Content-Sensitive Supervoxels via Uniform Tessellations on Video Manifolds (Supplemental Material)

Ran Yi, Yong-Jin Liu*
Tsinghua University, China
{yr16,liuyongjin}@tsinghua.edu.cn

Yu-Kun Lai
Cardiff University, UK
Yukun.Lai@cs.cf.ac.uk

S1. Overview

This supplemental material contains two parts:

- the proofs of theoretic results in the main paper (Section S2),
- more experimental results, including both qualitative and quantitative results, on three video datasets (Section S3),

S2. Proofs of Main Results

Definition 1. *The diameter d_i of a cell $C_{\mathcal{M}}(s_i)$, $s_i \in S_K$, is the maximum Euclidean distance between pairs of points in the cell, i.e.,*

$$d_i = \max_{\forall x,y \in C_{\mathcal{M}}(s_i)} \|x - y\|_2 \tag{S1}$$

Denote by $p_1(d_i)$ and $p_2(d_i)$ the two points in $C_{\mathcal{M}}(s_i)$ satisfying $\|p_1(d_i) - p_2(d_i)\| = d_i$.

Theorem 2. *Let s_m, s_i, s_j be three generators in an $RVT(S_K, \mathcal{M})$. For the cells $C_{\mathcal{M}}(s_m)$, $C_{\mathcal{M}}(s_i)$ and $C_{\mathcal{M}}(s_j)$, let m_m, m_i, m_j be their masses, s'_m, s'_i, s'_j be their mass centroids, respectively. For any partitioning of $C_{\mathcal{M}}(s_m)$ into two new cells $C'(p_1(d_m))$ and $C'(p_2(d_m))$, which satisfies $p_1(d_m) \in C'(p_1(d_m))$, $p_2(d_m) \in C'(p_2(d_m))$, $C'(p_1(d_m)) \cap C'(p_2(d_m)) = \emptyset$ and $C'(p_1(d_m)) \cup C'(p_2(d_m)) = C_{\mathcal{M}}(s_m)$, let s'_k, s'_p and s'_q be the mass centroids of $C_{\mathcal{M}}(s_i) \cup C_{\mathcal{M}}(s_j)$, $C'(p_1(d_m))$ and $C'(p_2(d_m))$, respectively. If $\|s'_p - s'_m\|_2 > \tau_{m,i,j}$ and $\|s'_q - s'_m\|_2 > \tau_{m,i,j}$, where*

$$\tau_{m,i,j} = \sqrt{\frac{m_i m_j}{m_m(m_i + m_j)}} \|s'_i - s'_j\|_2 \tag{S2}$$

then the pair of operations $(S, M) : (s_m, (s_i, s_j)) \to ((s'_p, s'_q), s'_k)$ does not increase the tessellation energy $\mathcal{E}$.

Proof. First, we consider the change of the tessellation energy by applying a merging operation $M : (s_i, s_j) \to s'_k$. This operation merges $C_{\mathcal{M}}(s_i)$ and $C_{\mathcal{M}}(s_j)$ into a new cell $C'(s'_k) = C_{\mathcal{M}}(s_i) \cup C_{\mathcal{M}}(s_j)$, whose mass centroid is s'_k. A merging operation will always increase the energy and the energy change is:

$$\begin{aligned}
\Delta\mathcal{E}_m &= \mathcal{E}(s'_k, C'(s'_k)) - \mathcal{E}(s_i, C_{\mathcal{M}}(s_i)) - \mathcal{E}(s_j, C_{\mathcal{M}}(s_j)) \\
&= \int_{x \in C'(s'_k)} \|x - s'_k\|_2^2 dx - \int_{x \in C_{\mathcal{M}}(s_i)} \|x - s_i\|_2^2 dx \\
&\quad - \int_{x \in C_{\mathcal{M}}(s_j)} \|x - s_j\|_2^2 dx \\
&= \int_{x \in C_{\mathcal{M}}(s_i)} \|x - s'_k\|_2^2 dx + \int_{x \in C_{\mathcal{M}}(s_j)} \|x - s'_k\|_2^2 dx \\
&\quad - \int_{x \in C_{\mathcal{M}}(s_i)} \|x - s_i\|_2^2 dx - \int_{x \in C_{\mathcal{M}}(s_j)} \|x - s_j\|_2^2 dx \\
&\leq \int_{x \in C_{\mathcal{M}}(s_i)} \|x - s'_k\|_2^2 dx + \int_{x \in C_{\mathcal{M}}(s_j)} \|x - s'_k\|_2^2 dx \\
&\quad - \int_{x \in C_{\mathcal{M}}(s_i)} \|x - s'_i\|_2^2 dx - \int_{x \in C_{\mathcal{M}}(s_j)} \|x - s'_j\|_2^2 dx
\end{aligned}$$

