Jean-Marie Beaulieu

[Bea1990a]
Versatile And Efficient Hierarchical Clustering For Picture Segmentation

Author: **Beaulieu** Jean-Marie

Conference: International Geoscience and Remote Sensing Symposium, IGARSS'90

The University of Maryland, College Park,Maryland

May 20-24, 1990, pp. 1663-1663

URL: https://ieeexplore.ieee.org/document/688832

DOI: 10.1109/IGARSS.1990.688832

Versatile And Efficient Hierarchical Clustering For Picture Segmentation, **Beaulieu** Jean-Marie, *International Geoscience and Remote Sensing Symposium, IGARSS'90*, The University of Maryland, College Park,Maryland, May 20-24, 1990, pp. 1663-1663.

Published in: 10th Annual International Symposium on Geoscience and Remote Sensing
Date of Conference: 20-24 May 1990
Date Added to IEEE Xplore: 06 August 2002
DOI: 10.1109/IGARSS.1990.688141

Conference Location: College Park, MD, USA
Publisher: IEEE

VERSATILE AND EFFICIENT HIERARCHICAL CLUSTERING FOR PICTURE SEGMENTATION

Jean-Marie Beaulieu
Département d'Informatique

Université Laval, Pavillon Pouliot,
Québec, (Québec), Canada, G1K-7P4.

Abstract: An efficient implementation of a segmentation algorithm based upon hierarchical clustering is presented. The algorithm starts with an initial picture partition, and at each iteration, the two most similar segments are merged by optimizing a "step-wise criterion". This yields a hierarchical decomposition of the picture. The implementation avoids recalculation by updating the only values that are modified by a segment merger. Moreover, data structures are employed to organize image data and to reduce computing time. The versatility of the algorithm is illustrated by the combination of a constant value approximation criterion with two segment shape criteria.

HIERARCHICAL SEGMENTATION

In hierarchical clustering, the number of clusters is sequentially reduced by merging. At each iteration, the similarity measures $C_{i,j}$ are calculated for all clusters pairs, and the clusters of the pair that minimizes the measure are merged. Beaulieu and Goldberg (IEEE Trans. Pattern. Anal. Mach. Int., Vol. 11, pp. 150-163, 1989) show the importance of hierarchical clustering to solve precisely defined picture segmentation problem, such as 1) finding the partition witch minimizes the approximation error, or 2) making statistical decisions with the minimum probability of error.

In hierarchical segmentation, only adjacent segments could be merged, greatly reducing the number of segment pairs to compare. However, computing time could still be excessive. In the implementation of the algorithm, recalculation are avoided by 1) making explicit the information needed, and 2) updating the only values that are modified by a segment merger.

EFFICIENT IMPLEMENTATION

Data structures are employed to organize image data and to reduce computing time, memory space replacing CPU time. A descriptive structure, D_i, contains pixel value descriptive parameters for segment S_i, e.g. the segment mean and size. The criterion $C_{i,j}$ is calculated directly from D_i and D_j, e.g. mean difference. Moreover, when two segments are merged, the descriptive parameters, D_v, of the new segment are calculated from the old ones.

Each segment have also a neighbour list, B_i. The list is used to scan the segment pairs. For each segment S_i, we examine all its neighbours to find the one with the lowest criterion value, the best neighbour. The lowest criterion value is stored in a criterion list, Cr(i). A tree structure is employed to find the lowest value of the criterion list. When two segments are merged, the new neighbour list, B_v, is produced from the fusion of old neighbour lists. Thus, D_i and B_i contain all the required information about the image, no more reference to the image is needed.

After each merge, we find the best neighbour of the new segment, S_v, and store the criterion value in the criterion list. We then update the tree structure by going from the leaf node v to the top node. The top node corresponds now to the next best segment pair to merge. After a merge, the best neighbour of the surrounding segments should be recalculated. We check and do it only when the segment is selected for the next merge.

