Consensus Region Merging for Image Segmentation

Frank Nielsen
Sony Computer Science Laboratories Inc
3-14-13 Higashi Gotanda, Shinagawa-ku Tokyo, Japan
Email: Frank.Nielsen@acm.org

Richard Nock
UAG CEREGMIA
Campus de Schoelcher
Martinique, France
Email: rnock@martinique.univ-ag.fr

***Abstract*—Image segmentation is a fundamental task of image processing that consists in partitioning the image by grouping pixels into homogeneous regions. We propose a novel segmentation algorithm that consists in combining many runs of a simple and fast randomized segmentation algorithm. Our algorithm also yields a soft-edge closed contour detector. We describe the theoretical probabilistic framework and report on our implementation that experimentally corroborates that performance increases with the number of runs.**

I. Introduction and prior work

Segmenting is the fundamental task of finding homogeneous regions (called *segments*) in images. Humans segment images using both a bottom-top and top-bottom cognitive process linked with recognition and scene understanding among others. In computer vision, segmentation is often tackled as a low-level clustering task (*e.g.*, region merging [1], [2], [3], mode seeking by mean shift [4], Gaussian blobworlds [5], spectral clustering [6], [7]). Those primitive low-level computer vision engineering segmentation techniques bring tools for more complex tasks (*e.g.*, super-pixels in image analysis [8], [9], object recognition [10], or image annotation [11], etc.). To fairly evaluate and compare segmentation algorithms, several data-sets with human annotated segmentations (defining "ground-truth") have been assembled. See for example the *Berkeley Segmentation Data Set and Benchmarks 500* (BSDS500 for short, see [12]). Figure 1 displays the results of segmenting an image by several people. Observe that each person yields a different segmentation result for this "open" task. In [13], the authors state that "There does not appear to be a consensus about which of these [segmentation] algorithms is best." In fact, even constraining segmentation as a pure axiomatically well-defined clustering problem does not bring a final unique solution. Indeed, Kleinberg [14] proved that under three simple clustering intuitive properties there does not exist such a clustering function to optimize.

In this work, we propose a simple *probabilistic segmentation algorithm* that generates each time a *different* segmentation output for the same input image. We then aggregate those segmented results using the flavor of *ensemble segmentation* [15], [16]. Ideally, we would like to have a *population of segmentation results*, and characterize the segmentation using probability segments. Our algorithm, termed Consensus Region Merging (CRM), relies on a straightforward random extension of the Statistical Region Merging (SRM) [2], [3], that we name RRM (for Random Region Merging). Every time, we call RRM on the source image, RRM returns a different segmentation. By running RRM multiple times, we thus get a population of segmentations (multiple segmentation is also considered in [17] by varying the number of segments). We then aggregate all those sample segmentations using a deterministic CRM using a simple voting scheme to get an area segmentation. As a byproduct, our technique interestingly yields also a high-quality contour detector as shown in experiments. Note that Arbeláez et al. [12] presented a generic way to transform the output of *any* contour detector (not necessarily closed-contour) into a hierarchical region tree [13]. We give preliminary theoretical arguments that favor a reduction of the variance of CRM compared to SRM [2], [3].

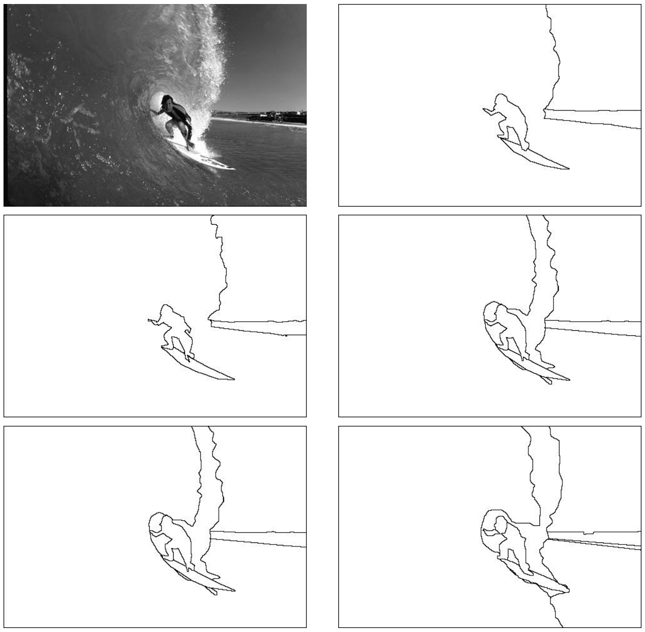

Fig. 1. Several "ground-truth" segmentations of an intensity image by different users (from BSDS500 [12]). Each user defines its own segmentation (with potentially a different number of segments; here $6, 11, 13, 9$ and 14 segments).

