Hierarchical Image Segmentation by Polygon Grouping

Lakshman Prasad

Sriram Swaminarayan

Los Alamos National Laboratory
PO Box 1663,
Los Alamos, NM 87545

Abstract

We present a simple cascading algorithm for rapid hierarchical image segmentation based on perceptually driven contour completion and scene statistics. We start with an initial fine-scale segmentation of an image obtained by perceptual completion of partial contours into polygonal regions using region-contour correspondences established by Delaunay triangulation of edge pixels. The resulting polygon size distribution is analyzed for a dominant mode of granularity of the image. Polygons whose sizes are less than or equal to this granularity are merged with their spectrally closest neighbors to obtain the next level of polygonal segments in the hierarchy. The iterative application of this process precipitates textured regions as polygons with highly convolved boundaries and helps distinguish them from objects which typically have more regular boundaries.

1 Introduction

In general, there is no single segmentation of an image that captures all possible features of interest since they occur at multiple scales and varying spectral prominence that determine their saliencies. Further, what is of interest in an image is quite often determined by what is sought in an image. Many application-driven approaches segment a specific class of spectrally or structurally distinctive objects and achieve satisfactory results via supervised or unsupervised methods. Other methods, while not restricting the types of features sought, require specification of the number of objects of interest [1], regions, or the specification of tolerances and parameters [2]. Typically, the choice of these inputs is ad hoc as it is not clear a priori what is best for segmenting a particular image. It is desirable to have a segmentation scheme that reveals all potentially salient objects at different scales in a hierarchical manner so that general queries about image content can be addressed without a priori knowledge about the image. Furthermore it is desirable to have the content information available in a context-friendly manner so that parts of objects that are salient by themselves can be related to the objects they belong to or interact with at different scales in the hierarchy. Cour et al. [1] have proposed an efficient multiscale version of the well-known normalized cuts graph decomposition approach [3]. The advantage of this approach is that it works on multiple scales of an image in parallel while preserving consistency across scales. However, as mentioned before, this method requires specification of the number of regions to be segmented and the segmentation results for two different regional specifications are not necessarily consistent Recently, E. Sharon et al. [4] have proposed an efficient hierarchical segmentation scheme that aggregates pixels into salient regions of varying scales using an algebraic multigrid method. They start by solving the normalized cut problem on a select subset of seed pixels, about half as many as the total number of image pixels, to get an initial segmentation. They then recursively compute aggregate properties of the segment pixels and obtain successively coarser aggregates corresponding to salient image regions. The latter progressive coarsening method preserves hierarchical containment of segmented regions at the different scales.

In this paper we will present an alternate approach to hierarchical image segmentation based on an earlier work of ours [5, 6, 7] to obtain an initial fine scale 'seed' segmentation that faithfully conforms to image contours. We then iteratively group these polygons based on their size distributions and spectral proximities to yield a hierarchy of image segmentations wherein each polygon at each level is an aggregate of one or more polygons at the previous finer level. Both object and texture regions are segmented in this method, with the successively coarser polygons inheriting the fidelity to image edges of the initial segmentation. Our approach yields a rapid hierarchical segmentation and is capable of addressing very large, complex images.

This paper is organized as follows: section 2 gives a brief overview of an edge-based segmentation algorithm we will use to obtain an initial segmentation. Section 3 describes our hierarchical polygon grouping scheme. Section 4 discusses how object and texture polygons are distinguishable from their contour characteristics. Section 5 concludes with

Filter	Property
TooLong	length > Median Delaunay edge length
Shortest	Shortest edge of either flanking trixel
Canny	Connects neighboring Canny edge points
EndLink1	Connects (with minimum turn) end point of a contour to the interior of another contour (transversality)
EndLink2	Connects two contour endpoints
Junction	At least one flanking trixel has all its vertices on different contours

Table 1: Boolean perceptual edge filters [7].

experimental results and comparison to other methods.

