MULTIPROCESSOR IMPLEMENTATION OF A TEXTURE SEGMENTATION SCHEME FOR SATELLITE RADAR IMAGES

R D Paget (+)
I D Longstaff (+)
B C Lovell (+)

(+) Department of Electrical and Computer Engineering, University of Queensland, Brisbane, Australia
This work is supported by the Co-Operative Research Centre for Sensor Signal and Information Processing.

ABSTRACT
This paper presents a scheme for segmenting images on the basis of differences in localised measures of spatial texture. The scheme used was originally proposed by Wilson and Spann [1] but incorporates a new clustering algorithm which gives improved overall segmentation performance. The Wilson and Spann [1] algorithm uses a clustering algorithm which proved susceptible to initial input parameters and gave poor segmentation on our images. Our algorithm uses a modification of the Koontz, Narendra and Fukunaga [2] clustering algorithm. By linking the clustering to the resolution of the image, significant clusters were able to be realised, yielding a more robust segmentation scheme. The adaptation also resulted in a significant reduction in run-time. The paper is directed towards the problem of segmenting satellite synthetic aperture radar (SAR) images and we give comparisons of the techniques on SAR and other images.

1 INTRODUCTION
A number of remote sensing satellites (*e.g.,* ERS-1, JERS, ALMAZ) use synthetic aperture radar (SAR) to generate high resolution images (20m) of the earth's surface. These radar systems have the ability to produce images even when the earths surface is cloud covered, or in darkness. A disadvantage is that they only operate at one (radar) band, so the conventional multi-spectral techniques for identifying surface cover can not be used. The multi-spectral techniques use the relative spectral response in each pixel as a feature vector. This paper evaluates a technique which attributes features to each pixel based on a localised measure of spatial frequency content to provide the basis for a texture segmentation algorithm.

The texture features are derived from the two dimensional Fourier transform of the whole image. This Fourier transformed image is then segmented into a number of windowed areas. These areas are then inverse transformed (with padding) back to the spatial domain. We now have various measures per pixel of the local spatial frequencies which should contain the information needed to describe the local texture. The recognition algorithm relies on finding separable clusters of these feature vectors. This is an unsupervised pattern recognition process.

2 WILSON AND SPANN TEXTURE SEGMENTATION SCHEME

The Wilson and Spann [1] algorithm has four stages: 1) Feature Extraction 2) Quadtree Smoothing, to reduce noise in the "feature images"; 3) Local Centroid Clustering, which identifies different groups of textures from the "feature images"; 4) Boundary Estimation, takes the segmented image back down the Quadtree, increasing the resolution of the image and reevaluating the boundaries. Figure 1 shows a symbolic representation of the algorithm.

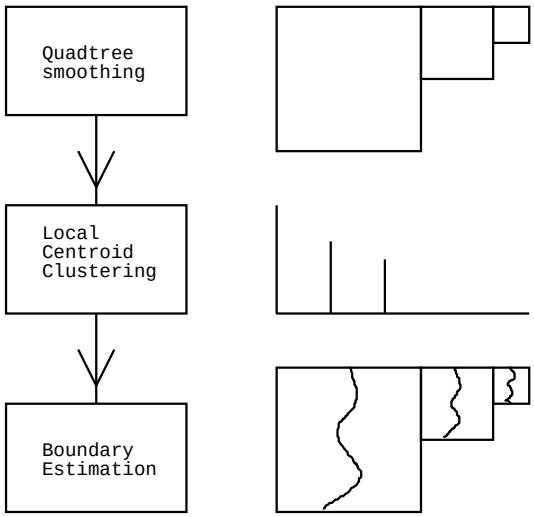


Figure 1: Neighbourhood and its 2-D histogram

2.1 Texture feature extraction

Texture features are designed to depict the correlation between a pixel in question and its neighbours. Wilson and Spann [1] use 2-dimensional "Finite Prolate Spheroidal Sequences" (f.p.s.s) to create filters that are optimally bounded both in the spatial frequency domain and the spatial domain. This means a directional spatial frequency filter can be created which has local finite support only in the spatial domain. In mathematical terms, the f.p.s.s. is the largest eigenvector obtained from equation (1), where B is a one-dimensional band-limiting operator in the spatial frequency domain, I is a one-dimensional index-limiting operator in the spatial domain, and F is the Fourier matrix.

$$B * F * I * F' * B v_\Omega = \lambda v_\Omega \tag{1}$$

$$V_\Omega = v_\Omega * v_\Omega^T \tag{2}$$

where v_Ω is a column vector, and V_Ω in equation (2) is a 2-dimensional spatial frequency filter. The filter can be represented both in the spatial frequency domain and the spatial domain. A convolution of the filter in the spatial domain is equivalent to a windowing in the spatial frequency domain. Another special property of the filter is that it is scalable, *i.e.,* a scaled result from one image filtering is equivalent to scaling both image and filter before filtering.