The outline of the paper is as follows: Section II recalls the statistical region merging segmentation algorithm using a weighted graph framework. It is followed by Section III describing the random segmentation extension. Section IV describes the voting-scheme consensus segmentation, CRM, based on multiple runs of RRMs. Section V reports on preliminary theoretical results. Finally, section VI concludes and hints at future work.

II. Graph Region Merging

We recall the Statistical Region Merging (SRM) algorithm [2], [3] using the broader setting of weighted graphs.

SRM starts by considering each pixel v as its own region (a singleton $\{v\}$), and eventually merge following a predefined order of the regions relying on a statistical merging predicate To describe SRM using the framework of graphs, we first convert an image I into a weighted graph $G = (V, E, w)$ using the 4-connectivity of pixels. The nodes $v \in V$ correspond to pixels, and edges $e = (a, b) \in E$ iff $\|a - b\|_1 = 1$ where $\| \cdot \|_1$ denote the L_1 norm (C_4 South/North/East/West neighboring pixels). This yields $2(w-1)(h-1)+w+h-2$ unoriented edges for an image of size $w \times h$. The weight $w(e) = |I_a - I_b|$ of an edge $e = (a, b)$ is set to the difference of the intensity channel, where I_v denotes the intensity at pixel v. Region merging is efficiently implemented using Tarjan's disjoint-set [18] data-structures[1], which requires in practice constant amortized time per merge or find operation. Algorithm 1 summarizes the graph segmentation algorithm, GRM.

Algorithm 1 Graph Region Merging (GRM)

```
// Input: Weighted graph $G = (V, E, w)$
Create disjoint set data-structure on vertex set $V$
Sort edges in increasing order of their weights in queue $Q$
// Greedy region merging segmentation
while $Q \neq \emptyset$ do
  $e = (a, b) \leftarrow Q.\mathrm{head}()$
  // Use disjoint set data-structure to find region [18]
  $R_a \leftarrow \mathrm{FindRegion}(a)$
  $R_b \leftarrow \mathrm{FindRegion}(b)$
  if $(R_a \neq R_a)$ and $\mathrm{MergePredicate}(R_a, R_b)$ then
    // Merge the two regions (average intensity)
    $\mathrm{Merge}(R_a, R_b)$
  end if
end while
```

In [2], [3], the merging predicate is derived from statistical concentration inequalities. For a region R_x, let I_x denote the average channel value, and n_x the number of pixels contained inside that region. For our purpose, we shall use the following simpler predicate:

$$\mathrm{MergePredicate}(R_a, R_b) = \begin{cases} \mathrm{true} & \text{if } |I_a - I_b| < \frac{255}{2 \log \max(n_a, n_b)}, \\ \mathrm{false} & \mathrm{otherwise.} \end{cases} \quad (1)$$

For color RGB images (or hyper-spectral images), we merge regions if and only if the predicate is true for each channel, independently of the others. As the pseudo-code of Algorithm 1 emphasizes, the greedy GRM segmentation algorithm relies on two principles:

1) A fixed predefined order on the weighted edges, and
2) a (deterministic) merging predicate.

By changing either the inspection of adjacent region pairs (linked with edges), or the predicate, we thus get a different segmentation result. The following section randomizes GRM by:

1) Shuffling randomly the order using a permutation, and
2) designing a random (non-deterministic) merging predicate.

Although the GRM algorithm makes greedy decisions to decide whether to merge or not regions, one can show that GRM segmentations satisfy global properties. For example, Nielsen and Nock [20], [2], [3] proved that with high probability that the algorithm yields an over-segmentation when using a statistical predicate. Similarly, Felzenswalb and Huttenlocher [1] proved that GRM is not too fine nor too coarse for their predicate.

