

# From Contours to Regions: An Empirical Evaluation \*

Pablo Arbeláez<sup>1</sup>, Michael Maire<sup>1</sup>, Charless Fowlkes<sup>2</sup>, and Jitendra Malik<sup>1</sup>

<sup>1</sup>University of California at Berkeley - Berkeley, CA 94720

<sup>2</sup>University of California at Irvine - Irvine, CA 92697

{arbelaez,mmaire,malik}@eecs.berkeley.edu, fowlkes@ics.uci.edu

## Abstract

We propose a generic grouping algorithm that constructs a hierarchy of regions from the output of any contour detector. Our method consists of two steps, an Oriented Watershed Transform (OWT) to form initial regions from contours, followed by construction of an Ultrametric Contour Map (UCM) defining a hierarchical segmentation. We provide extensive experimental evaluation to demonstrate that, when coupled to a high-performance contour detector, the OWT-UCM algorithm produces state-of-the-art image segmentations. These hierarchical segmentations can optionally be further refined by user-specified annotations.

## 1. Introduction

Applications such as object recognition [34, 25, 1, 19] and monocular inference of 3D structure [20, 41] have led to a renewed interest in algorithms for automatic segmentation of an image into closed regions. Segments come with their own scale estimates and provide natural domains for computing features used in recognition. Many visual tasks can also benefit from the reduction in complexity achieved by transforming an image with millions of pixels into a few hundred or thousand “superpixels” [39].

A broad family of approaches to segmentation involve integrating features such as brightness, color, or texture over local image patches and then clustering those features based on, *e.g.*, fitting mixture models [5, 47], mode-finding [10], or graph partitioning [42, 24, 44, 15]. While this is by no means the only approach taken (see *e.g.* the vast literature inspired by variational formulations [30, 29] and level set techniques [26]), three algorithms in this category appear to be the most widely used as sources of image segments in recent applications, due to a combination of reasonable performance and publicly available implementations:

- Felzenszwalb and Huttenlocher’s graph based region merging [15]

- Comaniciu and Meer’s Mean Shift [10]
- Shi and Malik’s Normalized Cuts [42] and its multi-scale variant due to Cour *et al.* [11]

There does not appear to be a consensus about which of these algorithms is best. Felzenszwalb and Huttenlocher [15] is typically used in high recall settings to create a gross oversegmentation into thousands of superpixels. Mean Shift and Normalized Cuts provide better precision, but often produce artifacts by breaking large uniform regions (*e.g.* sky) into chunks.

The problem of oversegmentation is common across approaches based on feature clustering since smooth changes in texture or brightness due to perspective or shading can cause patches to appear dissimilar despite belonging to the same image region. Contour detection ignores such smooth variations by directly searching for locations in the image where brightness or other features undergo rapid local changes [9, 33]. These high-gradient edge fragments can then be linked together in order to identify extended, smooth contours [32, 46, 13, 38].

There have been significant advances in contour detection in the last few years [16, 37, 27, 12, 48, 22, 36]. This progress is best summarized by the benchmark results in Figure 1 (left), which show the precision and recall of a contour detector with respect to human-marked boundaries based on the evaluation methodology proposed in [27]. Despite this progress, without some mechanism for enforcing closure, a segmentation built up from locally detected contours will often mistakenly join regions due to leaks in the bounding contour, resulting in an under-segmentation.

In this paper, we propose an algorithm that produces a hierarchical segmentation from the output of any contour detector, while avoiding these difficulties. We introduce a new variant of the watershed transform [6, 31], the Oriented Watershed Transform (OWT), for producing a set of initial regions from contour detector output. We then construct an Ultrametric Contour Map (UCM) [2] from the boundaries of these initial regions. This sequence of operations (OWT-UCM) can be seen as generic machinery for going from

---

\*This work was supported by ONR MURI N00014-06-1-0734.