Since s'_i and s'_j are the mass centroids of cells $C_{\mathcal{M}}(s_i)$ and $C_{\mathcal{M}}(s_j)$, respectively, we have

$$\begin{aligned}
\int_{x \in C_{\mathcal{M}}(s_i)} \|x - s'_k\|_2^2 dx - \int_{x \in C_{\mathcal{M}}(s_i)} \|x - s'_i\|_2^2 dx = \\
\int_{x \in C_{\mathcal{M}}(s_i)} \|s'_i - s'_k\|_2^2 dx + \\
\int_{x \in C_{\mathcal{M}}(s_i)} \langle x - s'_i, s'_i - s'_k \rangle \, dx \\
= \int_{x \in C_{\mathcal{M}}(s_i)} \|s'_i - s'_k\|_2^2 dx = m_i \|s'_i - s'_k\|_2^2
\end{aligned}$$

and

$$\begin{aligned}
\int_{x \in C_{\mathcal{M}}(s_j)} \|x - s'_k\|_2^2 dx - \int_{x \in C_{\mathcal{M}}(s_j)} \|x - s'_j\|_2^2 dx = \\
m_j \|s'_j - s'_k\|_2^2
\end{aligned}$$

*Corresponding author

Note that s'_k is the mass centroid of $C_{\mathcal{M}}(s_i) \cup C_{\mathcal{M}}(s_j)$, $s'_k = \frac{m_i s'_i + m_j s'_j}{m_i + m_j}$. Therefore

$$\begin{aligned}\Delta\mathcal{E}_m &\leq m_i \|s'_i - s'_k\|_2^2 + m_j \|s'_j - s'_k\|_2^2 \\ &= \frac{m_i m_j}{m_i + m_j} \|s'_i - s'_j\|_2^2\end{aligned}$$

Second, we consider the change of the tessellation energy by applying a splitting operation $S : s_m \rightarrow (s'_p, s'_q)$. This operation splits a cell $C_{\mathcal{M}}(s_m)$ into two new cells $C'(p_1(d_m))$ and $C'(p_2(d_m))$, whose mass centroids are s'_p and s'_q, respectively. A splitting operation always decreases the energy and the energy change is:

$$\begin{aligned}\Delta\mathcal{E}_s &= \mathcal{E}(s'_p, C'(p_1(d_m))) + \mathcal{E}(s'_q, C'(p_2(d_m))) \\ &\quad - \mathcal{E}(s_m, C_{\mathcal{M}}(s_m)) \\ &= \int_{x \in C'(p_1(d_m))} \|x - s'_p\|_2^2 dx \\ &\quad + \int_{x \in C'(p_2(d_m))} \|x - s'_q\|_2^2 dx \\ &\quad - \int_{x \in C_{\mathcal{M}}(s_m)} \|x - s_m\|_2^2 dx \\ &= \int_{x \in C'(p_1(d_m))} \|x - s'_p\|_2^2 dx \\ &\quad + \int_{x \in C'(p_2(d_m))} \|x - s'_q\|_2^2 dx \\ &\quad - \int_{x \in C'(p_1(d_m))} \|x - s_m\|_2^2 dx \\ &\quad - \int_{x \in C'(p_2(d_m))} \|x - s_m\|_2^2 dx \\ &\leq \int_{x \in C'(p_1(d_m))} \|x - s'_p\|_2^2 dx \\ &\quad + \int_{x \in C'(p_2(d_m))} \|x - s'_q\|_2^2 dx \\ &\quad - \int_{x \in C'(p_1(d_m))} \|x - s'_m\|_2^2 dx \\ &\quad - \int_{x \in C'(p_2(d_m))} \|x - s'_m\|_2^2 dx\end{aligned}$$

Let m_p and m_q be the masses of $C'(p_1(d_m))$ and $C'(p_2(d_m))$, respectively. We have $m_p + m_q = m_m$.

Since s'_p and s'_q are mass centroids of cells $C'(p_1(d_m))$ and $C'(p_2(d_m))$, respectively, we have

$$\Delta\mathcal{E}_s \leq -m_p \|s'_p - s'_m\|_2^2 - m_q \|s'_q - s'_m\|_2^2$$

If $\|s'_p - s'_m\|_2 \geq \tau_{m,i,j}$ and $\|s'_q - s'_m\|_2 \geq \tau_{m,i,j}$, then

$$\begin{aligned}\Delta\mathcal{E}_s &\leq -m_p \|s'_p - s'_m\|_2^2 - m_q \|s'_q - s'_m\|_2^2 \\ &\leq -(m_p + m_q)\tau_{m,i,j}^2 \\ &= -m_m \frac{m_i m_j}{m_m(m_i + m_j)} \|s'_i - s'_j\|_2^2 \\ &= -\frac{m_i m_j}{m_i + m_j} \|s'_i - s'_j\|_2^2 \\ \Rightarrow \Delta\mathcal{E}_m + \Delta\mathcal{E}_s &\leq 0\end{aligned}$$

Therefore, the pair of operations $(S, M) : (s_m, (s_i, s_j)) \rightarrow ((s'_p, s'_q), s'_k)$ does not increase the tessellation energy $\mathcal{E}$. □

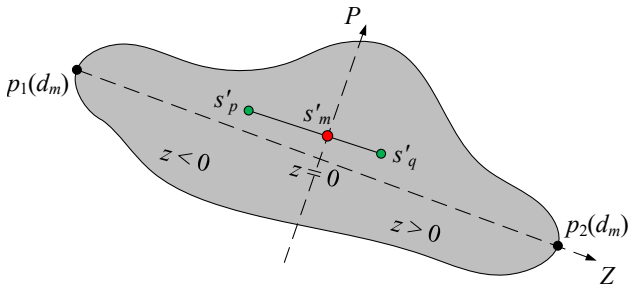


Figure S1. Illustration used for proof of Corollary 1. The shaded area is $C_{\mathcal{M}}(s_m)$.