III. Random region merging

In order to get a "population of segmentations," we add random behavior in GRM. First, we apply an arbitrary random permutation on the edge pairs (we purely discard the former $w(e)$ values). Second, we consider the following non-deterministic *random predicate*:

$$\mathrm{RandomPredicate}(R_a, R_b) = \begin{cases} \mathrm{true} & \text{if } |I_a - I_b| < \frac{255U}{\log \max(n_a, n_b)}, \\ \mathrm{false} & \mathrm{otherwise.} \end{cases}, \quad (2)$$

where U denotes a uniform random variate in $[0, 1)$. Note that $E[U] = \frac{1}{2}$, and that on average the merging predicate becomes the deterministic predicate of Eq. 1. Algorithm 2 describes the algorithm, and Figure 2 displays several runs on a given color image that shows a segmentation population.[2] The idea of choosing a random predicate has been inspired by random dithering techniques of grey images (see [21], page 265).

Algorithm 2 Random Region Merging (RRM)

```
// Input: Graph $G = (V, E)$
Let queue $Q$ be a random permutation of $E$
// Greedy region merging segmentation
while $Q \neq \emptyset$ do
  $e = (a, b) \leftarrow Q.\mathrm{head}()$
  // Use disjoint set data-structure
  $R_a \leftarrow \mathrm{FindRegion}(a)$
  $R_b \leftarrow \mathrm{FindRegion}(b)$
  if $(R_a \neq R_a)$ and $\boxed{\mathrm{RandomPredicate}(R_a, R_b)}$ then
    // Merge the two regions
    $\mathrm{Merge}(R_a, R_b)$
  end if
end while
```

We now turn to the consensus process that combines arbitrarily many segmentation results into one segmentation by using a simple voting scheme.

IV. Consensus region merging

Consensus Region Merging (CRM) outputs (1) a segmentation and (2) a soft contour map. The CRM algorithm consists in computing l independent random segmentations using RRM, and then combine the sample outcome into one overall segmentation. The key idea consists in adding *one vote*

[1] Disjoint-set forest with union by rank compression [19].

[2] Loosely speaking, note that by iteratively computing a random segmentation of a still image, we obtain a never-ending "video" segmentation.

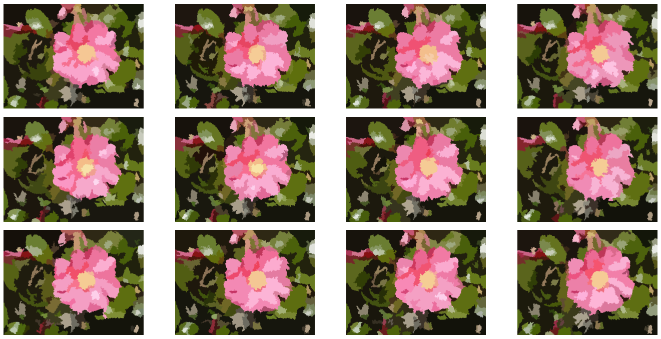

Fig. 2. A population of segmentations: Several random segmentation variates ($l = 12$) using non-deterministic RRM.

to an edge e whenever its merging predicate was evaluated to true. We then set $w(e)$ to the number of votes, and run GRM to export a hard[3] segmentation. In addition, we also compute a soft contour detector as follows: Each edge adds to the weights of its two extremity pixels the number of times it was selected for merging. (Thus each inner[4] pixel receives four values from its two incident edges.) Indeed, whenever an edge was voted, it means the pixels belong to the same region for the precise segmentation, and therefore are not on a picture "edge." We take l minus the same region value, and divide by l to get the *probability map* of being a contour. Note that since GRM provides closed regions (and therefore closed contours), the soft contour consensus region is guaranteed to have closed contours. Figure 3 displays the contour detector results for $l = 1000$. We can run Arbeláez et al. [12] generic algorithm to transform the output of a contour detector into a *hierarchical segmentation region* tree.

