Hierarchical Segmentation of Remotely Sensed Imagery Data using Massively Parallel GNU-LINUX Software

James C. Tilton
NASA's Goddard Space Flight Center
Mail Code 935
Greenbelt, MD 20771 USA
Telephone: (301) 286-9510
E-Mail: James.C.Tilton@nasa.gov

What is a Hierarchical Image Segmentation?

A set of image segmentations that

i. consist of segmentations at different levels of detail, in which
ii. the coarser segmentations can be produced from merges of regions from the finer segmentations, and
iii. the region boundaries are maintained at the full image spatial resolution.

Presentation Outline

I. Image Segmentation Overview

II. Hierarchical Segmentation (HSEG)

- A. Overview
- B. Example and Demo
- C. Algorithmic Details

III. Recursive Hierarchical Segmentation (RHSEG)

IV. Parallel Implementation of RHSEG

V. Analysis of MODIS Images of the Iberian Peninsula

Image Segmentation Overview

Let $\boldsymbol{X}$ be a two-dimensional array representing an image. The *optimum* segmentation of $\boldsymbol{X}$ into N regions can be defined as a partition of $\boldsymbol{X}$ into disjoint subsets $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$, such that

1) $\bigcup_{i=1}^{N} \boldsymbol{X}_i = \boldsymbol{X}$,

2) $\boldsymbol{X}_i$, $i = 1, 2, \ldots, N$ is connected,

3) $\sum_{i=1}^{N} \mathrm{G}(\boldsymbol{X}_i) =$ *MINIMUM* over all partitions into N regions, and

4) $\mathrm{G}(\boldsymbol{X}_i \cup \boldsymbol{X}_j) > \mathrm{G}(\boldsymbol{X}_i) + \mathrm{G}(\boldsymbol{X}_j)$ for $i \neq j$, where $\boldsymbol{X}_\mathrm{i}$ and $\boldsymbol{X}_\mathrm{j}$ are adjacent.

Image Segmentation Overview (cont'd)

G($\boldsymbol{X}_i$) is a function that assigns a cost to partition $\boldsymbol{X}_i$ depending on the image data values in $\boldsymbol{X}_i$.

G($\boldsymbol{X}_i$) can be any function measuring the dissimilarity of the region feature values of $\boldsymbol{X}_i$ from the image data values in $\boldsymbol{X}_i$. For example, a cost function based on the vector 2-norm is:

$$\mathrm{G}(\boldsymbol{X}_i) = \left(\frac{1}{n_i * B} \sum_{x_k \in X_i} \left\| x_k - \overline{x}_i \right\|_2 \right),$$

where

$$\left\| x_k - \overline{x}_i \right\|_2 = \left(\sum_{b=1}^{B} \left(x_{kb} - \overline{x}_{ib} \right)^2 \right)^{1/2}$$

where n_i is the number of pixels in region i, and $\overline{x}_i$ is the mean vector for region i. The division by B, the number of spectral bands, could be ignored as a constant factor.

Image Segmentation Overview (cont'd)

Directly determining the mathematically optimal image segmentation for a given level of detail or number of regions is not possible due to computational constraints, since one would have to search over all possible partitions of N regions.

Some indirect approach is necessary.

Image Segmentation Overview (cont'd)

Approach	Difficulty with the Approach
Spectral Feature Clustering	Spatial Information not Utilized
Edge Detection	No Guarantee of Closed Connected Regions
Region Growing	Process is Computationally Intensive and Global Convergence is Difficult

Image Segmentation Overview (cont'd)

I prefer region growing despite the difficulties with computational intensity and global convergence because it utilizes spatial information and can provide closed connected regions.

Hierarchical Segmentation (HSEG) addresses the global convergence problem by producing a hierarchy of segmentation results rather than one global result.

The recursive formulation of HSEG, with its effective parallel implementation, largely solves the computational difficulties.
(more later)

Image Segmentation Overview (cont'd)

The most widely used definition of image segmentation by region growing follows from a definition offered nearly 30 years ago by Horowitz and Pavlidis:

Reference: S. L. Horowitz and T. Pavlidis, "Picture segmentation by a directed split-and-merge procedure," *Proceedings of the Second International Joint Conference on Pattern Recognition*, pp. 424-433, 1974.

Image Segmentation Overview (cont'd)

Let $\boldsymbol{X}$ be a two-dimensional array representing an image. A segmentation of $\boldsymbol{X}$ can be defined as a partition of $\boldsymbol{X}$ into disjoint subsets $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$, such that

1) $\bigcup_{i=1}^{N} \boldsymbol{X}_i = \boldsymbol{X}$

2) $\boldsymbol{X}_i$, $I = 1, 2, \ldots, N$ is connected

3) $P(\boldsymbol{X}_i)$ = TRUE for $i = 1, 2, \ldots, N$ and

4) $P(\boldsymbol{X}_i \bigcup \boldsymbol{X}_j)$ = FALSE for $i \neq j$, where $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are adjacent.

