Reprinted from

Ninth International Symposium

Machine Processing of

Remotely Sensed Data

with special emphasis on

Natural Resources Evaluation

June 21-23, 1983

Proceedings

Purdue University
The Laboratory for Applications of Remote Sensing
West Lafayette, Indiana 47907 USA

SEGMENTATION OF REMOTELY SENSED DATA USING PARALLEL REGION GROWING

J.C. TILTON

Computer Sciences Corporation
Silver Spring, Maryland

S.C. COX

Goddard Space Flight Center
Greenbelt, Maryland

ABSTRACT

The improved spatial resolution of the new earth resources satellites will increase the need for effective utilization of spatial information in machine processing of remotely sensed data. One promising technique is scene segmentation by region growing. Region growing can use spatial information in two ways: (1) only spatially adjacent regions merge together, and (2) merging criteria can be based on region-wide spatial features. We describe a simple region growing approach in which the similarity criterion is based on region mean and variance (a simple spatial feature). An effective way to implement region growing for remote sensing is as an iterative parallel process on a large parallel processor. We explore a straightforward parallel pixel-based implementation of our algorithm and compare its efficiency with sequential pixel-based, sequential region-based and parallel region-based implementations. Experimental results from on aircraft scanner data set are presented followed by a brief discussion of proposed improvements to the segmentation algorithm.

I. INTRODUCTION

The new earth resources observation satellites of the 1980's will provide imagery with much higher information content than provided by the Landsat Multispectral Scanners (MSSs) of the 1970's. For example, the Thematic Mapper (launched in July 1982) has a spatial resolution of 30 meters in 6 spectral bands (and 120 meters in one thermal infrared band), while the Landsat MSS has a spatial resolution of about 80 meters in 4 spectral bands. Planned satellites include the French SPOT with 20 meter resolution in three spectral bands (10 meter resolution in a panchromatic band), and NASA's proposed Multispectral Linear Array (MLA) instrument with 10 to 20 meter ground resolution in six or more spectral bands.

Concurrent with the increase of information content in data from earth resources observation satellites has been an increase in (ground based) computer processing capability. Of particular relevance to image processing and classification are several large parallel processors under development. One such processor is the Massively Parallel Processor (MPP) being developed for the NASA Goddard Space Flight Center by Goodyear Aerospace [1,2]. The MPP contains 16,384 processing elements logically connected in a 128-by-128 array with each element having data transfer interconnections with its four nearest neighbors. This fixed configuration architecture contrasts with the reconfigurable architecture of another (not quite so) large parallel processor: the ZMOB at the University of Maryland [3]. The ZMOB consists of 256 (a "mob") Z80A-based microprocessors that communicate via a fast shift-register bus.

Present operational image processing and classification algorithms for remote sensing applications are generally designed for use with Landsat MSS data on serial computers. In particular, these algorithms make little use of spatial information (an exception is ECHO [4]). The improved spatial resolution of the new earth resources satellites will increase interest in the utilization of spatial information. In the past it has proved difficult to implement algorithms exploiting spatial information because of constraints imposed by the available serial computers. Fortunately, new parallel computers such as the MPP and ZMOB are well suited for algorithms that utilize spatial information.

As a part of NASA Goddard's MLA supporting science program, we are exploring region growing as an approach to image segmentation. Region growing can use spatial information in two ways. First, region merging is restricted by the relative spatial location of the regions. Only regions that are spatially adjacent can merge together. Second, merging criteria can be designed to use regional spatial features. A simple spatial feature we use is the region variance. Our segmentation algorithm performs region growing as an iterative parallel process which can take full advantage of the parallel architectures of either the MPP or ZMOB. The algorithm can be thought of as essentially a spatially-constrained clustering algorithm that can use spatial, spectral and/or temporal information as clustering criteria.

II. RELATED IMAGE SEGMENTATION RESEARCH

Since the initial development of machine processing of image data in the late 1960s, many image segmentation approaches have been investigated. Fu and Mui [5] present a detailed survey of image segmentation from the point of view of the biomedical application of cytology. They categorize segmentation approaches into three classes:

1. characteristic feature thresh- olding or clustering,

2. edge detection, and

3. region extraction.