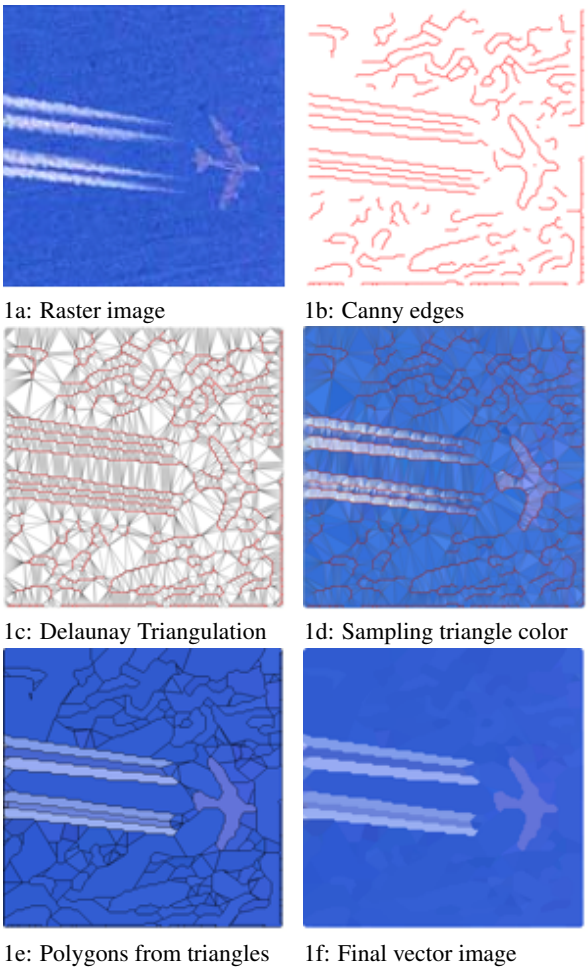
2 Vectorized Image Segmentation

We will use our vectorized image segmentation framework [5, 6, 7] to obtain our initial edge-based seed segmentation. In the above work we propose a departure from the common approach of grouping pixels for segmenting images and, instead, obtain polygonal segmentations by perceptual grouping of image edge-adaptive triangular mesh elements. Briefly, we first obtain image edges (Figure 1b) by a Canny edge detection procedure [8]. Next, a constrained Delaunay triangulation [9] of the edge point set is performed so that no triangle edge intersects an edge between neighboring edge pixels (Figure 1c). The proximity graph property of Delaunay triangulations is used to establish regional correspondences between contour chains and provide candidate edge completions of the Canny contours, which are typically fragmented and open curves. The pixels in each triangle are sparsely sampled to estimate an average color to be attributed to the triangle (Figure 1d). These color-attributed triangles (trixels) are the new image primitives that will be grouped to yield polygonal image segments.

A grouping graph G is constructed, with nodes representing trixels and links representing adjacency of trixels. Cuts in G are introduced by identifying those edges between trixels that support desirable contour completions and deleting corresponding links in G. The evaluation of a trixel edge for contour completion is based on six elementary perceptual criteria (see Table 1), modeled as Boolean filters on the trixel edges and is a logical concatenation of these criteria. The cut criterion C is then given by:

$$C = \quad (\sim TooLong) \wedge [Canny \vee EndLink1 \vee EndLink2 \vee (Junction \wedge Shortest)] \quad (1)$$

This criterion has been experimentally found to yield optimal results in terms of both image fidelity and polygon economy. A link l in G is cut if and only if C is equal to 1 for the corresponding edge between two trixels whose



1a: Raster image 1b: Canny edges

1c: Delaunay Triangulation 1d: Sampling triangle color

1e: Polygons from triangles 1f: Final vector image

Figure 1: Steps in vectorized image segmentation

representative nodes in G are linked by l. Trixels in connected components of the resulting cut graph are grouped to obtain polygonal image segments, with each segment attributed the area-weighted average color of all constituent trixels (Fig. 1e).

2.1 Remarks

It is worth noting that the grouping is achieved based on purely structural considerations with color playing no part in computing the cuts. The trixel colors are used only to compute the aggregate color of the polygons they constitute. The number of canny edge pixels is typically much smaller than the total number of image pixels, and the number of triangles produced in the triangulation of edge points is at most twice the number of edge points. The Canny edge detection algorithm is linear in the number N of image pixels, and the Delaunay triangulation of a set of $n << N$ edge points takes $O(nlog(n))$ time. The trixel edge filtering to

determine the cuts is linear in the number of edge points as the number of trixel edges is at most three times the number of edge points. Thus the algorithm is highly efficient in its overall time and space complexity allowing the segmentation of very large images. However, the segmentation uses only local region-edge correspondences and thus does not have long range information. Hence the segmentations are typically fine scale and do not in general capture larger features fully without further grouping of the polygons.

3 Hierarchical Polygon Grouping

Nevertheless, the high efficiency of the initial segmentation (which can be prohibitively expensive with a method such as normalized cuts approach [3] for segmenting large images) coupled with the fact that the resulting polygons conform to image contours, are compelling reasons for employing this method as the starting point for a hierarchical segmentation scheme based on region growing. Fine-scale segments conforming to image contours assures edge integrity of coarser regions when successively merging polygons, unlike pixel-based region growing methods.

