Image Segmentation by Size-Dependent Single Linkage Clustering of a Watershed Basin Graph
Supplementary Material

****

****
****
****
****

1 The Watershed Transform Algorithm

We present the watershed transform algorithm from **Section 2** of the main manuscript. We assume there is an ordering of the vertices in V. Usually the information about the vertices is stored in an array, where the i-th element of the array contains the vertex with index i.

1. Apply the threshold $T_{\min}$ and $T_{\max}$:
   (a) Remove each $\{u, v\}$ from E if $w(\{v_i, u\}) > T_{\max}$.
   (b) For each $\{u, v\}$ from E set $w(\{v_i, u\}) = 0$ if $w(\{v_i, u\}) < T_{\min}$.
   (c) Remove singleton vertices (vertices with no incident edges in E). Mark them as background.
2. Create G'. Set $V' = V$ and $E' = \emptyset$. For each vertex $v_i \in V'$:
   (a) Calculate $M_i = \min_{u \in V', \{v_i, u\} \in E} \{w(\{v_i, u\})\}$.
   (b) For each $u \in V'$ such that $\{v_i, u\} \in E$ add (v_i, u) to E' if $w(\{v_i, u\}) = M_i$.
3. Modify G' to remove all saddle vertices. For each vertex $v \in V$ that has more than one outgoing edge, keep only one outgoing edge pointing to a vertex with the minimal index.
4. Modify G' to split the non-minimal plateaus:
   (a) Initialize a FIFO queue Q.
   (b) For each vertex $v \in V'$, check whether the vertex has at least one outgoing edge and one bidirectional edge (check whether the vertex is a plateau corner). If it does mark it as visited and add it to the end of Q.
   (c) While $Q \neq \emptyset$, let u be the first element of Q and remove it from Q. For all v such that $(u, v) \in E'$ and $(v, u) \in E'$ remove (u, v) from E'. If v is visited remove (v, u) from E' as well, otherwise mark v as visited and add it to the end of Q.
5. Replace all unidirectional edges with bidirectional edges. For each $(u, v) \in E'$ add (v, u) to E' if not already there.
6. Return connected components of the modified G'.

In the first step of the algorithm we apply the thresholds and mark the singleton vertices as background, as desired by the output. We expect all the other vertices to be assigned to a watershed basin. In the second step we create the steepest descent graph. In the steps 3 we make sure that all the saddle vertices are removed. After this step there will be no vertices with more than one outgoing edge. The step 4 splits the plateaus. In the breadth first search, while examining the plateau vertices, we make sure that all the vertices have only one outgoing edge. The breadth first search order will ensure that we keep the edge on the path to the closest plateau corner (with the minimal index). After the step 4 all vertices that don't belong to a regional minima will have a single outgoing edge, hence there will be an unique steepest descent path to a single regional minima. Therefore, all the vertices will be uniquely assigned to a basin. The step 5 just makes sure there's also a path from a regional minima to all vertices in its basin of attraction so that we can just apply connected components in order to obtain the watershed basins.

As we operate on a connected graph we assume $O(|E|) \geq O(|V|)$. The first three steps of the algorithm visit each edge constant number of times. In the step 4, we visit each vertex at most once. While visiting each vertex we visit edges incident to it constant number of times. The step 5 also visits each edge at most once. As the complexity of connected components is $O(|E|)$, we get our overall complexity to be $O(|E|)$.

Equivalence of $T_{\min}$. In the manuscript we claim that a segmentation obtained by:

1. Applying watershed transform and then merging all basins with saliency smaller than $T_{\min}$, and
2. Applying watershed on a graph where with all the edges with weight smaller than $T_{\min}$ are replaced with edges with weight of 0

are equivalent. We first prove the following lemma:

Lemma 1 *Examine a watershed transform W of G and watershed transform W' obtained by applying a threshold $T_{\min}$ on G. Let $\{B_1, B_2, \ldots\}$ be the watershed basins of W and $\{B'_1, B'_2, \ldots\}$ the watershed basins of W'. For each B_i there exist B'_j such that $B_i \subseteq B'_j$.*

Proof. The steepest descent graph containing only vertices in V that are not incident to any edge with the weight smaller than $T_{\min}$ will stay unmodified, even after splitting the plateaus and getting rid of the saddle vertices. All other vertices will became a part of a regional minima (locally minimal plateau in the steepest descent graph). All previous locally minimal edges will stay locally minimal. Therefore applying the threshold $T_{\min}$ just introduces bidirectional edges to the steepest descent graph. Therefore, a connected component in the modified steepest descent graph of G has to be a subset of a connected component of the modified steepest descent graph of G after applying $T_{\min}$.

Watershed basins of W' are connected. Hence, they can be obtained by merging watershed basins of W. We prove that the result of 1. and 2. are equivalent by showing that two neighboring basins of W, B_i and B_j will be merged in 1. if and only if they are merged in 2.