At each iteration, the labels of the two merged segments are stored into a merge list. From the complete merge list, a partition with n segments can be quickly obtained, where n is specified by the user. The computing time of the algorithm is mainly proportional to the initial number of segments (which often corresponds to the number of pixels in the picture).

VERSATILE SEGMENTATION

The versatility of the algorithm should also be stressed. The segment similarity criterion, $C_{i,j}$, could be easily adapted to the specific application. In remote sensing, a segment is generally represented by its mean value. Thus, picture segmentation could be regarded as a piecewise constant value approximation problem. It could also be advantageous to consider other aspects, such as the segment shape. Hence, following an approach similar to Brice and Fennema, a first criterion will limit the increase of the segment contour length. It promotes the merge of the segments with a large common boundary. A modification is made to the neighbour list, B_i, to contain also the boundary length. The other shape criterion promotes the formation of compact regions instead of elongated ones. This two shape criteria are combined with the constant approximation criterion in an heuristic manner. Segmentation results will be presented at the conference.

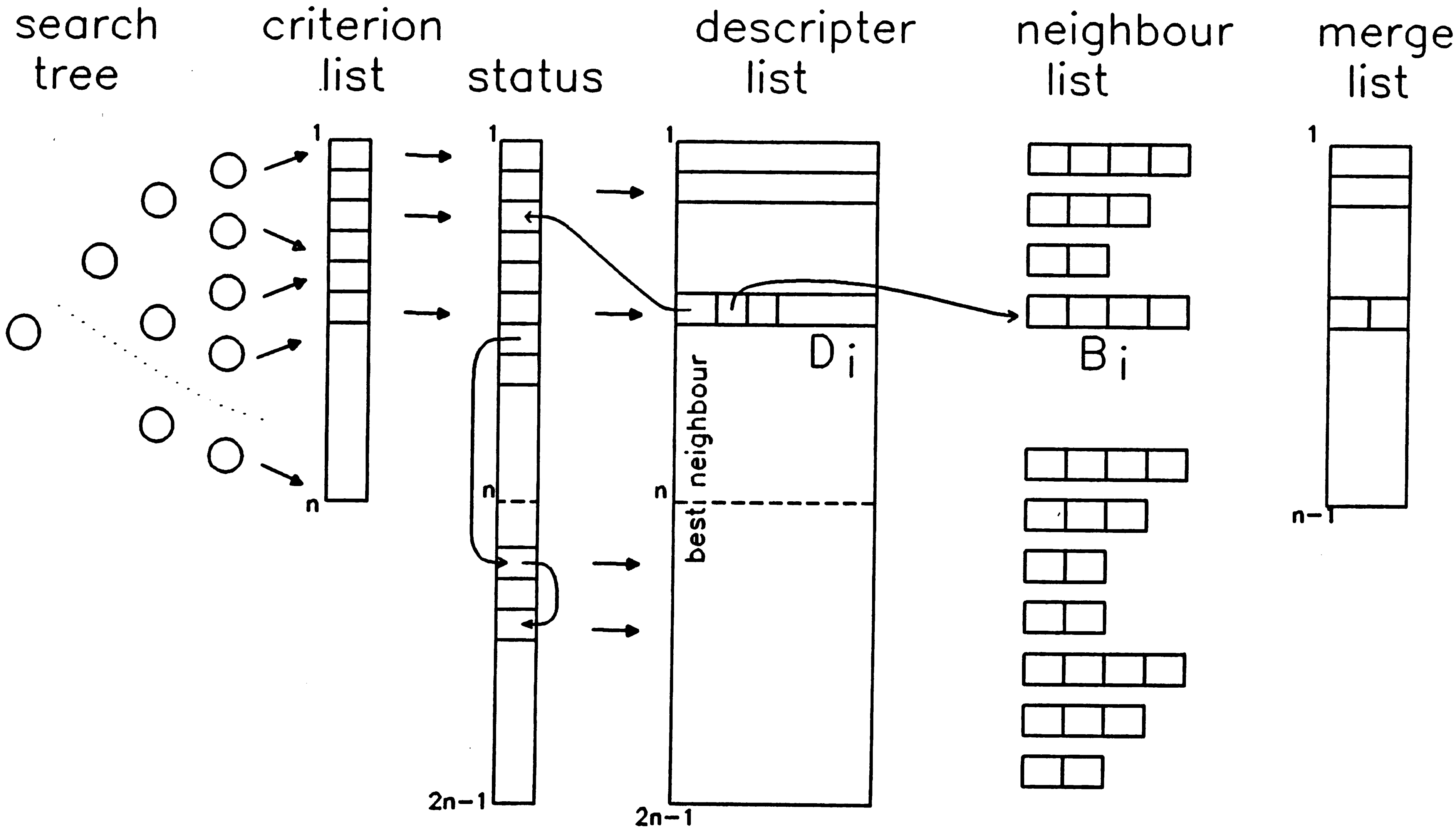
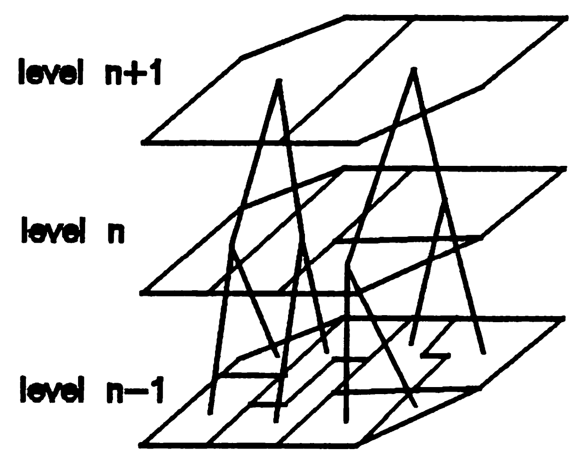