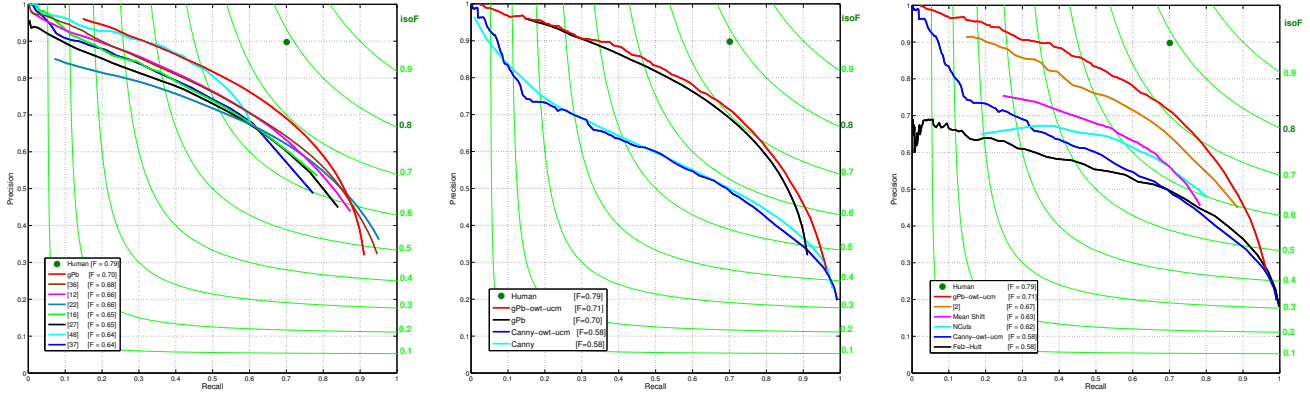


Figure 1. **Evaluating boundaries on the Berkeley dataset [17]. Left: Contours.** Leading approaches to contour detection are ranked according to their F-measure (harmonic mean of precision and recall) with respect to human ground-truth boundaries. Iso-F curves are shown in green. The best detector, *gPb* [23], does not produce closed boundaries required to form a segmentation. **Middle: Hierarchical Regions.** Our algorithm (OWT-UCM) produces a hierarchical segmentation from the output of any contour detector. Comparing the resulting segment boundaries to the original contours shows that this method constructs regions without losing performance on the boundary benchmark. In fact, we obtain a boost in performance when using the *gPb* detector as input. The quality of the contour detector (*gPb* vs Canny) on which we build significantly influences the quality of the resulting segmentation. **Right: Segmentations.** Paired with *gPb* as input, our hierarchical segmentation algorithm *gPb-owt-ucm* produces segments whose boundaries match human ground-truth better than those produced by alternative segmentation approaches such as Mean Shift, Normalized Cuts, or region-merging (Felz-Hutt).

contours to a hierarchical region tree. Contours encoded in the resulting hierarchical segmentation retain real-valued weights indicating their likelihood of being a true boundary. For a given threshold, the output is a set of closed contours that can be treated as either a segmentation or as a boundary detector for the purposes of benchmarking.

To establish the value of the OWT-UCM algorithm, we examine a number of different benchmark metrics and standard datasets for both boundary and region detection. Based on this extensive testing, we report two important results, illustrated in Figure 1 (middle) and (right), respectively:

- Weighted boundary maps can be converted into hierarchical segmentations without loss of boundary precision or recall. (Section 2)
- Using the *gPb* contour detector [23] as input, our method, *gPb-owt-ucm* provides a powerful mid-level grouping mechanism which outperforms all existing segmentation algorithms. (Section 3)

## 2. Contours to Hierarchical Regions

We consider a contour detector, whose output  $E(x, y, \theta)$  predicts the probability of an image boundary at location  $(x, y)$  and orientation  $\theta$ . We build hierarchical regions by exploiting the information in this contour signal using a sequence of two transformations, the Oriented Watershed Transform (OWT) and Ultrametric Contour Map (UCM), detailed below.

### 2.1. Oriented Watershed Transform

Using the contour signal, we first construct a finest partition for the hierarchy, an over-segmentation whose regions determine the highest level of detail considered. This is done by computing  $E(x, y) = \max_{\theta} E(x, y, \theta)$ , the maximal response of the contour detector over orientations. We take the regional minima of  $E(x, y)$  as seed locations for homogeneous segments and apply the watershed transform used in mathematical morphology [6, 31] on the topographic surface defined by  $E(x, y)$ . The catchment basins of the minima, denoted  $\mathcal{P}_0$ , provide the regions of the finest partition and the corresponding watershed arcs,  $\mathcal{K}_0$ , the possible locations of the boundaries.

Next, we transfer the strength of the boundaries, given by  $E(x, y, \theta)$ , to the locations  $\mathcal{K}_0$ . For this purpose, we approximate the watershed arcs with line segments, and weight each point in  $\mathcal{K}_0$  by the  $E(x, y, \theta)$  value at that point, in the direction  $\theta$  given by the orientation of the corresponding line segment, as detailed in Figure 2. This procedure, which we call the Oriented Watershed Transform (OWT), enforces consistency between the strength of the boundaries of  $\mathcal{K}_0$  and the underlying  $E(x, y, \theta)$  signal and removes artifacts of the standard watershed algorithm.

### 2.2. Ultrametric Contour Map

Contours have the advantage that it is fairly straightforward to represent uncertainty in the presence of a true underlying contour, *i.e.* by associating a binary random variable to it. It is not immediately obvious how to rep-