Corollary 1. *Denote a voxel in the video Ξ as v_i and let*

$$m_{\min} = \min_{v_i \in \Xi} \{V(\Phi(\boxdot_{v_i}))\}, \tag{S3}$$

$$d_{\max} = \max_{v_i \in \Xi} \{d(\Phi(\boxdot_{v_i}))\}, \tag{S4}$$

where $d(\Phi(\boxdot_{v_i}))$ is the diameter of $\Phi(\boxdot_{v_i}) \subset \mathcal{M}$. Let s_m, s_i, s_j be three generators in an $RVT(S_K, \mathcal{M})$ and s'_m be the mass centroid of $C_{\mathcal{M}}(s_m)$. Let P be the hyperplane which passes through s'_m and is perpendicular to the line connecting $p_1(d_m)$ and $p_2(d_m)$. P partitions $C_{\mathcal{M}}(s_m)$ into $C'(p_1(d_m))$ and $C'(p_2(d_m))$. If $d_m \geq w(\frac{m_m}{m_{\min}}\tau_{m,i,j} + d_{\max})$, where $m_m = V(C_{\mathcal{M}}(s_m))$, $w = \max\{1+\lambda, \frac{1+\lambda}{\lambda}\}$, $\lambda = \frac{\|p_1(d_m) - s'_m\|_2}{\|p_2(d_m) - s'_m\|_2}$ and $\tau_{m,i,j}$ is defined in Eq.(S2), then the pair of operations $(S, M) : (s_m, (s_i, s_j)) \rightarrow ((s'_p, s'_q), s'_k)$ does not increase the tessellation energy $\mathcal{E}$, where s'_p, s'_q and s'_k are the mass centroids of $C'(p_1(d_m))$, $C'(p_2(d_m))$ and $C_{\mathcal{M}}(s_i) \cup C_{\mathcal{M}}(s_j)$, respectively.

Proof. Refer to Figure S1. We construct a local coordinate system by defining the z-axis along the line from $p_1(d_m)$ to $p_2(d_m)$. Then the hyperplane P is perpendicular to the z-axis. Let the intersection point of z-axis and P be the origin of the coordinate system. Since s'_m, s'_p and s'_q are the mass centroids of $C_{\mathcal{M}}(s_m)$, $C'(p_1(d_m))$ and $C'(p_2(d_m))$ respectively, s'_m must lie on the line segment connecting s'_p

and s'_q. Let the z-coordinate of a point x be $z(x)$. Then we have

$$\begin{aligned}
&\|s'_p - s'_m\|_2 \geq z(s'_m) - z(s'_p) = -z(s'_p) \\
&= -\frac{\int_{x\in C'(p_1(d_m))} z(x)dx}{\int_{x\in C'(p_1(d_m))} dx} = -\frac{\int_{x\in C'(p_1(d_m))} z(x)dx}{Vol(C'(p_1(d_m)))} \\
&\geq \frac{(-z(p_1(d_m))-d_{\max})m_{\min}}{Vol(C'(p_1(d_m)))} \geq \frac{(-z(p_1(d_m))-d_{\max})m_{\min}}{m_m} \\
&= \left(\frac{\lambda}{\lambda+1}d_m - d_{\max}\right)\frac{m_{\min}}{m_m} \geq \tau_{m,i,j}
\end{aligned}$$

Similarly, we have

$$\|s'_q - s'_m\|_2 \geq \left(\frac{1}{\lambda+1}d_m - d_{\max}\right)\frac{m_{\min}}{m_m} \geq \tau_{m,i,j}$$

That completes the proof. □

Theorem 3. *By selecting $(1+\varepsilon)K$ generators, $\varepsilon > 0$, Algorithm 2 is a bi-criteria $\left(1+\varepsilon, 8\left(1+\frac{1+\sqrt{5}}{2\varepsilon}\right)\right)$-approximation algorithm in expectation.*

Proof. Let $S_K^{opt} = \{s_i^{opt}\}_{i=1}^K$ and $\{C_i^{opt}\}_{i=1}^K$ be the (unknown) optimal generator set and tessellation on $\mathcal{M}$, which minimize the energy $\mathcal{E}$. Let $\mathcal{E}_{OPT} = \mathcal{E}(\{(s_i^{opt}, C_i^{opt})\}_{i=1}^K)$. A simple adaptation of the proof in [24] (Theorem 1 and Corollary 1) can show that for any $K' = (1+\varepsilon)K$ generators selected by Algorithm 1, the expected tessellation energy $\mathcal{E}$ satisfies

$$\frac{\mathbb{E}(\mathcal{E}(\{(s_i, C_i)\}_{i=1}^{K'}))}{\mathcal{E}_{OPT}} \leq 8\left(1+\frac{1+\sqrt{5}}{2\varepsilon}\right) \tag{S5}$$

