

Online Journal of Space Communication

Volume 2
Issue 3 *Remote Sensing of Earth via Satellite*
(Winter 2003)

Article 18

January 2003

Hierarchical Image Segmentation

James Tilton

Follow this and additional works at: <https://ohioopen.library.ohio.edu/spacejournal>



Part of the [Astrodynamics Commons](#), [Navigation, Guidance, Control and Dynamics Commons](#), [Space Vehicles Commons](#), [Systems and Communications Commons](#), and the [Systems Engineering and Multidisciplinary Design Optimization Commons](#)

Recommended Citation

Tilton, James (2003) "Hierarchical Image Segmentation," *Online Journal of Space Communication: Vol. 2 : Iss. 3* , Article 18.

Available at: <https://ohioopen.library.ohio.edu/spacejournal/vol2/iss3/18>

This Research Reports is brought to you for free and open access by the OHIO Open Library Journals at OHIO Open Library. It has been accepted for inclusion in Online Journal of Space Communication by an authorized editor of OHIO Open Library. For more information, please contact deborded@ohio.edu.

Hierarchical Image Segmentation

James Tilton

Segmentation, the partitioning of image data into related sections or regions, is a key first step in a number of approaches to data analysis and compression. In image analysis, the group of image data points contained in each region provides a statistical sampling of image data values for more reliable labeling based on image feature values. In addition, the region shape can be analyzed as an additional clue for the appropriate labeling of the region. In data compression, the regions form a basis for compact representation of the image data. The quality of the prerequisite image segmentation is a key factor in determining the level of performance of most of these image analysis and data compression approaches.

Most image segmentation approaches can be placed in one of three categories [1]:

- Characteristic feature thresholding or clustering
- Boundary detection
- Region growing

Characteristic feature thresholding or clustering does not exploit spatial information, and thus ignores information that could be used to enhance the segmentation results. While boundary detection does exploit spatial information by examining local edges found throughout the image data, it does not necessarily produce closed connected region boundaries. For simple noise-free data, detection of edges usually results in straightforward region boundary delineation. However, edge detection on noisy, complex image data often produces missing edges and extra edges that cause the detected boundaries to not necessarily form a set of close connected curves that surround connected regions. Region growing approaches to segmentation are preferred here because region growing exploits spatial information and guarantees the formation of closed, connected regions.

Different analysis applications require different levels of segmentation detail, and a particular application may require different levels of segmentation detail depending on the image data objects being segmented. Applications such as data mining or knowledge discovery can benefit from being provided segmentations at different levels of detail [2]. The hierarchical image segmentation approach described here automatically provides segmentations for image data at several levels of segmentation detail.

The hierarchical image segmentation approach described herein, called HSEG, is a hybrid of region growing and spectral clustering that produces a hierarchical set of image segmentations based on detected natural convergence points (see also [3]). A hierarchical set of image segmentations is a set of several image segmentations at different levels of segmentation detail in which the segmentations at coarser levels of detail can be produced from simple merges of regions from segmentations at finer levels of detail.

The HSEG algorithm is very computationally intensive, and cannot be performed in a reasonable amount of time (less than a day) on moderately sized data sets, even with the most

powerful (single processor) computer currently available. For example, for a 6-spectral band Landsat TM image, a 128x128 pixel section takes about 25 minutes to process on a 1.2 GHz single processor computer. A 256x256 pixel section of the same image takes over 7.5 hours to process on the same computer. By extrapolation, a 512x512 pixel section of the same image would easily take several days.

The recursive formulation of HSEG (RHSEG), however, can process moderately sized data sets in a reasonable amount of time on currently available PCs and workstations. For example, for a 6-spectral band Landsat TM image, a 128x128 pixel section takes under 1 minute to process on a 1.2 GHz single processor computer with RHSEG. A 256x256 pixel section of the same image takes about 3.7 minutes to process on the same computer, a 512x512 pixel section of the same image takes about 16.3 minutes to process, and a 1024x1024 pixel section of the same image takes about 1.2 hour to process. Larger data sets require the use of a special parallel implementation of RHSEG on a parallel computing system, as disclosed under NASA Case No. GSC 14,305-1 [4]. On a 64-processor system, RHSEG takes about 3.2 minutes to process a 1024x1024 pixel section of a 6-spectral band Landsat TM image, and about 1.9 hours to process a full Landsat TM scene (roughly 7000x6500 pixels).