Image Segmentation Overview (cont'd)

P($\boldsymbol{X}_i$) is a logical predicate that assigns the value TRUE or FALSE to $\boldsymbol{X}_i$, depending on the image data values in $\boldsymbol{X}_i$.

For example, let $P(\boldsymbol{X}_i) = \left(\frac{\left\| x_j - \bar{x}_i \right\|_2}{B} \leq T \, \forall x_j \in \boldsymbol{X}_i \right)$,

where B is the number of spectral bands, $\bar{x}_i$ is the mean vector for region i, x_j is a pixel vector in region i, and T is a threshold.

Call this the "classical" definition of image segmentation by region growing.

Image Segmentation Overview (cont'd)

This definition of region growing is usually implemented taking a single pass through the image data, growing regions until the predicate can no longer be satisfied.

No attempt is made to find the "optimal" segmentation. Any segmentation that satisfies the logical predicate is acceptable.

(In my implementation, if the pixel being considered satisfies the logical predicate for more than one region, I choose the region for which the predicate is satisfied by the largest margin.)

Image Segmentation Overview (cont'd)

Example : A 512x512 pixel section of Landsat ETM+ data
obtained on May 28, 1999 over Washington, DC, U.S.A.

"Classical" region growing with logical predicate based on the vector 2-norm with $T = 0.350$.

Produced 4366 regions with global dissimilarity criterion value = 0.1304. (Took 4 minutes on a 1.2 GHz computer.)

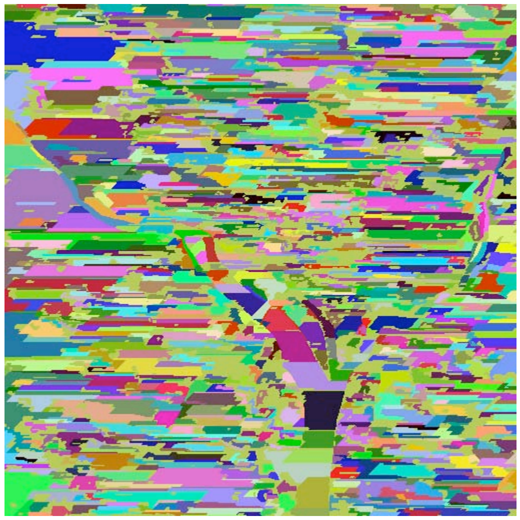


Image Segmentation Overview (cont'd)

A better result can be obtained by using a different criterion for region merging. As the image is scanned, merge a pixel x_j into the adjacent region $\boldsymbol{X}_i$ if $G(x_j, \boldsymbol{X}_i) < G(x_j, \boldsymbol{X}_k)$ $\forall$ regions adjacent to pixel x_j ($i \neq k$), and $G(x_j, \boldsymbol{X}_i) < T$. An example the cost function $G(x_j, \boldsymbol{X}_i)$, based on the vector 2-norm is:

$$G(\boldsymbol{X}_i) = \frac{\left\| x_j - \bar{x}_i \right\|_2}{B}.$$

where B is the number of spectral bands, and $\bar{x}_i$ is the mean vector for region i. Otherwise, start a new region with pixel x_j.

Image Segmentation Overview (cont'd)

Example: A 512x512 pixel section of Landsat ETM+ data Obtained on May 28, 1999 over Washington, DC, U.S.A.

"Classical" region growing with cost function based on the vector 2-norm with $T = 0.310$.

Produced 4262 regions with global dissimilarity criterion value = 0.1088. (Took 4 seconds on a 1.2 GHz computer.)

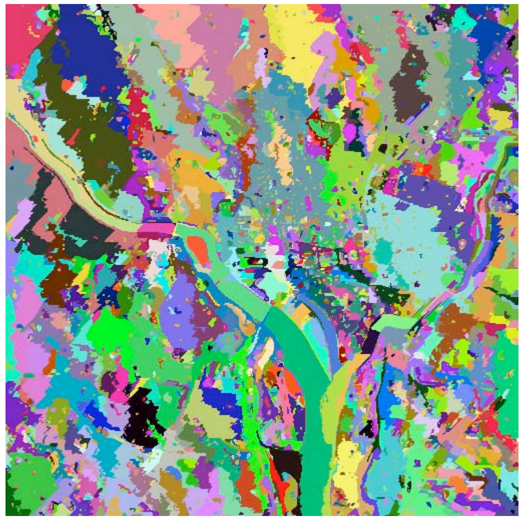


Image Segmentation Overview (cont'd)

Hierarchical Step-Wise Optimal Segmentation (Beaulieu and Goldberg, 1989) or Iterative Parallel Region Growing (Tilton and Cox, 1983) is essentially a compromise between “classical” image segmentation and the ideal image segmentation.

Instead of attempting to find the overall ideal image segmentation, these approaches find the best *N* region segmentation, given a pre-specified *N*+1 region segmentation.

