Constrained Image Segmentation from Hierarchical Boundaries

Pablo Arbeláez
University of California, Berkeley
Berkeley, CA, 94720, USA
arbelaez@eecs.berkeley.edu

Laurent Cohen
CEREMADE, UMR CNRS 7534
Université Paris Dauphine, 75775 Paris, France
cohen@ceremade.dauphine.fr

Abstract

In this paper, we address the problem of constrained segmentation of natural images, in which a human user places one seed point inside each object of interest in the image and the task is to determine the object boundaries. For this purpose, we study the connection between seed-based and hierarchical segmentation. We consider an Ultrametric Contour Map (UCM), the representation of a hierarchy of segmentations as a real-valued boundary image. Starting from a set of seed points, we propose an algorithm for constructing Voronoi tessellations with respect to a distance defined by the UCM. As a result, the main contribution of the paper is a method that allows exploiting the information of any hierarchical scheme for constrained segmentation. Our algorithm is parameter-free, computationally efficient and robust. We prove the interest of the approach proposed by evaluating quantitatively the results with respect to ground-truth data.

1. Introduction

Image segmentation is one of the fundamental and most studied problems in computer vision. However, automatic segmentation remains essentially unsolved and, as will be discussed below, state-of-the-art algorithms are still far from human performance. Furthermore, in many contexts (*e.g*. medical imaging), the quality of a segmentation can only be judged *a posteriori*, by its pertinence for a particular application. In these cases, an interactive approach is often preferred, where a human operator interprets the semantic contents of the image, selects the objects of interest and the segmentation algorithm is used to extract them automatically from the background. In this paper, we address this *constrained* segmentation problem, illustrated in Figure 1, in a quantitative framework.

There is a large literature on interactive segmentation. A classical approach is the *markers and watersheds* method from morphology [3], where the watershed lines are computed on a topographic surface (usually a gradient modulus)

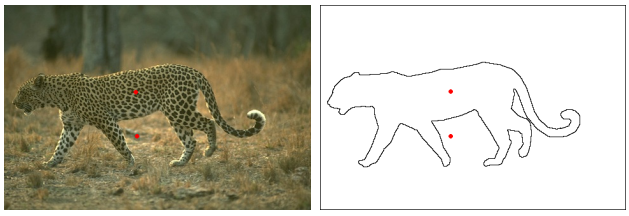

Figure 1. The problem studied in this paper: *With what accuracy can the true object boundaries be predicted, starting only from a single point inside each object?*

obtained by imposing a selected set of markers as only regional minima. Many recent approaches rely on graph-cuts, building on the min-cut / max-flow algorithm of [6]. The seminal work of [5] was improved in [4], where a Gaussian Mixture Markov Random Field model is proposed in order to learn parameters of color and contrast for figure and background. In [2], the segmentation is obtained by computing weighted geodesic distances to the user input.

Research in automatic boundary detection goes back to the early days of computer vision. Local approaches to the problem predict the presence of an edge at a given location by examining the image information on a neighborhood around it. Classic local detectors, such as Canny's [7], look for discontinuities in the brightness channel, while more recent approaches consider also color and texture information and rely on learning techniques for cue combination [14, 9]. Another category of methods considers global image information to address the task. This goal can be achieved through graph partitioning formulations [20, 12, 13, 21]. Other recent techniques are [10], where an approximate solution to the minimum-cover problem is used to extract smooth salient curves and [18], where curvilinear continuity of contours is enforced with Conditional Random Fields.

In previous work [1], we proposed a boundary-based formulation for hierarchical segmentation relying on the Ultrametric Contour Map (UCM), the real-valued image defined by weighting the boundary of each region in a hierarchy of segmentations by its scale of disappearance. The

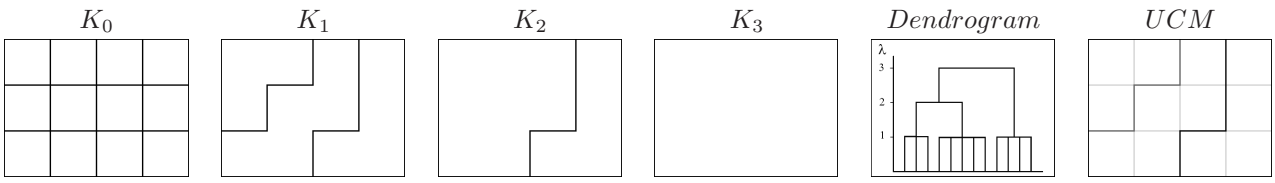

Figure 2. From left to right: Indexed hierarchy of segmentations, its representation as a dendrogram and as an Ultrametric Contour Map.