Spectral similarity alone is too simplistic a criterion for segmenting complex real world images and often leads to false regions corresponding to nonexistent features. Indeed, the human visual system is adept at factoring out texture fields in spite of significant variability in spectral values in such regions and looking beyond for objects that stand out. A polygon merging scheme that is tuned to first 'flattening' textured areas of an image before subsuming features into larger regions would likely better handle the emergence of saliency hierarchies. To test this hypothesis, we first model texture characteristics in an image, whether it is due to actual texture contained in it or texture induced by oversegmentation, by a statistical estimation of image granularity. Next we allow polygonal segments of a certain granularity to merge with their spectrally closest neighbors to obtain the next, coarser level polygonal segmentation. This process is repeated to produce a hierarchy of segmentations wherein each polygon at an intermediate level of the hierarchy is contained in another polygon (possibly itself) at the next coarser level. This nested scheme of segmentation is necessary to determine the context of an object or feature such as an eye in a face or a window in an automobile.

We represent each polygon obtained from the fine scale segmentation described in section 2 as a node in a polygon adjacency graph. Its adjacency to other polygon nodes is derived readily from the adjacency of its constituent trixels to the trixels constituting other polygons in the original trixel graph G before introducing cuts. Next, we compute the polygon areas as the cumulative areas of their constituent trixels. The distribution of polygons with respect to their areas provides a picture of the granularities comprising the image at that scale of segmentation. By choosing the highest mode of this distribution we identify the perceptually dominant granularity of the image. We then flatten the image to this level of granularity by merging polygons whose areas are equal to or less than the dominant grain size with their most similar neighbors, where the measure of similarity can be structural as well as spectral; for example, grain size, shape, orientation and or color/intensity. Indeed, the quality of our segmentation can be significantly improved by easily factoring in other perceptual cues. However, in this paper, we would like to highlight the fact that even our simple and straightforward polygon agglomeration approach without embellishments, when performed on the initial contour completion-based vectorized segmentation described above, by itself yields notably high quality segmentation and real-time performance.

To compute the distribution of polygons with respect to area we construct the histogram of the polygon numbers over the range of their areas. For the histogram to faithfully represent the distribution we need to estimate an appropriate bin size. We appeal to the recent work of H. Shimazaki and S. Shinomoto [10] to do this. Briefly, they propose an algorithm which minimizes the mean integrated squared error of the histogram of a finite sample from an unknown distribution with respect to the distribution. In practical terms they show this is equivalent to finding the bin size Δ that minimizes the cost function:

$$\Theta(\Delta) = \frac{2\mu(\Delta) - \nu(\Delta)}{\Delta^2} \quad (2)$$

Where μ and ν are the mean and variance of the histogram of the sample with bin size Δ. Similarly, to determine an automatic color threshold for merging polygons based on the dominant spectral difference, we compute the optimal histogram of the distribution of hue/intensity differences between neighboring polygons and pick the location of the dominant histogram peak as the upper bound for the spectral difference between adjacent polygons to be merged. This ensures that the merging of a polygon with its closest spectral neighbor does not take place if their spectral difference is greater than the dominant spectral difference. That is to say, a polygon that is salient due to its spectral distinction with respect to its surroundings does not get merged merely on account of its grain size falling below the dominant granularity of the image. Thus the granularity and spectral difference bounds help preserve the structurally and spectrally salient features, respectively, during the texture flattening process of our hierarchical segmentation scheme.

Other polygon properties such as orientation, aspect ratio, etc., can be similarly analyzed for a more careful control of the polygon merging process. However, in this paper, we restrict ourselves to color and size to demonstrate the efficacy of our simple method which is purely bottom up with no large-scale look-ahead computations performed as in [1, 4]. This process of merging is computed, well past the

Small Image: 0.15 MP (320×480) Average: 0.48 Mpixel/sec (Level 0: 0.90 Mpixel/sec)				
Level	N	dominant grain size	segmentation time	cumulative time
0	1814	0.000	0.165	0.165
1	461	0.154	0.032	0.198
2	117	0.158	0.023	0.221
3	36	0.166	0.023	0.244
4	10	0.170	0.015	0.260
5	4	0.142	0.014	0.274

Large Image:8 MP (2903×2896) Average: 0.31 Mpixel/sec (Level 0: 0.70 Mpixel/sec.)				
Level	N	dominant grain size	segmentation time	cumulative time
0	131428	0.000	11.625	11.625
1	31704	0.008	3.053	14.679
2	6885	0.008	2.061	16.740
3	1478	0.008	1.675	18.415
4	301	0.008	1.647	20.062
5	77	0.008	1.470	21.532
6	22	0.023	1.714	23.246

Table 2: Performance metrics for hierarchical segmentation of a small image and a large image done on a 2.13 GHZ Pentium M with 2GB of RAM. N is the number of polygons at a given level, dominant grain size is the area normalized peak granularity, segmentation time and the cumulative time are in seconds.

