Fast Computation of Content-Sensitive Superpixels and Supervoxels using q-distances (Supplemental Material)

Zipeng Ye[1*], Ran Yi[1*], Minjing Yu[2†], Yong-Jin Liu[1†], Ying He[3]
[1]Tsinghua University [2]Tianjin University [3]Nanyang Technological University

S1. Introduction

This supplemental material contains four parts:

- the proofs of theoretic results presented in Section 4 of the main paper (Section S2),
- full details of comparison of 31 superpixel methods on five image datasets (Section S3),
- full details of comparison of 8 supervoxel methods on four video datasets (Section S4),
- full details of comparisons in one image and two video applications (Section S5).

S2. Proofs of Theoretic Results in Section 4

In the main paper, both images and videos are mapped into ζ-dimensional manifold M_ζ embedded in $\mathbb{R}^d$, $\zeta = 2, 3$, $\zeta < d$, that is, mapping an image to $M_2 \subset \mathbb{R}^5$ and a video to $M_3 \subset \mathbb{R}^6$. A common characteristic of these manifolds $M_\zeta \subset \mathbb{R}^d$ is that the geodesic metric — which defines the lengths, area or volumes on M_ζ — gives a good measure of the content density in images and videos. Let $X = \{x_i\}_{i=1}^N$ be an N-atom media (i.e., either an image or a video) and $\mathcal{M}_\zeta = \Phi(X)$ the stretched manifold in $\mathbb{R}^d$.

In this section, we prove a proposition, indicating that if the shape of manifold $\mathcal{M}$ satisfies certain assumptions (characterized by the edge length ratio $\frac{l_{\max}(V_i)}{l_{\min}(V_i)}$) and K is sufficiently large (characterized by the working area $\Xi(\widehat{c}_i)$), the clustering $\{V_i\}_{i=1}^K$ on $\mathcal{G}$ is exactly the same for using either shortest distance or q-distance.

In Section 4 of the main paper, we present three properties. In this section, we also present their proofs.

Recall that we discretize the manifold $\mathcal{M}_\zeta$ by a graph $\mathcal{G} = \{V, E\}$, V is the vertex set $\{v_i = \Phi(x_i)\}_{i=1}^N$, $\forall x_i \in X$, and there is an edge $e = (v_i, v_j)$ in the edge set E, if $\Phi^{-1}(v_i)$ and $\Phi^{-1}(v_j)$ are n_ζ-neighbors in X:

- $n_\zeta = 8$ for $\zeta = 2$: i.e., for X being an image, $\Phi^{-1}(v_i) = (r_i, s_i)$ and $\Phi^{-1}(v_j) = (r_j, s_j)$ are 8-neighbors, if $\|r_i - r_j\|_2 \le 1$ and $\|s_i - s_j\|_2 \le 1$.
- $n_\zeta = 26$ for $\zeta = 3$: i.e., for X being a video, $\Phi^{-1}(v_i) = (r_i, s_i, t_i)$ and $\Phi^{-1}(v_j) = (r_j, s_j, t_j)$ are 26-neighbors, if $\|r_i - r_j\|_2 \le 1$, $\|s_i - s_j\|_2 \le 1$ and $\|t_i - t_j\|_2 \le 1$.

Given the fixed set of centers $C = \{c_i\}_{i=1}^K \subset V$ as multiple sources, we assign an index I_i to each vertex $v_i \in V$ based on the predefined traversal order $\{v_1 = c_1, v_2 = c_2, \cdots, v_K = c_K, v_{K+1}, v_{K+2}, \cdots, v_N\}$ by the FIFO queue. Then the q-path $\widetilde{cv_i} = \{v_{I_{i_1}} = c, v_{I_{i_2}}, \cdots, v_{I_{i_n}} = v_i\}$ from a center $c \in C$ to a vertex $v_i \in V \setminus C$, the output from Algorithm 2, satisfies that $\forall i_a, i_b$, $1 \le a < b \le n$, the indices $I_{i_a} < I_{i_b}$.

Proposition 1. *Given a fixed set of centers $C = \{c_i\}_{i=1}^K \subset V$, denote the clustering of V w.r.t. C on $\mathcal{G}$ as $\{V_i\}_{i=1}^K$, where $V_i = \{v \in V : d(c_i, v) < d(c_j, v), i \ne j\}$ and d is an arbitrary distance. Denote the clustering under geodesic distance as $\{V_i|_{d_g}\}_{i=1}^K$. For each center $c_i \in C$, we define the maximal and minimal edge lengths in $V_i|_{d_g}$ as $l_{\max}(V_i|_{d_g}) = \max_{e=(v_a,v_b)\in E}\{l_e = \|v_i - v_j\|_2, v_a, v_b \in \Phi(CH(\Phi^{-1}(V_i|_{d_g})))\}$ and $l_{\min}(V_i|_{d_g}) = \min_{e=(v_a,v_b)\in E}\{l_e = \|v_i - v_j\|_2, v_a, v_b \in V_i \cup Nb(V_i)|_{d_g}\}$, where $CH(A)$ is the convex hull of a region $A \in X$ and $Nb(V_i)$ is the union of all neighboring V_a of V_i, $a \ne i$. Denote the inverse mapping of $C = \{c_i\}_{i=1}^{n_c} \subset V$ back into the media X as $\widehat{C} = \{\widehat{c}_i\}_{i=1}^{n_c} \subset I$, $\widehat{c}_i = \Phi^{-1}(c_i)$, $\forall i$. For each mapped center $\widehat{c}_i \in \widehat{C}$, we define its working area in X as $\widehat{\Xi}(\widehat{c}_i) = \bigcap_{\widehat{c}_j \in \widehat{C}, i \ne j} \widehat{\Xi}(\widehat{c}_i, \widehat{c}_j)$, where $\widehat{\Xi}(\widehat{c}_i, \widehat{c}_j) = \{x \in I : \frac{l_{\max}(V_i|_{d_g})}{l_{\min}(V_j|_{d_g})}\|x, \widehat{c}_i\|_M \le \|x, \widehat{c}_j\|_M\}$ and $\|\ \|_M$ is the Manhattan distance. Then the clustering $\{V_i\}_{i=1}^K$ on $\mathcal{G}$ is the same for the geodesic distance $d = d_g$ and the q-distance $d = d_q$, if each cluster $V_i \subseteq \Xi(\widehat{c}_i)$, where $\Xi(\widehat{c}_i) = \{\Phi(x_j) \in V : x_j \in \widehat{\Xi}(\widehat{c}_i)\}$.*

Proof. The basic idea in the proof is that for any two centers $c_i, c_j \in C$, $i \ne j$, we want to determine an area $\Xi(c_i, c_j) \in V$ such that using either geodesic distance d_g or q-distance

*Joint first authors
†Corresponding authors

d_q, all the vertices in $\Xi(c_i, c_j)$ have smaller distance to c_i than to c_j. Let $\Xi(c_i) = \cap_{c_j \in C, i \neq j} \Xi(c_i, c_j)$. Then all the vertices in $\Xi(c_i)$ have smaller distance to c_i than to all other centers in C. If the cluster $V_i \in \Xi(c_i)$, then this cluster is the same using either d_g or d_q. That completes the proof.

To find $\Xi(c_i, c_j)$, we note that between any two vertices, their q-distance cannot be smaller than their geodesic distance. Then for any vertex $v \in \Xi(c_i, c_j)$, we look for a upper bound $U(c_i, v)$ of q-distance from c_i to v, and a lower bound $L(c_j, v)$ of geodesic distance[1] from c_j to v. Then what we need to do is to ensure $U(c_i, v) \leq L(c_j, v)$.

To setup such an upper bound and a lower bound, we map the center set $C \in V$ back into the media X, denoted as $\widehat{C} = \{\widehat{c}_i\}_{i=1}^{n_c} \subset X$, $\widehat{c}_i = \Phi^{-1}(c_i)$, $\forall i$. We also denote the inverse map of $\Xi(c_i, c_j)$ from the graph $\mathcal{G}$ to the media X as $\widehat{\Xi}(\widehat{c}_i, \widehat{c}_j) = \{x \in X : \Phi(x) \in \Xi(c_i, c_j)\}$.