By Theorem 2, the splitting and merging operation does not increase the energy $\mathcal{E}$. Furthermore, in each step of the Lloyd iteration process, computing the RVT and updating the positions of generators to their mass centroids do not increase the energy $\mathcal{E}$. Therefore for any tessellation $RCVT(S_{K'}, \mathcal{M})$ output from Algorithm 2, its expected tessellation energy $\mathcal{E}$ satisfies

$$\mathbb{E}(\mathcal{E}(RCVT(S_{K'}, \mathcal{M}))) \leq 8\left(1+\frac{1+\sqrt{5}}{2\varepsilon}\right)\mathcal{E}_{OPT}$$

That completes the proof. □

Theorem 4. *By selecting $(1+\varepsilon)K$ generators, $0 < \varepsilon < 1$, the time and space complexities of Algorithm 2 are $O(NK)$ and $O(N+K)$, respectively.*

Proof. In Algorithm 1, the initialization by Algorithm 1 (line 1) takes $O(NK)$ time and $O(N+K)$ space. In the iteration (lines 3-19),

- randomly picking three generators by Algorithm 4 (line 7) takes $O(N)$ time and space;
- both checking the splitting-merging feasibility (line 8) and applying the splitting-merging operation (line 10) take $O(1)$ time and space;
- by using a local search strategy in [15], the time and space complexities of computing/locally updating RVT in line 4 are both $O(N)$;
- moving all generators in RVT to corresponding mass centroids (lines 15-17) takes $O(N)$ time;
- storing and updating RVT takes $O(N)$ space.

As a summary, the time and space complexities of Algorithm 2 are $O(NK + iter_{max}(N + num_{rand}N))$ and $O(N+K)$, respectively. Since we used fixed values $iter_{max} = 20$ and $num_{random} = 20$, the time complexity reduces to $O(NK)$. That completes the proof. □

Theorem 5. *If $(1+\varepsilon)K$ generators, $0 < \varepsilon < 1$, are selected by Algorithm 2, Algorithm 5 is $(O(1), O(1))$-approximation.*

Proof. By Theorem 3, selecting $(1+\varepsilon)K$ generators, $0 < \varepsilon < 1$, makes Algorithm 2 an expected bi-criteria $(O(1), O(1))$-approximation algorithm. Theorem 3.1 in [1] states that if Algorithm 2 is an (a, b)-approximation, the two-level Algorithm 5 is an $(a, 2b + 4b(b+1))$-approximation. Accordingly, Algorithm 5 is $(O(1), O(1))$-approximation. That completes the proof. □

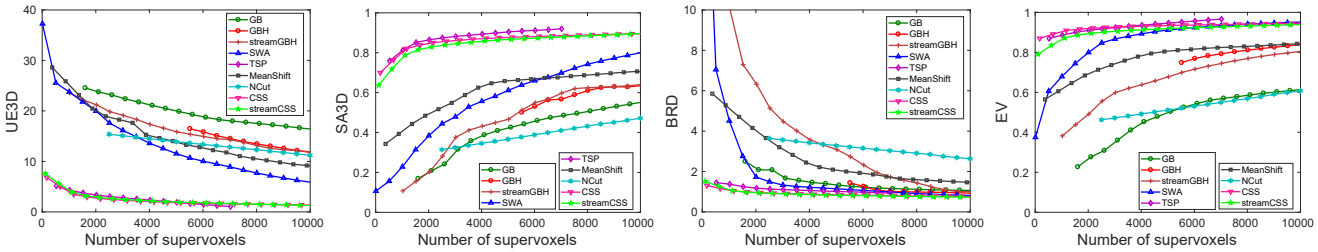
S3. More Experimental Results

We compare our methods (CSS and streamCSS) with seven representative methods selected in [25], including NCut [21, 8, 7], SWA [19, 20, 5], MeanShift [17], GB [6], GBH [10], streamGBH [26] and TSP [3]. Since CSS and streamCSS adopt a random initialization, we report the average results of 20 initializations. The performance evaluated on the BuffaloXiph dataset [4] is presented in the main paper. In this section, we present more experimental results on three additional video datasets, including SegTrack v2 [14], BVDS [23, 9] and CamVid [2].