emergence of the last meaningful salient feature, until a single polygonal region corresponding to the image rectangle remains. This will form the topmost coarsest node in our hierarchical segmentation tree. Each salient object/feature can be accessed from this node by traversing the tree and the path consisting of coarser parent segments provide the contextual embedding of the object/feature for a semantic assessment of it. If the merging process stagnates with more than one polygon, we expand the merge criterion to include the next dominant mode in the relevant distribution. At each stage of merging, graph representing the polygons as nodes and their adjacencies as edges is pruned. An edge is deleted if it fails the statistically computed threshold for the different merge criteria. The connected components of the resulting graph yield the next coarser level. The performance of our scheme on a small and a large image is shown in Table 2, providing the hierarchy level number, the number of segments, the grain size relative to the image, the segmentation time, and the total times. The differences in the rates are due to the texture and level differences in the two images. We illustrate our method with layers 2 to 4 from the hierarchical segmentations of two images in the Berkeley image segmentation data set [11] in Fig. 2.

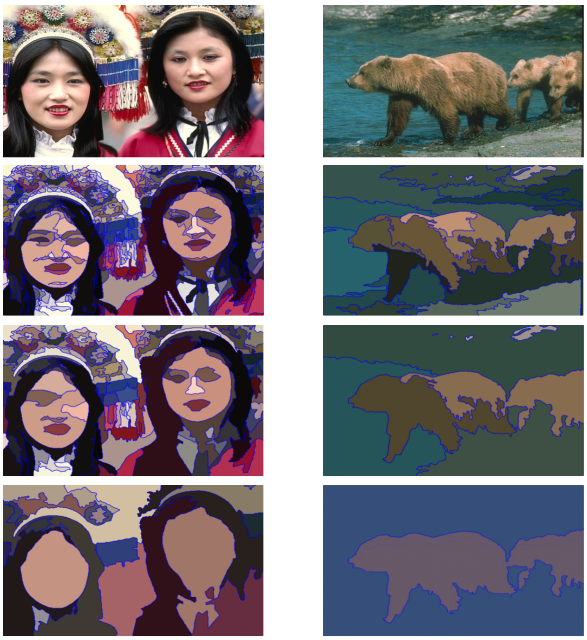


Figure 2: Example hierarchical segmentations on a high contrast low texture image (0.447 seconds) and a low contrash high texture image (0.493 seconds).

4 Detecting Objects and Textures

Our method of hierarchical segmentation clearly preserves boundary fidelity at all levels as illustrated in Figure 2. This is due to the inheritance of Canny edges by polygons from those constituting them at the previous level. As segmentation progresses into coarser scales, we observe that our method yields polygons with distinctly different boundary types for object and texture polygons. Object polygons tend to have smoother boundaries than texture polygons which tend to be significantly more oscillatory in nature. This is because Canny edges of objects are typically longer and smoother than texture elements which are typically smaller and tend to clump into polygons with wiggly meandering contours. This feature can be utilized as a perceptual cue to detect texture content in an image and prevent texture polygons from merging with object polygons at coarser scales. We define a simple measure of boundary 'wiggle' for a polygon A as:

$$W(A) \quad = \quad 1 - \frac{L_S(A)}{L(A)} \tag{3}$$

Where $L(A)$ is the boundary length of polygon A and $L_S(A)$ is the length of the averaged and decimated (i.e., low-pass filtered by a Haar wavelet transform) boundary of A, with the smoothing and decimation performed a few times (in our case, 4). For two adjacent polygons A and B at sufficiently coarse scales we adopt a 'win-win' strategy of merging them at the next scale into a larger polygon

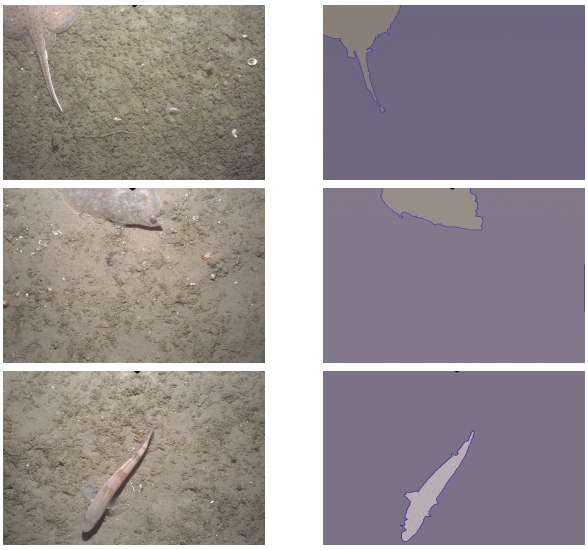


Figure 3: Example segmentations of camouflaged fish using wiggle filtering at coarse scales. Hierarchical segmentation time for each of these 512×768 images is $\sim$2.75 seconds.

