GPU-based Supervoxel Generation with a Novel Anisotropic Metric (Appendix)

Xiao Dong, Zhonggui Chen, Yong-Jin Liu, Junfeng Yao, and Xiaohu Guo

I. Anisotropic Supervoxel Algorithm

In this section, we further analyze the design of anisotropic distance metrics. The Section four of main paper shows the concept and the computation of anisotropic distance metric. Given the anisotropic matrix of a seed, we hope that the main direction of its iso-distance surface projected on image plane could be as consistent as possible with the moving direction of the seed. In Fig. 5 of the main paper, we show a toy model which satisfies the positive semidefinite (PSD) constraint and captures the object motion precisely.

Here, we discuss the situation where the matrix does not satisfy the PSD constraint. For example, we have a rolling soccer video and its corresponding optical flow fields. We want to see in detail where the anisotropic matrix of the seed may not meet the constraint. As shown in Fig. 1, the soccer is rolling with fast speed and the background is also moving due to the movement of the camera. We calculate the anisotropic matrix at every pixel on this frame (which is not necessary in the algorithm), and show those pixels that do not satisfy PSD constraint in black. We can see that very few pixels fail, and most of the failure cases are on the edge of image. It usually occurs at a pixel which has similar color and similar motion with its surroundings. We know that the seed will be updated to the center of its supervoxel after one iteration during the optimization, so there is almost not possible for the seed to be located to the edge of a frame. For the failure cases, we will discuss the remedy strategies later.

TABLE I: Non-PSD matrices

Total seed number	Percentage(%)	Angle α after correction(°)
500	4.24	17.82
1000	4.83	19.76
2000	5.18	22.47

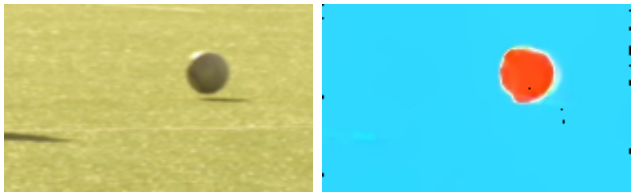
Next, we discuss the relationship between the anisotropic matrices and optical flow fields. Having a PSD matrix, we get the iso-distance surface, then we calculate the projection of its main direction on the image plane, denoted as D. We already have the optical flow vector F at the seed. Taking the results on SegTrackv2 dataset [13] as an example, we show the

Xiao Dong and Zhonggui Chen are with the School of Informatics, Xiamen University, Xiamen, China (e-mail: dongxiao0401@gmail.com; chenzhonggui@xmu.edu.cn). Junfeng Yao is with the School of Film, Xiamen University (e-mail: yao0010@xmu.edu.cn).

Yong-Jin Liu is with the Department of Computer Science and Technology, Tsinghua University, Beijing, China (e-mail:liuyongjin@tsinghua.edu.cn).

Xiaohu Guo is with the Department of Computer Science, University of Texas at Dallas, Dallas, USA (e-mail: xguo@utdallas.edu).

Corresponding authors: Zhonggui Chen, Junfeng Yao and Xiaohu Guo.



(a) Original frame (b) Optical flow

Fig. 1: One frame and the corresponding optical flow field of the rolling soccer video. We calculate the anisotropic matrix of every pixel, and show the pixels whose matrix do not satisfy the PSD constraint in black.

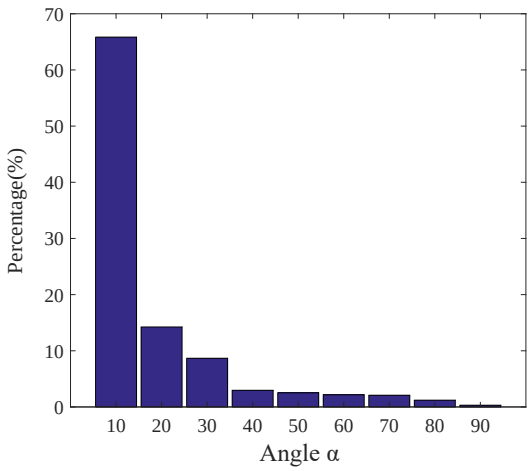
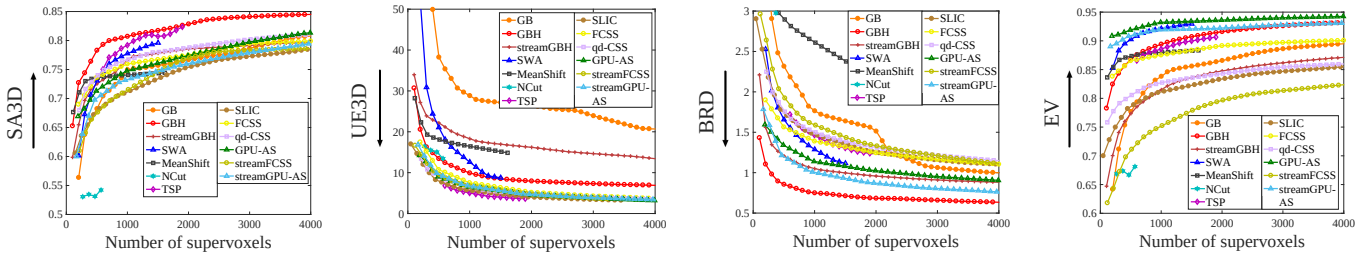


