[Bea1983]

Step-Wise Optimization for Hierarchical Picture Segmentation

Authors: **Beaulieu** Jean-Marie, Morris Goldberg

Conference: Conference on Computer Vision and Pattern Recognition

Washington, D.C.

1983, vol. 59, p. 64

Abstract: Hierarchical picture segmentations are very useful in picture analysis. We present a sequential segment merging algorithm for picture segmentation. Each iteration merges two segments which optimize a step-wise criterion. We relate picture segmentation to optimization problems. The implementation of the segmentation algorithm is examined, and results are presented and discussed.

Step-Wise Optimization for Hierarchical Picture Segmentation,
Beaulieu Jean-Marie, Morris Goldberg,
Conference on Computer Vision and Pattern Recognition, Washington, D.C., 1983, p. 64.
[Bibtex]

DOWNLOAD **the published version**

STEP-WISE OPTIMIZATION FOR HIERARCHICAL PICTURE SEGMENTATION

Jean-Marie Beaulieu and Morris Goldberg

Electrical Engineering Dept., University of Ottawa,
Ottawa, Canada, K1N-6N5.

ABSTRACT

Hierarchical picture segmentations are very useful in picture analysis. We present a sequential segment merging algorithm for picture segmentation. Each iteration merges two segments which optimize a step-wise criterion. We relate picture segmentation to optimization problems. The implementation of the segmentation algorithm is examined, and results are presented and discussed.

1. INTRODUCTION

A central problem in picture analysis is that of segmentation, i.e. partitioning a picture into disjoint regions that are homogeneous in some sense. Let a vector (x,y) represent a point (i.e. a pixel) of the picture plane I, $(x,y) \in I$. Let $\bar{f}(x,y)$ represent the multi-spectral value of the pixel (x,y). A picture partition $\mathbb{P}=\{P_1, P_2, \ldots P_n\}$ involves the decomposition of the picture plane I into disjoint regions $P_1, P_2, \ldots P_n$, i.e. $P_i \subset I$, $P_i \cup P_j = \emptyset$ for $i \neq j$, and $\cup P_i = I$. Some sequential and hierarchical picture segmentation algorithms are now presented.

Kettig and Landgrebe [1], and Gupta and Wintz [2] propose a sequential approach for picture segmentation. The picture is first divided into small cells (e.g. 2x2 pixels), and then sequentially examined. If a cell is statistically similar to a connected region, then this cell is merged with the region. The statistics of the region are updated in order to improve the results of the following statistical decisions. On the other hand, if the cell is not found to be similar to any connected regions, then the cell is considered as the seed point of a new region. The process continues until every cell is examined and assigned to a region. A problem with this algorithm is that the results are dependent of the order on which the cells are examined.

In hierarchical picture segmentation, new segments are formed by the division or the fusion of old ones. The segmentation process is often defined by predicate equations:

$$Q(P_i) = \text{true} \qquad \text{for } P_i \in \mathbb{P} \qquad \text{(eq. 1)}$$

$$Q(P_i \cup P_j) = \text{false} \qquad \text{for } P_i, P_j \in \mathbb{P} \qquad \text{(eq. 2)}$$

where P_j must be adjacent to P_i. The predicate $Q(P_i)$ can correspond, for example, to an evaluation of the homogeneity of the region P_i. The utilization of the predicate yields an hierarchical process which splits a region if $Q(P_i)$ is false, or merges two regions if $Q(P_i \cup P_j)$ is true. Some hierarchical segmentation algorithms are now reviewed.

Ohlander, Price and Reddy [3] propose a top-down segmentation algorithm based on histogram analysis. The picture is broken into smaller and smaller regions until eq. 1 holds. The predicate $Q(P_i)$ and the splitting are based on feature histograms of the regions. The feature histograms are examined to find the best peak for division. Lower and higher thresholds are chosen to mark the peak limits, and the region is divided into sub-parts corresponding to inside and outside peak limit values. The sub-parts are again checked for further division. If a region P_i can not be subdivised, the predicate $Q(P_i)$ is set to a true value, and the region P_i becomes an element of the final partition. Many modifications and improvements to this algorithm are proposed [4],[5].

