Harvey A. Cohen and Cung H. Duong, *Gray-scale Image Segmentation Using a Parallel Graph-Theoretic Algorithm*, Australian Pattern Recognition Society, Segment'96, Sydney Dec 13, 1996, pp 105-110.

Gray-scale Image Segmentation Using a Parallel Graph-Theoretic Algorithm

Harvey A. Cohen
Cung Hy Duong
Computer Science and Computer Engineering
La Trobe University
Bundoora 3083
Email: H.Cohen@latrobe.edu.au FAX: 03 9479 3060

Abstract

Gray-scale image segmentation involves attributing a common region label to all pixels of the same gray-scale that are (digitally) connected. A new algorithm for segmentation is described that involves labelling each pixel with an index label on a forward raster pass, while at the same time constructing a minimal adjacency table (MAT) for the image. For parallel implementation, this algorithm is applied separately to sub-images, with overlapping boundaries, to yield labelled sub-images with MATs, together with adjacency tables for each overlapping boundary. The algorithm has been implemented using PVM on a local network.

1. INTRODUCTION

Gray-scale image segmentation involves attributing a common region label to all pixels of the same gray-scale that are (digitally) connected. The classic Rosenfeld-Pfaltz algorithm involving two passes through an image can be readily extended to gray-scale, but is not simply parallelisable. The problem with parallelisation is that for geometric decomposition of an image into sub-images not only are some regions broken into two, with some pixels in adjoining regions, but within a given region the pixels in a (whole-image) connected region may be connected only via chains of connected pixels in adjoining regions.

The first author developed a one-pass image segmentation algorithm[..] that involved

being determined as for the Roenfeld-Pfaltz first pass. However, in the new algorithm, during this pass a minimal adjacency array is produced and progressively updated during the pass. Consequently, at the end of the pass, all adjacency relations are known, and the segmentation is complete. This segmentation algorithm was also extended to include connectivity of gray-scale regions.

In this paper a graph-theoretic algorithm is described for segmentation of binary or gray-scale images that is parallellizable. The algorithm is an

The key idea of the Rosenfeld-Pfaltz algorithm is the assigning to pixels of a "provisional" region label during a raster-scan, new labels being assigned to pixels not adjacent, to "North" or "West" of previously labelled regions. In the Rosenfelf-Pfaltz Segmentation Algorithm, labels are corrected during a so-called backwards pass, where starting from bottom right hand image corner, the image is reverse raster scanned, and labels of adjacent pixels to "South" and "East" used to correct the labelling.

A one-pass image segmentation algorithm was developed by the first author which involves the use of a Minimal Adjacency Array, similar to the array used in the well-known Kruskal Algorithm for Minimum Spanning Trees. In this algorithm, a forward pass through an image is used to give "provisional" region labels to each pixel, the labels more other regions.
extension of Cohen's One Pass Algorithm, but involves further graph theoretic operations.

The algorithm has been implemented in C, using PVM [8], and experimental data is presented over a local network of SUN Workstations.

1 Kruskal Adjacency Table

The standard graph-theoretic adjacency matrix for a graph has elements that indicate whether the corresponding vertices are connected. In the absence of loops and multiple edge,

if $A_{ij} = 0$ V_i and V_j are not connected if A_{ij} = 1 V_i and V_j are connected

In developing his minimal spanning tree algorithm, Kruskal [9][10] introduced what we call a Minimal Adjacency Table, which is computationally more useful way of representing connection relations. In a minimal adjacency K each entry indicates the smallest vertex number to which vertex is connected:

eg, if K[j] = [i]

then i =<j, and i is the vertex of smallest numeric label to which vertex V_j is connected. Of course, if K[i] =i, then i is not connected to any other vertex.
In Kruskal's minimum spanning tree algorithm, edges are progressively added to a graph, and the K table is updated for each addition..
In the algorithms discussed here, as adjacency is determined of various regions, the table is progressively updated.

1.1 UPDATE ALGORITHM for Minimum Adjacency Table

In Kruskal's minimum spanning tree algorithm, the minimal adjacency table K is initialised with K[i] =I for all vertices of a graph bare of edges, and edges are progressively added. The Update Algorithm, specifying the change in K when the edge connecting V_i and V_j is inserted, is:

```
if   m = K[i] ;  n = K[j]
        if m == n   do nothing(*)
else
        m_max = max (m,n) and
m_min = min (m,n).
        for ( region =1; region <
region =< region_max; region ++)
                if K[region] ==
m_max then K[region] = m_min
```

where the number of vertices is called region_max, consistent with algorithm below.
(*) In minimum spanning tree algorithm, this edge is not added to the edge list.