(in addition to meeting the constraints on granularity and spectral proximity) if and only if the wiggle of the polygon $A \cup B$ is less than or equal to the minimum of the individual wiggles of A and B. Noting that

$$L(A \cup B) \quad = \quad L(A) + L(B) - 2L(A \cap B) \tag{4}$$

And

$$L_S(A \cup B) \quad \leq \quad L_S(A) + L_S(B) - 2L_S(A \cap B) \tag{5}$$

We get

$$W(A \cup B) \geq \frac{L(A)W(A) + L(B)W(B) - 2L(A \cap B)W(A \cap B)}{L(A) + L(B) - 2L(A \cap B)} \tag{6}$$

We note that although equation 6 is an inequality, the difference between the two sides is typically small and due to the differences in smoothing at the points where the adjacency of the polygons begins and ends along their common interface. We therefore take the right hand side of equation 6 as a good approximation of the wiggle of the union of two polygons and use it to compute the latter quantity without actually merging the two polygons first. The wiggle and length of the interface can be computed at a lower cost from the polygon adjacency graph constructed at each level of the hierarchy. The additional wiggle computation is performed at coarser levels only when the segmentation is ascertained to have significant textured areas by the predominance of polygons of high wiggle.

We illustrate in figure 3 the use of wiggle to suppress texture and extract camouflaged fish in ocean floor imagery.

	Our Method		Multiscale Ncuts		Mean Shift	
Image	T	N	T	N	T	N
Elk	0.30	3	41.9	5	15.4	65
Tiger	0.52	8	57.5	7	25.7	105
Horse	0.55	2	105.6	13	15.4	121
Marmot	0.56	16	91.2	12	19.9	207
Baboon	0.34	5	69.2	7	15.9	371
Penguin	0.33	72	182.6	17	18.0	791
Owl	0.51	4	121.4	10	43.8	1255
Leopard	0.40	11	156.0	17	47.6	1952

Table 3: Comparison of run times (T) in seconds and the number of segmented regions (N) obtained by our method, the multiscale normalized cuts method, and the mean shift method for the eight images in figures 4 and 5. All timings were taken on a 2.13 GHz Intel Pentium M processor with 2 GB of RAM.

The parts of the boundaries of segments that coincide with the image rectangle were excluded in the computation of wiggle as this would not faithfully reflect the intrinsic characteristics of such segments.

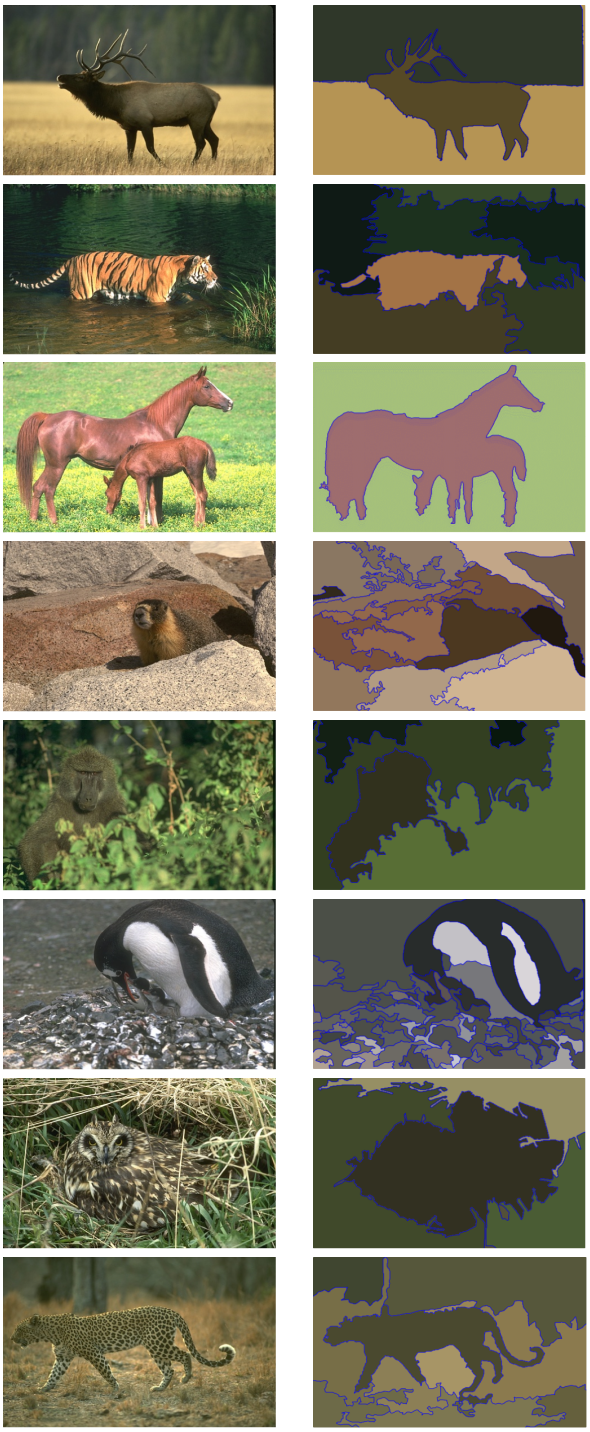
Thus our simple and efficient hierarchical segmentation scheme not only preserves contour integrity across coarsening scales but also provides an easy way to detect and isolate texture from salient objects even in very challenging imagery.