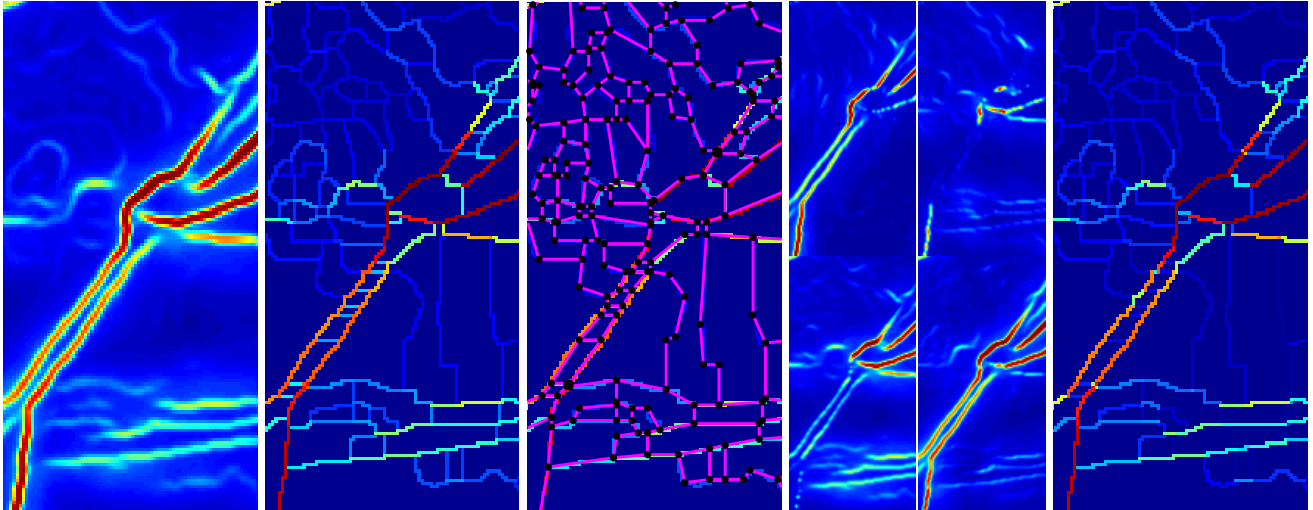


Figure 2. **Oriented Watershed Transform.** **Left:** Input boundary signal  $E(x, y) = \max_{\theta} E(x, y, \theta)$ . **Middle Left:** Watershed arcs computed from  $E(x, y)$ . Note that thin regions give rise to artifacts. **Middle:** Watershed arcs with an approximating straight line segment subdivision overlaid. We compute this subdivision in a scale-invariant manner by recursively breaking an arc at the point maximally distant from the straight line segment connecting its endpoints. Subdivision terminates when the distance from the line segment to every point on the arc is less than a fixed fraction of the segment length. **Middle Right:** Oriented boundary strength  $E(x, y, \theta)$  for four orientations  $\theta$ . In practice, we sample eight orientations. **Right:** Watershed arcs reweighted according to  $E$  at the orientation of their associated line segments. Artifacts, such as the horizontal contours breaking the long skinny segment, are suppressed as their orientations do not agree with the underlying  $E(x, y, \theta)$  signal.

resent uncertainty about a segmentation. One possibility, which we exploit here, is the Ultrametric Contour Map (UCM) [2] which defines a duality between closed, non-self-intersecting weighted contours and a hierarchy of regions. Making this shift in representation from a single segmentation to a nested collection of segmentations turns out to be very powerful.

The hierarchy is constructed by a greedy graph-based region merging algorithm. An initial graph is defined, where the nodes are the regions in  $\mathcal{P}_0$ , the links join adjacent regions and are weighted by a measure of similarity between regions. The algorithm proceeds by sorting the links by similarity and iteratively merging the most similar regions. This process produces a tree of regions, where the leaves are the elements of  $\mathcal{P}_0$ , the root is the entire image domain, and the regions are ordered by the inclusion relation.

We define similarity between two adjacent regions as the average strength of their common boundary in  $\mathcal{K}_0$ , initialized by the OWT. Since this value cannot decrease during the merging process, the above algorithm is guaranteed to produce an ultrametric distance on  $\mathcal{P}_0 \times \mathcal{P}_0$  [2]. As a consequence, the constructed region tree has the structure of an indexed hierarchy and can be described by a dendrogram, where the height of each segment is the value of the similarity at which it first appears and the distance between two regions is the height of the smallest segment in the hierarchy containing them. Furthermore, the whole hierarchy

can be represented as an Ultrametric Contour Map (UCM), the real-valued image obtained by weighting each boundary between two regions by its scale of disappearance.

Figure 3 presents an example of our method. The UCM is a weighted contour image that, by construction, has the remarkable property of producing a set of closed curves for any threshold. Conversely, it is a convenient representation of the region tree since the segmentation at a scale  $k$  can be easily retrieved by thresholding the UCM at level  $k$ . Since our notion of scale is the average contour strength, the UCM values reflect the contrast between neighboring regions.

### 2.3. Contour Detector Choice

While the OWT-UCM algorithm can use any source of contours, *e.g.* the Canny edge detector before thresholding, for the input  $E(x, y, \theta)$  signal, for best results, we employ the *gPb* detector introduced in our previous work [23].

