-means clustering and co-occurrence features. In *XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007)*, pages 43–44. Brazilian Symposium on Computer Graphics and Image Processing, October 2007.

[4] S. Bi and D. Liang. Texture segmentation using adaptive gabor filters based on hvs. *SPIE*, 6057, 2006.

[5] P. Brodatz. *Textures: A Photographic Album for Artists & Designers.* Dover Publications,Inc., New York, 1966.

[6] K. I. Chang, K. W. Bowyer, and M. Sivagurunath. Evaluation of texture segmentation algorithms. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1. IEEE, 1999.

[7] D. A. Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Can. J. Remote Sensing*, 28(1):45–62, 2002.

[8] G. R. Cross and A. K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):25–39, 1983.

[9] G. Dougherty. Image analysis in medical imaging: recent advances in selected examples. *Biomedical Imaging and Intervention Journal*, 6(3), July-Sepember 2010.

[10] M. F. A. Fauzi and P. H. Lewis. A fully unsupervised texture segmentation algorithm. In *British Machine Vision Conference 2003*, pages 519–528. British Machine Vision Association (BMVA), September 2003.

[11] J. R. Ferguson. *Using the Grey-level Co-occurrence Matrix to Segment and Classify Radar Imagery.* ProQuest, Information and Learning Company, Ann Arbor, MI, US, 1 edition, 2007.

[12] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):176–188, 1936.

[13] D. Garcia. Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational Statistics and Data Analysis*, 54(4):1167–1178, 2010.

[14] I. Guzide and S. V. Kamarthi. Feature extraction through discrete wavelet transform coefficients. In *Proc. SPIE 5999*, pages 27–35. Intelligent Systems in Design and Manufacturing VI, November 2005.

[15] K. Hammouda and E. Jernigan. *Texture segmentation using gabor filters.* Canada: Center for Intelligent Machines, McGill University, 2000.

[16] H. Handels, A. Horsch, and H.P. Meinzer. Advances in medical image computing. *Methods Inf Med*, 46(3):251–253, 2007.

[17] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions On Systems, Man, and Cybernetics*, 3(6):610–621, November 1973.

[18] J. Ian. *Principal component analysis.* Wiley Online Library, 2005.

[19] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, October 1991.

[20] A. K. Jain and K. Karu. Learning texture discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:195–205, 1996.

[21] U. Kandaswamy, D. A. Adjeroh, and M. C. Lee. Efficient texture analysis of sar imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43(9):2075–2083, 2005.

[22] I. Karoui, R. Fablet, J.M. Boucher, W. Pieczynski, and J. M. Augustin. Fusion of textural statistics using a similarity measure: application to texture recognition and segmentation. *Pattern Analysis & Applications*, 31(3-4):425–434, 2008.

[23] H. B. Kekre, K. Shah, T. K. Sarode, and S. D. Thepade. Performance comparison of vector quantization technique kfcg with lbg, existing transforms and pca for face recognition. *International Journal of Information Retrieval (IJIR)*, 2(1):64–71, 2000.

[24] O. Lahdenoja, M. Laiho, and A. Paasio. Local binary pattern feature vector extraction with cnn. In *9th International Workshop on Cellular Neural Networks and Their Applications*, pages 202–205. IEEE Cat., May 2005.

[25] A. Lev, S. W. Zucker, and A. Rosenfeld. Interactive enhancement of noisy images. *IEEE Trans. On Systems, Man and Cybernetics*, 7(6):435–422, 1977.

[26] G. A. Mastin. Adaptive filters for digital image noise smoothing: An evaluation. *Computer Vision,Graphics and Image Processing*, 31(1):103–121, 1985.

[27] K. S. Niel. Panel discussion-machine vision in industry: Barriers and pitfalls. In *Machine Vision Applications in Industrial Inspection XIII.* Electronic Imaging 2005, January 2005.

[28] K. S. Niel. Industrial computer vision. In *Proceedings of Image and Vision Computing*, pages 198–204. New Zealand 2007, December 2007.

[29] D. A. Reynolds. Gaussian mixture models. *Encyclopedia of Biometric Recognition,Springer, Journal Article*, February 2008.

[30] S. J. Roan, J. K. Aggarwal, and W. N. Martin. Multiple resolution imagery and texture analysis. *Pattern Recognition*, 20(1):17–31, 1987.

[31] J. Shlens. A tutorial on principal component analysis. *Systems Neurobiology Laboratory, University of California at San Diego*, 2005.

[32] L. Soh and C. Tsatsoulis. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *Geoscience and Remote Sensing, IEEE Transactions on*, 37(2):780–795, 1999.

[33] P. Soille. *Morphological Image Analysis: Principles and Applications.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 edition, 2003.

[34] D. Stekel. *Microarray Bioinformatics.* Cambridge University Press, UK, 2003.

[35] R. Taylor, A. Micolich, and D. Jonas. Fractal expressionism. *Physics World*, pages 25–28, October 1999.

[36] Q. Xu, J. Yang, and S. Ding. Texture segmentation using LBP embedded region competition. *Electronic Letters on Computer Vision and Image Analysis*, 5(1):41–47, 2005.

[37] P. Yu, D. A. Clausi, and K. Qin. Unsupervised polarimetric sar image segmentation and classification using region growing with edge penalty. *IEEE Transactions on Geoscience and Remote Sensing*, 50(4):1302 – 1317, 2012.

[38] D. Zheng, Y. Zhao, and J. Wang. Feature extraction using a gabor filter family. In *Proceedings of the 6th IASTED International Conference on Signal and Image Processing.* The International Association of

Science and Technology for Development, August 2004.

[39] Y. Zheng and K. Chen. A hierarchical algorithm for multiphase texture image segmentation. *ISRN Signal Processing*, 2012, 2012.