UCMs proposed in that paper achieve the best performance reported to date on the Berkeley Segmentation Dataset and Benchmark (BSDS) [11]. Furthermore, in contrast to most of the other leading boundary detectors, thresholding an UCM provides always a set of closed curves, the boundaries of a segmentation in regions.

In this paper, we study the connection between seed-based and hierarchical segmentation. We propose a front propagation algorithm on the UCM that constructs Voronoi tessellations with respect to collections of subsets of the image domain. Consequently, the main contribution of the paper is a parameter-free, computationally efficient and robust algorithm for constructing constrained segmentations from any hierarchical approach.

The quantitative evaluation of interactive segmentation has received little attention in the past. Although the Precision-Recall curves proposed in [14] have become a standard methodology for the evaluation of automatic boundary detection, the empiric validation of interactive approaches depends in great extent on the amount of user intervention allowed. As an example, in the framework proposed by Blake *et al*. [4], a user trimap (dividing the image into a single connected figure, a background and an unknown region) is provided. The algorithms are evaluated by their ability to determine the location of the boundary in the unknown band surrounding it, while allowed using the information in the rest of the image. In this paper, we are interested in the more challenging problem of determining the boundaries of any number of objects. Furthermore, in order to evaluate machine performance for this task separately from the amount of human intervention, we restrict the latter to the selection of a single point inside each object. Since there is not, to the best of our knowledge, a study in which such a problem has been evaluated quantitatively, we suggest a method to extract ground-truth seeds from the human segmentations of the BSDS, which allows measuring performance with the Precision-Recall methodology.

The rest of this paper is organized as follows. Section 2 recalls the definition of Ultrametric Contour Maps. Section 3 presents our algorithm for constrained segmentation and Section 4 contains the results. Section 5 presents some concluding remarks.

2. Ultrametric Contour Maps

In this section, we recall the formulation of hierarchical segmentation in terms of boundaries. For this purpose, we follow the presentation of [1].

A traditional way of thinking about hierarchical segmentation is to consider a family of nested partitions $\{\mathcal{P}_\lambda\}_\lambda$ of the image domain Ω, associated to a *scale parameter* $\lambda \in [0, \lambda_M]$, a finest partition $\mathcal{P}_0$ (*e.g*., the pixel grid) and a coarsest partition $\mathcal{P}_M = \{\Omega\}$. This imposes a hierarchical structure to the set of all the regions in the family $\mathcal{H} = \{R \in \Omega \,|\, \exists \lambda : R \in \mathcal{P}_\lambda\}$. The hierarchy $\mathcal{H}$ can be represented by a tree, where the root is Ω, the leaves are the elements of $\mathcal{P}_0$ and the regions are ordered by inclusion.

The scale of appearance of each region in the hierarchy can then be used to define a *stratification index* f:

$$f(R) = \inf\{\lambda \in [0, \lambda_M] \mid R \in \mathcal{P}_\lambda\},\ \forall\ R \in \mathcal{H} \quad (1)$$

The couple $(\mathcal{H}, f)$ is called an *indexed hierarchy*. Defining a stratification index f amounts to assigning a unique height $f(R)$ to each node R in the tree of regions, which can then be represented as a dendrogram (see Figure 2).

A classical theorem of hierarchical data analysis states that the structure of indexed hierarchy $(\mathcal{H}, f)$ is equivalent to the definition of an *ultrametric distance* Υ between elements of $\mathcal{P}_0$, given by the index of the smallest region in the hierarchy containing them:

$$\Upsilon(x, y) = \inf\{f(R) \mid (x \subseteq R) \wedge (y \subseteq R) \wedge (R \in \mathcal{H})\}. \quad (2)$$

Alternatively, a segmentation can be defined by considering the boundaries, rather than the regions of a partition. This is the approach proposed by Morel *et al*. [16] on a continuous domain, where a segmentation K is a finite set of rectifiable Jordan curves and the regions are the connected components of $\Omega \setminus K$. A segmentation can therefore be expressed equivalently by its contours K or by the partition $\mathcal{P} = \{\overline{R_i}\}_i$ of Ω.