First, let's consider the upper bound $U(c_i, v)$. For any atom $\Phi^{-1}(v)$, denote the scan conversion of the line (in the media domain X) from c_i to v as $L_{sc}(c_i, v) = \{x_{j_1} = \widehat{c}_i, x_{j_2}, x_{j_3}, \cdots, x_{j_{n'}} = \Phi^{-1}(v)\}$. It can readily be seen that the indices[2] in $L_{sc}(c_i, v)$ satisfy $j_a < j_b$, $\forall j_a, j_b$, $1 \leq a < b \leq n'$. Then the mapped path $\Phi(L_{sc}(c_i, v)) = \{\Phi(p_{j_1}) = c_i, \Phi(p_{j_2}), \Phi(p_{j_3}), \cdots, \Phi(p_{j_{n'}}) = v\}$ is in the search space of Algorithm 2 and therefore the q-distance from c_i to v is not larger than the length of $\Phi(L_{sc}(c_i, v))$. To characterize the local geometry around $\Xi(c_i, c_j)$, we map the cluster $V_i|_{d_g}$ back into the media X, i.e., $\Phi^{-1}(V_i|_{d_g}) = \{x_j : \Phi(x_j) \in V_i|_{d_g}\}$, and consider the convex hull of $\Phi^{-1}(V_i|_{d_g})$, which we denote as $CH(\Phi^{-1}(V_i|_{d_g}))$. For any $v \in V_i$, we have $L_{sc}(c_i, v) \subset CH(\Phi^{-1}(V_i|_{d_g}))$. We define the maximal edge length in $V_i|_{d_g}$ as $l_{\max}(V_i|_{d_g}) = \max_{e=(v_a,v_b)\in E}\{l_e = \|v_i - v_j\|_2, v_a, v_b \in \Phi(CH(\Phi^{-1}(V_i|_{d_g})))\}$. Then we have the length of $\Phi(L_{sc}(c_i, v))$ is bounded by $l_{\max}(V_i|_{d_g})\|x, \widehat{c}_i\|_M$, where the pixel $x = \Phi^{-1}(v)$, $\| \ \|_M$ is the Manhattan distance in the media. We set $l_{\max}(V_i|_{d_g})\|x, \widehat{c}_i\|_M$ to be the upper bound $U(c_i, v)$.

Now, let's consider the lower bound $L(c_j, v)$. In the clustering $\{V_i|_{d_g}\}_{i=1}^K$, denote the set of neighboring clusters of each $V_i|_{d_g}$ as $Nb(V_i)|_{d_g}$. We consider the working area $\widehat{\Xi}(\widehat{c}_i, \widehat{c}_j)$ in which the corresponding clusters $V_i|_{d_g}$ and $V_j|_{d_g}$ are neighbors. We define the minimal edge length in $V_j|_{d_g}$ as $l_{\min}(V_j|_{d_g}) = \min_{e=(v_a,v_b)\in E}\{l_e = \|v_i - v_j\|_2, v_a, v_b \in V_j \cup Nb(V_j)|_{d_g}\}$. For any atom $x = \Phi^{-1}(v) \in \widehat{\Xi}(\widehat{c}_i, \widehat{c}_j)$, the geodesic distance from c_j to v in the graph $\mathcal{G}$ is not smaller than $l_{\min}(V_j|_{d_g})\|x, \widehat{c}_j\|_M$, which we set to be the lower bound $L(c_j, v)$.

Finally to ensure $U(c_i, v) \leq L(c_j, v)$, we define the working area $\widehat{\Xi}(\widehat{c}_i, \widehat{c}_j)$ with the following condition:

$$l_{\max}(V_i|_{d_g})\|x, \widehat{c}_i\|_M \leq l_{\min}(V_j|_{d_g})\|x, \widehat{c}_j\|_M \qquad \text{(S1)}$$

i.e., for each atom $x \in \widehat{\Xi}(\widehat{c}_i, \widehat{c}_j)$, it satisfies $\frac{l_{\max}(V_i|_{d_g})}{l_{\min}(V_j|_{d_g})}\|x, \widehat{c}_i\|_M \leq \|x, \widehat{c}_j\|_M$. That completes the proof. □

[1] In graph $\mathcal{G}$, the geodesic distance equals to the shortest path distance.

[2] We assume there is only one center c_i in the set C. The case of multiple centers can be handled in a similar way.

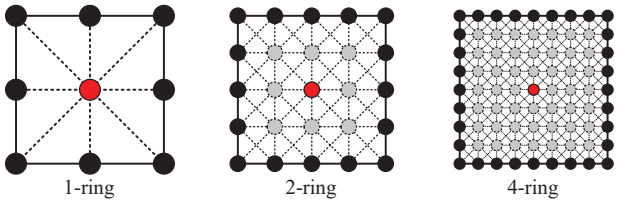


Figure S1. i-ring (black disks) of a center (red disks), $i = 1, 2, 4$, in the graph $\mathcal{G}$ representing an image.

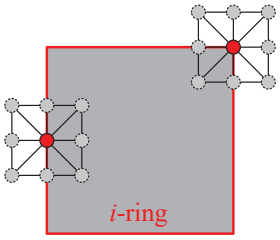


Figure S2. Proof of Property 4 in the image case. Assume that the vertex v (the red dot) lies on the i-ring (the red line). The grey area contains all j-rings, $j \leq i$. There are two possible locations of v on the i-ring: on the corner or on the edge.

Property 2. *For any $c \in C$, $v \in V \setminus C$ and a shortest path $\overline{cv} = \{v_{I_{j_1}} = c, v_{I_{j_2}}, \cdots, v_{I_{j_{n'}}} = v\}$ between c and v on $\mathcal{G}$, the q-path $\widetilde{cv}$ output from Algorithm 2 is exactly the shortest path $\overline{cv}$, if and only if $\forall a, b$, $1 \leq a < b \leq n'$, the indices $I_{j_a} < I_{j_b}$.*

Proof. If a shortest path $\overline{cv} = \{v_{I_{j_1}} = c, v_{I_{j_2}}, \cdots, v_{I_{j_{n'}}} = v\}$ between c and v on $\mathcal{G}$ satisfies that $\forall a, b$, $1 \leq a < b \leq n'$, the indices $I_{j_a} < I_{j_b}$, then this path is in the search space of Algorithm 2. Since Algorithm 2 uses the predefined traversal order to find a shortest path in the search space, the q-path output from Algorithm 2 can exactly find the shortest path $\overline{cv}$. On the other hand, if the indices of the vertices in the shortest path $\overline{cv}$ are not embedded in the predefined traversal order, it is not in the search space and therefore cannot be output from Algorithm 2. □

Definition 1. *For each vertex $v_i \in V$, we define an allowable region $\Omega(v_i)$ of v_i, which is a set of vertices satisfying $\Omega(v_i) = \{v_j \in V : j < i\}$.*

The allowable region can be visualized using the following i-ring concept.

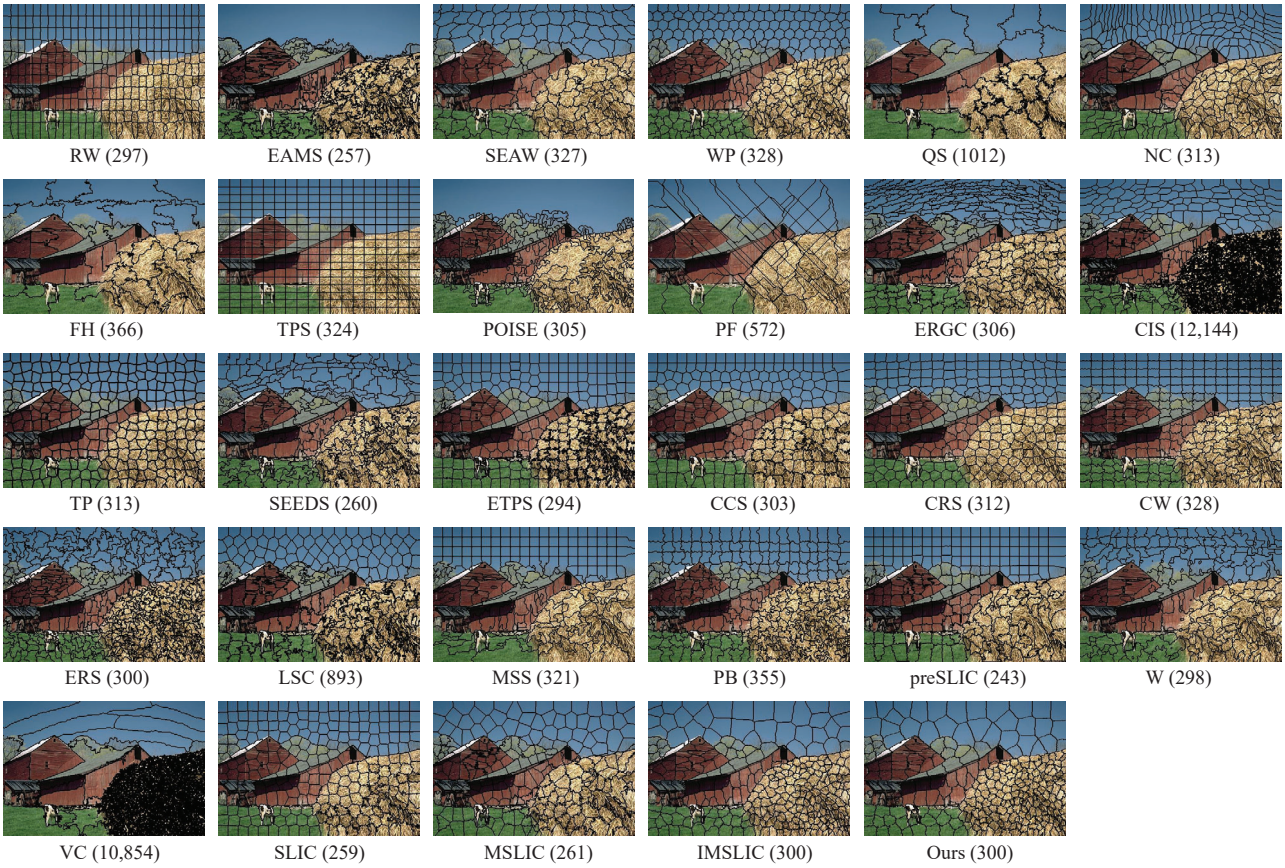


Figure S3. Visual comparison of 29 superpixel methods (another two methods are only suitable for RGBD depth images and do not show here). The number of superpixels specified by the user is 300 and the actual numbers of output superpixels are in parentheses. The figure is of high resolution and can zoom-in for details.