Figure 1: Schematic description of the data structures.

VERSATILE AND EFFICIENT HIERARCHICAL CLUSTERING FOR PICTURE SEGMENTATION

Jean-Marie Beaulieu
Département d'Informatique
Université Laval, Québec, Canada.

A hierarchical segmentation begins with an initial partition P^0 (with N segments), and then sequentially merges these segments.



SEGMENT TREE

STEPWISE OPTIMIZATION

A criterion, corresponding to a mesure of segment similarity, is employed to define which segments to merge.

At each iteration, an optimization process finds the two most similar segments and merges them.

This can be represented by a segment tree, one node per iteration, where only the two most similar segments are merged.

PICTURE APPROXIMATION

Each segment, S_i, is represented by an appromixation function, $r_i(x,y)$.

The approximation error is defined as

$$H(S_i) = \sum_{S_i} [\ f(x,y) - r_i(x,y)\ ]^2$$

The goal is to find the picture segmentation that produce the lowest overall approximation error.

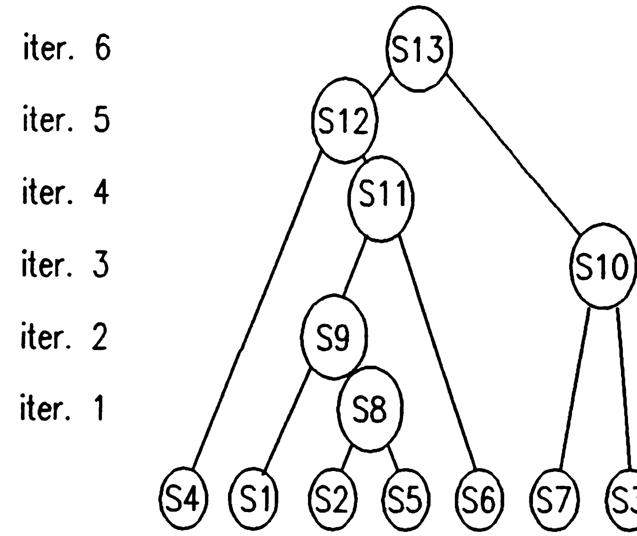
In hierarchical segmentation, this results in sequentially merging the segments that produce the smallest increase in the approximation error.