Image Segmentation Overview **(cont'd)**

References:

J.-M. Beaulieu and M. Goldberg, "Hierarchy in picture segmentation: A stepwise optimization approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 2, pp. 150-163, Feb. 1989.

J. C. Tilton and S. C. Cox, "Segmentation of remotely sensed data using parallel region growing," *Digest of the 1983 International Geoscience and Remote Sensing Symposium*, San Francisco, Ca, pp. 9.1-9.6, August 31 – September 2, 1983.

Image Segmentation Overview **(cont'd)**

HSEG is the same as HSWO, except that HSEG optionally alternates merges of spatially adjacent regions with merges spatially non-adjacent regions.

HSEG also offers a choice of cost functions:

- Currently implemented are cost functions based on vector norms (1-norm, 2-norm and infinity-norm), and mean squared error.
- Other cost functions can be implemented (e.g. statistical hypothesis testing, constraining image entropy, normalized vector distance, and others).

Image Segmentation Overview (cont'd)

Let $\boldsymbol{X}$ be a two-dimensional array representing an image. A segmentation of $\boldsymbol{X}$ into N regions is a partition of $\boldsymbol{X}$ into N disjoint subsets $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{N-1}, \boldsymbol{X}_N$:

1) $\bigcup_{i=1}^{N} X_i = X$, and

2) $\boldsymbol{X}_i$, $I = 1, 2, \ldots, N$ is connected.

Let $G(\boldsymbol{X}_i)$ be a function that assigns a cost to partition $\boldsymbol{X}_i$. Reorder the partition $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{N-1}, \boldsymbol{X}_N$, such that (next slide)

Image Segmentation Overview (cont'd)

$G(\boldsymbol{X}_{N-1} \cup \boldsymbol{X}_N) \leq G(\boldsymbol{X}_i \cup \boldsymbol{X}_j)$ for all $i \neq j$ where $\boldsymbol{X}_{N-1}$ and $\boldsymbol{X}_N$ are adjacent, and $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are adjacent. The optimal segmentation of $\boldsymbol{X}$ into N-1 regions (relative to the N region segmentation) is the partition

$$\boldsymbol{X}'_1, \boldsymbol{X}'_2, \ldots, \boldsymbol{X}'_{N-1}$$

where $\boldsymbol{X}'_i = \boldsymbol{X}_i$ for $i = 1, 2, \ldots, N$-2 and

$$\boldsymbol{X}'_{N-1} = \boldsymbol{X}_{N-1} \cup \boldsymbol{X}_N.$$

$G(X_i)$ is the same cost function introduced earlier in the discussion of the optimum image segmentation.

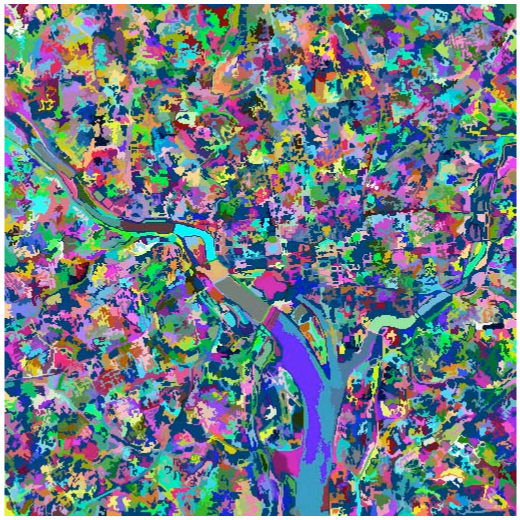
Image Segmentation Overview (cont'd)

Example: A 512x512 pixel section of Landsat ETM+ data obtained on May 28, 1999 over Washington, DC, U.S.A.

With convergence set at 4096 regions, HSWO produced a global dissimilarity criterion value = 0.0865. (Took 23 minutes on a 16 CPU Beowulf cluster.)

With convergence set at 1024 regions, HSWO produced a global dissimilarity criterion value = 0.1081. (Took 2 minutes on a 64 CPU Beowulf cluster.)

4096 regions



For Presentation to the Spanish Remote Sensing Conference, Sept. 18, 2003. 27

1024 regions

For Presentation to the Spanish Remote Sensing Conference, Sept. 18, 2003. 28

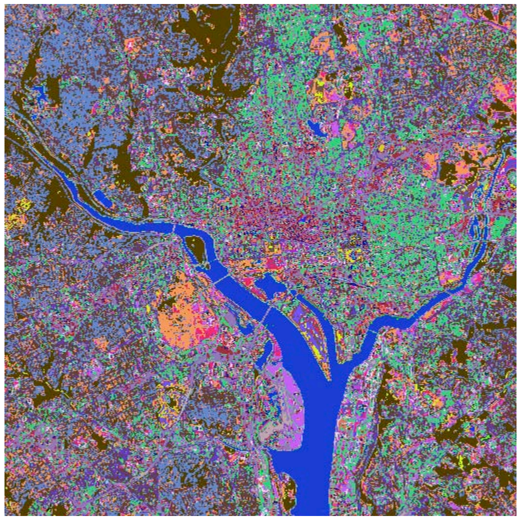
Hierarchical Segmentation (HSEG)

Addition of option for merging spatially non-adjacent regions:

The HSEG algorithm optionally alternates merging spatially adjacent regions with the merging of spatially non-adjacent regions. The merging threshold for the spatially non-adjacent merges is set equal to the dissimilarity of the previous spatially adjacent merge, time a weighting factor (*spclust_wght* ≤ 1).