5 Results, Comparison, and Conclusion

We demonstrate our results in in Fig. 4 by depicting a selected level in the hierarchy of segmentations that contains the salient features for each of the images shown on the left hand side from the Berkeley segmentation data set. The choice of images from this data set was motivated by difficulty, variability in contrast, texture, and color.

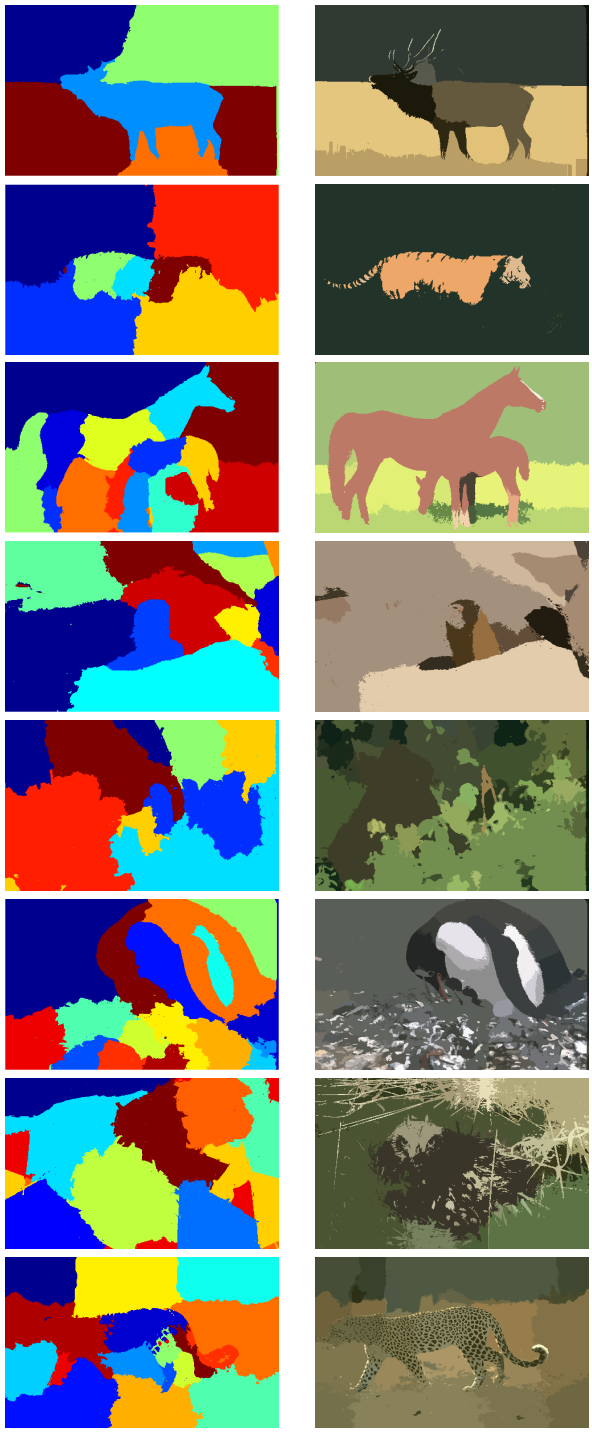
5.1 Comparison

We compare our results to those of two primarily spectral segmentation methods, namely the multiscale normalized cuts method [1] and the mean shift method [2]. We used the multiscale normalized cuts segmentation toolbox for MatLab [12] implementing the algorithm in [1] to obtain the results shown in the left column of Figure 5, and the edge detection and image segmentation system (EDISON) C++ executable [13] implementing the algorithm in [2] to obtain the results shown in the right column of Figure 5. In both cases we selected the required input parameters (by



Original Image | Our Segmentation

Figure 4: Example images (left) from the Berkeley data set and a level from the hierarchical segmentations (right) revealing salient objects and features