The *gPb* detector combines multiscale brightness, color, and texture gradients, with an oriented spectral signal computed from these cues. In particular, we define a weighted combination of multiscale local cues

$$mPb(x, y, \theta) = \sum_s \sum_i \alpha_{i,s} G_{i,\sigma(s)}(x, y, \theta) \quad (1)$$

where  $s$  indexes scales,  $i$  indexes feature channels (brightness, color, texture), and  $G_{i,\sigma(s)}(x, y, \theta)$  measures the difference in channel  $i$  between two halves of a disc of radius

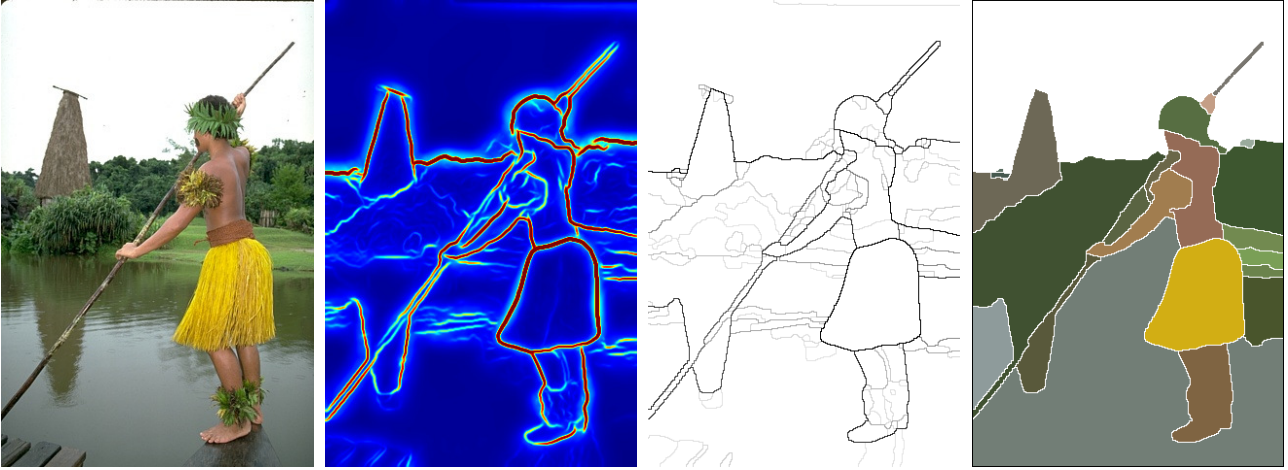


Figure 3. **Overview of our approach.** **Left:** Original image. **Middle Left:** Maximal response of contour detector  $gPb$  over orientations. **Middle Right:** Weighted contours resulting from the Oriented Watershed Transform - Ultrametric Contour Map (OWT-UCM) algorithm using  $gPb$  as input. **Right:** Segmentation obtained by thresholding the UCM at level 0.4, with segments represented by their mean color.

$\sigma(s)$  centered at  $(x, y)$  and divided by a diameter at angle  $\theta$ .

From  $mPb$ , we define an affinity matrix  $W$  between pixels using the intervening contour cue [18]. Writing  $D_{ii} = \sum_j W_{ij}$ , we solve for the generalized eigenvectors  $\{v_1, \dots, v_n\}$  of the system  $(D - W)v = \lambda Dv$  corresponding to the  $n$  smallest eigenvalues. Treating each eigenvector  $v_k$  as an image, we convolve with Gaussian directional derivatives to obtain oriented contour signals  $sPb_{v_k}(x, y, \theta)$ , and combine into

$$sPb(x, y, \theta) = \sum_{k=1}^n \frac{1}{\sqrt{\lambda_k}} \cdot sPb_{v_k}(x, y, \theta) \quad (2)$$

The final  $gPb$  detector is a weighted sum of local and spectral signals, which is subsequently rescaled using a sigmoid:

$$gPb(x, y, \theta) = \sum_s \sum_i \beta_{i,s} G_{i,\sigma(s)}(x, y, \theta) + \gamma \cdot sPb(x, y, \theta) \quad (3)$$

We report experiments using both  $gPb$  as well as the baseline Canny detector, and refer to the resulting segmentation algorithms as  $gPb$ -owt-ucm and Canny-owt-ucm, respectively. Figure 4 illustrates results of  $gPb$ -owt-ucm on images from the Berkeley Segmentation Dataset [17].

### 3. Empirical Evaluation

To provide a basis of comparison for the performance of the OWT-UCM algorithm, we make use of the Felzenszwalb and Huttenlocher (Felz-Hutt) [15], Mean Shift [10], and Multiscale Normalized Cuts (NCuts) [11] segmentation methods. We evaluate each method using multiple benchmark criteria. In this section, we present the details of our extensive evaluation framework.

### 3.1. Benchmarks

The Berkeley Segmentation Dataset (BSDS) [17] consists of 300 natural images, manually segmented by a number of different subjects. The ground-truth data for this large collection shows the diversity, yet high consistency, of human segmentation. We examine metrics for evaluating both boundaries and regions against human ground-truth.

#### 3.1.1 Precision-Recall on Boundaries

The boundary-based evaluation methodology developed by Martin *et al.* [27] on the BSDS has become a standard, as demonstrated by its widespread use [37, 16, 12, 2, 48, 22, 36, 23]. This framework considers two aspects of detection performance. Precision measures the fraction of true positives in the contours produced by a detector. Recall measures the fraction of ground-truth boundaries detected. For detectors that provide real-valued outputs, one obtains a curve parameterized by detection threshold, quantifying performance across operating regimes. The global F-measure, defined as the harmonic mean of precision and recall, provides a useful summary score for the algorithm.