With *spclust_wght* = 0.9 and convergence set at 64 regions, HSEG produced a global dissimilarity criterion value = 0.0741. (Took 45 seconds on a 64 CPU Beowulf cluster.)

64 regions



Hierarchical Segmentation (cont'd)

HSEG must address the problem of global convergence, i. e. when to stop growing regions?

HSEG monitors a global dissimilarity criterion (cost function) and outputs a hierarchical set of image segmentations based on the dynamic behavior of this criterion.

Hierarchical Segmentation (cont'd)

For example, let $G_{global}(X) = \frac{1}{N}\left[\sum_{X_i in X}\left(\sum_{x_j \in X_i} \left\| x_j - \bar{x}_i \right\|_2\right)\right]$,

where n_i is the number of pixels in region i, $\bar{x}_i$ is the mean vector for region i, and N is the total number of pixels in the image.

In HSEG, the segmentation result at iteration i-1 is saved as a hierarchical segmentation output when

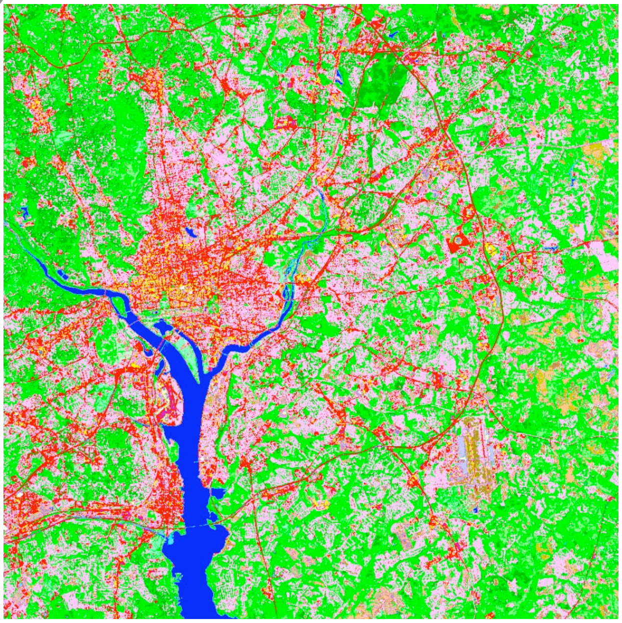
$$\frac{G^{i}_{global}}{G^{i-1}_{global}} > threshold.$$

Example: Landsat Thematic Mapper data

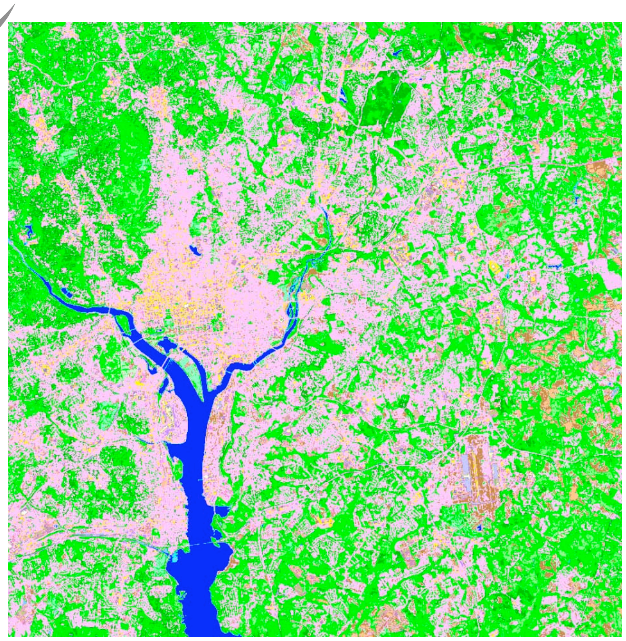
A 1024x1024 pixel section of Landsat ETM+ data obtained on May 28, 1999 over Washington, DC, U.S.A.

Parameters: *spclust_wght* = 0.9, *spatial_wght* =1.0, *min_nregions* = 256, *chk_nregions* = 64.

Produced 15 hierarchical segmentation levels in 2 minutes and 33 seconds on a 64 CPU 1.2 GHz clock Beowulf Cluster.