Horowitz and Pavlidis [6] present a split-and-merge algorithm applied on a picture pyramid. The predicate $Q(P_i)$ consists of the comparison of the region gray level range

$$r_i = \max_{(x,y) \in P_i} (f(x,y)) - \min_{(x,y) \in P_i} (f(x,y))$$

with a threshold t. Therefore, the predicate is true if the range r_i is lower or equal to the threshold, $Q(P_i) = (r_i \leq t)$.

The utilization of a pyramidal data structure defines the way in which regions can be merged or split. A pyramid is a stack of regular picture regions of decreasing sizes. The picture regions of one level are split into four regular sub-parts to yield the regions of the next lower level. Therefore, the split-and-merge algorithm consists of 1) splitting a region P_i into its four sub-parts $P_{i,1}$, $P_{i,2}$, $P_{i,3}$ and $P_{i,4}$ if the predicate $Q(P_i)$ is false, or 2) merging $P_{i,1}$, $P_{i,2}$, $P_{i,3}$ and $P_{i,4}$ if $Q(P_{i,1} \cup P_{i,2} \cup P_{i,3} \cup P_{i,4})$ is true.

CH1891-1/83/0000/0059$1.00

Freuder [7] presents a relative approach for hierarchical region merging. It is relative because the predicate $Q(P_i)$ is not only a function of the region P_i, but also of its surroundings. So, the value of $Q(P_i)$ will change if a neighbor of P_i is merged or split. The picture is first divided into a regular array of cells or initial regions. Then, for each region P_i, we choose the neighbor "most like" P_i. This involves first a measure of the affinity between P_i and each of its neighbors, and then the selection of the neighbor which optimizes the measure. Freuder uses the region mean difference multiplied by their area $|m_i - m_j| \times (A_i + A_j)$ as an affinity measure between two regions P_i and P_j. The neighbor P_j that minimizes this measure is chosen as the "most like" neighbor of P_i, and a "link" is made between the two regions.

After every region has been linked to one of its neighbors, the regions which are joined by double-links are merged into new composite regions. The process is then iterated on this new picture partition: links are made and double-linked regions are merged, ... This approach can be viewed as a replacement of a threshold based decision process, by a local optimization process (optimization on segment neighbors), for the selection of regions to merge.

In the following section, we show how optimization processes can be used in picture segmentation, and present a step-wise optimization algorithm for picture segmentation. Then in the next sections, we describe an implementation of the algorithm, examine its operation with a simple example, present some experimental results, and discuss the characteristics of the algorithm.

2. OPTIMIZATION AND HIERARCHICAL PICTURE SEGMENTATION

In this section, we first examine how optimization processes can be used for picture segmentation [8],[9]. Then we present a hierarchical segmentation algorithm based on step-wise optimization.

Let us define a simple picture partition cost function $\mathbb{R}(\mathbb{P})$:

$$\mathbb{R}(\mathbb{P}) = \sum_{P_i \in \mathbb{P}} R(P_i) \qquad \text{(eq. 3)}$$

where $\mathbb{P} = \{P_1, P_2, \ldots P_n\}$ and $R(P_i)$ is a measure of segment P_i cost, e.g. segment variance or approximation error. Segmentation can therefore be considered in terms of finding the partition $\mathbb{P}$ that minimizes $\mathbb{R}$.

The identification of a global optimum requires a search over the whole picture partition space $\{\mathbb{P}\}$, i.e. over all possible partitons. But, the implementation of this search is prohibited by the large size of $\{\mathbb{P}\}$ space. Therefore, we must constrain the search space S to a sub-set of $\{\mathbb{P}\}$, $S \subset \{\mathbb{P}\}$. Thus, we obtain only a sub-optimum which can be very close to the global optimum if the sub-set S is correctly selected.