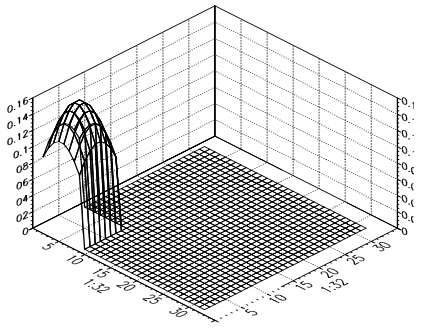


Figure 2: Spatial frequency domain representation of bandpass f.p.s.s.

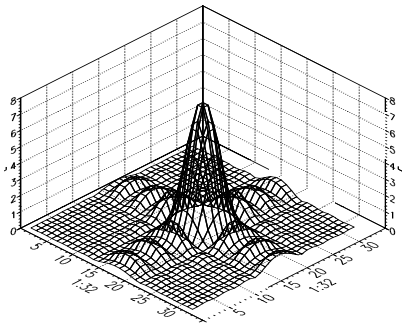


Figure 3: Spatial domain envelope of f.p.s.s.

To obtain the "texture feature images" a two-dimensional Fourier transform of the whole image is first taken. The Fourier transformed image is then separately multiplied by various different f.p.s.s. spatial frequency windowing functions (14 in our case) which are then inverse transformed (with padding) back to the spatial domain. There are now 14 measures per pixel of the local spatial frequencies to describe the texture.

2.2 Quadtree smoothing

Local area averaging or "blurring" reduces noise, and therefore reduces the variance of the features associated with each class. This method is often used with SAR images, as "speckle" noise is reduced by blurring. However blurring also reduces resolution. This is why the *multi-scale* quadtree technique is used. A quadtree is a layered version of the original image, each higher layer is a locally averaged and sampled version of the previous layer. The sampling is achieved via a two by two averaging array. The averaging at the top level gives the most well separated clusters, whereas the original unaveraged image has the highest spatial resolution.

2.3 Local centroid clustering

The *local centroid clustering* occurs at the highest quadtree level; here the image is segmented into its various classes. The classification algorithm does not require prior training nor *a priori* knowledge of the number of classes, which is an advantage if not all the textural characteristics of the terrain within the image are known.

2.4 Boundary estimation

Boundary estimation is required to work back down the quadtree after the original image has been segmented at the highest level. As the algorithm descends through the quadtree level-by-level, each segmentation boundary pixel is mapped to four pixels. Each of these new pixels must be assigned to either the interior of adjacent regions, or the new boundary.

3 MEAN-SHIFT CLUSTERING

Wilson and Spann [1] use a variation of the "mean-shift algorithm" that was original proposed by Fukunaga and Hostetler [3] as a special case of their "gradient clustering algorithm." Like most clustering algorithms, an input parameter has to be specified. This parameter determines the size of the windowing function, which in turn determines the amount of clustering. Silverman [4] says that there is an intuitive link between a windowing function and the size of a kernel in density estimating. The larger the kernel, the smoother the underlying density is estimated. Therefore it can be visualised that a large window will produce a few large clusters, while a small window will produce many little clusters.

In order to try and overcome the dependence of the algorithm on the window size, Wilson and Spann [1] increased the window size after each pass until successive runs produced consistent results. One pass was the continuous run of the "mean-shift algorithm" repeated until no more means were shifted.

The algorithm was found to be still very dependent on window size, due to the stopping mechanism. We found the stopping mechanism was unacceptable for real images. Consider the case where we have two important clusters close to each other in feature space, also included in this feature space are outlining points that are going to cluster. As the algorithm stands, the stopping mechanism is dependent on all points in the feature space whether they are forming significant clusters or not. Therefore it is quite probable that the two original clusters will be merged to one so as to allow all points to cluster; useful information has been lost for the sake of insignificant points.

4 HIERARCHICAL CLUSTERING WITH SIGNIFICANCE

The idea presented in this section is the inclusion of "significance" into the clustering algorithm. This "significance" is determined by whether the cluster that is formed in feature space can form a segmented region in the image of the original resolution. By Wilson and Spann's [1] requirements, such a segment occurs if the segment contains pixels that do not border pixels of other segments.