Thus, the stepwise criterion is

$$C_{i,j} = H(S_i \cup S_j) - H(S_i) - H(S_j)$$

For constant value approximation, this is the Ward criterion:

$$C_{i,j} = \frac{N_i \times N_j}{N_i + N_j} (\ \mu_i - \mu_j\ )^2$$



In hierarchical picture segmentation, only adjacent segments could be merged, greatly reducing the number of segment pairs to compare. However, computing time could still be excessive. In the implementation of the algorithm, recalculation are avoided by 1) making explicit the information needed, and 2) updating the only values that are modified by the segment merger.

The variables involved in the algorithm are:

1) B_i, the list of the segments S_j adjacent to S_i,
2) D_i, the parameters that describe the segment S_i,
3) $C_{i,j}$, the cost of merging segment S_i with S_j, where S_j is contained in B_i.

Initialisation:

1) $P^0 = \{ S_1, S_2, \ldots S_n \}$ (initial partition)

2) $k = 0$ and $m = n$.

3) calculate D_i and B_i for $\forall S_i \in P^0$.

4) calculate $CS = \{ C_{i,j} \mid S_i \in B_i$ and $i > j \}$.

Merge the two most similar segments:

5) $k = k + 1$ and $m = m + 1$.

6) find $C_{u,v} = \underset{C_{i,j} \in CS}{\text{Minimum}} (C_{i,j})$

7) $P^k = (P^{k-1} \cup \{ S_m \}) \cap \overline{\{ S_u, S_v \}}$.

8) calculate D_m from D_u and D_v.

9) $B_m = (B_u \cup B_v) \cap \overline{\{ S_u, S_v \}}$.

Update criterion and neighbor sets:

10) $\forall S_j \in B_m$, $B_j = (B_j \cup \{ S_m \}) \cap \overline{\{ S_u, S_v \}}$.

11) $CS = (CS \cup \{ C_{m,j} \mid S_j \in B_m \}) \cap \overline{\{ C_{i,j} \mid i,j = u \text{ or } v \}}$.

Stopping condition:

12) if continue merging then goto step 5.
else stop.