Two kinds of sub-sets S often used are 1) the neighborhood of an initial picture partition, and 2) the sub-set yielded by a hierarchical data structure. Thus, if we know that the optimum is close to an initial picture partition $\mathbb{P}^0$, we can surely constrain the search on the neighbors of $\mathbb{P}^0$. Furthermore, we can use a "gradient descent" like procedure [8],[9]. This consists of moving a pixel from one segment to another if such a move improves the cost function $\mathbb{R}$. This iterative process is terminated when a local optimum is found. But, in general, it is difficult to get a sufficiently good initial partition $\mathbb{P}^0$.

A hierarchical data structure can also be employed to define a useful subset of picture partitions. A hierarchy of segments can be represented by a segment tree in which nodes correspond to segments. Each segment P_i^L is linked to segments of a lower level $P_{i,1}^{L-1}$, $P_{i,2}^{L-1}$, ... which are disjoint sub-sets of P_i^L, and which are called "sons" of P_i^L. Therefore, a picture partition corresponds to a sub-set of these tree nodes.

We now present an algorithm for picture segmentation involving the construction of a segment tree. It consists of a sequence of step-wise optimizations. This is an adaptation of the algorithm of Ward [10] to picture segmentation. The algorithm needs an initial picture partition $\mathbb{P}^0 = \{P_1^0, P_2^0, \ldots P_n^0\}$ with n segments, and a cost function $\mathbb{R}(\mathbb{P})$ which reflects the cost or loss of information produced by the representation of the picture by the partition $\mathbb{P}$.

At each iteration k of the algorithm, two segments of $\mathbb{P}^{k-1}$ merge to produce a new partition $\mathbb{P}^k = \{P_1^k, P_2^k, \ldots P_{n-k}^k\}$. The number of segments is decreased by one at each iteration. Thus, $\mathbb{P}^k$ contains n-k segments, where n is the initial number of segments. The cost function must increase monotonically with k as the reduction of the number of segments. Furthermore, each iteration merges the two segments that produce the lowest increase of the cost function $\mathbb{R}$. This means that each iteration performs a search to find the optimum segment pair. This is a step-wise optimization.

If eq. 3 is used as a cost function then the increase $C_{i,j}$ of $\mathbb{R}$ resulting from the merging of two segments P_i and P_j can easily be calculated:

$$C_{i,j} = R(P_i \cup P_j) - R(P_i) - R(P_j) \qquad \text{(eq. 4)}$$

Thus, $C_{i,j}$ is the step-wise criterion to be optimized. So, each iteration k involves 1) the identification of all pairs of connected segments (P_i, P_j), 2) the calculation of $C_{i,j}$, 3) the selection of the lowest $C_{i,j}$, and 4) the merging of the two corresponding segments.

We must point out that the algorithm does not guarantee that $\mathbb{P}^k$ will optimize $\mathbb{R}(\mathbb{P})$ among the partitions with n-k segments. But, the optimization of the step-wise criterion $C_{i,j}$ assures that each iteration does the best to optimize the cost function $\mathbb{R}$. Therefore, the algorithm will probably yield a "good solution". This solution can furthermore be improved by a "gradient descent" like iterative process. Moreover, the implied hierarchical data structure can constitute an advantage for many applications.

A difficulty with our algorithm is to define when to stop the segment merging. The number of segments n-k of the result $\mathbb{P}^k$ can be used as a stopping criterion, if we know in advance the number of segments needed. On the other hand, the increase in the cost function ($\mathbb{R}(\mathbb{P}^{k+1}) - \mathbb{R}(\mathbb{P}^k)$) yielded by each iteration can also be employed as a stopping criterion. For example, the process is terminated if the increase exceeds a threshold value (which can correspond to the fact that the loss of information becomes too high).

3. STEP-WISE OPTIMIZATION SEGMENTATION ALGORITHM

In this section, we give a more complete description of the step-wise optimization segmentation algorithm. First, we present the step-wise criterion that is used.