Algorithm 3 Consensus Region Merging CRM)

```
// Input: Graph $G = (V, E)$ and $l$ number of segmentations
for $i = 1$ to $l$ do
    Perform a random region merging segmentation $\mathrm{RRM}(G)$
    Each time an edge $e$ is merged, add 1 to $n_e$
end for
// Export hard consensus segmentation
Build a weighted graph $G = (V, E, w)$ with $w_e = n_e \times |I_a - I_b|$, the number of times edge $e$ was merged
// Output 1: A segmentation
Call graph region merging on $G$.
// Output 2: A soft contour map
Let $s_v = 0 \; \forall v \in V$ // Strength of belonging to a contour
for $e = (a, b) \in E$ do
    $s_a \leftarrow s_a + w_e$
    $s_b \leftarrow s_b + w_e$
end for
// Rescale for exporting contour map
for $v \in V$ do
    $I_v = 255 \frac{s_v}{l}$
end for
```

In theory, the larger l, the better it produces a contour detector as it is less prone to sensitivity of the pair inspection

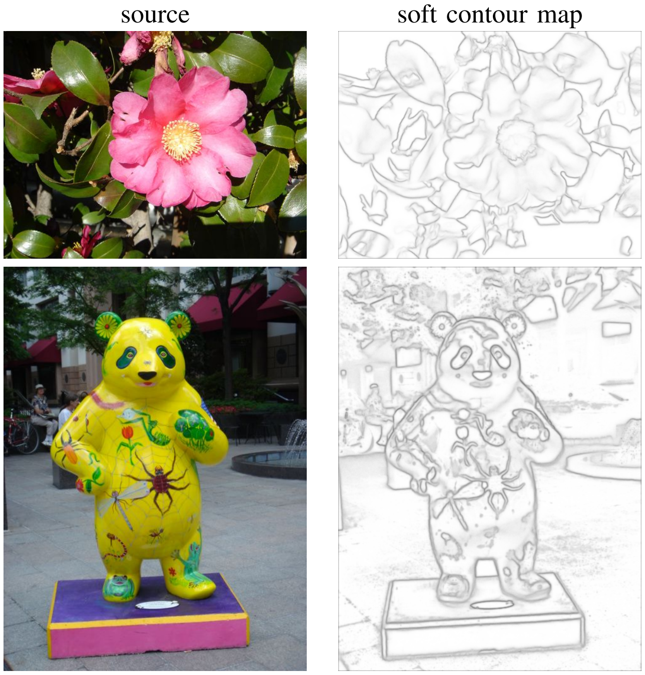

Fig. 3. Soft contour edge detector with guaranteed closed contours ($l = 1000$ segmentations).

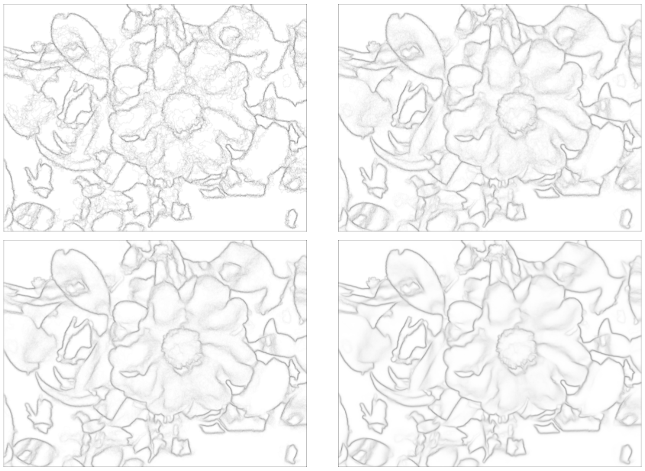

Fig. 4. Quality of soft contour detector according to the number $l \in (10, 50, 100, 1000)$ (from top to bottom, and left to right) of random segmentations. Observe the visible difference between the results for $l = 10$ and $l = 100$.

order. Figure 4 shows the impact of l for one input image. We observe empirically that a few dozen runs are enough to get a good contour detector in practice.