In our experiments, we report three different quantities for an algorithm: the **Optimal Dataset Scale (ODS)** or best F-measure on the dataset for a fixed scale, the **Optimal Image Scale (OIS)** or aggregate F-measure on the dataset for the best scale in each image, and the **Average Precision (AP)** on the full recall range (equivalently, the area under the precision-recall curve), shown in Table 1 for the BSDS.

This benchmarking methodology possesses the appealing property that it allows the comparison of region-based segmentation and contour detection methods in the same framework, as illustrated in Figure 1. Any segmentation al-



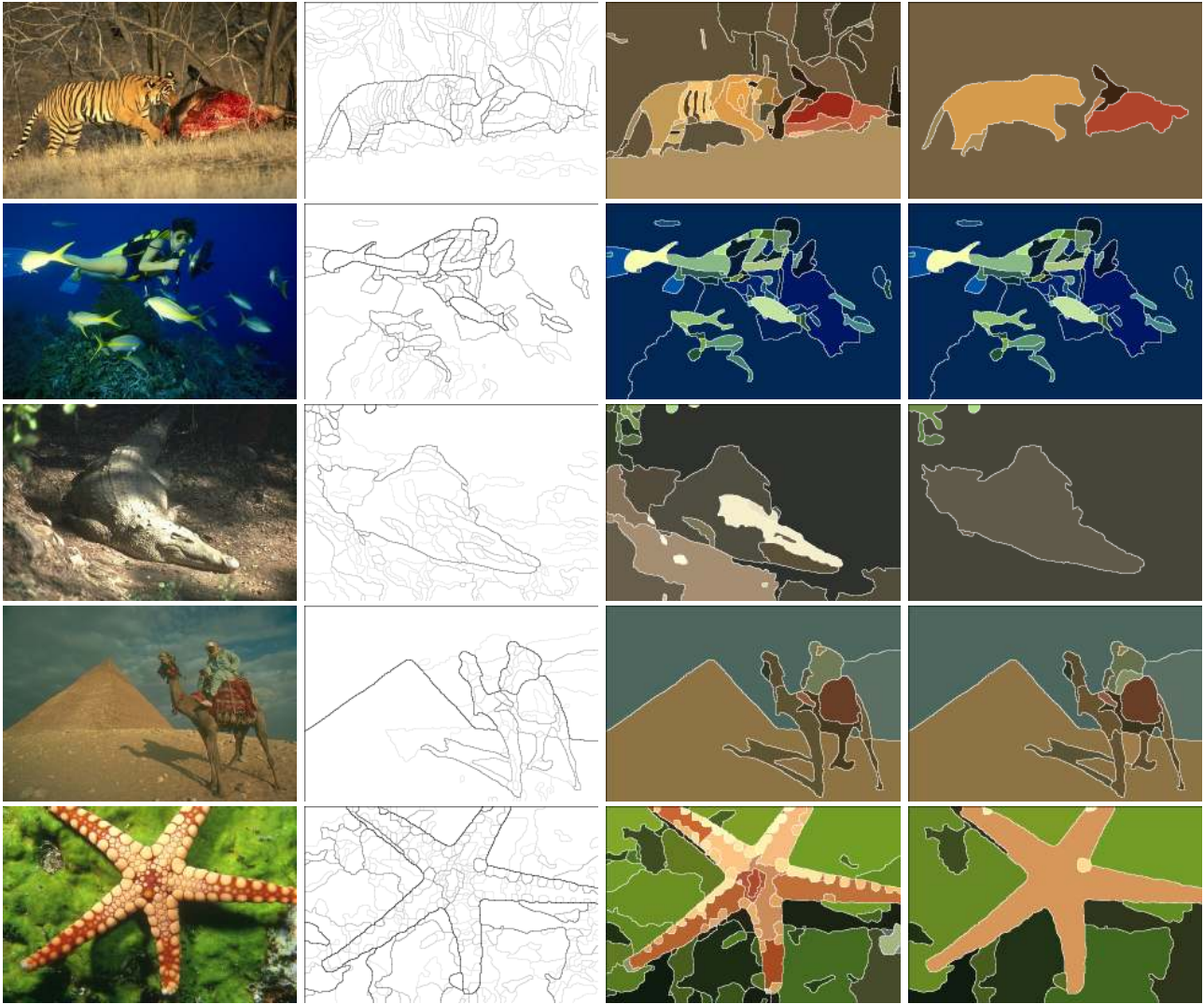


Figure 4. **Hierarchical segmentation results.** From left to right: Original image, Ultrametric Contour Map (UCM) produced by *gPb-owt-ucm*, and segmentations obtained by thresholding at the optimal dataset scale (ODS) and optimal image scale (OIS).

gorithm automatically provides contours in the form of the boundaries of the regions in the segmentation.

However, for segmentation algorithms, a methodology that directly evaluates the quality of the segments is also desirable. Some types of errors, *e.g.* a missing pixel in the boundary between two regions, may not be reflected in the boundary benchmark, but can have substantial consequences for segmentation quality, *e.g.* incorrectly merging two large regions. It can also be argued that the boundary benchmark favors contour detectors over segmentation methods, since the former are not burdened with the constraint of producing closed curves. We therefore also consider various region-based metrics.