Other Approaches to Image Segmentation by Region Growing: The most commonly used approach to image segmentation by region growing is based on a general definition of image segmentation based on four condition that can be summarized (following Zucker [6]) as follows: (i) Every picture element (pixel) must be in a region, (ii) each region must be connected, (*i.e.* composed of contiguous image pixels), (iii) all image pixels in a region must satisfy a specified property to be considered similar enough to be in the same region (*e.g.*, a vector norm of between each pixel and the region mean must be less than a specified threshold), and (iv) in the final segmentation result, any merging of any adjacent regions would violate the third condition.

A problem with this classic definition of image segmentation is that the segmentation so defined is not unique. The number and shape of the partitions depend on the order in which the image pixels are processed. In addition, there is no concept of optimality contained in this definition of image segmentation. Under this classic definition, all partitions that satisfy the conditions represent equally good or valid segmentations of the image.

A less commonly used approach to region growing image segmentation is the Hierarchical Stepwise Optimization algorithm of Beaulieu and Goldberg [7]. HSWO is best defined iteratively: Start with an image and a segmentation of that image into N regions in which (i) every picture element (pixel) is in a region, (ii) and each region is connected, (*i.e.* composed of contiguous image pixels). Then compare all spatially adjacent regions with each other (*e.g.*, compute a vector norm between the region means of the spatially adjacent regions). Merge the most similar pair of spatially adjacent regions. Continue to compare spatially adjacent regions and merge the most similar pair of spatially adjacent regions until either a specified number of regions are reached or the dissimilarity between the most similar pair of spatially adjacent regions reaches a specified threshold.

The initial partition may assign each image pixel to a separate region. Any other initial partition may be used, such as an over-segmented result from region growing based on the classic definition given above (*i.e.*, classic region growing segmentation with a low threshold value).

The region growing approach utilized by the hierarchical image segmentation (HSEG) algorithm described next is identical to that employed by Beaulieu and Goldberg's HSWO algorithm except that HSEG optionally alternates spectral clustering iterations with region growing iterations. In the spectral clustering iterations, non-adjacent regions are merged. In fact, in their paper on HSWO, Beaulieu and Goldberg [7] provide the theoretical basis for the HSEG algorithm in their theoretical analysis of HSWO. They show that the HSWO algorithm produces the globally optimal segmentation result if each iteration is statistically independent. Even though each iteration will generally not be statistically independent for natural images, the HSWO approach still produces excellent results. Beaulieu and Goldberg also point out that the sequence of partitions generated by this iterative approach reflect the hierarchical structure of the imagery data: the partitions obtained in the early iterations preserve the small details and objects in the image, while the partitions obtained in the latter iterations preserve only the most important components of the image. They further note that these hierarchical partitions may carry information that may help in identifying the objects in the imagery data.

Hierarchical Image Segmentation (HSEG): The following high-level description of the HSEG algorithm is based on the description given earlier in [3]:

HSEG Algorithm Description:

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 $dissim_val \leq thresh_val$.
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 $dissim_val \leq spclust_wght * thresh_val$.
6. If the number of regions remaining is less than the preset value *chk_nregions*, go to step 7. Otherwise, go to step 2.
7. Let *prevcritval* = *critval*. Calculate the current global criterion value and set *critval* equal to this value. If *prevcritval* = zero, go to step 2. Otherwise calculate *cvratio* = *critval/prevcritval*. If *cvratio* is greater than the preset threshold *convfact*, save the region label map from the previous iteration as a "raw" segmentation result. Also, store the 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.) If the number of regions remaining is two or less, save the region information from the current iteration as the

coarsest instance of the final hierarchical segmentation result and stop. Otherwise, go to step 2.

Spectral clustering (steps 4 and 5 above) is optional. When *spclust_wght* = 0.0, the HSEG algorithm is run with only region growing (steps 2 and 3), skipping spectral clustering (steps 4 and 5).