Notice that SRM/GRM is a hard segmentation algorithm that provides closed contours. By using several runs of RRM, we can export a high-quality soft contour map, as depicted in Figure 4. Figure 6 displays the results obtained on BSDS500 [12]. See http://anonymous It takes less than a minute for performing a CRM on a VGA color image size with $l = 500$ runs using a Intel® Pentium® i7-2640M at 2.8GHz (4GB RAM).

The next section hints at a theoretical analysis of CRM.

[3] Pixels belong to one and only one regions. Thus the segmentation is a partition of the image with closed contours.

[4] Pixel do not belonging to the image boundary.

V. Preliminary theoretical results

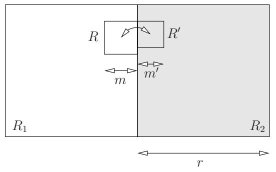

Fig. 5. Testing the merging of squared subregions R and R' of two regions R_1 and R_2 (see text).

The intuition of consensus region merging is similar to that of *bagging* in supervised learning [24]. In supervised learning, the error of a classifier can be decomposed into three terms: (1) a noise term, which describes fluctuations of true labels compared to their theoretical value, (2) a bias term, which models the average error of the classifier, and (3) a variance term, which models the fluctuations of the classifiers when the data at hand changes. With bagging, bootstrapping a dataset to induce classifiers, and finally averaging the output of the classifiers, is known to reduce the *variance* of the misclassification. There is certainly an analogy between the bias-variance trade-off in classification, and that in segmentation as carried out in the model of CRM. The objective of this section is to give a preliminary step towards understanding the way the bias-variance trade-off holds in this model, and to what extent CRM may reduce its variance.

Figure 5 presents the toy case study. We have two adjacent regions, R_1 and R_2, each having r^2 pixels. We assume that we have a single channel. Let Δ denote the smallest observed difference in absolute value between any two subregions of R_1 and R_2, so that for the two regions R and R' displayed in Figure 5, we have $|I_R - I_{R'}| \geq \Delta$. We simplify the merging predicate as merging R and R' if and only if:

$$|I_R - I_{R'}| \quad \leq \quad \frac{Kg}{\ln \max\{m, m'\}} \ . \tag{3}$$

for some constant $K > 1$ and $g = 256$ the number of grey levels. It comes that when:

$$\max\{m, m'\} \quad \geq \quad \exp\left(\frac{Kg}{\Delta}\right) \ , \tag{4}$$

the merging is not carried out. If we let $\sqrt{u}$ denote the right-hand side of (4), then the maximal number of mergings that can be accepted between squared subregions of R_1 and R_2 is of order $\Theta((r^3 - (r-u)^3)^2)$, out of $\Theta(r^6)$. The proportion of mergings that can be accepted is thus:

$$p \quad = \quad \Theta\left(\left(1 - (1 - (u/r))^3\right)^2\right) \ . \tag{5}$$

Plugging this in (4), we see that for a maximal proportion p of possible mergings to hold, it is sufficient that

$$\Delta \quad \geq \quad \Theta\left(\frac{g}{\ln(r(1 - (1-\sqrt{p})^{\frac{1}{3}}))}\right) \ . \tag{6}$$

Since $(1-\sqrt{p})^{\frac{1}{3}} \leq 1 - kp$ for some constant $k > 0$, we get the following Lemma on the problem statement illustrated in Figure 5.

Lemma 1: The proportion of mergings that can be accepted between R_1 and R_2 is no more than p if $\Delta > Kg/\ln(rp)$ for some constant K.

On the other hand, reversing the inequality in (4) to count the cases where merging is accepted, one sees that the minimal proportions of merging between R_1 and R_2 that can be accepted is $\Omega((r-u)/r^6) = \Omega((1/r^5)(1-(u/r)))$. The same reasoning as for Lemma 1 brings the following Lemma.

Lemma 2: The proportion of mergings that can be accepted between R_1 and R_2 is at least p if $\Delta < K'g/\ln(r(1-pr^5))$ for some constant K'.

Hence, repeating the randomized order in region merging, as it is carried out in CRM for its segmentation output, reduces the chance to merge first some parts of two distinct regions whose channel differences are not important (worst-case scenario of Lemma 2), while for regions with significant channel differences, it does not degrade on average the quality of merging compared to the situation where the order of SRM would be used (Lemma 1). This hints on the fact that averaging the randomized results, as carried out in CRM, may significantly reduce on average the risk of bad orders bringing bad segmentations, and thus, up to some extent, the variance of the differences with the ideal segmentation in SRM's model.