EFFICIENT IMPLEMENTATION

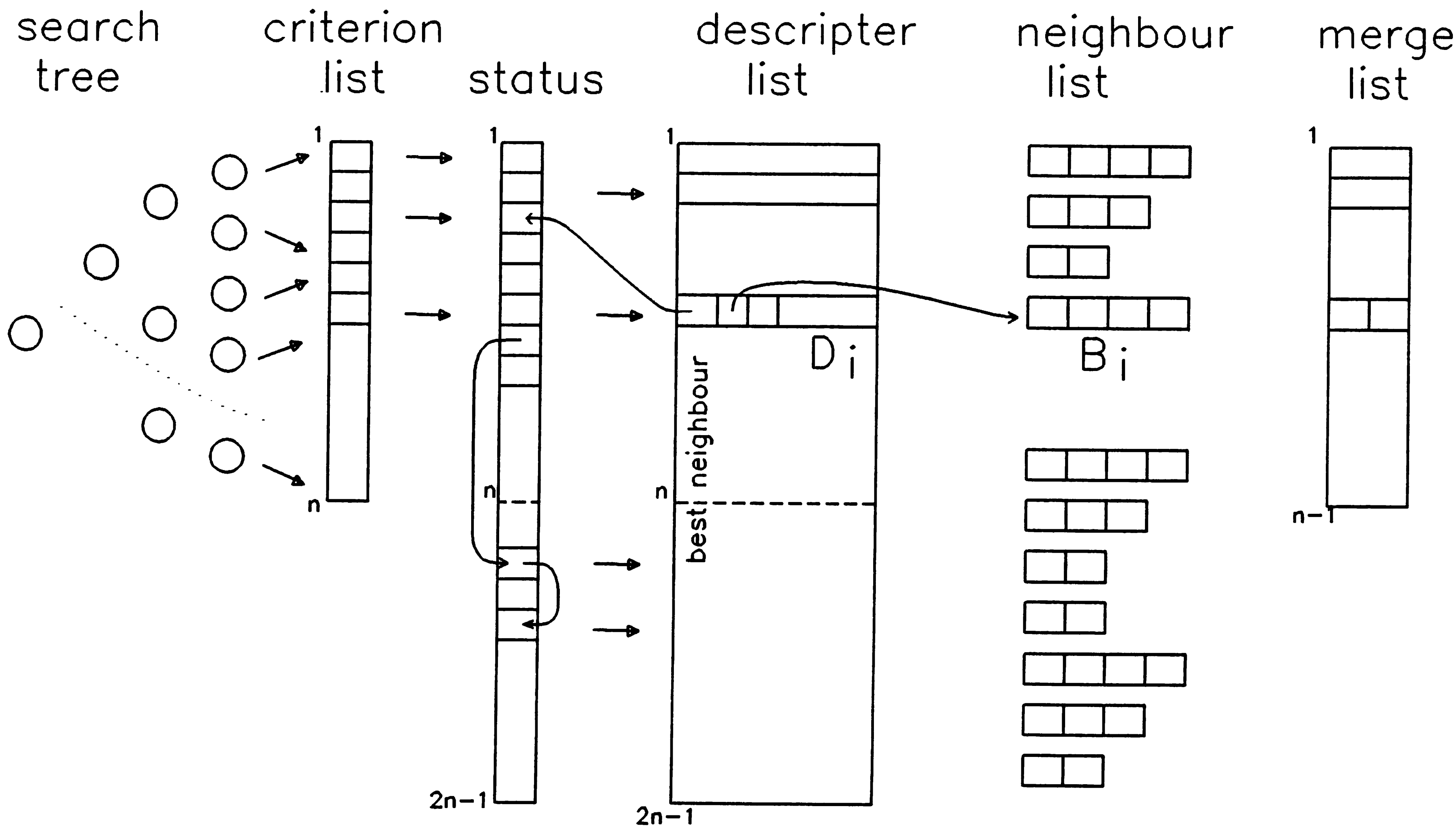
The sequence of segment merge is stored in a file, the merge list, and contains the required information to build the segment tree. Thus, a first program, cafus, does the segment merging and builds this merge list. A second program, anafus, cuts the segment tree to produce an image partition with the specified number of segments.