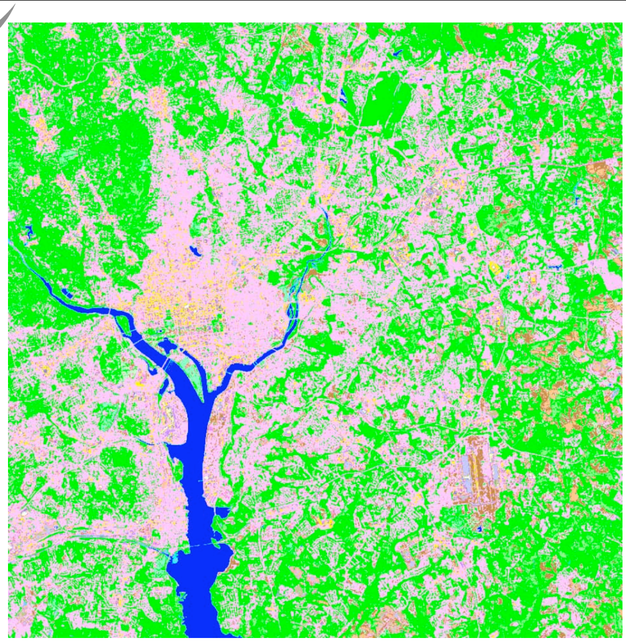


h_level	cvratio
0	-
1	1.017
2	1.065
3	1.101
4	1.080



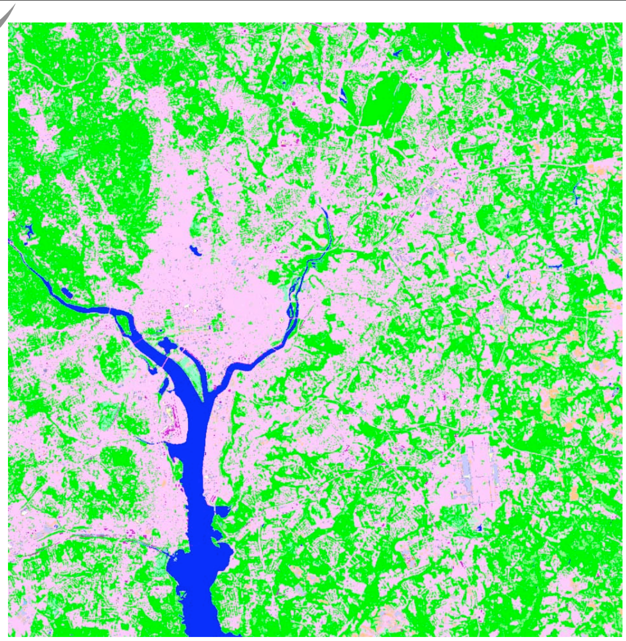
h_level	cvratio
5	1.063

For Presentation to the Spanish Remote Sensing Conference, Sept. 18, 2003. 37

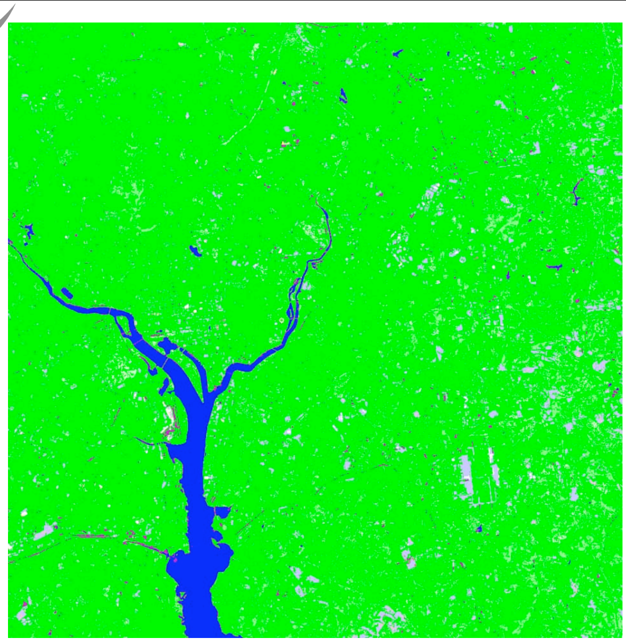


h_level	cvratio
6	1.020

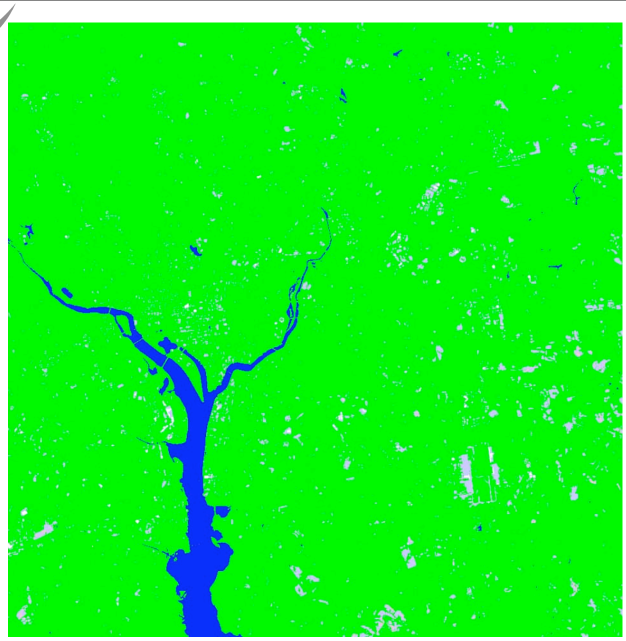
For Presentation to the Spanish Remote Sensing Conference, Sept. 18, 2003. 38