VI. Conclusion and perspectives

We described a soft contour detector and segmentation algorithm that relies on designing a random region merging algorithm (RRM) with a simple voting consensus scheme on several of its runs (CRM). Our algorithm is quasi-linear[5] in the image size and the number l of iterations. This work is a very first step in creating a truly probabilistic region segmentation model characterizing probabilistically segmented regions. Indeed, a good segmentation algorithm should be robust to small levels of perturbations [22]. CRM can be viewed as such as a robust SRM [2]. (In particular, CRM avoids flickering artifacts when segmenting video frame by frame.) Ongoing work investigates the persistence [23] of various segmentation algorithms using region-based metrics [13].

We end up with the following open problem by taking analogy with the field of Statistics: Can we design a generative probabilistic image model and report a randomized segmentation algorithm on random image variates that is asymptotically *consistent* (*e.g.*, segmentation converging to the true optimal segmentation as the number of runs increase) ? In computer vision, time has often been taken as the limiting resource to optimize. Can we design segmentation algorithms that always improve with time?

The Java$^{\text{TM}}$ code (about 1000 lines) with extensive results on the Berkeley data set BSDS500 [12] is available on-line at http://anonymous for reproducible research.

References

[1] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[5] For an image size of $n = w \times h$ pixels, it takes $O(ln\alpha(n))$ time to obtain l sample segmentations, where $\alpha(\cdot)$ is the functional inverse of Ackermann's function [18], [2].