One can then define a Hierarchical Segmentation Operator (HSO) as a family $\{K_\lambda\}_\lambda$ with a finest set of contours K_0 and such that all the contours vanish at finite scale. The hierarchical structure is expressed in this case by the principle of *strong causality*:

$$\lambda \leq \lambda' \Rightarrow K_\lambda \supseteq K_{\lambda'}, \quad (3)$$

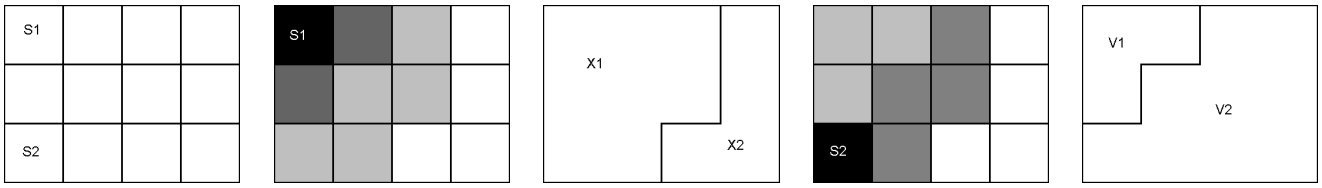

Figure 3. Example of classical metric tessellations on the ultrametric space of Figure 2. **(a):** two sites s_1 and s_2. **(b):** distance to s_1. **(c):** Tessellation obtained by thresholding the distance to s_1 at level 3. **(d):** distance to s_2. **(e):** Voronoi tessellation with respect to the two sites. Note that (e) cannot be obtained by thresholding the UCM at a constant level.

which establishes that the localization of contours is preserved through the scales.

An advantage of defining segmentations in terms of one-dimensional objects is that it allows representing a whole family of partitions as a single two dimensional image:

Let Υ be the ultrametric distance defined by a HSO. The **Ultrametric Contour Map (UCM)** associated to Υ is the application $\mathcal{C}(\Upsilon) : K_0 \rightarrow [0, \lambda_M]$ given by:

$$\mathcal{C}(\Upsilon)(\partial) = \inf\{\lambda \in [0, \lambda_M] \mid \partial \not\subseteq K_\lambda\}, \forall\, \partial \in K_0. \quad (4)$$

We call the number $\mathcal{C}(\Upsilon)(\partial)$ the **saliency** of contour ∂. Note the duality with the regions, the saliency of ∂ being its *scale of disappearance* from the hierarchy of contours.

The UCM is a representation of a HSO as a single real-valued image. Figures 2 and 4 present examples of UCMs. By definition, thresholding this soft boundary image at scale λ provides a set of closed curves, the segmentation K_λ.

Hence, the problem of hierarchical segmentation is formulated in this framework as defining an ultrametric distance from the image data such that the UCM models the boundaries of the objects. In [1], this is achieved by integrating local contour cues along the regions boundaries and combining this information with intra-region attributes.

3. Constrained Segmentation

The equivalence between ultrametric distances and indexed hierarchies gives rise to a type of metric tessellations specific to hierarchical segmentation: the set of all the balls of ultrametric radius λ corresponds to the segmentation K_λ and can be obtained by thresholding the corresponding UCM at level λ.

However, in order to study constrained segmentation, we are interested in partitioning the image domain with respect to an arbitrary set of points or subsets of $\mathcal{P}_0$. The classical tool for this purpose on general metric spaces is the Voronoi tessellation. Therefore, we focus in this section on the construction of this type of tessellations on ultrametric spaces.

3.1. Voronoi Tessellations

Let (X, D) be a metric space. The first obvious tessellation of X with respect to a subset S consists in choosing a threshold d for the distance to S and separating the points in two sets, depending on whether they are closer or farther than d from S: $X_1 = \{x \in X \mid D(x, S) \leq d\}$, $X_2 = X \setminus X_1$. X_1 is called a dilation of S of radius d and X_2 an erosion of its complementary set $X \setminus S$. If S contains only an isolated point s, then X_1 is the closed ball of radius d centered at s.

If we now have a collection of subsets of X instead of a single one, then we can partition the space with a Voronoi tessellation. Given a set of *sites* $S = \{S_i\}_{i \in I}$, one considers the region of influence, or Voronoi cell, of each S_i:

$$V_i = \{x \in X \mid D(x, S_i) < D(x, S_j), \forall j \in I \setminus \{i\}\}.$$

The Voronoi tessellation is then given by the set of Voronoi cells and the Voronoi diagram V, the points at the same distance of two or more sites:

$$V = \{x \in X \mid \exists i, j \in I \,:\, D(x, S_i) = D(x, S_j)\}.$$

Figures 3 and 4 present examples of the notions introduced above.