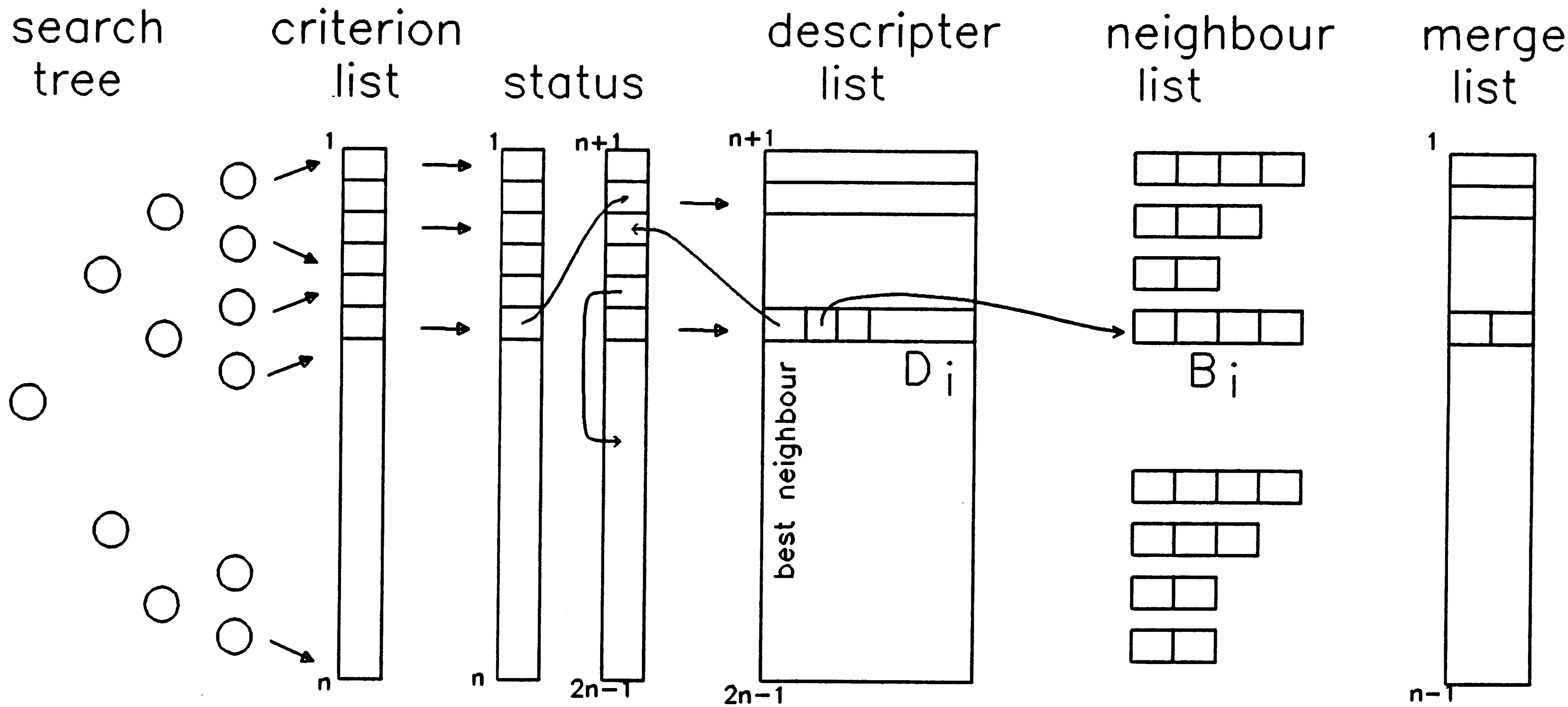
cafus program:

```
    get image and initial partition
    init_descripter_list()
    init_neighbour_list()
    init_criterion_list_and_status()
              /* witch call calcul_criterion() */
    init_search_tree()

    while( continue_merging )

         find_best_segment_pair()
         write_to_merge_list()
         new_segment = new_segment + 1
         merge_segment_descripter()
         merge_neighbour_list()
         find_best_neighbour()
              /* witch call calcul_criterion() */
         update_search_tree()
         update status information
```


search tree
criterion list
status
descripter list
neighbour list
merge list
1
n
1
n
2n−1
1
n
2n−1
best neighbour
D_i
B_i
1
n−1


search tree
criterion list
status
descripter list
neighbour list
merge list
1
n
1
n
n+1
2n-1
n+1
2n-1
best neighbour
D_i
B_i
1
n-1

A <u>descriptive structure</u>, D_i, contains pixel value descriptive parameters for segment S_i, e.g. the segment mean and size. The criterion $C_{i,j}$ is calculated directly from D_i and D_j, e.g. mean difference. Moreover, when two segments are merged, the descriptive parameters, D_v, of the new segment are calculated from the old ones.

Each segment have also a <u>neighbour list</u>, B_i. The list is used to scan the segment pairs. For each segment S_i, we examine all its neighbours to find the one with the lowest criterion value, the best neighbour. When two segments are merged, the new neighbour list, B_v, is produced from the fusion of old neighbour lists.