### 3.1.2 Variation of Information [28]

This metric was introduced for the purpose of clustering comparison. It measures the distance between two segmentations in terms of their average conditional entropy given by

$$VI(C, C') = H(C) + H(C') - 2I(C, C') \quad (4)$$

where  $H$  and  $I$  represent respectively the entropies and mutual information between two clusterings of data  $C$  and  $C'$ . In our case, the two clusterings are test and ground-truth segmentations. Although  $VI$  possesses some interesting theoretical properties [28], its perceptual meaning and applicability in the presence of several ground-truth segmentations remains unclear.

Method	ODS	OIS	AP
human	0.79	0.79	—
gPb-owt-ucm	<b>0.71</b>	<b>0.74</b>	<b>0.77</b>
Mean Shift	0.63	0.66	0.62
NCuts	0.62	0.66	0.59
Canny-owt-ucm	0.58	0.63	0.59
Felz-Hutt	0.58	0.62	0.54
gPb	0.70	0.72	0.75
Canny	0.58	0.62	0.60

Table 1. **Boundary benchmarks on the BSDS.** We benchmark boundaries produced by five different segmentation methods (upper table) and two contour detectors (lower table). Shown are the F-measures when choosing an optimal scale for the entire dataset (ODS) or per image (OIS), as well as the average precision (AP). Figure 1 shows the full precision-recall curves for the boundaries produced by these algorithms.

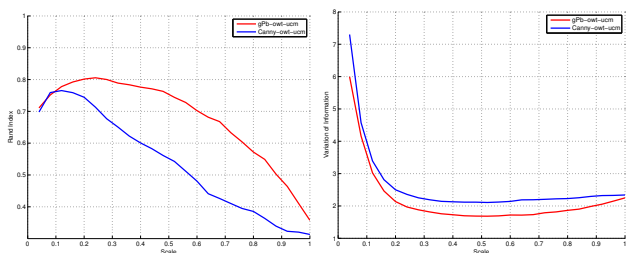


Figure 5. **Evaluating regions on the BSDS.** Contour detector influence on segmentation quality is evident when benchmarking the regions of the resulting hierarchical segmentation. **Left:** Probabilistic Rand Index. **Right:** Variation of Information.

### 3.1.3 Rand Index [35]

Originally, the Rand Index [35] was introduced for general clustering evaluation. It operates by comparing the compatibility of assignments between pairs of elements in the clusters. In our case, the Rand Index between test and ground-truth segmentations  $S$  and  $G$  is given by the sum of the number of pairs of pixels that have the same label in  $S$  and  $G$  and those that have different labels in both segmentations, divided by the total number of pairs of pixels. Variants of the Rand Index have been proposed [45, 47] for dealing with the case of multiple ground-truth segmentations. Given a set of ground-truth segmentations  $\{G_k\}$ , the Probabilistic Rand Index is defined as

$$PRI(S, \{G_k\}) = \frac{1}{T} \sum_{i < j} [c_{ij} p_{ij} + (1 - c_{ij})(1 - p_{ij})] \quad (5)$$

where  $c_{ij}$  is the event that pixels  $i$  and  $j$  have the same label and  $p_{ij}$  its probability. When the sample mean is used to estimate  $p_{ij}$ , (5) amounts to averaging the Rand Index among different ground-truth segmentations. However, the  $PRI$  has been reported to suffer from a small dynamic range

Method	ODS	OIS	Best	PRI	VI
human	0.73	0.73	—	0.87	1.16
gPb-owt-ucm	<b>0.58</b>	<b>0.64</b>	<b>0.74</b>	<b>0.81</b>	<b>1.68</b>
Mean Shift	0.54	0.58	0.64	0.78	1.83
Felz-Hutt	0.51	0.58	0.68	0.77	2.15
Canny-owt-ucm	0.48	0.56	0.67	0.77	2.11
NCuts	0.44	0.53	0.66	0.75	2.18

Table 2. **Region benchmarks on the BSDS.** For each segmentation method, the leftmost three columns report the score of the covering of ground-truth segments according to optimal dataset scale (ODS), optimal image scale (OIS), or Best covering criteria. The rightmost two columns compare the segmentation methods against ground-truth using the probabilistic Rand Index (PRI) and Variation of Information (VI) benchmarks, respectively.

[45, 47], and its values across images and algorithms are often very similar. In [45], this drawback is addressed by normalization with an empirical estimation of its expected value.

### 3.1.4 Segmentation Covering

The **overlap** between two regions  $R$  and  $R'$ , defined as:

$$\mathcal{O}(R, R') = \frac{|R \cap R'|}{|R \cup R'|} \quad (6)$$

has been used for the evaluation of the pixel-wise classification task in recognition [25, 14].

We define the **covering** of a segmentation  $S$  by a segmentation  $S'$  as

$$\mathcal{C}(S' \rightarrow S) = \frac{1}{N} \sum_{R \in S} |R| \cdot \max_{R' \in S'} \mathcal{O}(R, R'), \quad (7)$$

where  $N$  denotes the total number of pixels in the image.