The bulk of successful approaches in cytology were reported to fall in the first category; characteristic feature thresholding or clustering. No successful application was reported using region extraction, and edge detection approaches had difficulties because the "detected edges may not necessarily form a set of closed connected curves that surround connected regions."

Schachter et al [6] discuss image segmentation by clustering of local feature values; an approach that falls in Fu and Mui's category 1. Schachter et al's application is a natural scene consisting of a house, lawn, bushes and sky, which is more like scenes encountered in remote sensing than those encountered in cytology. Accordingly, their conclusions may have greater direct relevance to remote sensing applications than the results presented by Fu and Mui. Schachter et al attempted segmentation by clustering on local features such as gray level, gradient magnitude, Laplacian, regional standard deviation, regional total variation of gray level, and a texture transform. After examining segmentations based on these features and combinations of these features, they were dissatisfied with the results and suggested that "if we want to obtain better segmentation performance, we must make use not only of similarities among the image points, but also of their relative positions." They suggest doing a preliminary segmentation based on feature space clusters and then taking "the points belonging to the clusters as 'core points' of image regions and ...[completing] the segmentation of the image by a region growing process starting from these points." We might suspect from the above that a region growing (region extraction) approach may be fruitful in remote sensing applications where it may not be useful in biomedical applications such as cytology.

Both Schachter et al and Fu and Mui point out that region growing approaches have had the disadvantage that the regions produced depend on the order in which portions of the image are processed. But Schachter et al suggest that implementing region growing as "an iterative parallel process" would overcome the order dependent problem.

III. REGION SIMILARITY CRITERIA

A key part of any region growing technique is the similarity test used to determine whether a region should grow by merging with a neighboring region or pixel. The earliest attempt at region growing for remote sensing applications was reported by Muerle and Allen [7]. Muerle and Allen tried two different similarity criteria in their experiments. The first was the Kolmogorov-Smirnov two-tailed hypothesis test. This test "resulted in mediocre extraction of objects." The second test was based on "the average magnitude difference between the two cumulative distributions." This test performed better at extracting sensible regions, but Muerle and Allen were not at all satisfied with their results. They opined, nonetheless, that the optimal similarity criteria would be a function of the means and variances of the regions being considered for merging.

In their test of similarity, Muerle and Allen did not assume a form for the distribution of the samples in each region. In developing their similarity criterion, Kettig and Landgrebe [8,4] assumed their region samples followed a multivariate normal distribution. (A multivariate rather than univariate distribution was assumed since they were interested in segmenting multi-channel scanner data.) It is interesting to note that distributions with multivariate distributions can be completely specified by a mean vector and covariance matrix. The success Kettig and Landgrebe had with their multivariate normal assumption is consistant with Muerle and Allen's concluding remarks on optimal similarity criteria.

Kettig and Landgrebe used region growing as an integral part of a sample classification scheme. Their technique also incorporated a cell rejection test that split up initial regions that were not sufficiently homogeneous. They called their region growing approach conjunctive paritioning.

In defining their merging criterion, Kettig and Landgrebe discussed both supervised and unsupervised approaches. In the supervised approach, the multivariate normal object class distributions were assumed to be known **a priori**. (They were estimated by clustering techniques.) In the unsupervised approach, the multivariate normal object class distributions were estimated separately for each region at the time the merging decision was made. Kettig and Landgrebe proposed a multivariate similarity test based on the multi-channel mean vectors and covariance matrices of the regions being compared. This test was not used in the final unsupervised ECHO algorithm, however, because the test required that the number of pixels in the initial regions had to be more than the number of channels in the multispectral data being classified. Since Kettig and Landgrebe wanted to be able to use 2-by-2 initial regions, this proved to be a problem even for 4-channel Landsat MSS data.

To avoid this restriction on the smallest size of the initial regions, Kettig and Landgrebe opted to use a multiple univariate similarity test. This test is essentially a one-dimensional version of the multivariate similarity test applied separately for each channel. Regions are annexed or merged only when the test is satisfied for all channels separately.