3.2. Segmentation Algorithm

In order to construct Voronoi tessellations on ultrametric spaces and study their interest for constrained segmentation, we need a way of measuring an ultrametric distance to a collection of subsets of the image domain. The algorithm we now describe extracts this information from the UCM.

The core of our approach is the front propagation strategy employed in algorithms such as Dijkstra's [8] or the Fast Marching [19]. A general front propagation algorithm is shown in Table 1. This type of methods operate on a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of links between nodes. In the case of the Fast Marching, $\mathcal{N}$ is a regular sampling of a subjacent continuous space.

The objective of a front propagation algorithm is to calculate the minimal distance $\mathcal{U}$ from any node in $\mathcal{N}$ to a starting set of nodes $\mathcal{M}$. It acts by partitioning $\mathcal{N}$ into three subsets: *Alive*, the nodes where the final value of $\mathcal{U}$ has been

Figure 4. Original image, UCM, and ultrametric distance to three seed points.

computed; *Trial*, the nodes with an estimate of $\mathcal{U}$, and *Far*, all the other nodes. The algorithm is initialized by setting *Alive* the nodes in $\mathcal{M}$, *Trial* their neighbors and *Far* all the other nodes. The initial estimate of $\mathcal{U}$ is 0 in $\mathcal{M}$ and $+\infty$ in $\mathcal{N} \setminus \mathcal{M}$.

Table 1. General Front Propagation Algorithm

While the priority queue is not empty:

- Let p be the *Trial* node with the smallest priority $\mathcal{Q}(p)$
- Move p from the *Trial* to the *Alive* set
- For each neighbor q of the current node p:
  - if q is *Far*, then add it to *Alive* and compute a new value for $\mathcal{U}(q)$
  - if q is *Alive*, recompute the value $\mathcal{U}(q)$, and update it if the new value is smaller
  - recompute the priority $\mathcal{Q}(q)$

A particular front propagation algorithm is determined by the implementation of the following: (1) A priority map $\mathcal{Q}$ that orders the set of *Trial* points. (2) A way to update the value $\mathcal{U}$ at a given *Trial* node. See [17] for specific instantiations for the Dijkstra, Fast Marching, A* and other algorithms.

In our algorithm, the input is an UCM and a set of seed pixels $\mathcal{M}$, marked with labels. Several seed pixels can share the same label. The outputs are the ultrametric distance from any pixel to $\mathcal{M}$ and a label for each pixel.

We consider the weighted graph $\mathcal{G} = (\mathcal{N}, \mathcal{L}, \mathcal{W})$, where the nodes $n \in \mathcal{N}$ are the elements of $\mathcal{P}_0$, a link $l(p,q) \in \mathcal{L}$ joins two adjacent elements and the weight $w(l(p,q)) \in \mathcal{W}$ equals the saliency of the boundary between p and q in the UCM.

Our priority map $\mathcal{Q}(p)$ and our function $\mathcal{U}(p)$ are both equal to $\Upsilon(p, \mathcal{M})$, the ultrametric distance from p to $\mathcal{M}$. Their value at a node p is given by the maximum weight w along a shortest path between $\mathcal{M}$ and p.

The seeds labels are propagated with the front and are updated with $\mathcal{U}$. Hence, all the elements in the Voronoi cells are assigned a unique label by the algorithm. However, for the elements in the Voronoi diagram, there is a choice to be made, since they are at the same distance of two or more seeds. We address this issue by ordering the nodes with equal priority in $\mathcal{Q}$ by their current label. Thus, all the elements in each connected component of the Voronoi diagram obtain the same label. Although this heuristic can potentially modify the output of the algorithm for a particular image, in practice it does not affect the global performance reported in the next section.

The computational complexity of our algorithm is the usual in front propagation methods: $O(Nlog(N))$, where N is the total number of nodes of the graph (the number of elements of $\mathcal{P}_0$ in our case).

Note that, if the ultrametric is obtained by a region merging strategy, then the segmentation can be constructed simultaneously with the hierarchy by preventing the merging of regions with different labels. However, by working directly on the UCM, our algorithm can be used with any hierarchical scheme.

4. Results

In the rest of the paper, we use as input to our method the UCMs of [1], noted $\mathcal{C}_A$ and available at [11].