Similarly, the covering of a machine segmentation  $S$  by a family of ground truth segmentations  $\{G_i\}$  is defined by, first covering  $S$  separately with each human map  $\{G_i\}$  in turn, and then averaging over the different humans, so that to achieve perfect covering the machine segmentation must explain all of the human data.

We can then define two quality descriptors for regions: the covering of  $S$  by  $\{G_i\}$  and the covering of  $\{G_i\}$  by  $S$ . Due to space constraints, we only include results for the latter. For a family of machine segmentations  $\{S_i\}$ , corresponding to different scales of a hierarchical algorithm or different sets of parameters, we report the **Optimal Dataset Scale (ODS)**, **Optimal Image Scale (OIS)**, and the **Best possible covering** of the ground-truth by segments in  $\{S_i\}$ .

Figure 5 and Table 2 present region benchmarks on the BSDS. While the relative ranking of segmentation algorithms remains fairly consistent across different benchmark criteria, the boundary benchmark (Table 1) appears most capable of discriminating performance.

MSRC	ODS	OIS	Best
gPb-owt-ucm	<b>0.66</b>	<b>0.75</b>	<b>0.78</b>
Canny-owt-ucm	0.57	0.68	0.72
PASCAL08	ODS	OIS	Best
gPb-owt-ucm	<b>0.45</b>	<b>0.58</b>	<b>0.61</b>
Canny-owt-ucm	0.40	0.53	0.55

Table 3. **Region benchmarks on MSRC and PASCAL08.** Shown are scores for the segment covering criteria used in Table 2.

### 3.2. Additional Datasets

We concentrated experiments on the BSDS because it is the most complete dataset available for our purposes, has been used in several publications, and has the advantage of providing several human-labeled segmentations per image. Table 3 reports the comparison between Canny-owt-ucm and *gPb*-owt-ucm on two other publicly available datasets:

- **MSRC** [43] The MSRC object recognition database is composed of 591 natural images with objects belonging to 21 classes. We evaluate performance using the ground-truth object instance labeling of [25], which is cleaner and more precise than the original data.
- **PASCAL 2008** [14] We use the train and validation sets of the segmentation task on the PASCAL challenge 2008, composed of 1023 images. This is one of the most difficult and varied datasets for recognition. We evaluate performance with respect to the object instance labels provided. Note that only objects belonging to the 20 categories of the challenge are labeled, and 76% of all pixels are unlabeled.

## 4. Interactive Segmentation

Until now, we have only discussed fully automatic image segmentation. Human assisted segmentation is relevant for many applications, and recent approaches rely on the graph-cuts formalism [7, 40, 21] or other energy minimization procedure [4] to extract single foreground regions.

It turns out that the segmentation trees generated by the OWT-UCM algorithm provide a natural starting point for user-assisted refinement. Following the procedure of [3], we can extend a partial labeling of regions to a full one by assigning to each unlabeled region the label of its closest (in terms of the ultrametric distance) labeled region. This procedure, illustrated in Figure 6, allows a user to obtain high quality results with minimal annotation.

## 5. Conclusion

Our segmentation algorithm, *gPb*-owt-ucm, offers the best performance on every dataset and for every benchmark

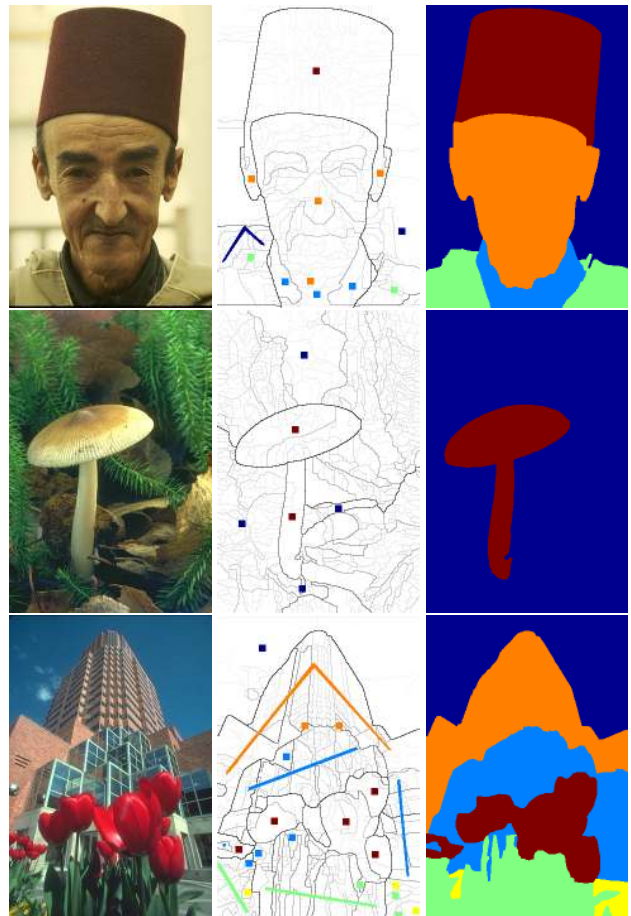


Figure 6. **Interactive segmentation.** **Left:** Original image. **Middle:** UCM produced by *gPb*-owt-ucm (grayscale) with additional user annotations (color dots and lines). **Right:** The region hierarchy defined by the UCM allows us to automatically propagate annotations to unlabeled segments, resulting in the desired labeling of the image with minimal user effort.

criterion we tested. In addition, this algorithm is straightforward, fast, has no parameters to tune, and supports interactive user refinement. Our generic grouping machinery has found use in optical flow [8] and object recognition [19] applications. We have made code available online<sup>1</sup>.