The similarity test used in our parallel segmentation algorithm is very similar to a one dimensional version of the multiple univariate test used by Kettig and Landgrebe, and can be extended to the multi-channel case, similarly. As did Kettig and Landgrebe, we use statistical hypothesis testing in the development of our similarity criterion. We test the null hypothesis that the distributions of the two regions are identical. Our test combines a test on the means of the two regions with a test on the variances of the two regions. Both tests assume the samples in each region are normally distributed.

Kettig and Landgrebe used a means test which assumed that the two regions had equal variances. In our means test, we allow for possibly unequal variances. We felt we could afford to use this somewhat more complicated, but more accurate, test because of anticipated computational gains from implementing our algorithm on parallel computers such as the MPP or ZMOB. The means test we use is referred to in the statistical literature (e.g. [9,10]) as the Behrens-Fisher problem. This problem has no exact solution, but according to Sachs [10], it is usually adequate to calculate an approximate degree of freedom value for the Student's t-statistic and perform the standard Student's t-test.

Let

ABS(x) = absolute value of x,
SQR(x) = square of x,
SQRT(x) = square root of x,
mi = mean of region i,
vi = variance of region i, and
di = degrees of freedom in region i.

Then

$$t = \frac{\text{ABS(m1-m2)}}{\text{SQRT(v1/(d1-1) + v2/(d2-1))}}$$

and

$$rd = \frac{\text{SQR(v1/(d1-1) + v2/(d2-1))}}{\dfrac{\text{SQR(v1/(d1-1)}}{\text{d1}} + \dfrac{\text{SQR(v2/(d2-1))}}{\text{d2}}}$$

where t is the Student's t-statistic and rd is rounded to the nearest integer and used as an approximate degrees of freedom in the Student's t-test.

We used the F-ratio test for our variance test. Kettig and Landgrebe used an identical test. With the above definitions, the F-ratio is simply F = v1/v2.

For our composite similarity criterion, we combine the means test and variance test by taking the geometric mean (or, equivalently, the product) of the probabilities of random occurence of the particular t- and F-values calculated. This is different than the usual practice in statistical hypothesis testing. Usually, one selects in advance a particular significance level and accepts or rejects the null hypothesis at that significance. In keeping with this approach, Kettig and Landgrebe preset a significance level and accepted the null hypothesis only if the two regions satisfied both the means test and the standard deviation test separately (for all channels). We prefer the geometric mean, because with it we can (to a certain extent) trade off similarity in mean with similarity in variance. This allows recognition of the more coarsely textured ground-cover classes in the segmentation.

We have only used the mean and variance of regions as features for the comparison of regions. Other regional features could be used, such as any of the various texture features discussed in the literature during the past decade (e.g. [11]). It is not clear, however, whether a simple, effective similarity criterion can be derived for comparing some of the more complicated texture features. It should be noted that the variance feature we use is sometimes used as a first-order texture feature.

IV. REGION GROWING STRATEGIES

As noted earlier, earlier region growing algorithms are processing order dependent. This is because earlier methods use a sequentially based region growing strategy. It was impractical on the serial computers then available to use a parallel region growing strategy.

Muerle and Allen [7], and Kettig and Landgrebe [8,4] both used a similar sequential region growing strategy. The whole image is initially divided into square regions of n-by-n pixels. Initial region sizes of size 2-by-2 up to 8-by-8 were tested. The processing started with the region in the upper left corner of the image. Starting with this region, all neighboring regions were compared in turn with the initial region and merged with the initial region if the similarity criterion was satisfied. This continued until the initial region couldn't grow any further. Then the next encountered unmerged initial region was considered for merging with its neighbors, and so on, until the entire image was processed.

In addition to the order dependence problem, this region growing strategy has the fault that the best match at each step is not sought. Kettig and Landgrebe [8] considered modifying their algorithm to find the best match, but found such a strategy too difficult to implement in a sequential manner.

The iterative parallel segmentation strategy that we have developed has the advantage over earlier methods of doing the best merges first. Since the globally best merges are done first, there is no processing order dependence. The basic idea of the algorithm is to identify the most similar pair of adjacent regions in the image, merge them, identify the most similar pair of adjacent regions in the resulting image and merge them, etc., until a desired number of regions remain, or until all pairs of adjacent regions are not similar enough to be merged according to a predetermined minimum value for the similarity criterion. A flow chart of the basic algorithm is given in figure 1.