Figure 5 shows how our constrained segmentation algorithm takes advantage of the coarseness of an ultrametric distance for its robustness. In this case, the boundary of the object to be extracted has the highest saliency in the UCM. As a consequence, the output of our algorithm remains unaffected, regardless of the exact location of the seed inside the object or in the background.

4.1. Ground-truth Seeds

In order to evaluate quantitatively a constrained segmentation algorithm, we need a point placed inside each object perceived in a large variety of natural images. Since there is not, to our knowledge, such an annotated database, we

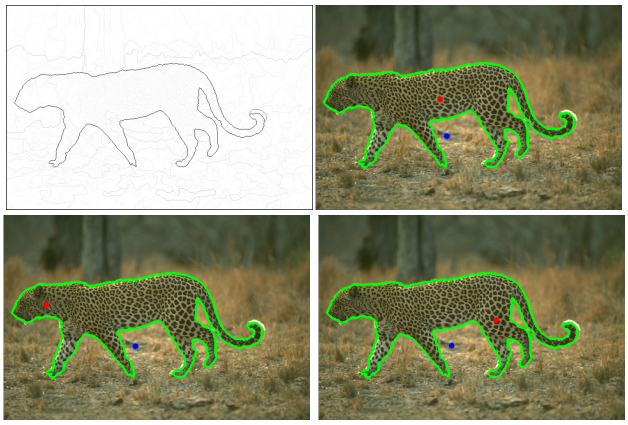

Figure 5. UCM and segmentations corresponding to three sets of seeds.

use for this purpose the human-labeled segmentations of the BSDS.

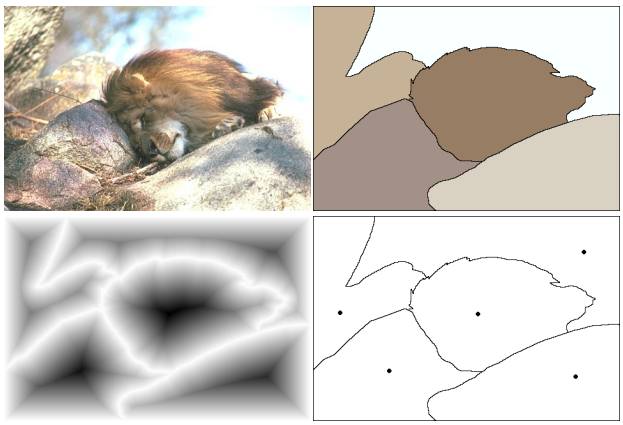

Figure 6. Determination of ground-truth seeds (see text)

The method for constructing ground-truth seeds is illustrated in Figure 6. Consider the natural image in Top-Left. Top-Right presents one of the human segmentations provided for this image, with segments represented by their mean color. For each segment, we pick as seed one of the points with highest Euclidean distance to the segment boundary. Bottom-Left presents the overall distance to the set of boundaries, with high values depicted with low intensities, and Bottom-Right shows the seeds for this segmentation. Now, the BSDS provides an average of five human segmentations for each image, and their level of detail and coherence can vary greatly from one another. In order to cope with the variability inherent to the dataset, we allow the use any of the segmentations provided for producing the ground-truth seeds of an image.

4.2. Evaluation

We measure the performance of a constrained segmentation algorithm by segmenting the images of the BSDS with respect to the ground-truth seeds and then evaluating the accuracy of the contours obtained with the precision-recall framework [14]. This standard methodology considers two quality descriptors. Precision, that measures the fraction of true positives in the contours obtained, and Recall, defined as the fraction of ground-truth boundaries detected. Since a constrained segmentation method produces a set of hard boundaries, a point in the precision-recall plane is obtained for each image. The overall performance of the method is measured by considering the global descriptors on the dataset. Finally, the global F-measure, defined as the harmonic mean of Precision and Recall, provides a score for the algorithm.

We use as a baseline the Matlab implementation of the *markers and watersheds* method from morphology [3]. The initial topographic surface is given by the Pb operator [14] before non-maxima suppression. The surface is modified with morphological reconstruction in order to make the ground-truth seeds its only regional minima and the watershed lines of this new surface provide the boundaries of the regions.

The evaluation results are presented in Figure 7. The black and red dots represent the performance of the watersheds (WS) and of our method respectively, extended along the iso-level lines of the function $F(P, R)$. Our method obtains a global recall of $R = 0.70$ and a global precision of $P = 0.76$, leading to an F-measure of $F = 0.73$, a significant improvement with respect to the watersheds, whose descriptors are: $R = 0.55$, $P = 0.81$ and $F = 0.65$.