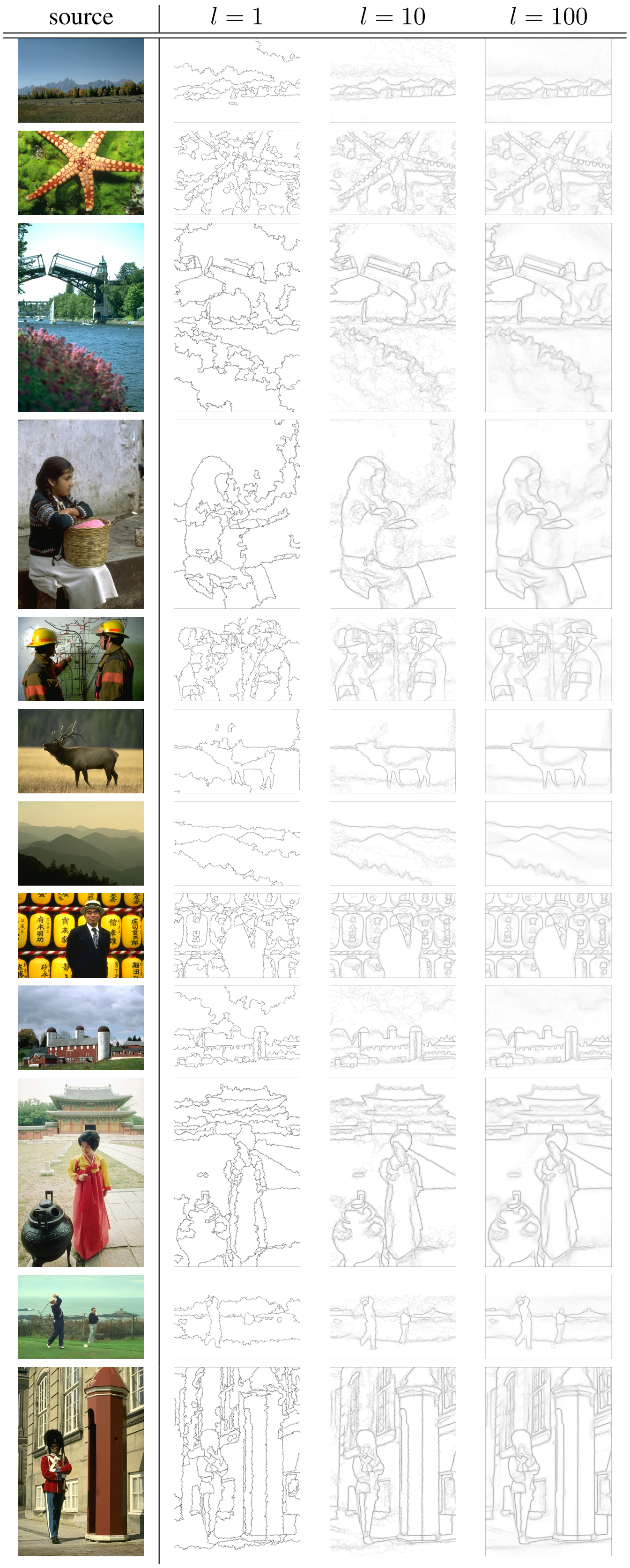

Fig. 6. Assessing the visual performance of the soft contour extractor of CRM on the Berkeley Segmentation Data Set (BSDS500) for $l = 1, l = 10$ and $l = 100$ run of RRM. Observe that the quality of contours increase with the number of runs.

[2] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1452–1458, 2004.

[3] F. Nielsen and R. Nock, "Fast Graph Segmentation Based on Statistical Aggregation Phenomena," *MVA*, pp. 150-153, 2007.

[4] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000*, vol. 2. IEEE Comput. Soc, 2000, pp. 142–149.

[5] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using Expectation-Maximization and its application to image querying," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026–1038, 2002.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 731–737.

[7] T. Cour, F. Bénézit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 1124–1131.

[8] S. Boltz, F. Nielsen, and S. Soatto, "Earth mover distance on superpixels," in *ICIP*, 2010, pp. 4597–4600.

[9] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*. Washington, DC, USA: IEEE Computer Society, 2003.

[10] K. Schindler and D. Suter, "Object detection by global contour shape," *Pattern Recogn.*, vol. 41, no. 12, pp. 3736–3748, Dec. 2008.

[11] W. Wu and J. Yang, "Semi-automatically labeling objects in images," *IEEE Transactions on Image Processing (TIP)*, vol. 18, no. 6, pp. 1340–1349, Jun. 2009.

[12] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[13] P. Arbelaez, M. Maire, C. C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *CVPR*, 2009, pp. 2294–2301.

[14] J. M. Kleinberg, "An impossibility theorem for clustering," in *Neural Information Processing Society (NIPS)*, 2002, pp. 446–453.

[15] V. Singh, L. Mukherjee, J. Peng, and J. Xu, "Ensemble clustering using semidefinite programming with applications," *Mach. Learn.*, vol. 79, no. 1-2, pp. 177–200, May 2010.

[16] A. Alush and J. Goldberger, "Ensemble segmentation using efficient integer linear programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 34, pp. 1966-1977, 2013.

[17] T. Suzuki and M. Hebert, "Estimating object region from local contour configuration," in *Joint Workshop on Visual and Contextual Learning from Annotated Images and Videos, and Visual Scene Understanding*, 2009, pp. 69–76.

[18] R. E. Tarjan, "Efficiency of a good but not linear set union algorithm," *J. ACM*, vol. 22, no. 2, pp. 215–225, Apr. 1975.

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.

[20] F. Nielsen and R. Nock, "On region merging: The statistical soundness of fast sorting, with applications," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2003, pp. 19–26.

[21] F. Nielsen, *Visual Computing: Geometry, Graphics, and Vision*. Charles River Media / Thomson Delmar Learning, 2005.

[22] H. Zhou, X. Wang, and G. Schaefer, "Mean shift and its application in image segmentation," in *Innovations in Intelligent Image Analysis*, ser. Studies in Computational Intelligence, H. Kwasnicka and L. Jain, Eds. Springer Berlin / Heidelberg, 2011, vol. 339, pp. 291–312.

[23] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba, "Persistence-based clustering in Riemannian manifolds," in *Symposium on Computational Geometry*, 2011, pp. 97–106.

[24] Y. Ganjisaffar, R. Caruana, and C.-V. Lopes, "Bagging gradient-boosted trees for high precision, low variance ranking models," in 35^{th} *ACM SIGIR*, 2011, pp. 85–94.