Definition 2. *In the graph $\mathcal{G}$, the 0-ring of a center c is the center itself, which is also a vertex in $\mathcal{G}$. The i-ring of c are those vertices sharing an edge with a vertex in $(i-1)$-ring and do not appear in any other j-ring, $0 \leq j < i, \forall i$.*

Figure S1 shows three examples of i-rings in an image.

For any $v_i \in V$, assume it is on the j-ring of a center c. It can readily be seen that allowable region $\Omega(v_i)$ of v_i includes all the k-rings of c, $0 \leq k < j$, and a portion of j-ring. On the other hand, all the k-rings of c, $k > j$, are not contained in $\Omega(v_i)$.

Property 3. *For any $c \in C$, $v \in V \setminus C$ and a shortest path $\overline{cv} = \{v_{I_{j_1}} = c, v_{I_{j_2}}, \cdots, v_{I_{j_{n'}}} = v\}$ between c and v on $\mathcal{G}$, the q-path $\widetilde{cv}$ output from Algorithm 2 is exactly the shortest path $\overline{cv}$, if and only if $\forall i$, $1 \leq i \leq n'$, the subpath $\overline{cv_{I_{j_i}}}$ of $\overline{cv}$ is contained in the allowable region of $v_{I_{j_i}}$.*

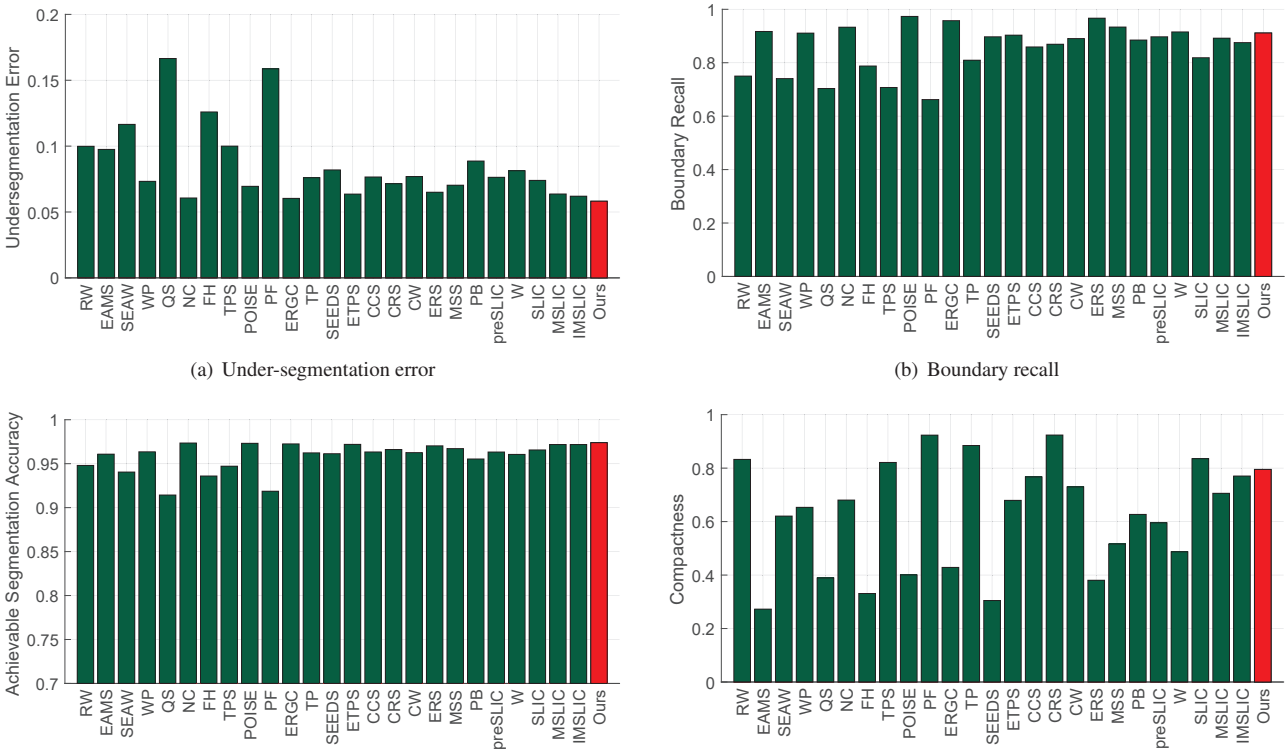
Proof. If $\forall i$, $1 \leq i \leq n'$, the subpath $\overline{cv_{I_{j_i}}}$ of $\overline{cv}$ is contained in the allowable region of $v_{I_{j_i}}$, then by Definition of allowable regions, $\forall 1 \leq a < i$, we have $I_{j_a} < I_{j_i}$ for all the vertices on the shortest path $\overline{cv}$. Accordingly, by Property 2, Property 3 is held. □

Property 4. *Assume $v \in V$ is in a general position, i.e., it has n_ζ neighbors in V. Then in these neighbors, half of them have indices larger than v.*

Proof. Referring to Figure S2, we prove this property in the image case (i.e., $\zeta = 2$ and $n_\zeta = 8$). The video case (i.e., $\zeta = 3$ and $n_\zeta = 26$) can be proved in a similar way. For any vertex $v \in V$ in a general position, assume that it lies on the i-ring of a center c. Then there are two possible locations of v on the i-ring: on the corner or on the edge.

Case 1: on the corner. In this case, five of eight neighbor vertices of v are outside of the i-ring and then they have indices larger than v.

Case 2: on the edge. In this case, three of eight neighbor vertices of v are outside of the i-ring and then have indices larger than v. In addition, there are two neighbor vertices of v on the edge of i-ring. In the Section 4 of the main paper, we set the rule that the neighboring vertices are visited in



(a) Under-segmentation error

(b) Boundary recall

(c) Achievable segmentation accuracy

(d) Compactness

Figure S4. Evaluation of 26 superpixel methods on the BSDS500 dataset, using the extended metric versions proposed in [50] that are independent of the number of superpixels (i.e., the area under the curve for $K \in [200, 700]$).

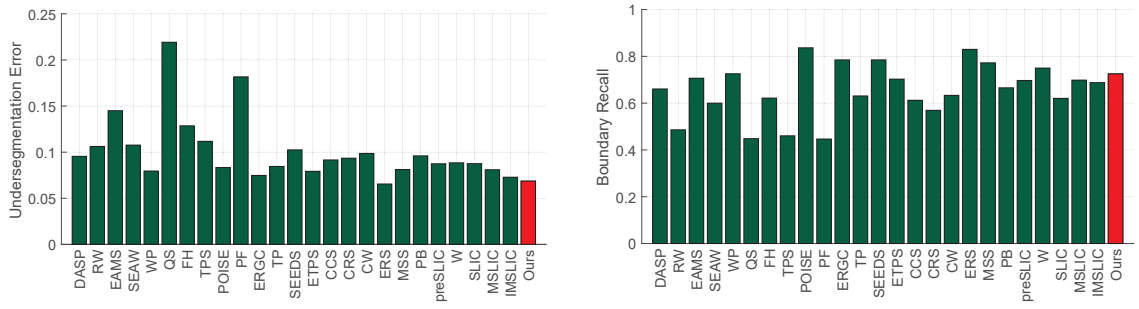
the order Γ = (N, W, S, E, NW, SW, SE, NE). Given this order, at least one of these two neighbor vertices on the edge of i-ring has the index larger than v. That completes the proof. □

S3. Comparison of 31 Superpixel Methods

We compared 31 superpixel methods (28 collected in [50], MSLIC [30], IMSLIC [31] and our qd-CSS) on five datasets: BSDS500 [2] and SBD [18] are outdoor RGB image datasets, NYUV2 [46] and SUNRBGD [47] are indoor RGBD depth image datasets, and Fashionista [58] is a clothing image dataset.