A. PARALLEL PIXEL-BASED IMPLEMENTATION

The segmentation algorithm can be implemented in parallel in a straightforward manner on the MPP for image sizes with up to 128-by-128 initial regions. (Image size would be 512-by-512 pixels with 4-by-4 initial regions.) Larger images could be processed by folding the image in several layers onto the 128-by-128 processor MPP architecture, or by hooking initial 128-by-128 region portions together using an edge initialization scheme. Specifying the algorithm for large images is beyond the scope of this paper. We will limit ourselves here to images that produce initial regions of size 128-by-128 or less.

In the parallel pixel-based implementation on the MPP, the entire image is initially subdivided into n-by-n regions (n=3 and n=4 were used in our tests). The means and variances (or standard deviations) of each region are calculated as region features and stored in separate image planes in the MPP. The degrees of freedom and a unique region label for each region are stored in other image planes. These image planes map directly onto individual MPP processors.

Each iteration of this parallel pixel-based implementation is started by calculating, in parallel, the best similarity criterion values for each nonredundant neighbor direction (east, southwest, south and southwest). As the criterion values are calculated for each direction, the criterion values are set to zero for neighboring pixels within the same region. The criterion values are stored in a real image plane in the MPP which overlays the region label, degree of freedom, mean and variance image planes. If the globally best similarity criterion value is less than a preset minimum, the algorithm ends. Otherwise, the pair of regions labels associated with the globally best similariy criterion value is noted, and these two regions are merged by making the appropriate changes in the region label, degree of freedom, mean and variance image planes. If the remaining number of regions is equal to a preset minimum, processing ends. Otherwise, a new iteration is started.

```
                     ---------
                     | START |
                     ---------
                         |
                        \|/
     -------------------------------------
     | set minimum for similarity criterion |
     -------------------------------------
                         |
                        \|/
     -------------------------------------
     | set minimum final number of regions |
     -------------------------------------
                         |
                        \|/
               ---------------------------------
  _________\| compute similarity criterion for |
  |        /| all spatially adjacent regions   |
  |         ---------------------------------
  |                      |
  |                     \|/
  |           ----------------------------
  |           |   find maximum criterion  |
  |           |    value for all pairs of |
  |           | spatially adjacent regions|
  |           ----------------------------
  |                 |
  |                \|/
  |                /-\
  |               / is \                  --------
  |             / value of\               | print |
  |           /  criterion  \             | current|
  |         /   greater than  \_____\     | region |
  |         \    the preset   / NO  /     | map and|
  |           \   minimum   /             |   END  |
  |             \    ?    /               --------
  |               \     /
  |                \_/
  |                 |  YES
  |                \|/
  |       ------------------------------------
  |       | identify the pair of regions with |
  |       | the maximum similarity criterion  |
  |       |  value and merge the two regions  |
  |       ------------------------------------
  |                 |
  |                \|/
  |                /-\
  |               /   \                   --------
  |             / is the \                | print |
  |           /  remaining \              | current|
  |_________/  # of regions  \_____\      | region |
     YES    \  greater than  / NO  /      | map and|
              \ the preset /              |   END  |
               \ minimum /                --------
                 \  ?  /
                   \_/
```

Figure 1. Basic flow chart of iterative parallel segmentation algorithm.

Each iteration can be performed on the MPP with O(1) (the order of magnitude of one) operations. Since one pair of regions is merged each iteration, the total number of iterations necessary to end up with, say, ENDREGIONS number of regions is NS*NL-ENDREGIONS. NS and NL are the number of columns and rows, respectively, in the mean and standard deviation images. Since ENDREGIONS is generally much less than NS*NL, the number of operations for the entire parallel pixel-based implementation of the segmentation algorithm is O(NS*NL).