Multiscale Ncuts | Mean Shift

Figure 5: Results of multiscale normalized cuts (left) and mean shift (right) methods applied to the original images in figure 4

trial and error,) that produced the visually best segmentation for each image such that the salient features of interest were depicted with the least number of polygons. In the case of the multiscale Ncuts method, we selected the number of regions that obtained best results for each image, whereas, in the case of the mean shift method we selected the spatial and spectral bandwidth parameters that obtained the best results, again on a per image basis. In contrast, we did not vary any parameters of our method across the images tested on in this paper. The Canny edge detection parameters that govern the initial segmentation as described in [7] were fixed at sigma = 1 and the high and low hysteresis thresholds were fixed at 0.6 and 0.0, respectively. Both the multiscale Ncuts and the mean shift methods yield over-segmentations of salient objects of interest compared to our method. This necessitates further processing to assemble the parts into a meaningful whole for object recognition. Although the mean shift method produced visually better looking segmentations, the number of regions produced by it is significantly larger than the other two methods. For instance, in the case of the leopard image, each spot is a different segment in the mean shift result, and the leopard does not correspond to a single polygonal segment as in our method's case. Our method also produces a hierarchy of segmentation images at multiple scales with features of different saliencies extracted at different levels whereas the other two methods produce a single segmentation image for a given input image and parameter set. Thus, to obtain finer or coarser scale features, these methods would require re-segmentation of the image. The segments obtained at different scales by our method form a tree with finer scale segments nested in segments at a coarser level, providing contextual information. The segments obtained by the multiscale normalized cuts and the mean shift methods do not have this hierarchical containment property across scales. Thus the same object may be segmented in two different ways for differing parameters with the segments intersecting across the results. Our method explicitly provides the contours of the segments, while the other two methods provide pixel masks for the segments requiring further processing to obtain contours. Finally, as illustrated in Table 3, our hierarchical method is an order of magnitude faster than the other two methods. This is even allowing for a ten-fold speedup in a C implementation of the multiscale normalized cuts method over the MatLab implementation tested in this paper.

5.2 Other methods

There are too many interesting and successful approaches to image segmentation to enumerate here. We touch upon a few methods and comment briefly upon some differences of our approach to these. Tu and Zhu [15] assume a size distribution for regions to determine how cluttered or 'busy' an image is, whereas we non-parametrically determine image granularity at each scale. Paris and Durand [14] detect singularities in a 5D color-position space to determine stable manifolds and image regions. We obtain our image 'singularities' in 3D (position-intensity space) as gradient maxima via a Canny edge detection algorithm to separate regions in an image. A comparison of the two methods is presented in Fig. 6. The force field transform of Tabb and Ahuja [16] yields edges at locations of repulsion that close to form regions. In the method used by us to obtain a fine scale segmentation the edges are readily obtained via an edge detection algorithm, but since they are not necessarily closed, a perceptual approach is taken to complete them into regions based on gestalt principles modeled explicitly on edges of a Delaunay grid. Ren and Malik [17] obtain initial regions (superpixels) by normalized cuts and then train a linear classifier with correct and wrong segmentations. we do not use training or learning to segment images. The ObjCut method of Kumar et al [18] uses Markov Random Fields along with prior knowledge in terms of object/shape model to incorporate topdown information into the segmentation process. Our objective is to demonstrate the efficacy and efficiency of a purely bottom up perceptually driven agglomeration process in precipitating meaningful features at their natural scales.

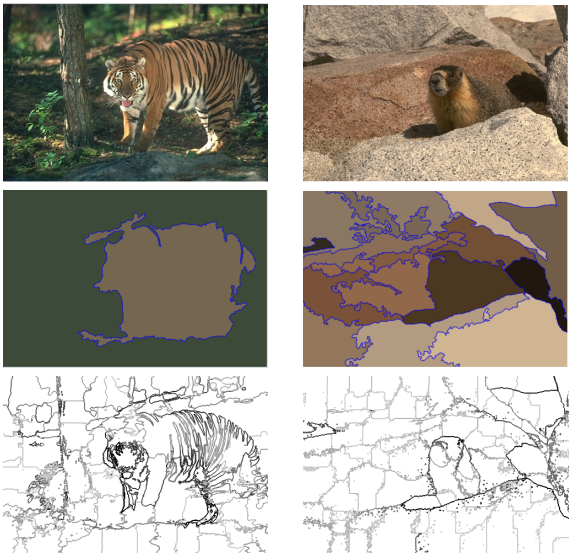


Figure 6: Comparison of the coarsest levels our hierarchical method and that of Paris-Durand [14]

5.3 Conclusion

In this paper we have presented a highly efficient linear time algorithm for hierarchical image segmentation based on [7]

that:

1. reveals objects and features at various saliencies,
2. preserves boundary integrity across scales,
3. provides contextual information in a segmentation tree of nested object with parent-child relationships,
4. provides boundary information without additional effort because of vector (polygonal) segments,
5. allows the detection of texture in images and handle them differently from object segments.