To interpret "significance" as part of the clustering algorithm, pixels from the image must hold the one-to-one relationship with their respective points in feature space throughout the clustering procedure. The clustering algorithm can then assign a class to each point, which can be interpreted both in the image and feature space. "Significance" is given to individual point/pixels if they fulfill certain criteria both in the feature space and image. The clustering algorithm must also adhere to a few rules:

1. Once a point/pixels has been labelled as "significant" it must keep its class for the remainder of the clustering algorithm
2. The "insignificant" point/pixels must be able to continue their clustering without being affected by "significant" point/pixels

To implement "significance" into the "Mean-Shift algorithm", "significant" points

must be kept stationary in feature space. Otherwise, if two different "significant" points were to merge, confusion would be created for those "insignificant" points that were to join this cluster as to which class they should belong. The "Mean-Shift algorithm" however requires all points to be moving so that they all converge onto the same mean of their chosen cluster. If a point stops before it reaches its correct mean, then in order for that cluster to form properly, all the other points that were to converge onto the same mean must now converge onto the stationary point. This may never happen if other converging points also stop. Therefore rule 2 in the clustering rules cannot be adhered to.

4.1 Hierarchical clustering

The hierarchical clustering algorithm presented by Koontz, Narendra, and Fukunaga [2] could be considered a distant cousin to the "Mean-Shift algorithm". It has already been said that the "Mean-Shift algorithm" is a special case of the more general algorithm presented by Fukunaga and Hostetler [3] which employs a gradient estimate. The hierarchical clustering algorithm also uses a gradient estimate. Silverman [4] gives a generalisation of the algorithm, but we used the one defined in Koontz, Narendra, and Fukunaga [2], due to its ease of computation and because there is no need to find a density estimate.

The basis of the algorithm is to assign a parent to each point in feature space. If no parent can be assigned, then that point becomes a root. Each root is given a different class, and those points with parents take on the class of their parent. The algorithm is called hierarchical because no point can have more than one parent, but a parent can have many children.

Let d_{ij} denote the distance between two points X_i and X_j in feature space. Define the neighbourhood η_θ^i of X_i as

$$\eta_\theta^i = \{k | d_{ik} \leq \theta, k \neq i\} \tag{3}$$

where θ is a given scalar. Define the (population) density N_i at X_i as

$$N_i = |\eta_\theta^i| = \text{ number of elements in } \eta_\theta^i. \tag{4}$$

Finally we define a gradient index g_{ij}

$$g_{ij} = \frac{N_j - N_i}{d_{ij}}. \tag{5}$$

The parent node of X_i is the one that gives the highest non-negative gradient. For cases where there is more than one suitable parent, a tie breaking rule is applied so only one parent exists. Nodes that have not been assigned a parent are labelled as roots. These roots can form the nucleus of new clusters.

The overall clustering of points is similar to that achieved via the "mean-shift algorithm." In the hierarchical algorithm the points are not moved. Therefore the previous concept of "significance" can be applied.

4.2 Implementation

The hierarchical algorithm is non-iterative and is completely determined by a single

control variable θ, or equivalently the window size. As it stands, starting with a small window and increasing its size after each calculation makes no sense, as each window size will produce its own unique solution independent of previous calculations. However this all changes if we consider "significance."

Hierarchical Clustering with Significance

- Step 1: Start with a small window size, and label all points as "insignificant"
- Step 2: Perform hierarchical clustering on all points, but only assign classes to the "insignificant" points
- Step 3: Map the classes to the image
- Step 4: Find image pixels that are completely surrounded by pixels of the same class (interior pixels)
- Step 5: Label interior pixels and their immediate neighbours as "significant", so they are not reassigned in future clustering
- Step 5: Test stopping criterion, if not finished increase window size and go to step 2

There are two stopping criteria that are suggested. The first tests whether the number of "significant" point/pixels is greater than a certain percentage of the image size. The second test determines whether each cluster contains "significant" point/pixels. If either is true the clustering is stopped.

4.3 Multiprocessor implementation

The complete segmentation scheme was built on a multiprocessor. For an $N \times N$ image, the clustering algorithm was able to be reduced from an order of N^3 to N^2 computations. With the new clustering algorithm, the modifications required very little extra computational time. Since fewer iterations were required it was running faster. The final run time for a 512x512 image was around ten minutes on a MasPar with 4096 processing elements.

5 RESULTS

The Wilson and Spann [1] algorithm performed well on simulated data. The segmentation of real images (SAR) required further tuning to optimise segment boundaries. With the replacement of the "mean-shift" clustering algorithm with the "significant hierarchical" algorithm, better segmentation could be obtained in a real image.

Advantages: 1) Small increments in the window size could be used without affecting the stopping mechanise. This allowed for the capture of fine texture differences in the image; 2) Worked well for feature vectors with small variances (which occur in real images); 3) A significant speed increase was observed, especially for the larger images.

Disadvantage: 1) Did not work as well for feature vectors with large variances *e.g.,* synthetic brodatz images, especially if a particular texture had a multimodal distribution

Further Research: 1) The use of edge detectors on the feature images and image itself to aid in segmentation; 2) Finding the correct Quadtree level to isolate each texture. To obtain the best feature vectors each filter must match the scale of the texture. However with satellite images, texture scale does not change very much.