Following the main paper, we use the commonly used quality metrics pertaining to supervoxels for evaluation [3, 13, 16, 25], including:

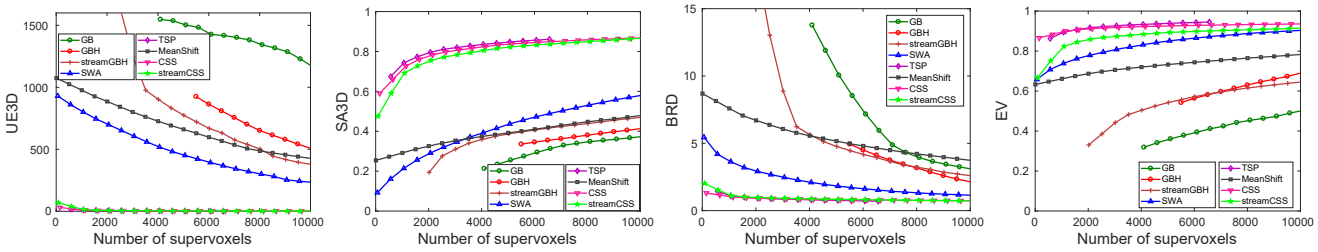
- *3D under-segmentation error (UE3D).* This metric measures the space-time leakage of supervoxels when overlapping groundtruth segments. Denote a ground-truth segmentation of a video as $\tilde{G} = \{\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_l\}$, and a supervoxel segmentation as $\tilde{S} = \{\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_r\}$. The UE3D metric is defined as

$$UE3D = \frac{1}{l}\sum_{\tilde{g}_i\in\tilde{G}} \frac{\sum_{\{\tilde{s}_j\in\tilde{S}: V(\tilde{s}_j\cap\tilde{g}_i)>0\}} V(\tilde{s}_j) - V(\tilde{g}_i)}{V(\tilde{g}_i)} \tag{S6}$$



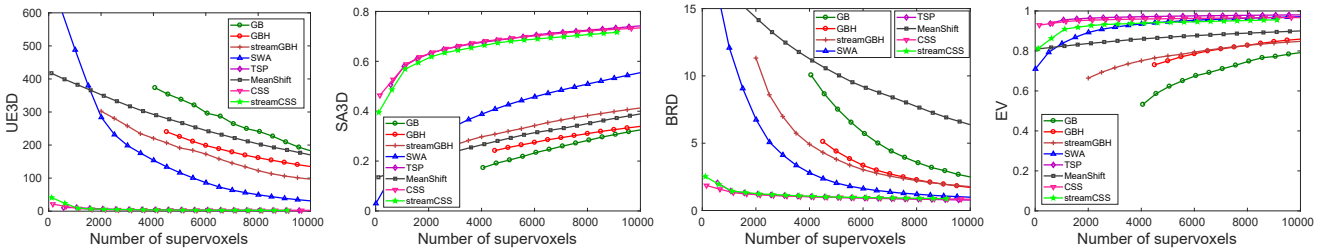
(i) 3D under-segmentation error (ii) 3D segmentation accuracy (iii) Boundary recall distance (iv) Explained variance

(a) SegTrack v2 dataset



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

(d) BVDS dataset



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

(c) CamVid dataset

Figure S2. Evaluation of different supervoxel results on the SegTrack v2, BVDS and CamVid datasets. CSS, streamCSS and TSP have the best performance on these measures, i.e., the highest SA3D, the smallest UE3D and BRD, and the largest EV. These results are consistent with the performance on the BuffaloXiph dataset (shown in the main paper).

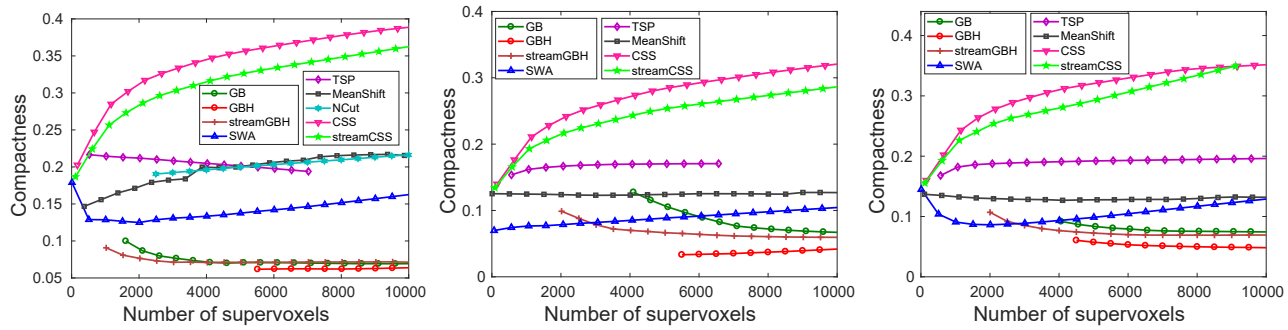
where $V(x)$ is the voxel number in a segment x. Equation (S6) takes the average score from all groundtruth segments $\tilde{G}$. A small under-segmentation error means that very few voxels are leaked from groundtruth segments.