B. SEQUENTIAL PIXEL-BASED IMPLEMENTATION

Our parallel iterative segmentation algorithm lends itself naturally to the parallel implementation discussed above. The algorithm can also be implemented sequentially, with parallelism mimicked by several passes through the image data. Alternatively, the number of passes through the image can be limited to two forward passes and one reverse pass in which the calculation results are passed up and down through the image row by row.

As the reader can probably imagine, an efficient sequential pixel-based segmentation algorithm is difficult to program, and even the most efficient takes a relatively large amount of processing time. If in each iteration we merge the pair of regions with the best global similarity criterion value, the number of iterations necessary is NS*NL-ENDREGIONS. The number of operations per iteration is O(NS*NL). Since ENDREGIONS would be generally much less than NS*NL, the number of operations for the entire sequential pixel-based implementation would be O((NS*NL)**2).

Alternatively we can merge, say, Q*100% of the remaining number of regions each iteration. The number of iterations in this case is approximately

$$\frac{\ln(NS*NL/ENDREGIONS)}{\ln(1/(1-Q))}$$

or O(ln(NS*NL)). The total number of operations for such a sequential pixel-based implementation is O(NS*NL*ln(NS*NL)). This compares to the O(NS*NL) operations required for the parallel pixel-based implementation.

C. SEQUENTIAL REGION-BASED IMPLEMENTATION

The segmentation algorithm can be implemented in a region-based mode that takes less than the O(NS*NL*ln(NS*NL)) operations required for the pixel-based mode. In this implementation we generate from the initial segmentation a "feature table" listing the region label, mean and variance feature values, and the number of degrees of freedom in each region. The label and similarity criterion value of the best neighoring region for merging is updated every iteration and stored in the feature table. A separate "neighbor table" is generated containing the region label, the number of regions with larger region label that are are spatially adjacent to the region, and a list of region labels of these adjacent regions. (The neighboring regions with lower region label are excluded from neighbor table list to avoid redundant comparisons.) A region label merge list is maintained so that the spatial extent of each remaining region can be generated after the last iteration.

The best merges are found each iteration by searching through the feature table and comparing regions that are neighbors according to the neighbor table. Merges are performed by modifying these two tables and adding to the merge list.

Since the size of these tables decrease with each iteration, the number of operations required per iteration drop with each iteration. The number of regions remaining at the start of each iteration is:

$$NS*NL*(1-Q)^{I-1}$$

where I is the number of iterations performed and NS, NL are as before. Q is the percentage of regions merged each iteration. The number of operations performed each iteration is directly proportional to the number of regions remaining at the start of each iteration. As was the case for the sequential pixel-based implementation, the total number of iterations for the sequential region-based implementation is approximately

$$\frac{\ln(NS*NL/ENDREGIONS)}{\ln(1/(1-Q))}$$

or O(ln(NS*NL)). The total number of operations for the sequential pixel-based implementation can be calculated by summing the number of regions remaining at the start of each iteration over the total number of iterations. This calculation reveals that the total number of operations is less than O(NS*NL/Q). Since Q is on the order of 0.10, this sequential implementation essentially requires an order of calculations which is similar to that required by the parallel pixel-based implementation. In practice, the sequential implementation takes more processing time because the constant multiplying the NS*NL term in the detailed number of operations equation is larger for the sequential version. This is because of the larger amount of disk I/O and searches required by the sequential version.

D. PARALLEL REGION-BASED IMPLEMENTATION

The success of the sequential region-based implementation as compared to the sequential pixel-based implementation naturally leads us to wonder if a parallel region-based implementation might be more efficient than the parallel pixel-based implementation discussed above. A parallel region-based algorithm is more naturally implemented on a reconfigurable parallel computer like the ZMOB [3] than on a fixed configuration parallel computer like the MPP [1,2]. We can only speculate on this implementation, for we have not had the opportunity to implement the segmentation algorithm on the ZMOB. (We have, however, previously implemented the segmentation algorithm in sequential and parallel pixel-based modes and the sequential region-based mode.)