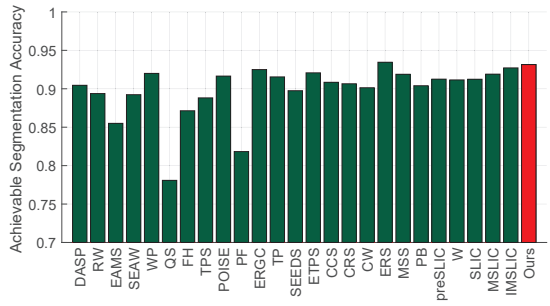
Following [50], these 31 superpixel methods can be classified into seven classes:

- Watershed-based: watershed method (W) [35], compact watershed (CW) [37], morphological superpixel segmentation (MSS) [3], water pixels (WP) [32];
- Density-based: edge-augmented mean shift (EAMS) [9], quick shift (QS) [53];
- Graph-based: normalized cuts (NC) [41], graph-based image segmentation (FH) [13], random walks (RW) [19], constant intensity superpixels (CIS) [54], entropy rate superpixels (ERS) [29], Boolean optimization superpixels (PB) [62], proposals for objects from improved seeds and energies (POISE) [21];
- Contour evolution: turbo pixels (TP) [25], Eikonal region growing clustering (ERGC) [6];
- Path-based: path finder (PF) [12], topology preserving superpixels (TPS) [16];
- Clustering-based: simple linear iterative clustering (SLIC) [1], depth adaptive superpixels (DASP) [56], VCells (VC) [23], voxel-cloud connectivity segmentation (VCCS) [38], preemptive SLIC (preSLIC) [37], linear spectral clustering (LSC) [27], MLSIC [30], IMSLIC [31] and our qd-CSS;
- Energy optimization: contour relaxed superpixels (CRS) [34], superpixels extracted via energy-driven sampling (SEEDS) [11], convexity constrained superpixels (CCS) [52], extended topology preserving segmentation (ETPS) [59];
- Wavelet-based: superpixels from edge-avoiding wavelets (SEAW) [49].

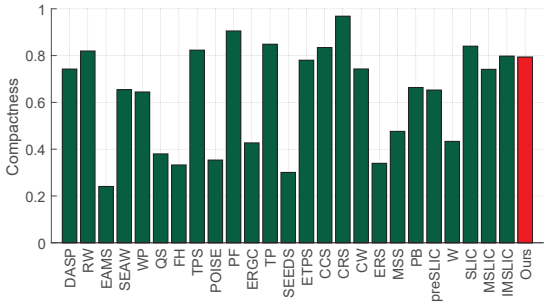


(a) Under-segmentation error

(b) Boundary recall

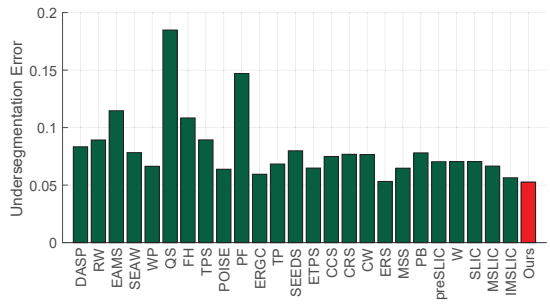


(c) Achievable segmentation accuracy

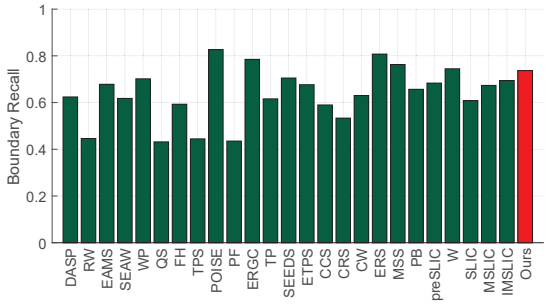


(d) Compactness

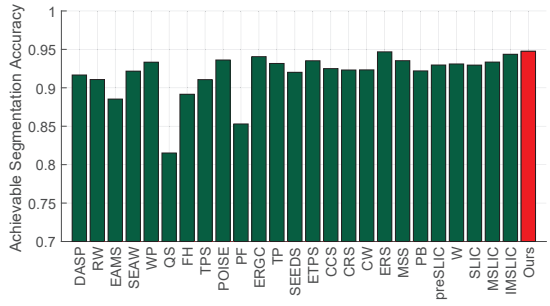
Figure S5. Evaluation of 26 superpixel methods on the NYUV2 dataset, using the extended metric versions proposed in [50] that are independent of the number of superpixels (i.e., the area under the curve for $K \in [200, 700]$).



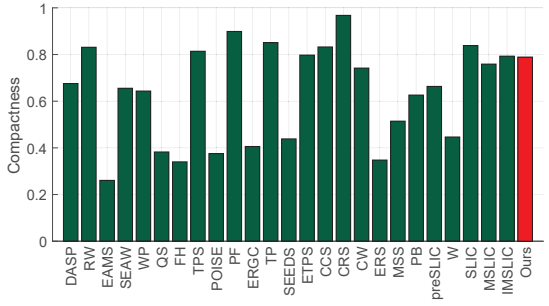
(a) Under-segmentation error



(b) Boundary recall

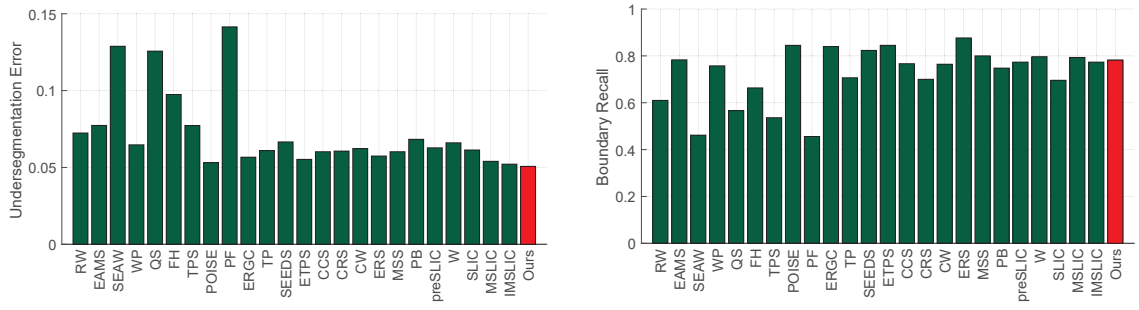


(c) Achievable segmentation accuracy



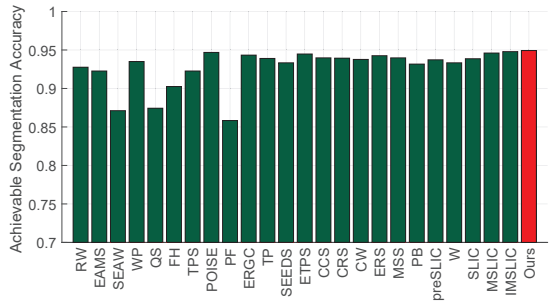
(d) Compactness

Figure S6. Evaluation of 26 superpixel methods on the SUNRBGD dataset, using the extended metric versions proposed in [50] that are independent of the number of superpixels (i.e., the area under the curve for $K \in [200, 700]$).

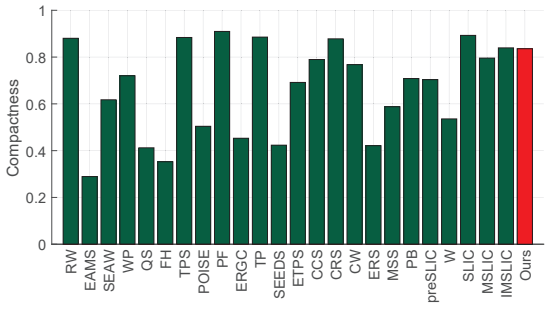


(a) Under-segmentation error

(b) Boundary recall

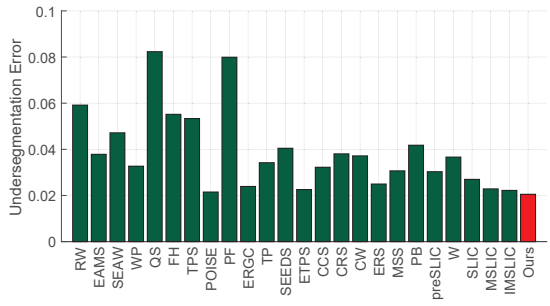


(c) Achievable segmentation accuracy

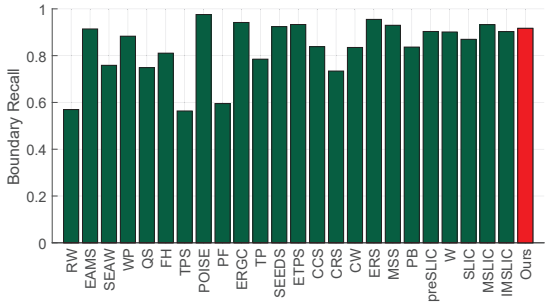


(d) Compactness

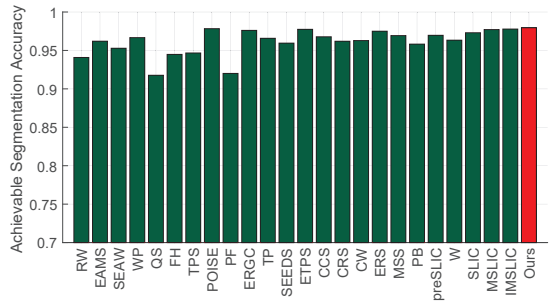
Figure S7. Evaluation of 25 superpixel methods on the SBD dataset, using the extended metric versions proposed in [50] that are independent of the number of superpixels (i.e., the area under the curve for $K \in [200, 700]$).