The criterion value for the best neighbour is stored in a <u>criterion list</u>, Cr(i). A tree structure (<u>search tree</u>) is employed to find the lowest value of the criterion list.

After each merge, we find the best neighbour of the new segment, S_v, and store the criterion value in the criterion list. We then update the tree structure by going from the leaf node v to the top node. The top node corresponds now to the next best segment pair to merge.

After a merge, the best neighbour of the surrounding segments should be recalculated. We check and do it only when the segment is selected for the next merge.

At each iteration, the labels of the two merged segments are stored into a <u>merge list</u>. From the complete merge list, a partition with n segments can be quickly obtained, where n is specified by the user.

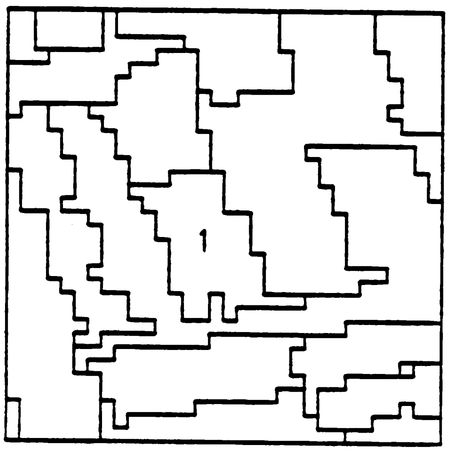
The computing time of the algorithm is mainly proportional to the initial number of segments (which often corresponds to the number of pixels in the picture).

VERSATILE SEGMENTATION

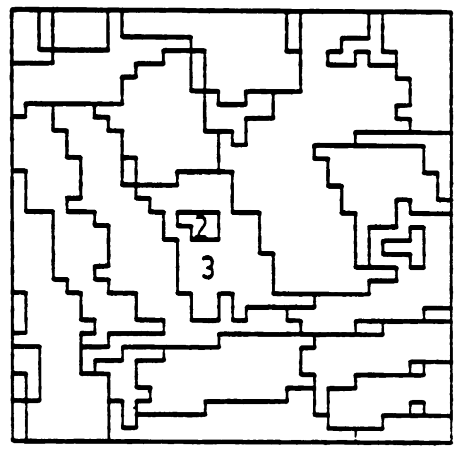
The segment similarity criterion, $C_{i,j}$, could be easily adapted to the specific application. The fonctions, that calcul the descriptive parameters and the stepwise criterion, are grouped into a file. Therefore, only this file should be replaced in order to modify the segmentation criterion.

LANDSAT PICTURE

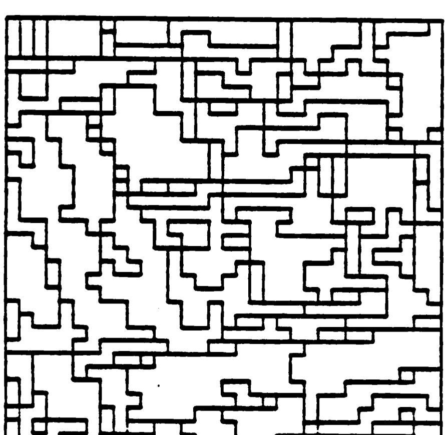
A Landsat satellite picture (32x32 pixels, 0.8-1.1 um) of an agricultural area near Melfort in Saskatchewan, Canada, is segmented into 18, 36 118 and 212 regions, in order to show the hierarchical aspect of the algorithm.