- *3D segmentation accuracy (SA3D).* This metric measures the fraction of groundtruth segments that is correctly covered by supervoxels. If a supervoxel $\tilde{s}_a$ coincides with a groundtruth segment $\tilde{g}_b$ and the majority part of $\tilde{s}_a$ is inside $\tilde{g}_b$, then $\tilde{s}_a$ belongs to $\tilde{g}_b$ and their overlapped volume is counted into the correct covered volume of $\tilde{g}_b$. Denote a ground-truth segmentation of a video as $\tilde{G} = \{\tilde{g}_1, \tilde{g}_2, ..., \tilde{g}_l\}$, and a supervoxel segmentation as $\tilde{S} = \{\tilde{s}_1, \tilde{s}_2, ..., \tilde{s}_r\}$. The SA3D metric is defined as

$$SA3D = \frac{1}{l}\sum_{\tilde{g}_i \in \tilde{G}} \frac{\sum_{\{\tilde{s}_j \in \tilde{S}: V(\tilde{s}_j \cap \tilde{g}_i) \geq 0.5V(\tilde{s}_j)\}} V(\tilde{s}_j \cap \tilde{g}_i)}{V(\tilde{g}_i)} \quad \text{(S7)}$$

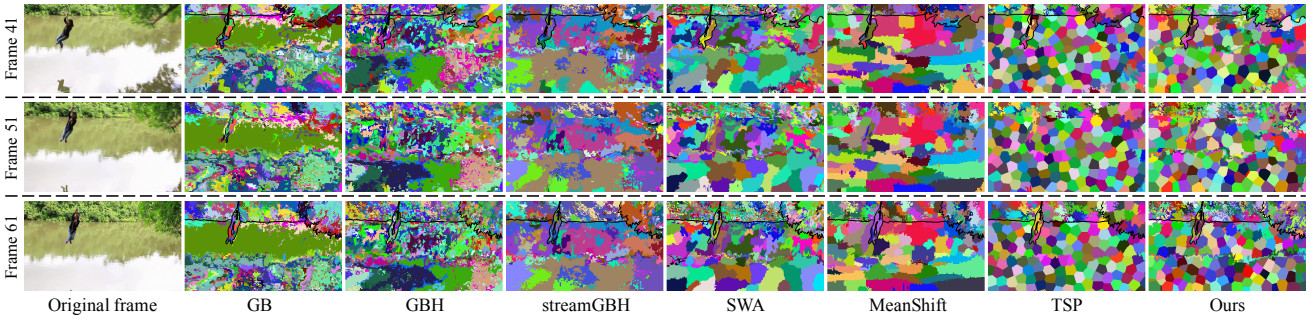
Equation (S7) takes the average score from all groundtruth segments $\tilde{G}$. The score range is in $[0, 1]$, where a larger value means a better over-segmentation result.

- *Boundary recall distance (BRD).* This metric measures the extent of groundtruth boundaries that are correctly retrieved by supervoxel boundaries. It is computed by averaging the distance from points on groundtruth boundaries to the nearest points on supervoxel boundaries in each frame. Denote the t-th frame's groundtruth segmentation as $\tilde{G}^t$, and the t-th

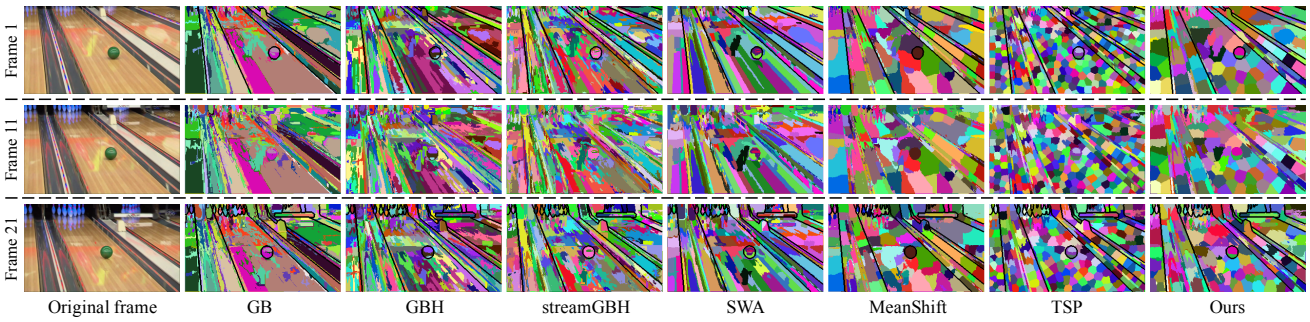


(a) SegTrack v2 dataset (b) BVDS dataset (c) CamVid dataset

Figure S3. Compactness measures of different supervoxel results on the SegTrack v2, BVDS and CamVid datasets. CSS and streamCSS and TSP have the best performance, i.e., the highest compactness values. These results are consistent with the performance on the BuffaloXiph dataset (shown in the main paper).



(a) Supervoxels clipped on frames #41, #51 and #61



(b) Supervoxels clipped on frames #1, #11 and #21

Figure S4. Superpixels (induced by clipping supervoxels on each image frame) obtained by GB [6], GBH [10], streamGBH [26], SWA [19, 20, 5], MeanShift [17], TSP [3] and our CSS method. Due to limited space, we only show the results of CSS here and the results of streamCSS are illustrated in the demo video. All the methods generate approximately 1,000 supervoxels. TSP and CSS produce regular supervoxels (and accordingly regular clipped superpixels), while other methods produce highly irregular supervoxels. Compared to TSP, CSS generates more supervoxels in content-rich areas and fewer supervoxels in content-sparse areas.