## References

- [1] N. Ahuja and S. Todorovic. Connected segmentation tree: a joint representation of region layout and hierarchy. *CVPR*, 2008.
- [2] P. Arbeláez. Boundary extraction in natural images using ultrametric contour maps. *POCV*, 2006.
- [3] P. Arbeláez and L. Cohen. Constrained image segmentation from hierarchical boundaries. *CVPR*, 2008.

<sup>1</sup><http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping>



- [4] S. Bagon, O. Boiman, and M. Irani. What is a good image segment? A unified approach to segment extraction. *ECCV*, 2008.
- [5] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color and texture-based image segmentation using EM and its application to content-based image retrieval. pages 675–682, 1998.
- [6] S. Beucher and F. Meyer. *Mathematical morphology in image processing*, chapter 12. Marcel Dekker, 1992.
- [7] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. *ICCV*, 2001.
- [8] T. Brox and J. Malik. Large displacement optical flow. *CVPR*, 2009.
- [9] J. Canny. A computational approach to edge detection. *PAMI*, 1986.
- [10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002.
- [11] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. *CVPR*, 2005.
- [12] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. *CVPR*, 2006.
- [13] J. Elder and S. Zucker. Computing contour closures. *ECCV*, 1996.
- [14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. PASCAL 2008 Results. <http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html>.
- [15] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based segmentation algorithm. *IJCV*, 2004.
- [16] P. Felzenszwalb and D. McAllester. A min-cover approach for finding salient curves. *POCV*, 2006.
- [17] C. Fowlkes, D. Martin, and J. Malik. The Berkeley Segmentation Dataset and Benchmark (BSDB). [www.cs.berkeley.edu/projects/vision/grouping/segbench/](http://www.cs.berkeley.edu/projects/vision/grouping/segbench/).
- [18] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches. *CVPR*, 2003.
- [19] C. Gu, J. Lim, P. Arbeláez and J. Malik. Recognition using regions. *CVPR*, 2009.
- [20] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. *ICCV*, 2005.
- [21] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *SIGGRAPH*, 2004.
- [22] J. Mairal, M. Leordeanu, F. Bach, M. Hebert and J. Ponce. Discriminative sparse image models for class-specific edge detection and image interpretation. *ECCV*, 2008.
- [23] M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. *CVPR*, 2008.
- [24] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 2001.
- [25] T. Malisiewicz and A. A. Efros. Improving spatial support for objects via multiple segmentations. *BMVC*, 2007.
- [26] R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: A level set approach. *PAMI*, 1995.
- [27] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *PAMI*, 2004.
- [28] M. Meila. Comparing clusterings: An axiomatic view. *ICML*, 2005.
- [29] J.-M. Morel and S. Solimini. *Variational methods in image segmentation*. Birkhäuser, 1995.
- [30] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions, and associated variational problems. *Comm. in Pure and Applied Math.*, pages 577–684, 1989.
- [31] L. Najman and M. Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *PAMI*, 1996.
- [32] P. Parent and S. Zucker. Trace inference, curvature consistency, and curve detection. *PAMI*, 1989.
- [33] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. *ICCV*, 1990.
- [34] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. *ICCV*, 2007.
- [35] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846850, 1971.
- [36] X. Ren. Multi-scale improves boundary detection in natural images. *ECCV*, 2008.
- [37] X. Ren, C. Fowlkes, and J. Malik. Scale-invariant contour completion using conditional random fields. *ICCV*, 2005.
- [38] X. Ren, C. Fowlkes, and J. Malik. Learning probabilistic models for contour completion in natural images. *IJCV*, 2008.
- [39] X. Ren and J. Malik. Learning a classification model for segmentation. *ICCV*, 2003.
- [40] C. Rother, V. Kolmogorov, A. Blake. “Grabcut”: Interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 2004.
- [41] A. Saxena, S. H. Chung, A. Y. Ng 3-D depth reconstruction from a single still image. *IJCV*, 2008.
- [42] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.
- [43] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. *ECCV*, 2006.
- [44] D. Tolliver and G. L. Miller. Graph partitioning by spectral rounding: applications in image segmentation and clustering. *CVPR*, 2006.
- [45] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *PAMI*, 2007.
- [46] L. Williams and D. Jacobs. Stochastic completion fields: a neural model of illusory contour shape and salience. *ICCV*, 1995.
- [47] A. Yang, J. Wright, Y. Ma, and S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *CVIU*, 2008.
- [48] Q. Zhu, G. Song, and J. Shi. Untangling cycles for contour grouping. *ICCV*, 2007.