This algorithm is designed to process multi-channel pictures $\bar{f}(x,y) = (f_1(x,y), f_2(x,y),\dots f_K(x,y))$. The cost function employed is defined as the sum of segment costs as in eq. 3. The segment cost $R(P_i)$ is itself defined as the weighted sum of the squared difference between the pixel values and the segment mean:

$$R(P_i) = \sum_{k=1}^{K} w_k \sum_{(x,y)\in P_i} [\, f_k(x,y) - m_{k,i} \,]^2 \qquad \text{(eq. 5)}$$

where K indicates the number of channels, $\bar{m}_i=(m_{1,i}, m_{2,i},\dots m_{K,i})$ is the mean of segment P_i, and w_k is a weighting factor which takes into account the difference between channel ranges. The step-wise criterion is as given in eq. 4. The step-wise minimization of $C_{i,j}$ yields the two segment merge that minimizes the increase in the overall pixel variance around the segment means.

We now present the algorithm, that starts with an initial partition $\mathbb{P}^0$ with Nseg segments. The algorithm is divided into three parts. Part I is the only one which uses the multi-channel picture $\bar{f}(x,y)$. Part II employs the results of part I to initialize the criterion values $C_{i,j}$. Part III performs the hierarchical merging of segments.

Part I : Calculation of description parameters

$\forall\, P_i \in \mathbb{P}^0$ (for all segments of $\mathbb{P}^0$)

1 - calculation of moments (M_i) and cost R_i

N_i = number of pixels in segment P_i

$$Sf_{k,i} = \sum_{(x,y)\in P_i} f_k(x,y) \qquad \text{for } k=1\dots K$$

$$Sf^2_{k,i} = \sum_{(x,y)\in P_i} f_k(x,y)^2 \qquad \text{for } k=1\dots K$$

$$R_i = R(P_i) = \sum_{k=1}^{K} w_k\, [\, Sf^2_{k,i} - (Sf_{k,i})^2 / N_i \,]$$

2 - calculation of neighbors of P_i

$$B_i = \{\, P_j \mid P_j \text{ is a neighbor of } P_i \,\}$$

Part II : Initialization of criterion values $C_{i,j}$

$\forall\, P_i \in \mathbb{P}^0$ (for all segments of $\mathbb{P}^0$)

calculate $C_i = \{\, C_{i,j} \mid P_j \in B_i \text{ and } j > i \,\}$
(since $C_{i,j} = C_{j,i}$ only one is calculated)

The calculation of $C_{i,j}$ is done as follow:

$$C_{i,j} = R(P_i \cup P_j) - R_i - R_j$$

where :

$$R(P_i \cup P_j) = \sum_{k=1}^{K} w_k\, [\, Sf^2_{k,i} + Sf^2_{k,j} - (Sf_{k,i} + Sf_{k,j})^2 / (N_i + N_j) \,]$$

Part III : Hierarchical merging of segments

1 - Nseg = number of segments in $\mathbb{P}^0$

2 - find m, n such as $C_{m,n} = \mathrm{Min}\ C_{i,j}$

3 - define a new segment P_v corresponding to the fusion of P_m and P_n

$$N_v = N_m + N_n$$

$$Sf_{k,v} = Sf_{k,m} + Sf_{k,n} \qquad \text{for } k = 1\dots K$$

$$Sf^2_{k,v} = Sf^2_{k,m} + Sf^2_{k,n} \qquad \text{for } k = 1\dots K$$

$$R_v = \sum_{k=1}^{K} w_k\, [\, Sf^2_{k,v} - (Sf_{k,v})^2 / N_v \,]$$

$$B_v = B_m \cup B_n \cap \overline{\{P_m, P_n\}}$$

4 - remove segments P_m and P_n

5 - update neighbor lists B_j and criteria $C_{i,j}$

$\forall\, P_j \in B_v$ (for all neighbors of P_v)

i) $B_j = B_j \cap \overline{\{P_m, P_n\}} \cup \{P_v\}$
ii) delete $C_{j,m}$ and/or $C_{j,n}$
iii) calculate $C_{v,j}$ ($C_{v,j} \in C_v$) using the equations of part II

6 - Nseg = Nseg - 1
Stop here, if a final partition with Nseg segments is required.
Otherwise, return to step 2.