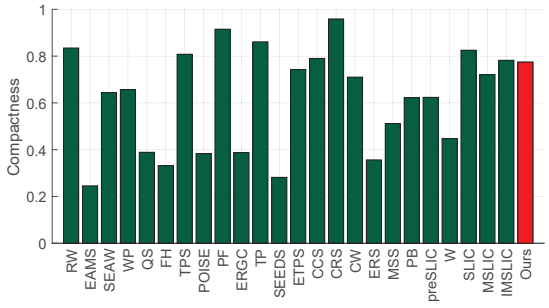
(a) Under-segmentation error



(b) Boundary recall

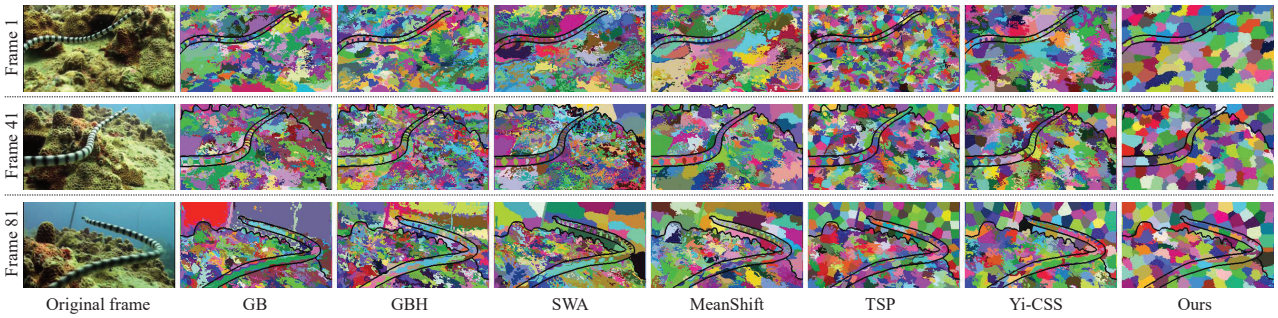


(c) Achievable segmentation accuracy

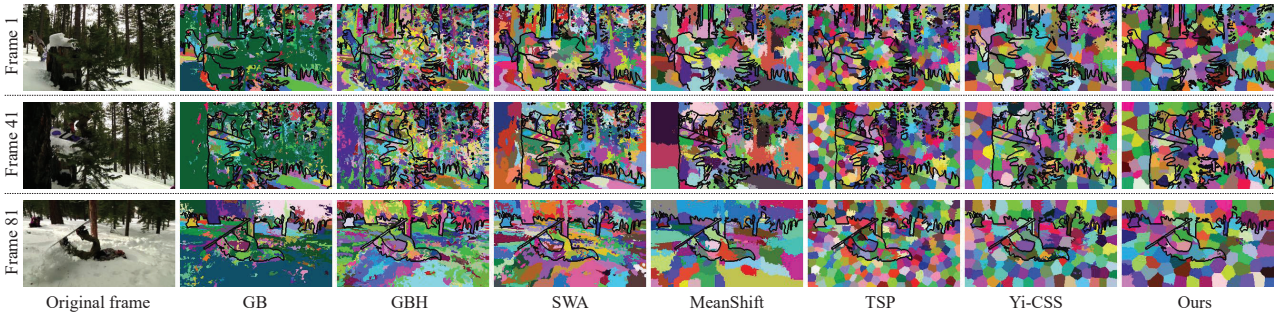


(d) Compactness

Figure S8. Evaluation of 25 superpixel methods on the Fashionista dataset, using the extended metric versions proposed in [50] that are independent of the number of superpixels (i.e., the area under the curve for $K \in [200, 700]$).



(a) Example video clip 1: supervoxels clipped on frames #1, #41 and #81



(b) Example video clip 2: supervoxels clipped on frames #1, #41 and #81

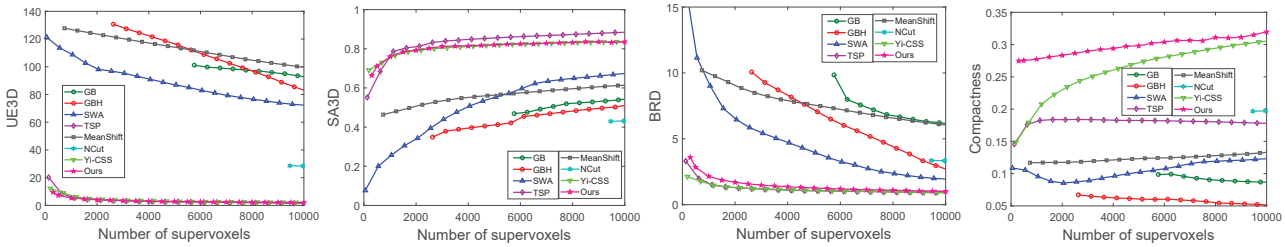
Figure S9. Visual comparison of superpixels (induced by clipping supervoxels on each image frame) obtained by GB [13], GBH [20], SWA [43, 44, 10], MeanShift [39], TSP [7], Yi-CSS [60] and our method qd-CSS. All the methods generate approximately 1,000 supervoxels. TSP, Yi-CSS and qd-CSS produce regular supervoxels (and accordingly regular clipped superpixels), while other methods produce highly irregular supervoxels. Compared to TSP, Yi-CSS and qd-CSS generates more supervoxels in content-rich areas and fewer supervoxels in content-sparse areas. In terms of UE3D, BRD, SA3D and CO on four video datasets (Figure S10), qd-CSS is better than Yi-CSS on average.

Among the above 31 methods, only ERS [28], IMSLIC and qd-CSS can output the exact number of superpixels as desired by users. Furthermore, only W, MSLIC, IMSLIC and qd-CSS have one parameter. All other methods have 2-6 parameters. We use the parameters optimized in [50] to evaluate these methods. Figure S3 shows a qualitative result of 29 superpixel methods on a RGB image (another two methods VCCS and DASP can only work for RGBD depth images).

In the introduction section of the main paper, connectivity is listed as one of five key criteria for evaluating superpixel algorithms. Stutz et al. [50] also suggest to strictly enforce connectivity by relabelling disjoint components in a superpixel as separated superpixels. Liu et al. [31] show that VC frequently outputs a much larger number of superpixels than the number desired by the user (Table 3 in [31]). Our experiment shows that LSC, CIS and VCCS have similar behavior: e.g., on BSDS500 dataset, when users input a desired number 300, the minimum, maximum and average numbers of superpixels produced by LSC, VC and CIS are $(275, 1884, 542.002)$, $(41, 35535, 3102.614)$ and $(220, 13275, 3311.968)$, respectively. See also Figure S3 for an example. VCCS can only work on RGBD depth images and allow users to input the number of supervoxels[3]. However, the number of supervoxels does not relate to the output number of superpixels. So we exclude these four methods for further comparison.

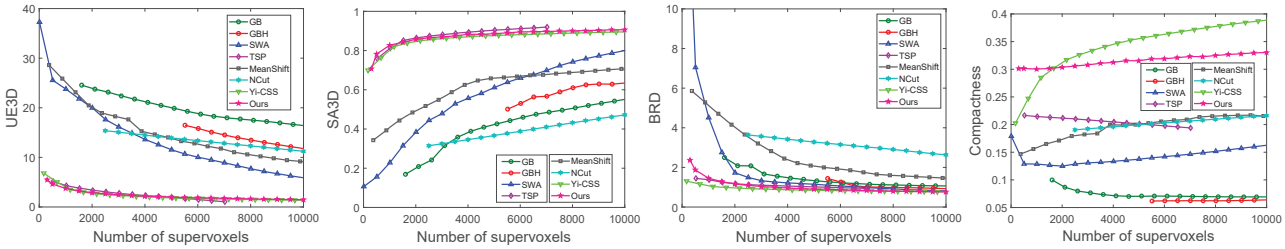
In the main paper, the metrics of under segmentation error (UE) [1, 25] and boundary recall (BR) [33] are used to measure the over-segmentation accuracy, and here we add one more metric, achievable segmentation accuracy (ASA) [28, 55]: (1) a lower UE value means that superpixels are better overlapped with a ground-truth segmentation, (2) a higher BR value means that fewer true ground-truth edges are missed, and (3) a higher ASA value means a better achievable accuracy when superpixels are used for subsequent segmentation. For a concise comparison, we use the extended version of these metrics proposed in [50] that are independent of the number of superpixels. Since NC is much slower than the other methods, we only evaluate it on BSDS500. Aslo noting that DASP only works on RBGD depth images, we evaluate 26 methods on NYUV2 and SUNRBGD depth image datasets, evaluate 26 methods on BSDS500 and evaluate 25 methods on SBD and Fash-

[3] This type of supervoxels is defined on 3D point cloud.