2 THE ONE PASS SEGMENTATION ALGORITHM

The basic idea of the one-pass algorithm is to use a graph whose vertices correspond to possibly connected regions in an image. When two regions are found to be connected, then a edge is inserted in the graph connecting the corresponding verticies.

Using E[i][j] to denote image pixels, and R[i][j] the corresponding region labels the algorithm is as follows: (the algorithm is written in sufficient generality so as to include both the case of binary images and gray-scale images. Pascal code formulation was presented in [6].)

```
        /*Initialisation */
        no_of_regions =0;

/* raster scan through an image - pixels "outside" an image treated as "background" pixels */

        If   E[i][j] == BACKGROUND   R [i][j]  =
INFINITY
        else
        {
        If  E[i][j]  is adjacent to both NORTH and
WEST pixels:
        {
        R[i][j]     =  min  {  m  =R[i-1]  [j],  n  =
R[i][j+1]}          /* Compare Rosenfeld-Pfaltz) */
        /* Update Maximum Adjacency Table */
                if  m != n   then
                        {
                        m_max = max (m,n) and
m_min = min (m,n)
                        for ( region =1; region <
region =< region_max; region ++)
                                if K[region] ==
m_max then K[region] = m_min
                        }
        if  E[i][j] is adjacent ONLY to NORTH
                R[i][j] = R[i-1][j]
        if  E[I][J] is adjacent only to west
                R[i][j]  = R[i][j-1]
        if   E[i][j] has no adjacent pixel to west or
north
                R[i][j]    =  ++ no_of_regions
        }
```

At end of raster scan all entries in region R have been made, and minimal adjacency table K has been finalised.

3 THE PARALLEL SEGMENTATION ALGORITHM

In this section, the new parallel algorithm for image segmentation is described as an extension of the single-image algorithm of the previous section.

Initialization stage of the parallel algorithm involves splitting the image into regions with overlapping common boundaries. -- the overlap being of one pixel width, as indicated in Fig 1:

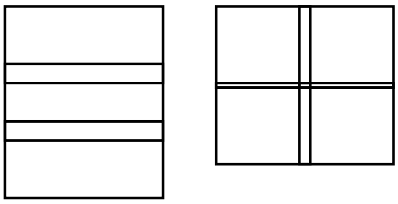

Fig 1. Image subdivision styles. To left, sub-images with overlapping “horizontal”boundaries, as used in experiment reported. To right, sub-division into regions with both horizontal and vertical boundaries.

Following this initialisation, sub-images are sent to daughter processors to perform Step 1 of the new algorithm.

Step 1 [Parallel Step]

The processor α performs the One Pass Segmentation Algorithm for sub-region α to determine a region label image R[α] and Minimal Adjacency Table [α] and no_of_regions[α] for sub-images.

Step 2
The sub-image label

The minimal Adjacency Table for each sub_image is returned to the processor overlord.

Step 2 [Graph-Theoretic Merge]

Reassign consecutive region numbers to table entries in each minimal adjacency table.
The tables now constitute a (partitioned) minimal adjacency table for the whole image.
Scan through each overlap region: for if pixel p is in sub-images α and β,
then if K[α] != K[β] , update the table.

Segmentation is now complete. However, as an optional further step, the relevant partitions of each table can be sent to the corresponding processor, and the region lables given the final form in the Region label Images.

4 CONCLUSION

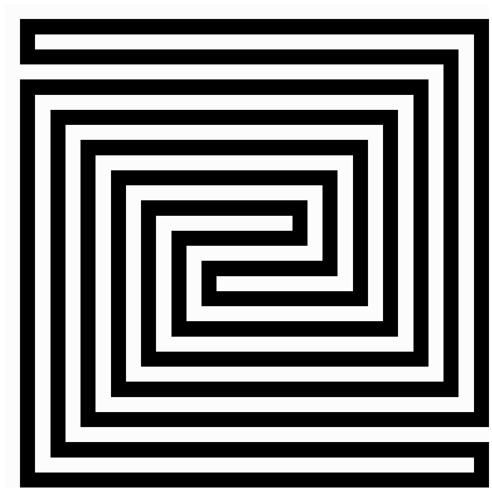

Fig 1: Perceptron -II Image
The basic image is a 32x32 thumbnail. Experiments have been performed with scaling via replication to 512x512. There are two connected regions plus background in this image.