The human consistency on this dataset, obtained by comparing the human segmentations between them, is $F = 0.79$, with overall recall of $R = 0.71$ and overall precision of $P = 0.90$ (green dot and line). Thus, our method detects roughly the same fraction of ground-truth boundaries as the human reference and the difference in the global scores comes from the lower Precision for the machine, indicating a higher fraction of false positives.

The other curves show for reference the leading boundary detection techniques evaluated on the BSDS. The score of $\mathcal{C}_A$, obtained by thesholding the UCMs at a fixed level (0.19) for all the images is the highest performance reported to date with this methodology for automatic segmentation. The global descriptors for $\mathcal{C}_A$ are $R = 0.69$, $P = 0.65$ and $F = 0.67$.

Although the problems addressed by $\mathcal{C}_A$ and our algorithm are different (automatic and constrained segmentation respectively), the evaluation shows that the introduction of semantic information through the selection of a set of ground-truth seeds and the application of our semi-automatic method improve the performance with respect to

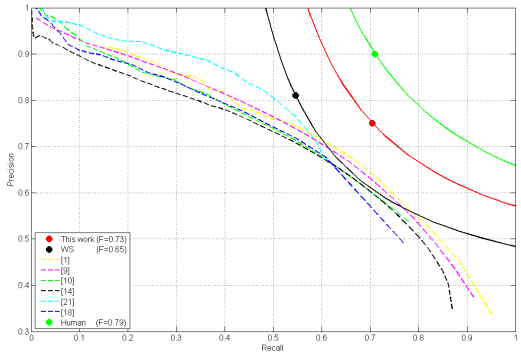

Figure 7. Benchmark Results

a fully automatic approach. The improvement is obtained mainly by reducing the fraction of false positives. Our results in this challenging database are promising for the application of our algorithm to interactive segmentation.

In terms of efficiency, the evaluation was carried out on a Pentium 4 at 3.2 Mhz. The average number of ground-truth seeds per image on the test set of the BSDS is 28. Starting from the UCMs and the seeds, the average computation time for the segmentations with our current C++ implementation is 0.2 s per image.

Figure 8 presents some benchmarked results and Figure 9 shows an example of interface of our method for interactive Figure/Ground labeling. In this case, the user can mark the objects of interest directly on the UCM, by assigning one label to each object and one label to the background. The human intervention in this case consists in drawing lines of different colors. Note that, by marking pixels on both sides of a boundary, the user can easily correct errors in the original UCM.

5. Conclusion

We proposed a straight-forward, robust and efficient algorithm for constrained segmentation. Its key idea is to exploit the information provided by the contours of a hierarchical segmentation method. We validated empirically our approach with respect to ground-truth segmentation data, by using a standard methodology and the UCMs of a leading hierarchical method. The evaluation results prove the potential of our algorithm for interactive segmentation applications.

Aknowledgement

This work was partially supported by ANR grant SURF-NT05-2-45825.

References

[1] P. Arbeláez. Boundary extraction in natural images using Ultrametric Contour Maps. In *POCV*, 2006.

[2] X. Bai and G. Sapiro. A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. In *ICCV*, 2007.

[3] S. Beucher and F. Meyer. *Mathematical Morphology in Image Processing*, chapter 12. Marcel Dekker, 1992.

[4] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*,2004.

[5] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, 2001.

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *TPAMI*, 26(9):1124–1137, September 2004.

[7] J. Canny. A computational approach to edge detection. *TPAMI*, 8(6):679–698, Nov. 1986.

[8] E. Dijkstra. A Note on two problems in connection with graphs. *Numerische Mathemetic*, 1:269–271, 1959.

[9] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *CVPR*, 2006.

[10] P. Felzenszwalb and D. McAllester. A min-cover approach for finding salient curves. In *POCV*, 2006.

[11] www.cs.berkeley.edu/projects/vision/grouping/segbench/.

[12] S. Mahamud, L. Williams, K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. *TPAMI*, 25(4):433–444, April 2003.

[13] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 43(1):7–27, June 2001.

[14] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *TPAMI*, 26(5):530–549, 2004.

[15] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application. In *ICCV*, 2001.

[16] J. Morel and S. Solimini. *Variational Methods in Image Segmentation*. Birkhauser, 1995.