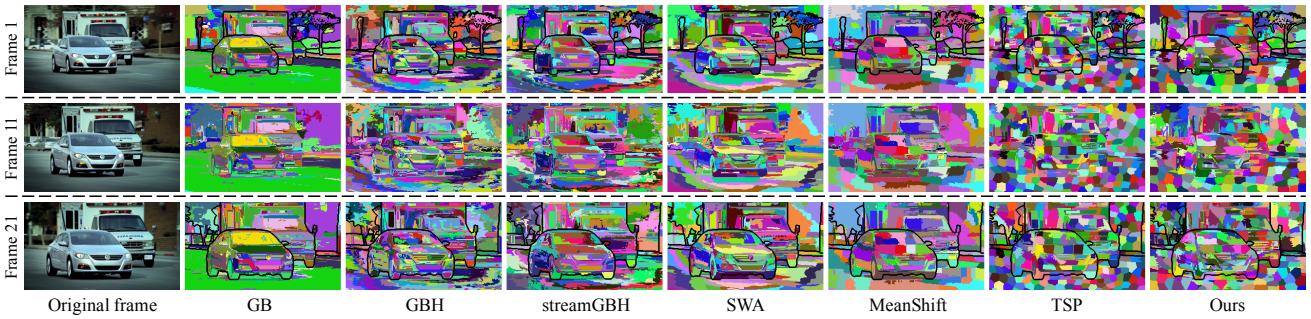
frame's supervoxel segmentation as $\tilde{S}^t$. The BRD metric is defined as

$$BRD = \frac{1}{\sum_t \left| B(\tilde{G}^t) \right|} \sum_t \sum_{p \in B(\tilde{G}^t)} \min_{q \in B(\tilde{S}^t)} d(p, q) \tag{S8}$$

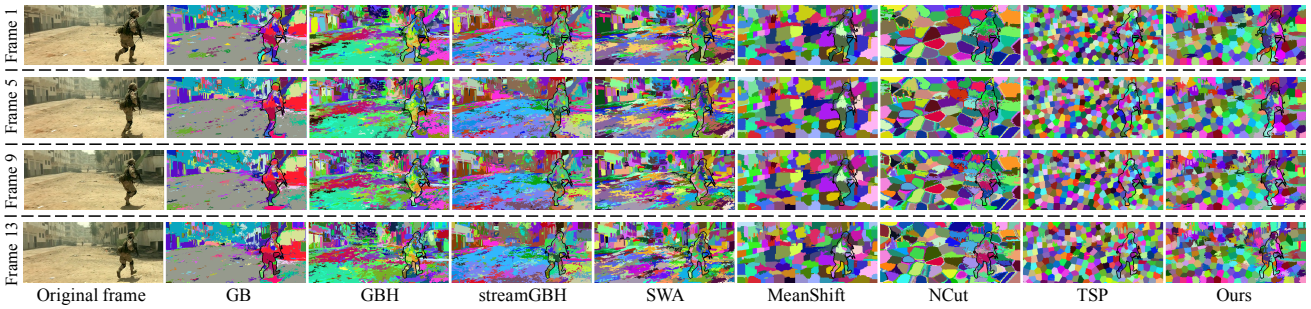
where $B(\cdot)$ returns the 2D boundaries in a frame, $d(\cdot, \cdot)$ measures Euclidean distance between two points, and $|\cdot|$ returns the number of pixels in a 2D boundary.

- *Explained variation (EV).* This metric measures the extent of supervoxels to represent voxels in the color domain. It is defined as

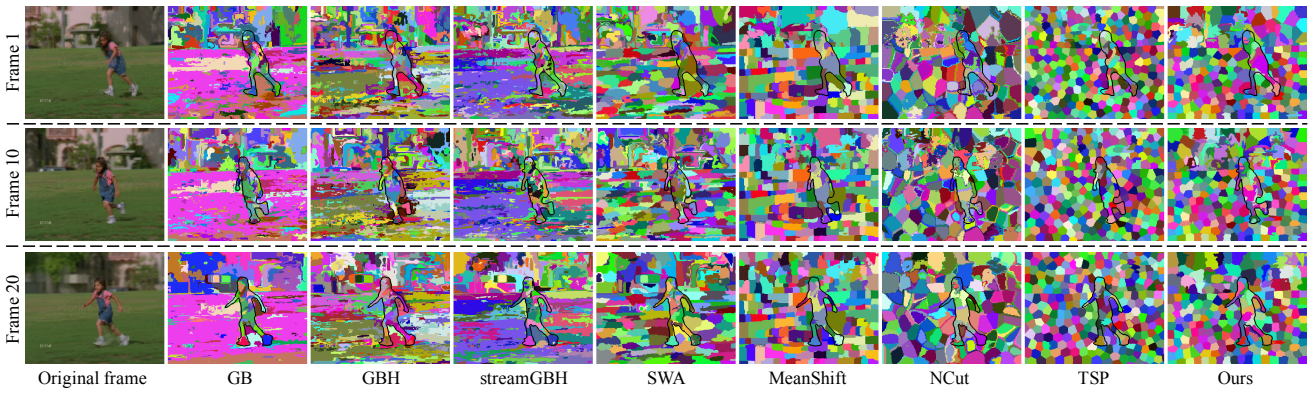
$$EV = \frac{\sum_{\tilde{s}_i \in \tilde{S}} (\mu(\tilde{s}_i) - \mu) \left| \tilde{s}_i \right|}{\sum_j (x_j - \mu)} \tag{S9}$$