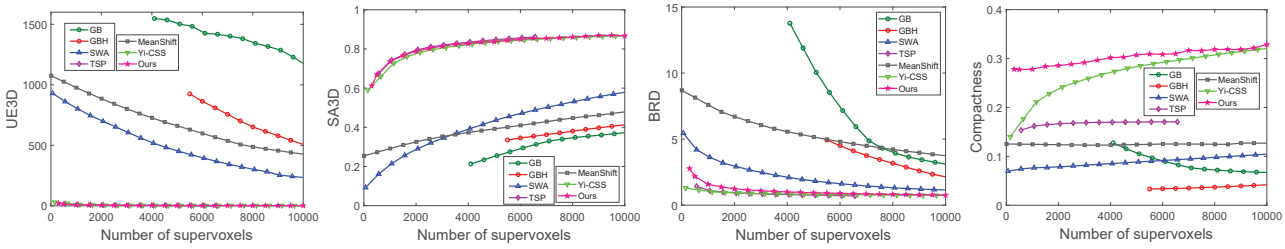
(i) 3D under-segmentation error (ii) 3D segmentation accuracy (iii) Boundary recall distance (iv) Compactness

(a) **BuffaloXiph dataset:** qd-CSS has the smallest UE3D and BRD, the highest CO and the second highest SA3D.



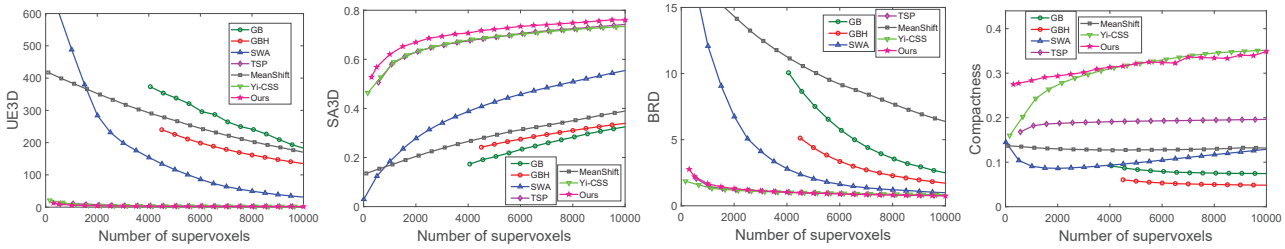
(i) 3D under-segmentation error (ii) 3D segmentation accuracy (iii) Boundary recall distance (iv) Compactness

(b) **SegTrack v2 dataset:** qd-CSS has the smallest UE3D, the second smallest BRD, the second highest CO and SA3D.



(i) 3D under-segmentation error (ii) 3D segmentation accuracy (iii) Boundary recall distance (iv) Compactness

(c) **BVDS dataset:** qd-CSS has the smallest UE3D and BRD, the highest CO and SA3D.



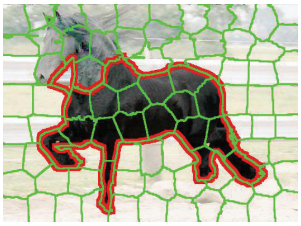
(i) 3D under-segmentation error (ii) 3D segmentation accuracy (iii) Boundary recall distance (iv) Compactness

(d) **CamVid dataset:** qd-CSS has the smallest UE3D and BRD, the highest CO and SA3D.

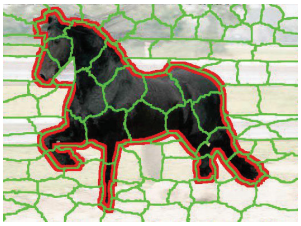
Figure S10. Evaluation of 8 supervoxel methods on the BuffaloXiph, SegTrack v2, BVDS and CamVid datasets. Our method qd-CSS have the best overall performance on the measures UE3D, BRD, SA3D and CO.

ionista datasets.

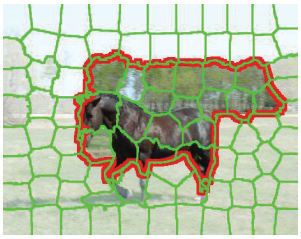
The quantitative results are summarized in Figures S4, S5, S6, S7 and S8. In terms of UE metric, qd-CSS is ranked No.1, 1, 2, 1, 1 in five datasets. In terms of BR metric, qd-CSS is ranked No.8, 8, 8, 10, 6 in five datasets. In terms of ASA metric, qd-CS is ranked No.1, 1, 2, 1, 1 in five datasets. The average rank of qd-CSS over 15 measures (i.e., three metrics vs. five datasets) is 3.47. There are only five methods whose all 15 measures are ranked within Top 10. We sort these five methods using their



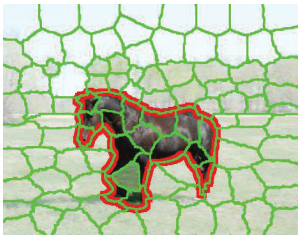
(a) ETPS (0.8161)



(b) qd-CSS (0.9264)



(c) ETPS (0.6019)



(d) qd-CSS (0.9113)

Figure S11. Contour closure results on two examples in the WHD dataset using 100 superpixels generated by qd-CSS and ETPS. The optimal closure contours are shown in red, and the boundaries of superpixels are shown in green. The F measure value for each closure contour is shown below each image; the range of the F-measure values is $[0, 1]$, and larger values indicate better results.

average ranks: qd-CSS(3.47), POISE(3.73), ERS(3.87), ERGC(4.20), ETPS(6.07). Overall, qd-CSS is ranked No. 1.

Compactness [42] is another important metric which measures shape regularity for superpixels. It was observed [31, 42] that compact superpixels usually have regular neighboring relations and then better segment foregrounds in images. In all 25 superpixel methods evaluated on five datasets, the average ranking of compactness for the top five methods are qd-CSS (8.6), POISE (21.2), ERS (22.8), ERGC (20.2), ETPS (11.2), showing that only qd-CSS and ETPS have good compactness. Then We further compare qd-CSS and ETPS in the application of optimal image contour (Section S5).

S4. Comparison of 8 Supervoxel Methods

By setting $\zeta = 3$ for the manifold $\mathcal{M}_\zeta$, qd-CSS can also generate supervoxels in video. We compared qd-CSS with Yi-CSS [60] and six representative supervoxel methods collected in [57], which are NCut [45, 15, 14], SWA [43, 44, 10], MeanShift [39], GB [13], GBH [20] and TSP [7], and evaluated them on four video datasets, i.e., BuffaloXiph [8], SegTrack v2 [26], BVDS [51, 17] and CamVid [5], all of which have ground truth labels drawn by human annotators. Some qualitative results of these supervoxel methods are shown in Figures S9.

We use three commonly used quality metrics pertaining

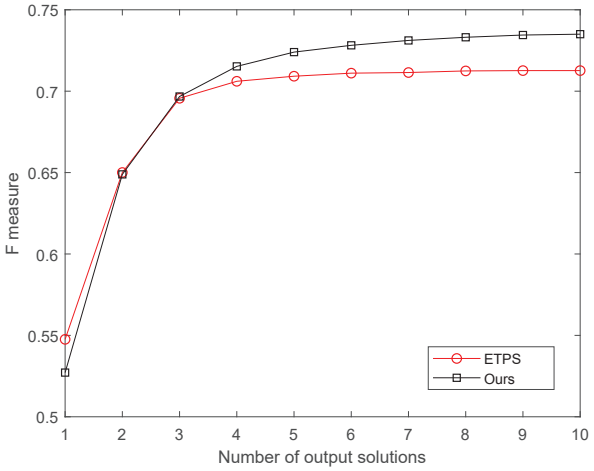


Figure S12. F-measure values with respect to the number of output solutions (from 1 to 10) in the framework developed by Levinshtein et al. [24] on the WHD dataset. The number of superpixels is fixed to 100. The range of the F-measure values is $[0, 1]$, and larger values indicate better results.