Recursive Hierarchical Image Segmentation (RHSEG): As was noted earlier, the HSEG algorithm is very computationally intensive, and cannot be performed in a reasonable amount of time (less than a day) on moderately sized images, even with the most powerful (single processor) computer currently available. The recursive formulation of HSEG (RHSEG), however, can process moderately sized images in a reasonable amount of time on currently available PCs and workstations.

Description of the RHSEG algorithm (assumes 2-dimensional image data):

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.)

Outline of *recur_hseg(level,data)*:

1. If *level* = *rnb_levels*, go to step 3 below. 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 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 "raw" segmentation results.

The above divide-and-conquer approach limits the number of regions that are processed at any time in step 4 of the HSEG algorithm. This limit leads to a significant reduction in processing time versus the non-recursive approach for even relatively small data sets [3]. An efficient parallel implementation of RHSEG leads to additional significant reduction in processing time [4].

Another problem emerges when the RHSEG algorithm is used to process moderate to large images. Processing window artifacts may arise from the recursive division of the image data into four equal subsections. This artifacts can be eliminated, however, by the addition of a fourth step to the outline of *recur_hseg(level,data)* as follows:

4. If *level* = *rnb_levels*, exit. Otherwise, switch the region assignment of certain pixels in the following manner: For each region, determine which other regions may contain pixels

that may more similar to it than the region to which they are currently assigned. (Details of how this determination is made are the subject to a pending patent application [8].) Then for each of these regions compute the dissimilarity each pixel contained in the region to its current region (*own_region_dissim*) and to each region to which it may potentially be more similar. If a pixel is found to have $own_region_dissim > switch_pixels_factor * other_region_dissim$, switch the region index for that pixel to the region with the minimum *other_region_dissim* value. Exit.

Comparative segmentation results on Landsat Thematic Mapper (TM) data: (I would like to show and contrast example results from each region growing-based image segmentation approach described above – however, the results from the classical region growing approach will not be available until the end of January.)

Exploiting the segmentation hierarchy: The exploitation of the region membership relationships between the regions at different levels of the segmentation hierarchy is being actively explored in a current research project[2]. In a hierarchical segmentation, an object of interest may be represented by multiple image segments in finer levels of detail in the segmentation hierarchy, and may be merged into a surrounding region at coarser levels of detail in the segmentation hierarchy. If the segmentation hierarchy has sufficient resolution, the object of interest will be represented as a single region segment at some intermediate level of segmentation detail. A current research goal is to identify the hierarchical level at which the object of interest is represented by a single region segment. The object may then be identified through its spectral and spatial characteristics. Additional clues for object identification may be obtained from the behavior of the image segmentations at the hierarchical segmentation level above and below the level at which the object of interest is represented by a single region.

References

- [1] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition*, Vol 13, pp. 3-16, 1981.
- [2] James C. Tilton (PI), Giovanni Marchisio (Co-I) and Mihai Datcu (Co-I), "Knowledge Discovery and Data Mining Based on Hierarchical Segmentation of Image Data, " a research proposal submitted October 23, 2000 in response to NRA2-37143 from NASA's Information Systems Program. (This project proposal was selected for funding through December 2003.)
- [3] James C. Tilton, "Method for recursive hierarchical segmentation by region Growing and spectral clustering with a natural convergence criterion, " *Disclosure of Invention and New Technology: NASA Case No. GSC 14,328-1*.
- [4] James C. Tilton, "Method for implementation of recursive hierarchical segmentation on parallel computers," *Disclosure of Invention and New Technology: NASA Case No. GSC 14,305-1*.
- [5] Steven L. Horowitz and Theodosios Pavlidis, "Picture segmentation by a directed split-and-merge procedure," *Proceedings of the Second International Joint Conference on Pattern Recognition*," pp. 424-433, 1974.
- [6] Steven W. Zucker, "Region growing: Childhood and adolescence," *Computer Graphics and Image Processing*, Vol. 5, pp. 382-399, 1976.

- [7] Jean-Marie Beaulieu and Morris Goldberg, "Hierarchy in picture segmentation: A stepwise optimization approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 2, pp. 150-163, February 1989.
- [8] James C. Tilton, "A Method for Recursive Hierarchical Segmentation which Eliminates Processing Window Artifacts," *Disclosure of Invention and New Technology: NASA Case No. GSC 14,681-1*.