Certain aspects of the algorithm may now be stressed. Firstly, only part I uses the multi-channel picture $\bar{f}(x,y)$ and needs to know the pixel membership, i.e. to which segment a pixel belongs. Part I calculates the segment moments M_i, the segment cost R_i and the neighbor set B_i for each initial segment $P_i \in \mathbb{P}^0$, and this is the only information needed by the remaining parts of the process. This information is sufficient for the calculation of 1) the criterion values $C_{i,j}$, and 2) the description parameters M_v, R_v and B_v of each new segment P_v produced by merging. This results in a reduction of the number of computer operations needed (CPU time). Moreover, the memory space and computing time of part II and III are only functions of 1) the number of initial and final segments, and 2) the number of neighbors per segment.

A large amount of memory space is required to store the description parameters and criterion values M_i, R_i, B_i, C_i for each segment P_i. Therefore, in many cases, all this information cannot remain in the main memories, but must reside on a secondary device (e.g. magnetic disk). This will result in a large amount of input/output operations, and will increase the response time.

4. AN EXAMPLE

We now use a simple example to show how the above algorithm operates. Fig. 1 shows a small picture (4x4 pixels) with 7 initial constant level segments. This is a one channel picture (K=1); therefore we omit the k indices for simplicity, and set the channel weight factor to 1 (w=1).

The algorithm starts with an initial partition P^0 of 7 segments. At the first iteration, two segments merge and yield a new segment labelled P_8. At the following iterations, segments P_9, P_{10}, ... are sequentially created. Fig. 2 shows a segment tree which indicates the sequence of segment merging.

We now show the step by step operation of the algorithm. Part I is the only one which requires the picture gray level matrix (fig. 1-a) and the initial partition picture (fig. 1-b). For each initial segment P_i, part I calculates the number of pixels N_i, the moments Sf_i and Sf_i^2, the segment cost R_i and the segment neighbors B_i. The part II of the algorithm calculates the criterion values $C_i=\{ C_{i,j} \}$ for the initial segments P_i. These calculations are shown in the upper part of Table 1.

Part III of the algorithm, the most important, does the hierarchical segment merging. At the first iteration, the available criterion list is that calculated in part II, and is reported in column "iteration 1" of Table 1. Step III-2 finds the minimum of this list, which is the value 1.2 enclosed by a rectangle. This value corresponds to $C_{2,5}$; therefore the segments P_2 and P_5 must be merged. The merge yields a new segment named P_8. The descriptive parameters of P_8 are calculated from those of P_2 and P_5 (step III-3), and are written in a new line of Table 1. The segments P_2 and P_5 are removed (step III-4, -5) by 1) deleting any appearances of P_2 and P_5 in B_i set (this is not reported in the table), and 2) by removing from the criterion list any $C_{i,j}$ involving P_2 or P_5; this can be see by an examination of the column "iteration 2" of Table 1. Then, the criteria $C_8 = \{ C_{8,j} \}$ involving P_8 and its neighbors are calculated (step III-5) and inserted in the criterion list. This completes the first iteration of part III.

This process is repeated at each iteration: find the minimum of the criterion list, create a new segment, and update the criterion list. The process stops when the number of segments has been reduced to the required value. In this example, we do not stop the process until only one segment remains.

5. EXPERIMENTAL RESULTS AND DISCUSSION

The algorithm described above is applied to aerial photography of land. Fig. 3 contains a LANDSAT satellite picture of an agricultural area in Saskatchwan (64x64 pixels). It is the .6-.7 um band (k=1) and .8-1.1 um band (k=2) of a four-channel multispectral scanner. We present the results yielded by our algorithm, and discuss them.

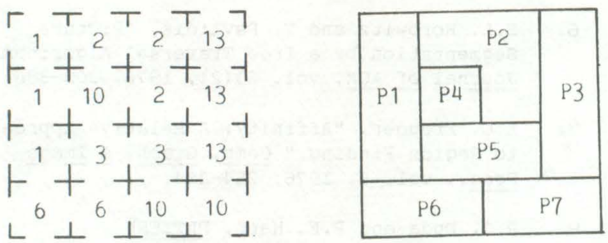
An initial segmentation of the picture (Fig. 4) with 1453 segments is produced by the grouping of pixels with a difference lower or equal to 3 such as to get small segments. This initial partition and the two channel picture (K=2) constitute the inputs to our algorithm. Fig. 5 shows three picture partitions yielded by the algorithm. They contain respectively 100, 50 and 25 segments. Their standard deviations from the segment means are, in the order, $\sigma_1 = 3.51$ and $\sigma_2 = 7.30$; $\sigma_1 = 4.45$ and $\sigma_2 = 8.93$; $\sigma_1 = 5.44$ and $\sigma_2 = 11.61$.