h_level	cvratio
7	1.032
8	1.066
9	1.011

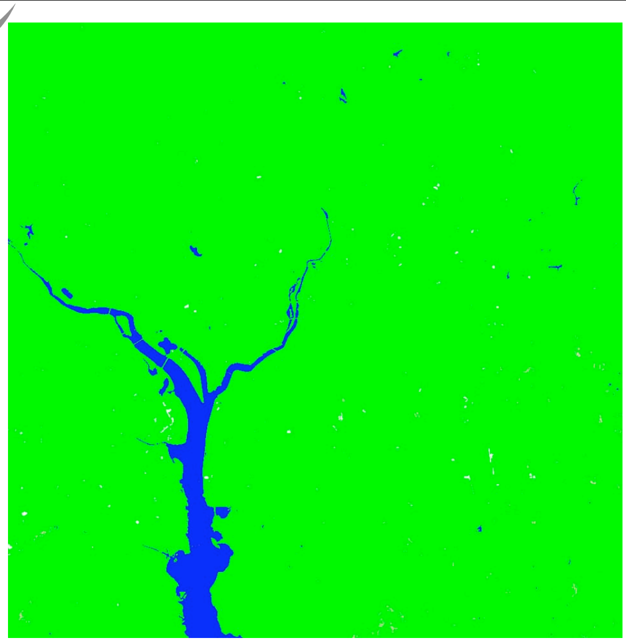


h_level	cvratio
10	1.374
11	1.046



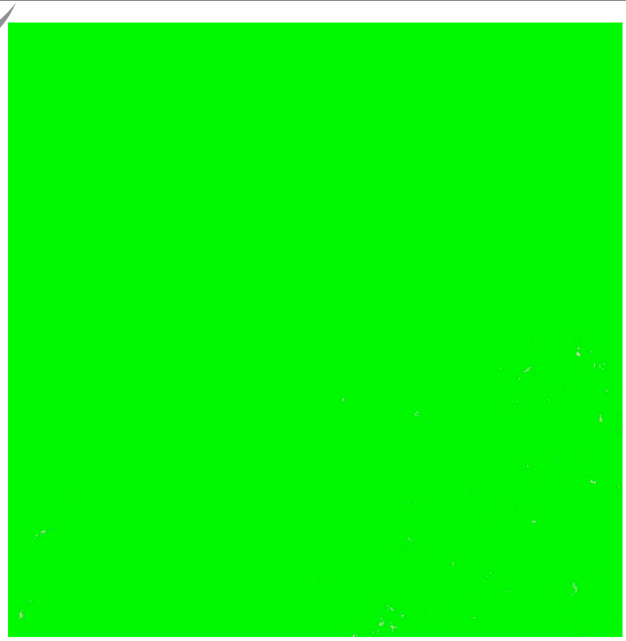
h_level	cvratio
12	1.074

For Presentation to the Spanish Remote Sensing Conference, Sept. 18, 2003. 41



h_level	cvratio
13	1.088

For Presentation to the Spanish Remote Sensing Conference, Sept. 18, 2003. 42



h_level	cvratio
14	-

Region Labeling Tool Example

A 1024x1024 pixel section of Landsat ETM+ data obtained on May 28, 1999 over Washington, DC, U.S.A.

Parameters: *spclust_wght* = 0.9, *spatial_wght* =1.0, *min_nregions* = 256, *chk_nregions* = 64.

Produced 15 hierarchical segmentation levels in 2 minutes and 33 seconds on a 64 CPU 1.2 GHz clock Beowulf Cluster.

Hierarchical Segmentation (cont'd)

Algorithmic Description of HSEG:

1. Give each data point a region label and set the global criterion value, *critval*, equal to zero. If a pre-segmentation is provided, label each data point according to the pre-segmentation. Otherwise, label each data point as a separate region.
2. Calculate the dissimilarity criterion value, *dissim_val*, between each spatially adjacent region.
3. Find the smallest *dissim_val* and set *thresh_val* equal to it. Then merge all pairs of spatially adjacent regions with *dissim_val* ≤ *thresh_val*.

Hierarchical Segmentation (cont'd)

4. If *spclust_wght* = 0.0, go to step 6. Otherwise, calculate the *dissim_val* between all pairs of non-spatially adjacent regions.
5. Merge all pairs of non-spatially adjacent regions with *dissim_val* ≤ *spclust_wght* thresh_val*.
6. If the number of regions remaining is less than the preset value *chk_nregions*, go to step 7. Otherwise, go to step 2.