The parallel pixel-based implementation of the segmentation algorithm on the MPP does not use all the processors efficiently in the later iterations. Processors that correspond to pixel locations interior to regions perform no useful work besides recording the image map locations covered by the region. The segmentation algorithm could be implemented on the ZMOB so that each processor corresponds to a region or set of regions rather than an image map location. The ZMOB can be reconfigured each iteration so that regions communicate only with their current neighboring regions, simplifying keeping track of which regions are neighbors to each other. As with the sequential version, a region label image merge list must be maintained to regenerate the spatial extent of each region after the last iteration.

The relatively small number of processors (256) contained in the ZMOB compared the MPP may offset the configuration advantage the ZMOB may have over the MPP. The MPP with 16,384 processors in a 128-by-128 array can efficiently process a 256-by-256 image with a simple image folding (each processor process four pixel locations). Even with a 128-by-128 inital region image, the ZMOB with its 256 processors would have to do a substantial amount of processing in serial, rather than parallel, in the early iterations of the algorithm.

Even with a 16,384 processor ZMOB, the reconfigurable architecture may still not offer a subtantial advantage over the fixed configuration MPP. This is because the limiting process for both the parallel pixel-based and parallel region-based implementations is the performing of merges. The order of operations required to run the segmentation algorithm on an expanded ZMOB would still be O(NS*NL) - the same as for the MPP. The constant multiplying NS*NL in the detailed operations equation may be slightly lower for the ZMOB implementation, however.

Even though the order of the number of operations is the same for the ZMOB and MPP, we still would anticipate a significant advantage for the ZMOB in the later iterations of the algorithm, when the number of regions drops closer to 256. We would suggest that an ideal implementation of our segmentation algorithm would use the MPP in the early iterations and shift to the ZMOB in the later iterations.

Figure 2. Panchromatic aircraft scanner image of study site shown at 10-meter resolution.

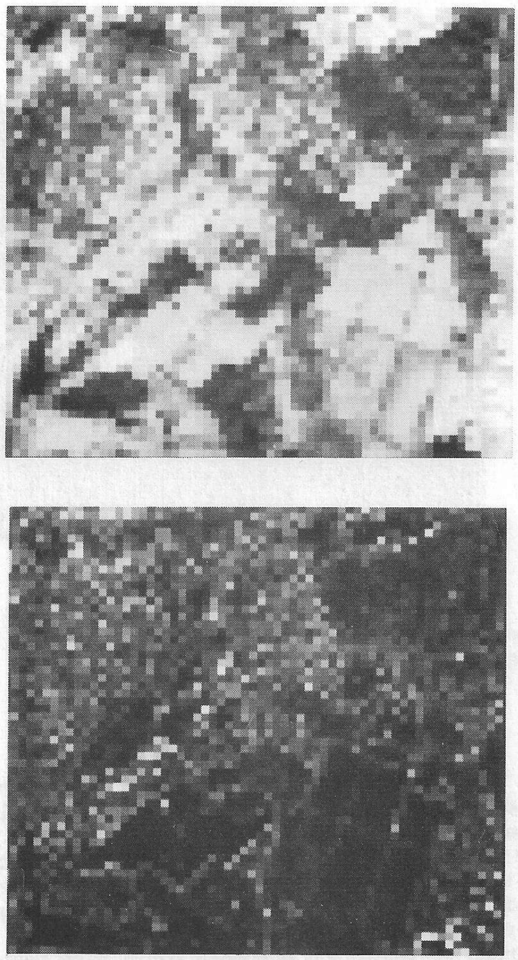
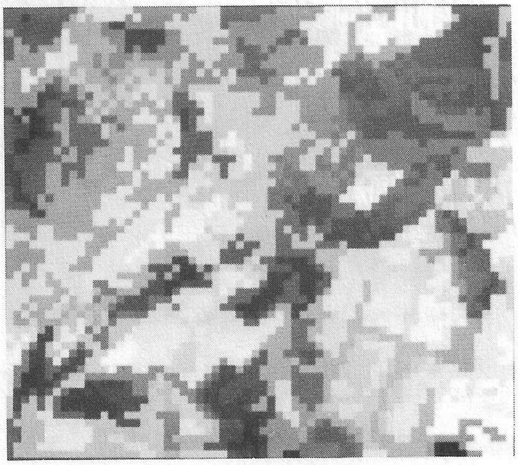


Figure 3. $-by-4 pixel region initialization. Mean (upper) and standard deviation (lower) images are shown.