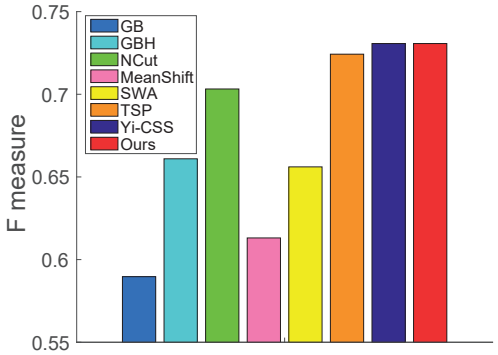


Figure S13. Average F measure in spatiotemporal closure application. The results are averaged on Stein et al. [48] dataset. qd-CSS and Yi-CSS achieves the best average F measure among eight methods, and qd-CSS is slightly better than Yi-CSS.

to supervoxels for evaluating the over-segmentation accuracy: 3D under-segmentation error (UE3D) and 3D segmentation accuracy (SA3D) [7, 25, 57] are complementary to each other and cooperatively measure how tight supervoxels overlapping with ground truth segmentation. Boundary recall distance (BRD) [36, 57] measures to what extent the ground truth boundaries are correctly retrieved by supervoxel boundaries. Compactness (CO) [61] measure the shape regularity of supervoxels. Better supervoxels' quality means lower values of UE3D and BRD, and higher values of SA3D, and CO.

The quantitative results of the UE3D, BRD, SA3D and CO metrics evaluated on four video datasets are summarized in Figure S10. The results show that among 8 supervoxel methods, our qd-CSS, Yi-CSS and TSP are top three methods, and qd-CSS has the best overall performance.

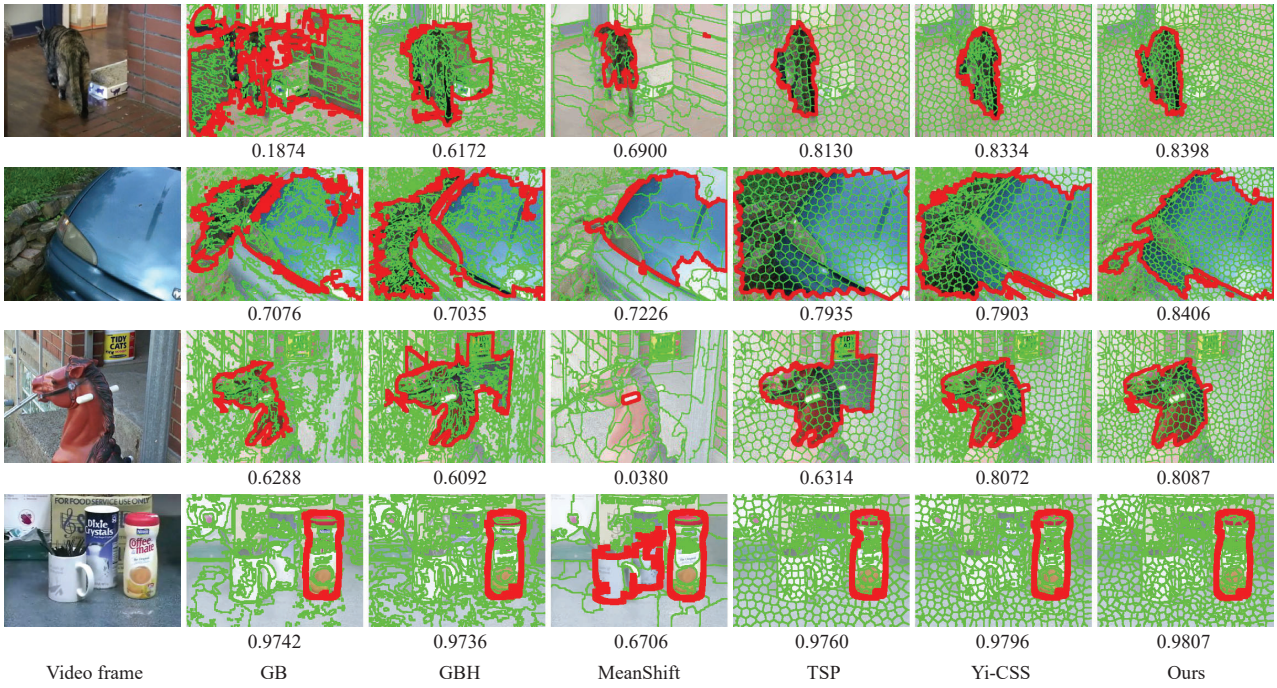


Figure S14. Spatiotemporal Closure results on four examples in Stein et al. dataset [48]. Results using six supervoxel methods are presented. The optimal spatiotemporal closure contours are shown in red, and the boundaries of supervoxels are shown in green. One representative frame is illustrated for each video. The F measure value for each spatiotemporal closure is shown below each frame; the range of the F measure values is $[0, 1]$, and larger values mean better results.

S5. Applications

In Section 6 of the main paper, superpixels and supervoxels are directly evaluated on one image and two video applications. Here we present the full details of the comparison.

Optimal image and video closure. To avoid the exhaustive searching in the entire image space of all pixels, Levinshtein et al. [24] propose a novel framework that separates an object from background by finding subsets of superpixels/supervoxels such that the contour of the union of these atomic regions has strong boundary support in the image/video. We use the source code provided by the authors[4] to compare different superpixels/supervoxel methods on an image dataset WHD [4] and a video dataset [48] with ground-truth segmentations. For image contour closure evaluated on the WHD dataset, we compare two superpixel methods – qd-CSS and ETPS as selected in Section S3 — and illustrate some qualitative results in Figure S11. The F-measure values averaged on the WHD dataset are summarized in Figure S12, showing that qd-CSS has better performance than ETPS. For optimal video closure by supervoxel grouping, the dataset of Stein et al. [48] in which each sequence has a ground truth segmentation mask, is used to perform a quantitative assessment. Seven representative methods (GB, GBH, NCut, MeanShift, SWA, TSP, Yi-CSS) and our CSS method are compared. The average F measures across all sequences are summarized in Figure S13. Some qualitative results are illustrated in Figure S14.

[4] http://www.cs.toronto.edu/~babalex/spatiotemporal_closure_code.tgz

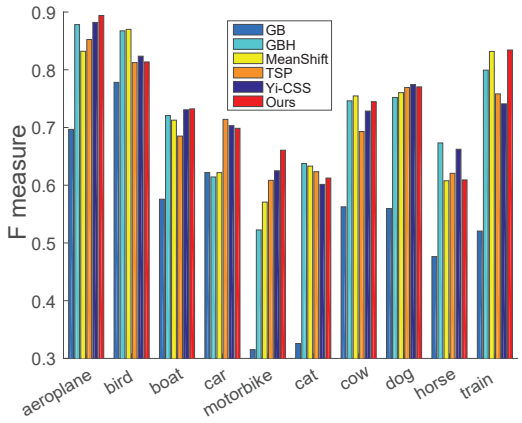


Figure S15. The average F measures of different supervoxel results on Youtube-Objects Dataset. The results are plotted per object class and each object class contains several video sequences. Larger F measure values mean better foreground propagation results. The results show that qd-CSS is ranked 1, 4, 1, 3, 1, 4, 3, 2, 4, 1 in ten object classes and achieves the best average performance.

Figure S16. Foreground propagation results of six supvoxel methods on two examples in Youtube-objects dataset [40]. For each example video, three representative frames are selected. The foreground masks are shown in green. The incorrectly labeled areas are circled in red. The average F measure for each example video is shown below three frames. the value of the F measure ranges in $[0, 1]$, and larger values mean better results.

These results show that qd-CSS achieves the best spatiotemporal closure performance.

Foreground propagation in videos. Given the first frame with manual annotation for the foreground object, a novel approach is proposed in [22] to propagate the foreground region through time, by using supervoxels to guide the estimates towards long-range coherent regions. We use the source code provided by the authors[5] to compare[6] five representative methods (GB, GBH, MeanShift, TSP and Yi-CSS) and our qd-CSS. Youtube-Objects dataset [40] (126 videos with 10 object classes) with foreground ground-truth, is used to perform a quantitative assessment. The average F measures of 10 classes are summarized in Figure S15. In particular, the quantitative results using F-measure reveals that qd-CSS achieves the best performance in four object classes, i.e., aeroplane, boat, motorbike, train. Some qualitative results are illustrated in Figure S16 and more results are presented in accompanying demo video. These results show that qd-CSS achieves the best performance averagely in ten classes.