Hierarchical Segmentation (cont'd)

7. Let *prevcritval* = *critval*. Calculate the current global criterion value and set *critval* equal to this value. If *prevcritval* = zero, save the results* from the <u>current</u> iteration and go to step 2. Otherwise calculate *cvratio* = *critval/prevcritval.* If *cvratio* is greater than the preset threshold *convfact*, save the results* from the <u>previous</u> iteration. If the number of regions remaining is two or less, save the results* from the <u>current</u> iteration as the coarsest instance of the final hierarchical segmentation result and stop. Otherwise, go to step 2.

* Results include region label map, region number of pixels list, region mean vector list and region criterion value list for this previous iteration. (Note: The region criterion value is the portion of the global criterion value contributed by the data points covered by the region.)

Hierarchical Segmentation **(cont'd)**

HSWO is computationally intensive and HSEG with non-adjacent region merging is even more computationally intensive (due a combinatorial explosion of required comparisons between regions).

The computational problem is solved through a recursive formulation of the HSEG algorithm. (Reduced by being able to deal with fewer regions from smaller image sections *and* joining disjoint regions.)

Recursive Hierarchical Segmentation (RHSEG)

RHSEG recursively divides an image into quarter sections until the image sections are small enough where the combinatorial explosion of inter-region comparisons is sufficiently reduced (about 1000 to 4000 pixels).

HSEG is performed on each section until a preset number of regions is reached – and the recursion is returned up until the image is fully reassembled (see NASA Case Number GSC 14,328-1).

A fast parallel implementation of RHSEG has been devised and is described in NASA Case Number GSC 14,305-1 (and U. S. patent application no. 5,965,879).

Recursive Hierarchical Segmentation (RHSEG)

Algorithmic Description of RHSEG:

1. Specify the number of levels of recursion required (*rnb_levels*) and pad the input data set, if necessary, so the width and height of the data set can be evenly divided by $2^{rnb\_levels-1}$. (A good value for *rnb_levels* results in a data section at *level* = *rnb_levels* consisting of roughly 1000 to 4000 data points.) Set *level* = 1.
2. Call *recur_hseg(level,data)*.
3. Execute the HSEG algorithm using as a pre-segmentation the segmentation output by the call to *rhseg*() in step 2. (Continue executing HSEG past the point that the number of regions reaches *chk_nregions* and save the segmentation results as specified.)

Recursive Hierarchical Segmentation (RHSEG)

Outline of *recur_hseg(level,data)*.:

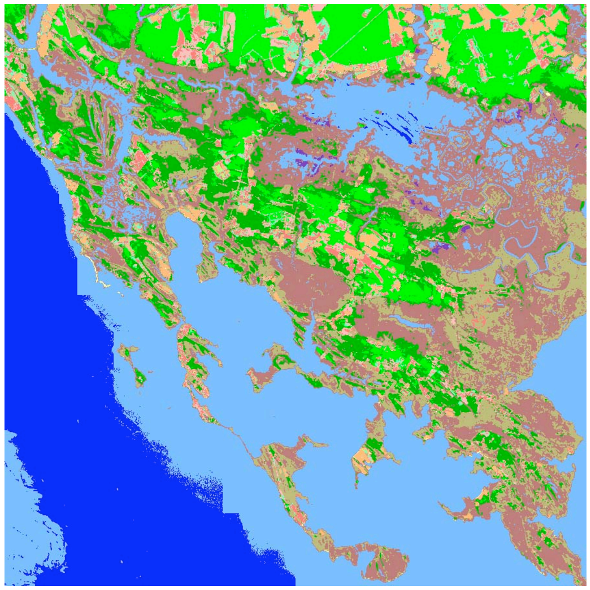
1. If *level* = *rnb_levels*, go to step 3. Otherwise, divide the data set into four equal subsections and call *recur_hseg(level+1,sub_data)* for each subsection of the data set (represented as *sub_data*).
2. After the calls to *recur_hseg( )* for each data set subsection from step 1 complete processing, reassemble the data segmentation results.
3. Execute the HSEG algorithm as described in the *HSEG Algorithm Description* above (using the reassembled segmentation results are as the pre-segmentation when *level* < *rnb_levels*), with the following modification: Terminate the algorithm when the number of regions reaches the preset value *min_nregions* (if *level* = 1, terminate at the greater of *min_nregions* or *chk_nregions*) and do not check for *critval* or output any segmentation results .

Recursive Hierarchical Segmentation (RHSEG)

In the initial stages of this project, a flaw was discovered in the current implementation of RHSEG:

Processing window artifacts often appeared in the results when RHSEG was applied to large images.

An example of this can be seen in the results from processing a Landsat ETM+ image obtained on May 28, 1999 covering an area south of Cambridge, MD.