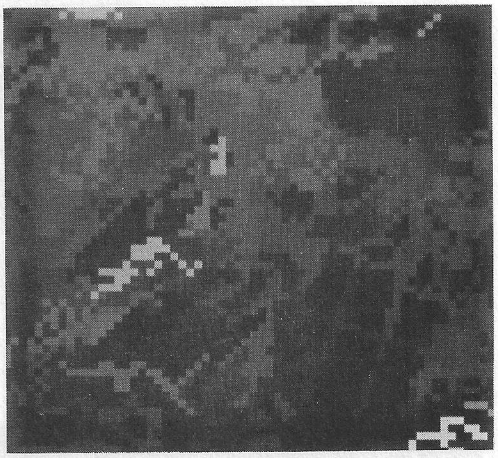


Figure 4. Segmentation result after 43 iterations. Mean (upper) and standard deviation(lower) images are shown.

V. EXPERIMENTAL RESULTS

We have tested our segmentation algorithm on Daedulus scanner data from over the Maryville, TN area which was collected by Geospectra Corp. of Ann Arbor, MI. We used a panchromatic combination of the three channel data that was formed by taking the square root of the sum of the squares of each channel and rescaling the result. (An average of the three channels could have been used just as well.) We selected the 232-by-256 portion of the 10-meter resolution panchromatic image shown in figure 2. We divided this image into 4-by-4 regions and calculated means and standard deviations for these regions. The resulting 40-meter resolution mean and standard deviation images are shown in figure 3. The darker areas in the panchromatic image are generally forested areas (esp. in the upper right), residential areas can be seen in the upper left, and agricultural fields can be seen mainly in the lower half of the image. The bright area near the lower right corner is a land fill.

We used the sequential region-based implementation of our algorithm to segment the image. After 43 iterations we had 160 remaining regions. We allowed merges down to a minimum similarity criterion value of 5.0E-5. Figure 4 shows the resulting mean and standard deviation images. The forested area in the upper right was not cleanly detected, with a portion of it being joined with a residential area. Other forested areas were detected, however. Residential areas were detected as several smaller regions rather as one large region. The algorithm worked best on the agricultural areas, where individual fields were cleanly picked out. These agricultural fields held together even when the minimum criterion value was allowed to drop to as low as 2.0E-10. The forest and residential areas were totally confused at this lower level, however.

VI. DIRECTIONS FOR FURTHER RESEARCH

As we mentioned earlier, Schachter et al suggest doing a preliminary segmentation based on feature based clusters and completing the segmentation using region growing. Zucker [12] suggests a similar, but less elaborate, global initialization technique for improving the performance of region growing approaches to image segmentation. Zucker suggests that histograms of local features could provide a basis for initialization of a region growing algorithm. Such local features could be pixel gray level, or the local region mean and variance features we use in our segmentation algorithm. Histograms on these local features could reveal image areas that are relatively constant in the select features. These areas could then be used as initial regions for region growing. We would hope that such an approach would not only improve the performance of our algorithm, but also reduce the total number of iterations required. We plan to pursue global initialization in the near future.

Schachter et al [6] suggest incorporating discrete or probabilistic relaxation into future segmentation algorithms based on region growing. We have made limited tests of a simple relaxation technique that allows a pixel on a region edge to change its region identification to that of a neighboring region at any iteration. The technique is based on the fact that as a region grows, its feature values gradually change. Occasionally the feature values may change sufficiently that some pixels the were previously merged into the region are actually more similar to a neighboring region. When such a pixel is on a region edge adjacent to a region that is now more similar to it than its current region, our augmented algorithm changes the region membership of that pixel to that of the adjacent region.

We must await the installation of the MPP at NASA Goddard's research facility before we can test this approach on images which generate initial region images that are much larger than 16-by-16 (e.g., a 64-by-64 original image initialized with 4-by-4 pixel regions). This region switching approach cannot be implemented in a region-based mode, so we must rely on the MPP to provide practical processing times. However, based on limited test results on 16-by-16 initial region images, we feel that this relaxation augmentation may reduce the confusion between forest and residential areas seen in the results discussed in section V above.