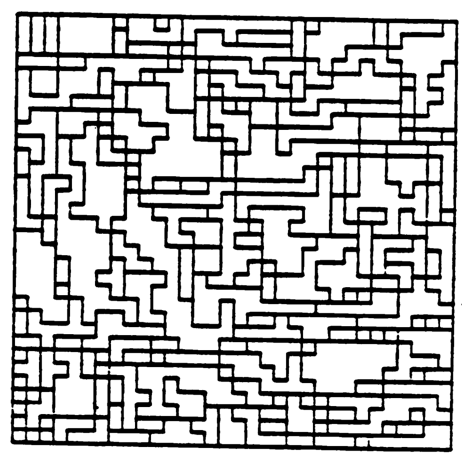
a) 18 segments



b) 36 segments



c) 118 segments

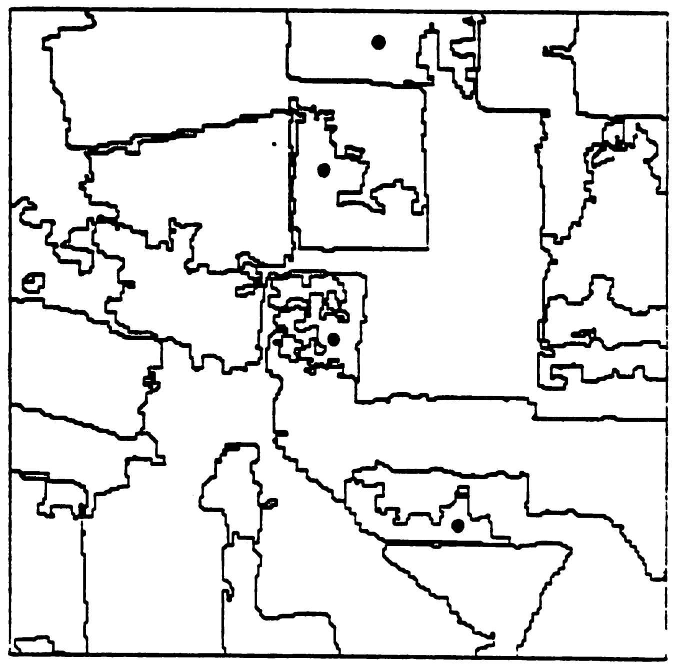


d) 212 segments

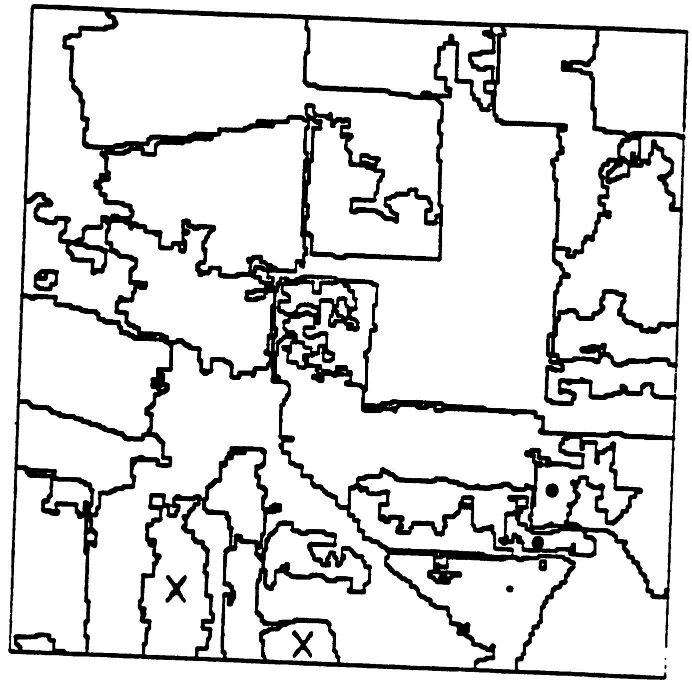
SAR PICTURE

A composite criterion is employed to segment an airborne X-band radar picture with vertical-vertical polarization, 256x256 pixels, and a 5 meter resolution. The picture covers a 1.28 km x 1.28 km area near Makofen, in the Federal Republic of Germany.

a) 25 segments



b) 37 segments



c) 86 segments