A comparison of the two techniques is presented in Figures 4 to 9, showing the segmentations of three different types of images: three brodatz images, a photographic image, and a SAR image.

6 CONCLUSION

In conclusion we find that the problem of the Wilson and Spann [1] algorithm arises from its characteristic of continuing the clustering of all feature points until a global measure is satisfied. This can allow quite separate clusters to become merged to accommodate just a few outliers.

The technique we propose which uses a "significance" criterior to identify clusters gives a much improved performance.

The technique relies on a one-to-one mapping between points in feature space and pixels. Hence a cluster in feature space can be tested to see whether it provides an appropriate segment in the image. Therefore a feature point can be continually reclustered until it is both represented in a cluster and as a pixel in an appropriate segment of the image, such a point/pixel was called "significant". This provided a local stopping mechanism for the clustering algorithm. The clustering technique proved more successful than the original, with the added advantage of being faster.

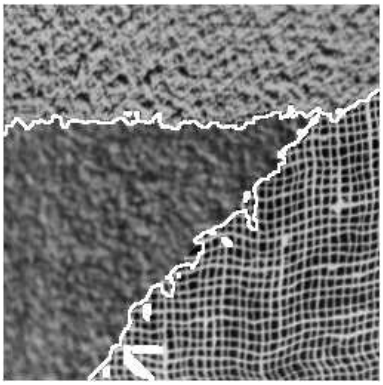
References

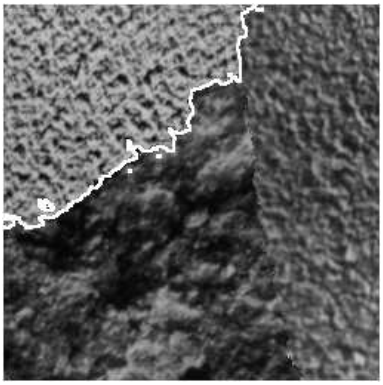
[1] R. Wilson and M. Spann, Image Segmentation and Uncertainty, Research Studies Press Ltd, 1988.

[2] W. L. Koontz, P. M. Narendra and K. Fukunaga, "A Graph-Theoretic Approach to Nonparametric Cluster Analysis", IEEE Transactions on Computers, C-25, pp. 936-944, 1976.

[3] K. Fukunaga, L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition", IEEE Transactions on Information Theory, IT-21, pp. 32-40, 1975.

[4] B. W. Silverman, Density Estimation, New York, Chapman and Hall, 1986.





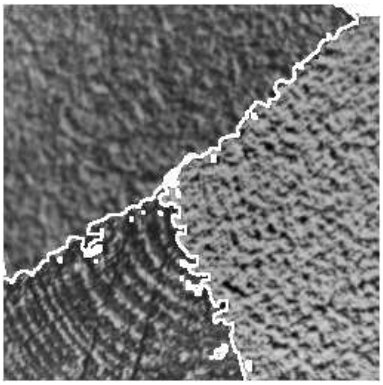
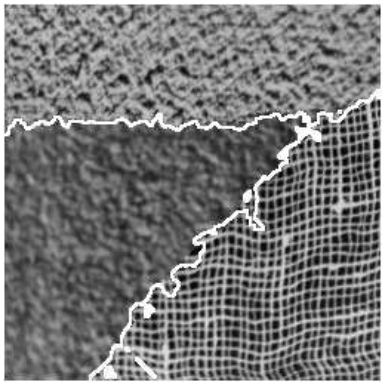
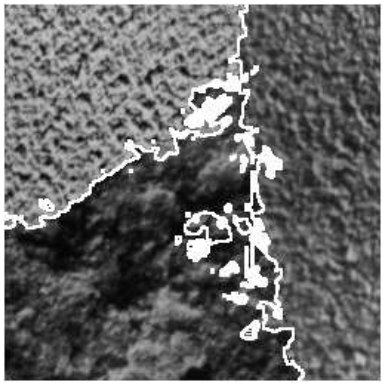


Figure 4: Segmentation of synthetic brodatz images using the Wilson and Spann algorithm





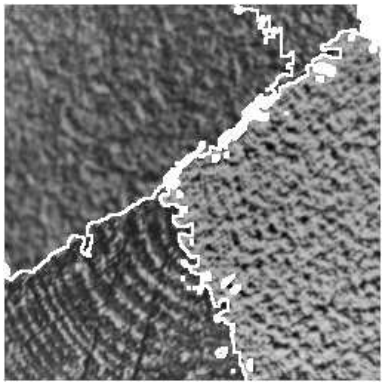


Figure 5: Segmentation of synthetic brodatz images with "significant hierarchical" clustering