For Presentation to the Spanish Remote Sensing Conference, Sept. 18, 2003. 55

Recursive Hierarchical Segmentation (RHSEG)

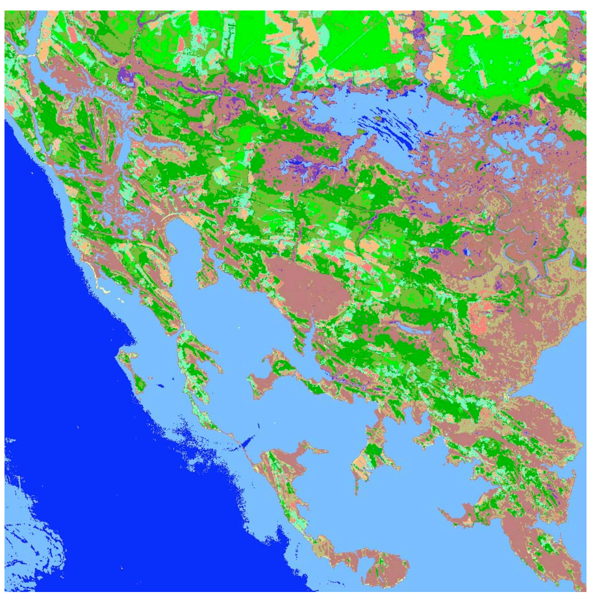
Initial attempts to eliminate the artifacts were not reliable and carried a large processing overhead.

It was decided a total rewrite of RHSEG was necessary…

A 1½ year effort resulted in a low processing overhead approach that reliably eliminated the artifacts.

For Presentation to the Spanish Remote Sensing Conference, Sept. 18, 2003. 56

For Presentation to the Spanish Remote Sensing Conference, Sept. 18, 2003. 57



For Presentation to the Spanish Remote Sensing Conference, Sept. 18, 2003. 58

Recursive Hierarchical Segmentation (RHSEG)

The method is described in "A Method for Recursive Hierarchical Segmentation which Eliminates Processing Window Artifacts," NASA Case Number GSC 14,681-1 (patent application in process).

With this artifact elimination, RHSEG can now reliably produce artifact free results for large images in acceptable processing times.

Recursive Hierarchical Segmentation (RHSEG)

Merges of small regions are accelerated by multiplying the dissimilarity function times an acceleration factor, *accel_factor*.

Define: $min\_npixels = \lfloor npixels * min\_npixels\_pct / 100.0 \rfloor$,

where *npixels* is the number of pixels in the section of data being processed, and *min_npixels_pct* is a user supplied parameter.

If *small_npix* is the number of pixels in the smaller of the two regions being compared, $accel\_factor = 1.0$ if $small\_npix \geq min\_npixels$ and

$$accel\_factor = 1.0 - ((min\_npixels - small\_npix)/min\_npixels)$$

otherwise.

Parallel Implementation of RHSEG

The recursive form of RHSEG lends itself to parallelization.

The number of parallel processes that can be utilized depends on the depth of recursion. If 32x32 pixels sections are processed at the deepest level of recursion:

rnb_levels	image size	# of sections	# of CPUs
1	32x32	1	1
2	64x64	4	4
3	128x128	16	16
4	256x256	64	64
5	512x512	256	64 to 256
6	1024x1024	1024	64, 256 or 1024
7	2048x2048	4096	64, 256, 1024 or 4096

Parallel Implementation of RHSEG

Processing times for 6-band Landsat ETM+ data.
The 64 CPU results are from the Medusa Beowulf cluster and
The 128 CPU results are from the Thunderhead Beowulf cluster.

rnb_levels	image size	# of CPUs	Processing Time
7	2048x2048	64	12 minutes, 24 secs.
7	2048x2048	256	2 minutes, 31 secs.
8	4096x4096	64	26 minutes, 22 secs.
8	4096x4096	256	8 minutes, 58 secs.
9	6912x6528	64	57 minutes, 27 secs.
9	6912x6528	256	*memory limitation*

Analysis of MODIS Images of the Iberian Peninsula

With the help of my colleagues Patrick Coronado and Alan Lunsford I obtained MODIS data from August 4, 2003 from over
the Iberian Peninsula.

Drs. Martinez and Plaza had suggested this dataset would be of interest to conference attendees, because it shows wildfires that plagued this area last summer.

Processed a 3072x4096 section of the 250m data (took 25 minutes on Thunderhead), and processed a 1536x2048 section of the 500m data (took 3½ minutes).

I will now explore the results from the 500m data using the Region Labeling Tool.

"Core" RHSEG and Region Labeling Tool Software Available

You just need to fill out and give to me the "Software Request Forms":

REQUEST FOR HIERARCHICAL SEGMENTATION SOFTWARE

Print, Complete, Sign and then FAX this form to James C. Tilton at (301) 286-1776.

Name

Company

Address

City State

Country Postal Code

Please provide a brief description of your intended use of the Region Labeling Tool software:

Contact Information

James C. Tilton
NASA's Goddard Space Flight Center
Mail Code 935
Greenbelt, MD 20771 USA
Telephone: (301) 286-9510
E-Mail: James.C.Tilton@nasa.gov
URL: http://code935.gsfc.nasa.gov/code935/tilton