PS1-32

Consensus Region Merging for Image Segmentation

Frank Nielsen（Sony CSL） Richard Nock (UAG-CEREGMIA)

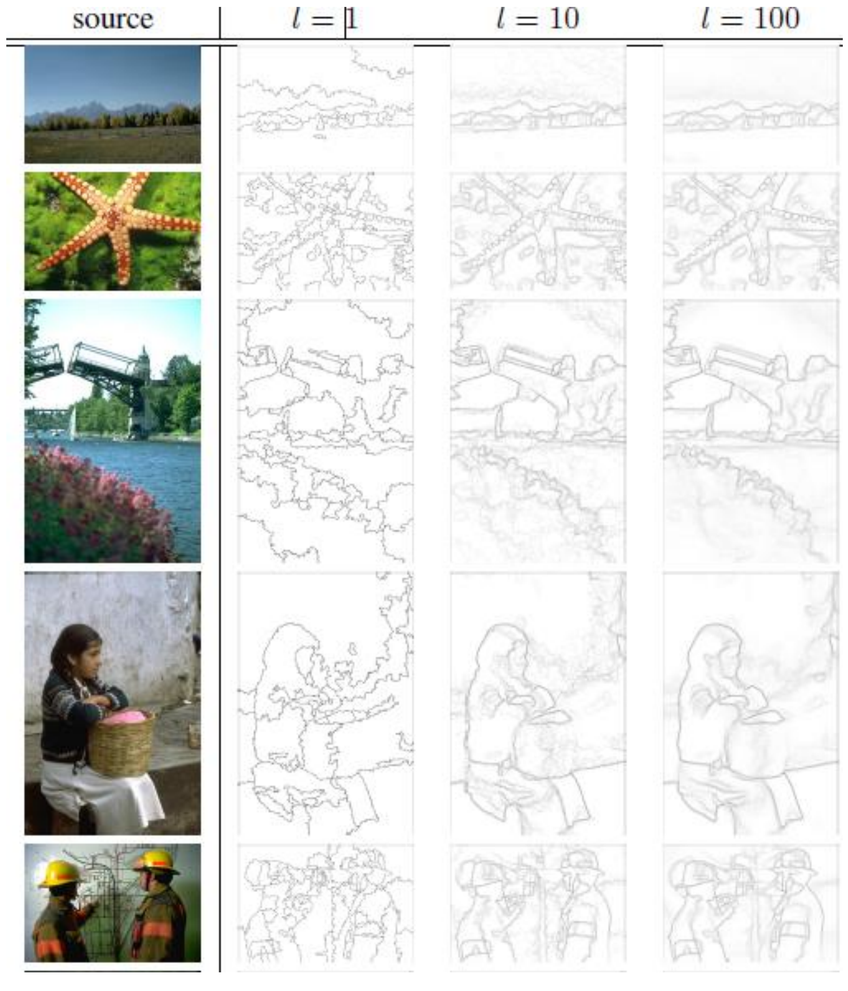
How can we design a segmentation algorithm that keeps improving with time?

Randomized algorithm ⇒ Population of segmentations

Then perform a consensus on the population (soft edge contour)

Source | Population of segmentations | Consensus (soft edge)

Random Region Merging (RRM) → Consensus Region Merging (CRM)



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**
  $e = (a, b) \leftarrow Q$.head()
  // Use disjoint set data-structure
  $R_a \leftarrow \text{FindRegion}(a)$
  $R_b \leftarrow \text{FindRegion}(b)$
  if $(R_a \neq R_a)$ **and** $\boxed{\text{RandomPredicate}(R_a, R_b)}$ **then**
    // Merge the two regions
    $\text{Merge}(R_a, R_b)$
  end if
end while

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

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 $\text{RRM}(G)$
  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

R. Nock and F. Nielsen, “Statistical region merging,” IEEE PAMI, vol. 26, no. 11, pp. 1452–1458, 2004.