(a) Supervoxels clipped on frames #1, #11 and #21



(b) Supervoxels clipped on frames #1, #5, #9 and #13



(c) Supervoxels clipped on frames #1, #10 and #20

Figure S5. Superpixels (induced by clipping supervoxels on each image frame) obtained by GB [6], GBH [10], streamGBH [26], SWA [19, 20, 5], MeanShift [17], TSP [3] and our CSS method. Due to limited space, we only show the results of CSS here and the results of streamCSS are illustrated in the demo video. All the methods generate approximately 1,000 supervoxels. TSP and CSS produce regular supervoxels (and accordingly regular clipped superpixels), while other methods produce highly irregular supervoxels. Compared to TSP, CSS generates more supervoxels in content-rich areas and fewer supervoxels in content-sparse areas.

where μ is the average color of all voxels in a video, $\mu(\tilde{s}_i)$ is the average color of the supervoxel $\tilde{s}_i$, and x_j is the color of the voxel j. The score range is in $[0, 1]$, where a larger value means a better representation.

The results summarized in Figure S2 show that the performances on three datasets (SegTrack v2, BVDS and CamVid) are consistent with that on BuffaloXiph dataset (summarized in the main paper): CSS, streamCSS and TSP have the best performance on these measures, i.e., the highest SA3D, the smallest UE3D and BRD, and the smallest EV.

We further apply a *compactness* metric to measure the regularity of supervoxels. Its formulation is given in Equations (19)-(20) in the main paper. The results are summarized in Figure S3, showing that CSS and streamCSS have the most regular shape (i.e., highest compactness values) and these results on the three datasets are consistent with the ones on the BuffaloXiph dataset summarized in the main paper.

More qualitative results are illustrated in Figures S4 and S5. By clipping supervoxels in each image frame, these results clearly show that CSS have regular shape, well capture object boundaries in a video and are content sensitive, i.e., supervoxels are small in content-dense regions and large in content-sparse regions. The content sensitive feature is due to the characteristic that regions of high appearance and motion variance have large volumes in $\mathcal{M}$. The qualitative results of streamCSS are shown in the demo video.

References

[1] N. Ailon, R. Jaiswal, and C. Monteleoni. Streaming k-means approximation. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, NIPS'09, pages 10–18, 2009. 3

[2] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *European conference on computer vision*, pages 44–57. Springer, 2008. 3

[3] J. Chang, D. Wei, and J. 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. 3, 5, 6

[4] A. Y. Chen and J. J. Corso. Propagating multi-class pixel labels throughout video frames. In *Proceedings of Western New York Image Processing Workshop*, 2010. 3

[5] J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. L. Yuille. Efficient multilevel brain tumor segmentation with integrated Bayesian model classification. *IEEE Trans. Med. Imaging*, 27(5):629–640, 2008. 3, 5, 6

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. 3, 5, 6

[7] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):214–225, 2004. 3

[8] C. C. Fowlkes, S. J. Belongie, and J. Malik. Efficient spatiotemporal grouping using the nystro"m method. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 231–238, 2001. 3

[9] F. Galasso, N. S. Nagaraja, T. J. Cárdenas, T. Brox, and B. 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. 3

[10] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. Efficient hierarchical graph-based video segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 2141–2148, 2010. 3, 5, 6

[11] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV*, pages 656–671. Springer International Publishing, 2014.

[12] A. Levinshtein, C. Sminchisescu, and S. J. Dickinson. Optimal image and video closure by superpixel grouping. *International Journal of Computer Vision*, 100(1):99–119, 2012.

[13] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(12):2290–2297, 2009. 3

[14] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2192–2199, 2013. 3

[15] Y.-J. Liu, C. Yu, M. Yu, and Y. He. Manifold slic: a fast method to compute content-sensitive superpixels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 651–659, 2016. 3

[16] A. P. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 3

[17] S. Paris and F. Durand. A topological approach to hierarchical segmentation using mean shift. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, 2007. 3, 5, 6

[18] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3282–3289. IEEE, 2012.

[19] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *2000 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1070–1077, 2000. 3, 5, 6

[20] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):810–813, 2006. 3, 5, 6

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. 3

[22] A. N. Stein, D. Hoiem, and M. Hebert. Learning to find object boundaries using motion cues. In *IEEE 11th International Conference on Computer Vision, ICCV 2007*, pages 1–8, 2007.

[23] P. Sundberg, T. Brox, M. Maire, P. Arbeláez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2233–2240. IEEE, 2011. 3

[24] D. Wei. A constant-factor bi-criteria approximation guarantee for k-means++. In *Annual Conference on Neural Information Processing Systems (NIPS) 2016*, pages 604–612, 2016. 3

[25] C. Xu and J. J. Corso. Libsvx: A supervoxel library and benchmark for early video processing. *Int. J. Comput. Vision*, 119(3):272–290, 2016. 3

[26] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, pages 626–639, 2012. 3, 5, 6