[17] G. Peyre and L. D. Cohen. Landmark-based geodesic computation for heuristically driven path planning. In *CVPR*, 2006.

[18] X. Ren, C. Fowlkes, and J. Malik. Scale-invariant contour completion using Conditional Random Fields. In *ICCV*, 2005.

[19] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[20] S. Wang, T. Kubota, J. Siskind, and J. Wang. Salient closed boundary extraction with ratio contour. *TPAMI*, 27(4):546–561, April 2005.

[21] Q. Zhu, G. Song, and J. Shi. Untangling cycles for contour grouping. In *ICCV*, 2007.

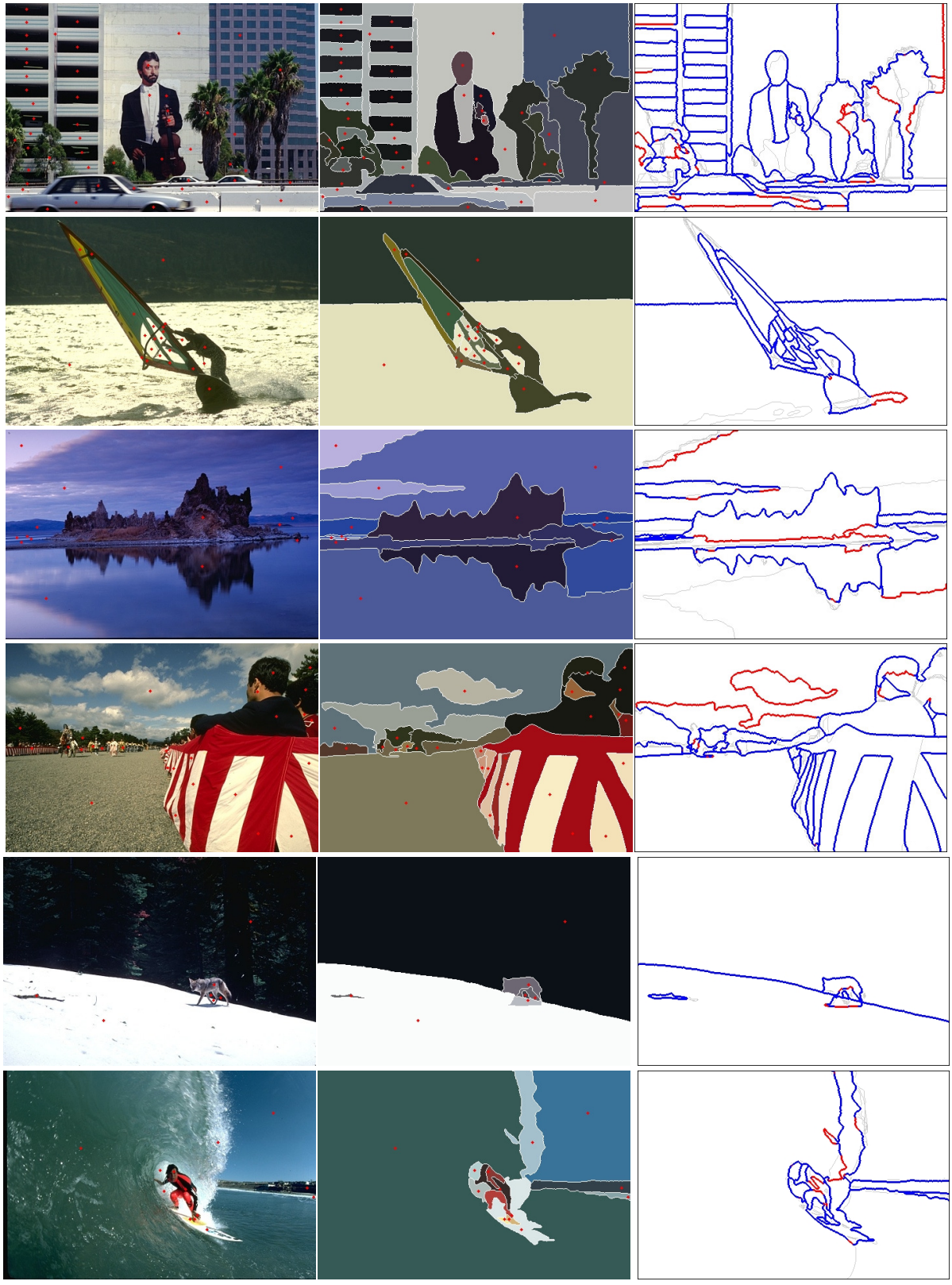

Figure 8. Examples of benchmarked results. From left to right: ground-truth seeds on original image, segmentation result and correspondence with ground-truth boundaries (true positives, false positives and missed detections represented in blue, red and gray respectively).