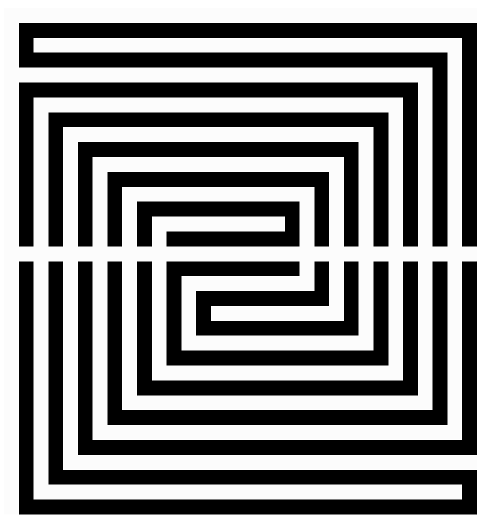

Fig 1: Perceptron -II Image split into two sub-images, with replication of the boundary line. Both upper and lower sub-image have 6 connected regions, plus background.

Data for 512x512 image is as follows:

No of processors	Duration of Parent secs	Duration -child processors secs
1	154.63	93.074
2	115.77	89.24, 65.62
3	90.49	60.66,65.38,58.48
4	81.51	60.32, 61.24,51.60,59.37

The speed-up is relatively modest for two reasons
(a) communication overhead
(b) the increasing LUT transfer and manipulation required.

It is proposed to modify the algorithm, so that
Note that the size of the child process is steadily decreasing, in the ratio 1: 0.5:

There is a trade-off effect -- as ultimately more effort0.3

A parallel segmentation algorithm has been developed, implemented, and tested for test images. The only comparable algorithm known is the parallel thinning algorithm of E. Trichina & A. Kolesnikov [7]; for more than two sub-images, their algorithm requires mutliple passes of Rosenfeld-Pfaltz operators (both forward and backwards) over all the sub-images involved until convergence.

The real utility of this algorithm is its use in convenient combination with the computation of the additive properties of connected regions -- area, first and second moments. - such as used in the well-known SRI parameters. [11] [12]

4 IMPLEMENTATION

The implementaion was performed using a Single Program Multiple Data programming model, using PVM. PVM, real utility of this algorithm is its use in

5 REFERENCES

[1] A. Rosenfeld and D. Pfaltz, "Sequential Operations in Digital Picture Processing" J.A.C.M. (1966) **13** No 4 pp 471-494.

[2] A. Rosenfled and D. Pfaltz, "Distance Transforms on Digital Pictures", CVGIP (Computer Graphics, Vision and Image Processing) Vol 34, 1968, pp 33-61.

[3] G. Borgefors, "Distance Transforms in Digital Image Processing" in CGVIP 34 (1986) pp 344-371.

[4] J. You and Harvey A. Cohen, "Classification and Segmentation of Rotated and Scaled Textured Images Using Texture Tuned Masks", Int'l Journal of Pattern Recognition, Vol 26, No 2, pp 245-258, 1993.

[5] Harvey A. Cohen and You Jia, "A multi-scale texture classifier based on multi-resolution "tuned" masks", Pattern Recognition Letters, Vol 13 No 8 August 1992, 599-603.

[6] Harvey A. Cohen, "One-pass gray-scale image segmentation", Proc DICTA-93, Sydney, December 1993, pp 572-578

[7] E. Trichina & A. Kolesnikov, "Parallel Implementation of Width-Independent Fast Thinning Algorithm", Proc First Int'l Conf. on Visual Information Systems, Visual-96, Melbourne, January, 1996, pp 384-393.

[8] W. Zwaenepoel et al, "Shared memory computing on networks of workstations", Technical Report, Rice University, 1994.

[9] A. Geist, et al, "PVM-3 user's guide and Reference manual", ONRL/TM-12187, May 1994.

[9] P.F. Dierker and W.L. Voxman, "Discrete Mathematics" Harcout, Brace, Jovanich, San Diego, 1986 pp 113-122.

[10] R. Skvarcius and W.B. Robinson, "Discrete Mathematics with Computer Science Applications", Benjamin & Cummings, Menlo Park, Calif, 1986. pp185-187

[11] Automatix Inc, "Autovision RAIL Reference Manual Rev 5.00" undated.

[12] S. Dudani, K.Breeding, and R.. McGhee, "Aircraft Identification by Moment Invariants", IEEE Trans Computers C-26, no 1, (Jan 1977) pp 39-45.