[5] www.cs.utexas.edu/∼suyog/code_release_public.tar

[6] NCut is not compared due to its high computational cost. SWA is not compared since there are many long videos in this dataset and SWA requires huge memory.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(11):2012–2281, 2012. 4, 7

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image

segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011. 4

[3] Wanda Benesova and Michal Kottman. Fast superpixel segmentation using morphological processing. In *Conference on Machine Vision and Machine Learning*, MVML '14, pages 67–1–9, 2014. 4

[4] Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *Proceedings of the 7th European Conference on Computer Vision-Part II*, ECCV '02, pages 109–124. Springer-Verlag, 2002. 10

[5] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision*, ECCV '08, pages 44–57. Springer, 2008. 9

[6] P. Buyssens, I. Gardin, and S. Ruan. Eikonal based region growing for superpixels generation: Application to semi-supervised real time organ segmentation in ct images. *IRBM*, 35(1):20–26, 2014. 4

[7] Jason Chang, Donglai Wei, and John W. Fisher III. A video representation using temporal superpixels. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 2051–2058, 2013. 7, 9

[8] Albert Y.C. Chen and Jason J. Corso. Propagating multi-class pixel labels throughout video frames. In *Proceedings of Western New York Image Processing Workshop*, 2010. 9

[9] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002. 4

[10] Jason J. Corso, Eitan Sharon, Shishir Dube, Suzie El-Saden, Usha Sinha, and Alan L. Yuille. Efficient multilevel brain tumor segmentation with integrated Bayesian model classification. *IEEE Trans. Med. Imaging*, 27(5):629–640, 2008. 7, 9

[11] Michael Van den Bergh, Xavier Boix, Gemma Roig, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. *International Journal of Computer Vision*, 111(3):298–314, 2015. 4

[12] Fabio Drucker and John Maccormick. Fast superpixels for video analysis. In *The Workshop on Motion & Video Computing*, pages 55–62, 2009. 4

[13] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. 4, 7, 9

[14] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):214–225, 2004. 9

[15] Charless C. Fowlkes, Serge J. Belongie, and Jitendra Malik. Efficient spatiotemporal grouping using the Nyström method. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '01, pages 231–238, 2001. 9

[16] Huazhu Fu, Xiaochun Cao, Dai Tang, Yahong Han, and Dong Xu. Regularity preserved superpixels and supervoxels. *IEEE Transactions on Multimedia*, 16(4):1165–1175, 2014. 4

[17] Fabio Galasso, Naveen Shankar Nagaraja, Tatiana Jiménez Cárdenas, Thomas Brox, and Bernt Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 3527–3534, 2013. 9

[18] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *IEEE 12th International Conference on Computer Vision*, ICCV '09, pages 1–8, 2009. 4

[19] Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 28(11):1768–1783, 2006. 4

[20] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan A. Essa. Efficient hierarchical graph-based video segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'10, pages 2141–2148, 2010. 7, 9

[21] Ahmad Humayun, Fuxin Li, and James M. Rehg. The middle child problem: Revisiting parametric min-cut and seeds for object proposals. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '15, pages 1600–1608, 2015. 4

[22] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *13th European Conference on Computer Vision*, ECCV '14, pages 656–671, 2014. 11

[23] Wang Jie and Wang Xiaoqiang. Vcells: simple and efficient superpixels using edge-weighted centroidal voronoi tessellations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 34(6):1241–7, 2012. 4

[24] Alex Levinshtein, Cristian Sminchisescu, and Sven Dickinson. Optimal image and video closure by superpixel grouping. *International Journal of Computer Vision*, 100(1):99–119, 2012. 9, 10

[25] Alex Levinshtein, Adrian Stere, Kiriakos N. Kutulakos, David J. Fleet, Sven J. Dickinson, and Kaleem Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(12):2290–2297, 2009. 4, 7, 9

[26] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV '13, pages 2192–2199, 2013. 9

[27] Zhengqin Li and Jiansheng Chen. Superpixel segmentation using linear spectral clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '15, pages 1356–1363, 2015. 4

[28] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 2097–2104, 2011. 7

[29] Ming Yu Liu, O Tuzel, S Ramalingam, and R Chellappa. Entropy rate superpixel segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 2097–2104, 2011. 4

[30] Yong-Jin Liu, Chengchi Yu, Minjing Yu, and Ying He. Manifold SLIC: a fast method to compute content-sensitive super-

pixels. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 651–659, 2016. 4

[31] Yong-Jin Liu, Minjing Yu, Bing-Jun Li, and Ying He. Intrinsic manifold SLIC: A simple and efficient method for computing content-sensitive superpixels. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(3):653–666, 2018. 4, 7, 9

[32] Vaïa Machairas, Matthieu Faessel, David Cárdenas-Peña, Théodore Chabardes, Thomas Walter, and Etienne Decencière. Waterpixels. *IEEE Transactions on Image Processing*, 24(11):3707–3716, 2015. 4

[33] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004. 7

[34] Rudolf Mester, Christian Conrad, and Alvaro Guevara. Multichannel segmentation using contour relaxation: Fast superpixels and temporal propagation. In *Scandinavian Conference on Image Analysis*, pages 250–261, 2011. 4

[35] Fernand Meyer. Color image segmentation. In *International Conference on Image Processing and Its Applications*, pages 303–306, 1992. 4

[36] Alastair Philip Moore, Simon Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones. Superpixel lattices. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 9

[37] Peer Neubert and Peter Protzel. Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. In *International Conference on Pattern Recognition*, ICPR '14, pages 996–1001, 2014. 4

[38] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation - supervoxels for point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 2027–2034, 2013. 4

[39] Sylvain Paris and Frédo Durand. A topological approach to hierarchical segmentation using mean shift. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'07, 2007. 7, 9

[40] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'12, pages 3282–3289, 2012. 11

[41] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *IEEE International Conference on Computer Vision*, ICCV '03, pages 10–17, 2003. 4

[42] Alexander Schick, Mika Fischer, and Rainer Stiefelhagen. An evaluation of the compactness of superpixels. *Pattern Recognition Letters*, 43(1):71–80, 2014. 9

[43] Eitan Sharon, Achi Brandt, and Ronen Basri. Fast multiscale image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '00, pages 1070–1077, 2000. 7, 9

[44] Eitan Sharon, Meirav Galun, Dahlia Sharon, Ronen Basri, and Achi Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):810–813, 2006. 7, 9

[45] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. 9

[46] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the 12th European Conference on Computer Vision*, ECCV '12, pages 746–760, 2012. 4

[47] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '15, pages 567–576, 2015. 4

[48] Andrew N. Stein, Derek Hoiem, and Martial Hebert. Learning to find object boundaries using motion cues. In *IEEE 11th International Conference on Computer Vision*, ICCV '07, pages 1–8, 2007. 9, 10

[49] Johann Strassburg, Rene Grzeszick, Leonard Rothacker, and Gernot A Fink. On the influence of superpixel methods for image parsing. In *VISAPP*, pages 518–527, 2015. 4

[50] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018. 4, 5, 6, 7

[51] Patrik Sundberg, Thomas Brox, Michael Maire, Pablo Arbeláez, and Jitendra Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 2233–2240. IEEE, 2011. 9

[52] H. Emrah Tasli, Cevahir Cigla, and A. Aydin Alatan. Convexity constrained efficient superpixel and supervoxel extraction. *Signal Processing Image Communication*, 33:71–85, 2015. 4

[53] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *European Conference on Computer Vision*, ECCV '08, pages 705–718, 2008. 4

[54] Olga Veksler, Yuri Boykov, and Paria Mehrani. Superpixels and supervoxels in an energy optimization framework. In *European Conference on Computer Vision*, ECCV '10, pages 211–224, 2010. 4

[55] Peng Wang, Gang Zeng, Rui Gan, Jingdong Wang, and Hongbin Zha. Structure-sensitive superpixels via geodesic distance. *International Journal of Computer Vision*, 103(1):1–21, 2013. 7

[56] D Weikersdorfer, D Gossow, and M Beetz. Depth-adaptive superpixels. In *International Conference on Pattern Recognition*, pages 2087–2090, 2012. 4

[57] Chenliang Xu and Jason J. Corso. Libsvx: A supervoxel library and benchmark for early video processing. *International Journal of Computer Vision*, 119(3):272–290, 2016. 9

[58] Kota Yamaguchi, M. Hadi Kiapour, Luis E. Ortiz, and Tamara L. Berg. Parsing clothing in fashion photographs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '12, pages 3570–3577, 2012. 4

[59] Jian Yao, Marko Boben, Sanja Fidler, and Raquel Urtasun. Real-time coarse-to-fine topologically preserving segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '15, pages 2947–2955, 2015. 4

[60] Ran Yi, Yong-Jin Liu, and Yu-Kun Lai. Content-sensitive supervoxels via uniform tessellations on video manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '18, pages 646–655, 2018. 7, 9

[61] Ran Yi, Yong-Jin Liu, and Yu-Kun Lai. Evaluation on the compactness of supervoxels. In *IEEE International Conference on Image Processing*, ICIP '18, pages 2212–2216, 2018. 9

[62] Yuhang Zhang, Richard Hartley, John Mashford, and Stewart Burn. Superpixels via pseudo-boolean optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '12, 2012. 4