These three partitions constitute interesting results. Each segment of these partitions corresponds to a distinct region of the two channel picture. According to the goal of the segmentation, we can prefer a partition with many small segments (e.g. the 100 segments partition) or one with fewer and larger segments (e.g. the 25 segments partition).

We can also note that the segment characteristics are not homogeneous over the picture plane. It means that a picture partition that contains good segments for a part of the picture can yield too small or too large segments for some other parts of the picture.

These results, as the preceding example, point out that more information is needed in order to select the "best" partition among the good ones yielded by our algorithm. Two promising ways to include this additional information are: 1) inclusion of this information in the step-wise criterion, 2) utilization of a post-processing which takes into account the context of a segment node in the complete segment tree, in order to find when to stop the segment merging. We project to study these two approaches and report any interesting results.

In conclusion, the step-wise optimization algorithm presented seems promising, and up to date, has yielded good results.



a) gray level values   b) initial partition

Fig. 1 : A small picture

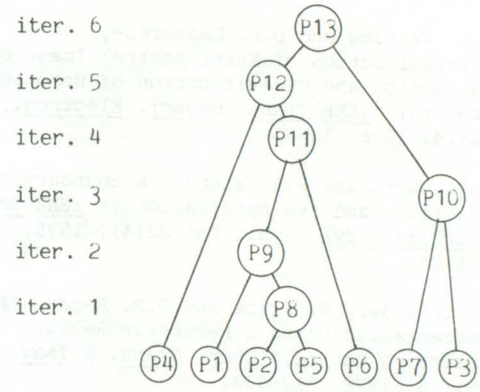


Fig. 2 : Segment tree.

Table 1 : Segment description parameters and criterion lists

	N_i	Sf_i	Sf_i^2	R_i	i,j	$C_{i,j}$	Lists of criteria at each iteration it.1	it.2	it.3	it.4	it.5	it.6
P1	3	3	3	0.	1,2	1.5	1.5					
					1,4	60.7	60.7	60.7				
					1,5	4.8	4.8					
					1,6	30.0	30.0	30.0				
P2	3	6	12	0.	2,3	181.5	181.5					
					2,4	48.0	48.0					
					2,5	1.2	$\boxed{1.2}$					
P3	3	39	507	0.	3,5	120.0	120.0					
					3,7	10.8	10.8	10.8	$\boxed{10.8}$			
P4	1	10	100	0.	4,5	32.7	32.7					
P5	2	6	18	0.	5,6	9.0	9.0					
					5,7	49.0	49.0					
P6	2	12	72	0.	6,7	16.0	16.0	16.0	16.0			
P7	2	20	200	0.								
P8	5	12	30	1.2	8,1	3.7		$\boxed{3.7}$				
					8,3	210.7		210.7				
					8,4	48.1		48.1				
					8,6	18.5		18.5				
					8,7	82.5		82.5				
P9	8	15	33	4.9	9,3	270.0			270.0			
					9,4	58.7			58.7	58.7		
					9,6	27.2			27.2	$\boxed{27.2}$		
					9,7	105.6			105.6			
P10	5	59	707	10.8	10,6	48.1				48.1		
					10,9	303.1				303.1		
P11	10	27	105	32.1	11,4	48.4					$\boxed{48.4}$	
					11,10	277.0					277.0	
P12	11	37	205	80.6	12,10	244.6						$\boxed{244.6}$
P13	16	96	912	336.								

REFERENCES

1. R.L. Ketting and D.A. Landgrebe, "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects," IEEE Trans. Geosci. Electron., vol. GE-14, 1976, 19-26.