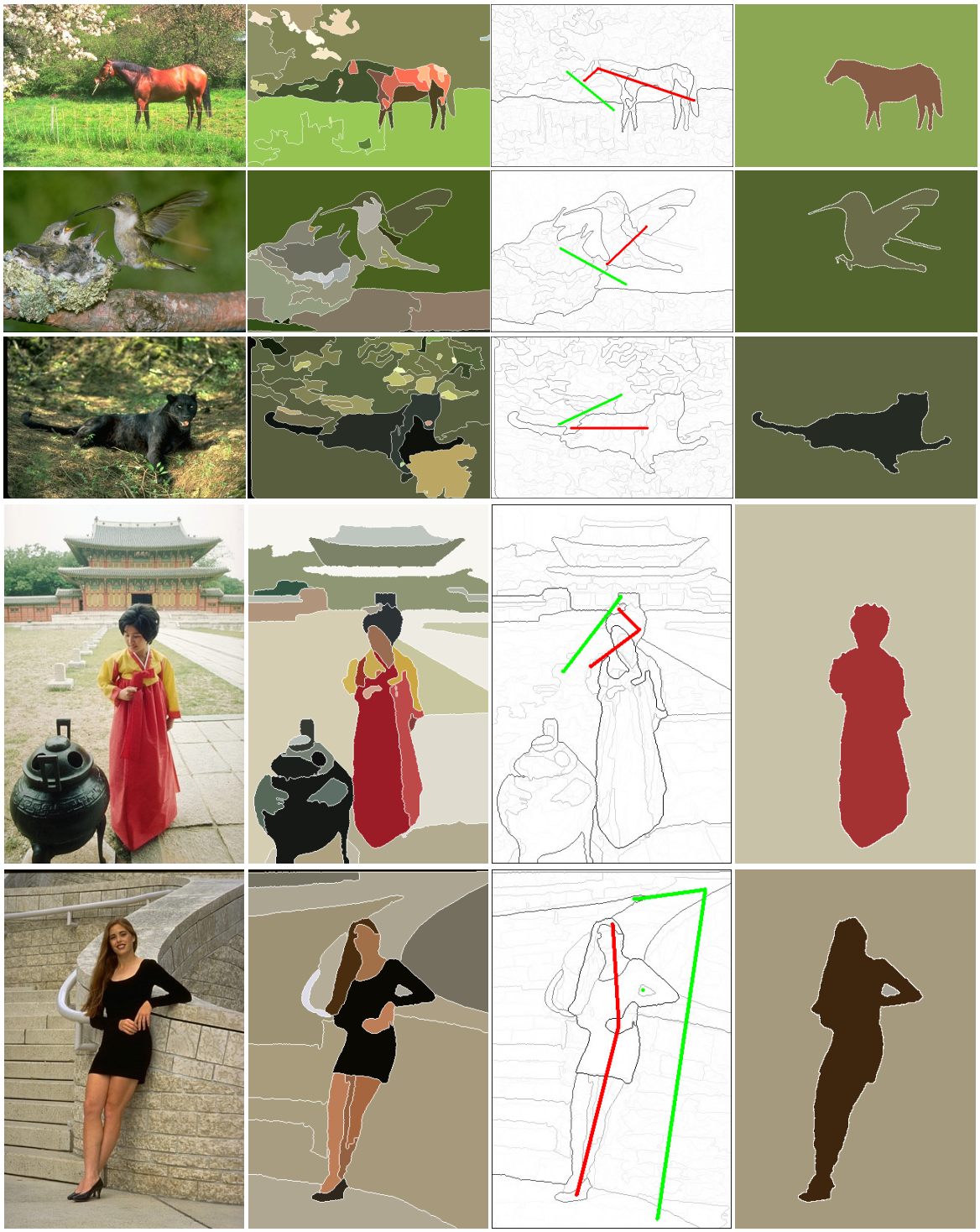

Figure 9. Application to interactive Figure / Ground segmentation. From left to right: Original image, automatic segmentation of $\mathcal{C}_A$ (for comparison), user input (thickened for better visualization) and result of our method.