VII. CLOSING REMARKS

Region growing, implemented as an iterative parallel process, is a promising segmentation technique that can effectively utilize the spatial information contained in the higher resolution imagery gathered by the newer earth resources satellites. We have described a simple region growing algorithm that use regional means and variances as merging features, and have discussed parallel and sequential implementations of this iterative parallel algorithm. This simple segmentation algorithm can be improved by appropriate region initialization and by adding on a form of relaxation to the merging process.

VIII. REFERENCES

1. K. E. Batcher, "Design of a Massively Parallel Processor," IEEE Trans. on Computers, Vol. C-29, pp. 836-840 (1980).

2. T. Kushner, A. Y. Wu and A. Rosenfeld, "Image processing on MPP: 1," Pattern Recognition, Vol. 15, No. 3, pp. 121-130 (1982).

3. A. Rosenfeld and A. Y. Wu, "Parallel computers for region-level image processing," Pattern Recognition, Vol. 15, No. 1, pp. 41-50 (1982).

4. R. L. Kettig and D. A. Landgrebe, "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects," IEEE Trans. on Geoscience Elect., Vol. GE-10, No. 1, pp. 19-26 (1976).

5. K. S. Fu and J. K. Mui, "A Survey on Image Segmentation," Pattern Recognition, Vol. 13, No. 1, pp. 3-16 (1981).

6. B. J. Schachter, L. S. Davis and A. Rosenfeld, "Some Experiments in Image Segmentation by Clustering of Local Feature Values," Pattern Recognition, Vol. 11, No. 1, pp. 19-28 (1979).

7. J. L. Muerle and D. C. Allen, "Experimental Evaluation of Techniques for Automatic Segmentation of Objects in a Complex Scene," in Pictorial Pattern Recognition (G. C. Cheng, R. S. Ledley, D. K. Pollock and A. Rosenfeld editors), pp 3-13, Thompson, Washington (1968).

8. R. L. Kettig and D. A. Landgrebe, "Computer Classification of Remotely Sensed Multispectral Image Data by Extraction and Classification of Homogeneous Objects," LARS Information Note 050975, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, IN (1975).

9. G. W. Snedecor and W. G. Cochran, Statistical Methods, The Iowa State University Press, Ames, IA (1967).

10. L. Sachs, Applied Statistics, Springer-Verlag, New York, NY (1982).

11. R. M. Haralick, "Statistical and Structural Approaches to Texture," Proceedings of the IEEE, Vol. 67, No. 5, pp. 786-804 (1979).

12. S. W. Zucker, "Region Growing: Childhood and Adolescence," Computer Graphics and Image Processing, Vol. 5, pp. 382-399 (1976).

James C. Tilton is a Senior Engineer with the System Sciences Division of Computer Sciences Corporation (CSC), Silver Spring, MD. He is currently CSC Task Leader for the Spatial Resolution Studies portion of NASA Goddard's Multispectral Linear Array Science Support Task. He received a Ph. D. degree in Electrical Engineering from Purdue University in 1981, and an M. S. degree in Optical Sciences from the University of Arizona in 1978. In 1976 he received both the M. E. E. degree and the B. A. degree in Electrical Engineering, Environmental Science and Engineering, and Anthropology from Rice University. His primary research interest is the development of optimal techniques for analyzing remotely sensed data.

Scott C. Cox is with the Interpretive Techniques Branch of the Information Extraction Division at Goddard Space Flight Center, Greenbelt, MD. He is presently Study Scientist for Spatial Resolution Studies for the Multispectral Linear Array Science Support Team. Prior to his joining the Interpretive Techniques Branch, he was State Program Manager at the Eastern Regional Remote Sensing Applications Center NASA/GSFC, responsible for technology transfer activities in Michigan, New York and New Jersey. Mr. Cox recieved a B. A. from Purdue University in 1972 and an M. S. in 1974. In addition, he holds an M. R. P. degree from the Pennsylvania State University which he received in 1978. Since coming to Goddard in 1978, he has been actively involved in remote sensing applications research and technology transfer.