We have used only size and color as the driving parameters of our segmentation scheme for most images (except for the camouflage images in figure 3 which additionally use the wiggle filter) to illustrate its efficacy. Additional structural, spectral, and statistical criteria may be easily incorporated to significantly enhance or specialize the segmentation performance. We do not vary any parameters to segment different images. The Canny edge detection was performed on all images using a sigma of 1 and with the high and low hysteresis thresholds fixed at 0.6 and 0, respectively. The number of segments at each level and the total number of levels is completely data adaptive. This also accounts for the variability of average processing rate from image to image.

Applications such as automated content-based classification, archival, and retrieval of images, video analysis, etc., require not only efficient but also robust image segmentation methods that do not assume a priori knowledge of image characteristics or parameters to segment and analyze them. Further, to accommodate a wide range of queries on image content, segmentations must preemptively provide features at multiple scales and in a manner in which they can be related to extract image semantics via object/feature context. The method presented in this paper is intended as a step in that direction. We believe the surprising effectiveness of our simple polygon grouping method, as illustrated by our results, is largely due to the power of the contour completion method in providing the basic building blocks for segmentation. Indeed, Elder and Goldberg [19] have studied the statistics of contour grouping cues in natural images and found that the law of proximity is by far the single most powerful cue for contour grouping. The basis of using Delaunay triangulation of contour points to achieve completions in [7] is to exploit proximity relationships, thus providing a robust starting point for our algorithm to propagate these contours to coarser scales and extract salient structures. This work was supported by the US Department of Energy. We would like to thank Scott Gallager of the Woods Hole Oceanographic Institution for kindly providing the marine images used in this paper.

References

[1] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," *Proceedings of CVPR 2005*, 2005.

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on PAMI*, vol. 24, no. 5, pp. 1–18, 2002.

[3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on PAMI*, vol. 22, no. 8, pp. 501–514, 2005.

[4] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and adaptivity in segmenting visual scenes," *Nature (online version)*, 2006.

[5] L. Prasad and A. Skourikhine, "A new image representation for compact and secure communication," *Proceedings of the Annual Meeting of the INMM*, vol. 45, 2004.

[6] L. Prasad and A. Skourikhine, "Vectorized image segmentation via trixel agglomeration," *Lecture Notes in Computer Science*, vol. 3434, pp. 12–22, 4 2005.

[7] L. Prasad and A. Skourikhine, "Vectorized image segmentation via trixel agglomeration," *Pattern Recognition*, vol. 39, no. 4, pp. 501–514, 4 2006.

[8] J. Canny, "A computational approach to edge detection," *IEEE Transactions on PAMI*, vol. 8, no. 6, pp. 679–698, 1986.

[9] J. Shewchuk, "Triangle: Engineering a 2d quality mesh generator and delaunay triangulator," *ACM 1st Workshop on Applied Computational Geometry, Philadelphia*, 1996.

[10] H. Shimazaki and S. Shinomoto, "A method for selecting the bin size of a time histogram," *Neural Computation*, vol. 19, pp. 1503–1527, 2007.

[11] "The berkeley segmentation dataset and benchmark," http://www.cs.berkeley.edu/projects/vision/grouping/segbench/.

[12] T. Cour, F. Benezit, and J. Shi, "Multiscale normalized cuts segmentation toolbox for matlab version 1.0," 2006, http://www.seas.upenn.edu/∼timothee.

[13] B. Georgescu and C. Christoudias, "Edge detection and image segmentation system (edison) v1.1," 2002, Center for Advanced Information Processing, Rutgers University.

[14] S. Paris and F. Durand, "A topological approach to hierarchical image segmentation using mean shift," *Proceedings of CVPR'07*, 2007.

[15] Z. Tu and S-C Zhu, "Image segmentation by data-driven markov chain monte carlo," *IEE Transactions on PAMI*, vol. 24, no. 5, pp. 657–673, 5 2002.

[16] M. Tabb and N. Ahuja, "Multiscale image segmentation by integrated edge and region detection," *IEEE Transactions on Image Processing*, vol. 6, no. 5, pp. 642–655, 5 1997.

[17] X. Ren and J. Malik, "Learning a classification model for segmentation," *Proceedings of ICCV'03*, 2003.

[18] M. P. Kumar, P.H.S Torr, and A. Zisserman, "Objcut," *Proceedings of CVPR'05*, 2005.

[19] J. H. Elder and R. M. Goldberg, "Ecological statistics of gestalt laws for the perceptual organization of contours," *Journal of Vision*, vol. 2, no. 4, pp. 324–353, 5 2002.