Let's examine two watershed basins of W, B_i and B_j. If there is an edge $\{u, v\}$ in G such that $u \in B_i$ and $v \in B_j$ with the weight smaller than $T_{\min}$, then u and v will be part of the same regional minima in W'. Hence, the two basins of W belong to a single basin of W'. We also know that the saliency between B_i and B_j has to be smaller than $T_{\min}$, therefore they will belong to the same segment when applying the algorithm 1. Similarly, if the saliency d_{ij} between B_i and B_j is below $T_{\min}$, then there exist an edge $\{u, v\}$ in G such that $u \in B_i$ and $v \in B_j$ with the weight equal to $d_{ij} < T_{\min}$. Therefore u and v will be part of the same regional minima, and B_i and B_j will be part of the same watershed basin in W'. □

2 Hierarchical Clustering of the Watershed Basin Graph

The following algorithm and theorem were introduced in the main manuscript.

Algorithm 1 Size-dependent single linkage clustering

1. Order E_W into $\pi(e_1, \dots, e_n)$, by non-decreasing edge weight.
2. Start with basins as singleton clusters $S^0 = \{C_1 = \{B_1\}, C_2 = \{B_2\}, \dots\}$
3. Repeat step 4 for $k = 1, \dots, n$
4. Construct S^k from S^{k-1}. Let $e_k = \{B_i, B_j\}$ be the k-th edge in the ordering. Let C_i^{k-1} and C_j^{k-1} be components of S^{k-1} containing B_i and B_j. If $C_i^{k-1} \neq C_j^{k-1}$ and $\Lambda(C_i^{k-1}, C_j^{k-1})$ is not satisfied then S^k is created from S^{k-1} by merging C_i^{k-1} and C_j^{k-1}, otherwise $S^k = S^{k-1}$.
5. Return the hierarchical segmentation $(S^0, \dots, S^n)$

Theorem 1 *The highest level of the hierarchical segmentation produced by algorithm (1) will have the predicate Λ satisfied for all pairs of the clusters. The complexity of the algorithm is $|E_W| \log |E_W|$. The algorithm can be modified to consider only the edges of the minimum cost spanning tree of G_W.*

Proof. First we prove that Λ will be satisfied for all pairs of the clusters in the highest level. Assume $\Lambda(C_i^n, C_j^n)$ is not satisfied for some pair of clusters, $d_{C_i^n, C_j^n} < \tau(\min\{S(C_i^n), S(C_j^n)\})$. That means there exist an edge $e \in E_W$ with the weight equal to $d_{C_i^n, C_j^n}$. Without losing generality assume e was examined at the $(k+1)$-th step of the algorithm and the following predicate was satisfied: $d_{C_p^k, C_q^k} \geq \tau(\min\{S(C_p^k), S(C_q^k)\})$. As $C_p^k \subseteq C_i^n$ and $C_q^k \subseteq C_j^n$ then $\min\{S(C_p^k), S(C_q^k)\} \leq \min\{S(C_i^n), S(C_j^n)\}$ and $\tau(\min\{S(C_p^k), S(C_q^k)\}) \geq \tau(\min\{S(C_i^n), S(C_j^n)\}) > d_{C_i^n, C_j^n} = d_{C_p^k, C_q^k}$. Contradiction. □

Assuming real values of the weights, the complexity of the algorithm is dominated by sorting the edges of E_W. If we use *union-find* data structure, an iteration of the step 4. takes $O(u(n))$, where n is the number of basins and u is the inverse Ackermann function.

Let E_T be the set of edges on an MST of the watershed basin graph. Assume we run the previous algorithm considering only edges in E_T. We want to prove that Λ will be satisfied for all pairs of the clusters at the highest level. Again, assume $\Lambda(C_i^n, C_j^n)$ is not satisfied for some pair of clusters, $d_{C_i^n, C_j^n} < \tau(\min\{S(C_i^n), S(C_j^n)\})$. That means there exist an edge $e \in E_W$ with the weight equal to $d_{C_i^n, C_j^n}$. If $e \in E_T$, then e was examined in the algorithm. We follow the same steps as above to show that when e was examined the predicate must have been unsatisfied meaning that C_i^n and C_j^n can't be two different clusters. If $e = \{B_p, B_q\}$ is not on the MST, then there is a path in the watershed basin graph from B_p to B_q where all the edges on the path have the weight less or equal to the weight of e. Without losing generality, assume $S(C_i^n) \leq S(C_j^n)$ and $B_p \in C_i^n$. Then $\tau(S(C_j^n)) > w(e)$. Let $e_x = \{B_a, B_b\}$ be an edge on the path from B_p to B_q in the MST such that $B_a \in C_i^n$ and $B_b \notin C_i^n$. Such edge must exist, otherwise B_p and B_q would be part of the same cluster. As e_x is in the MST, it was examined in the algorithm. When e_x was examined the size of the cluster C that B_a was part of was smaller or equal than $S(C_j^n)$). Hence $\tau(S(C)) \geq \tau(S(C_j^n)) > w(e) \geq w(e_x)$. Therefore e_x can't exist, and B_p and B_q have to be part of the same component. Contradiction. $\square$