2. J.N. Gupta and P.A. Wintz, "A Boundary Finding Algorithm and its Applications," IEEE Trans. Circuits & Sys., vol. CAS-22(4), 1975, 351-362.

3. R. Ohlander, K. Price and D.R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," Comp. Graph. & Image Proc., vol. 8, 1978, 313-333.

4. D.L. Milgram and D.J. Kahl, "Recursive Region Extraction," Comp. Graph. & Image Proc., vol. 9, 1979, 82-88.

5. S. Shafer and T. Kanade, "Recursive Region Segmentation by Analysis of Histograms," Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc., Paris (France), 1982, 1166-1171.

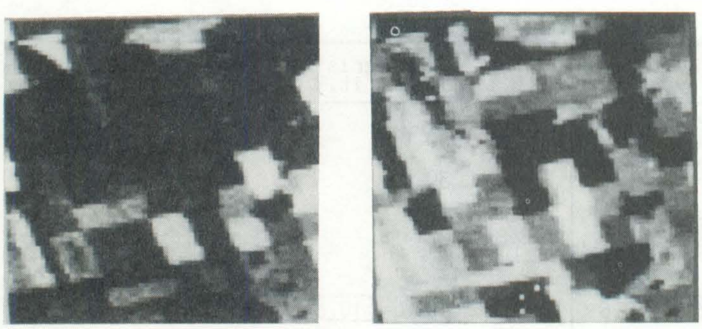
6. S.L. Horowitz and T. Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm," Journal of ACM, vol. 23(2), 1976, 368-388.

7. E.C. Freuder, "Affinity: A Relative Approach to Region Finding," Comp. Graph. & Image Proc., vol. 5, 1976, 254-264.

8. R.O. Duda and P.E. Hart, PATTERN CLASSIFICATION AND SCENE ANALYSIS, John Wiley & Sons, ed., New-York, 1973.

9. D.A. Pierre, OPTIMIZATION THEORY WITH APPLICATIONS, John Wiley & Sons, ed., New-York, 1969.

10. J.H. Ward, "Hierarchical Grouping to Optimize an Objective Function," J. Amer. Stat. Ass., vol. 58, 1963, 236-245.



a) .6-.7 um band   b) .8-1.1 um band

Fig. 3 : LANDSAT picture (64x64 pixels)

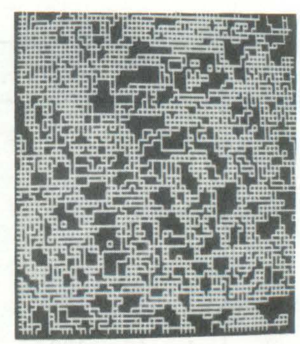
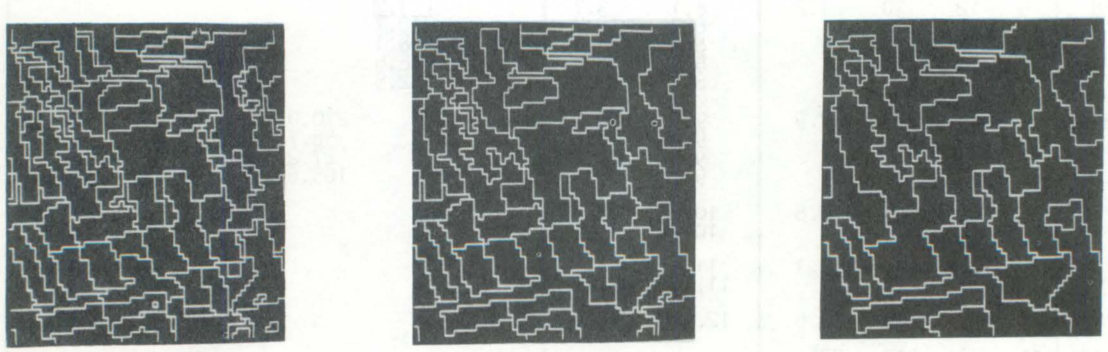


Fig. 4 : Initial partition of fig. 3



a) 100 segments   b) 50 segments   c) 25 segments

Fig. 5 : Final picture partitions