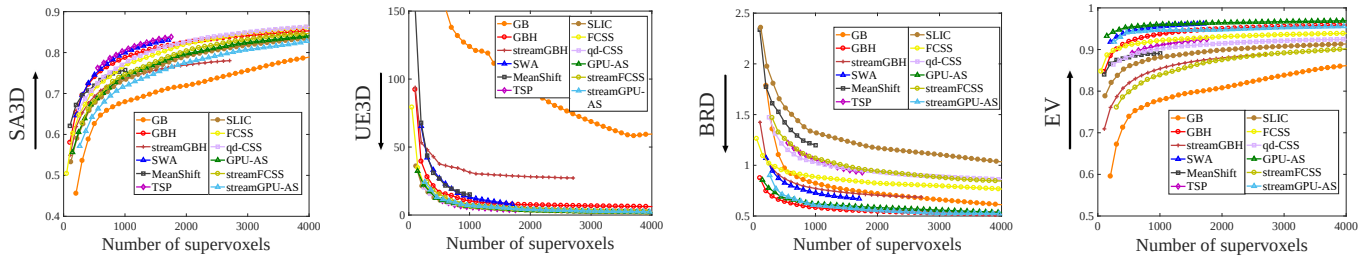
Fig. 2: Angle α between the projected direction D of iso-distance surface and the optical flow direction F at the seed.

average angle between seed matrix direction and the optical flow direction in Fig. 2. This experiment shows that the matrix can accurately capture the movement of the object when the angle α between D and F is within 30 degrees.

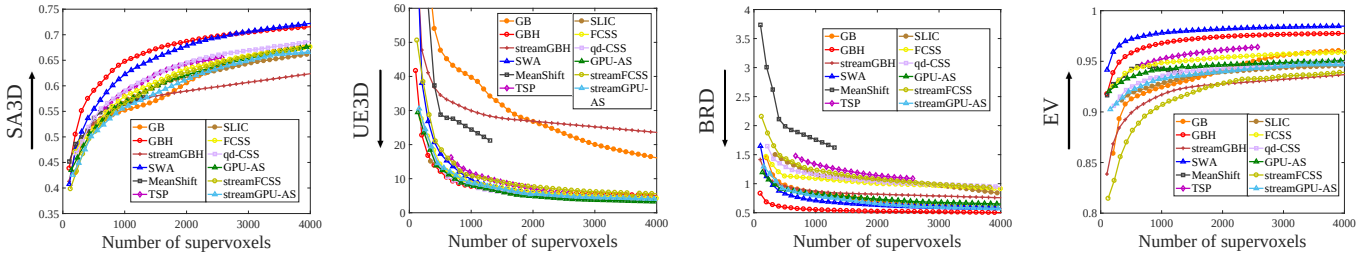
Since the algorithm cannot guarantee that all anisotropic matrices meet the PSD constraint, there is a small percentage of matrices that need to be post-processed. We adopt two strategies to guarantee the PSD nature: (1) replacing the current matrix with the average PSD matrices of neighboring voxels with the same moving direction; (2) calculating the nearest PSD matrix of the current matrix. We hope that there are only a few matrices that do not meet the PSD constraint, and for these matrices, they are able to capture the object motion after correction. Table I is the statistical results averaged on the SegTrackv2 dataset. The first column shows



(a) Evaluation on BuffaloXiph dataset



(b) Evaluation on BVDS dataset



(c) Evaluation on CamVid dataset

Fig. 3: Evaluation of the representative methods and our method on the BuffaloXiph [6], BVDS [11], [18] and CamVid [4] datasets. Our algorithm GPU-AS achieves a good balance on all performance such as SA3D, UE3D, BRD and EV.

the total number of seeds placed in a video, the second column shows the percentage of non-PSD seed matrices of the total seeds, and the last column shows the average angle α of these matrices after the correction strategies. From Table I and Fig. 2 we can see that there are about 5 percent of seeds that do not meet the PSD constraint. The algorithm then fixes this problem to get positive semi-definite matrix and the angle α after correction is within 30 degrees. This experiment shows that the PSD correction strategies are working well.

II. More Experiments

In the main paper, we compare the performance between our method with the representative algorithms on the SegTrackv2 dataset [13], including NCut [9], [10], [17], MeanShift [14], GB [8], GBH [12], streamGBH [20], SWA [7], [15], [16], TSP [5],SLIC [1], [2], qd-CSS [21] and FCSS [22]. For comprehensive understanding, we will introduce the definition of the quality metrics and also report the performance of our algorithm on other three datasets: BuffaloXiph [6], BVDS [11], [18] and CamVid [4]. In addition, we show a demo of LC comparison on MiddleburyFlow [3] dataset, and the supervoxel segmentation of some videos.

A. Evaluation Metrics

The common performance evaluation metrics include 3D segmentation accuracy (SA3D), 3D undersegmentation error (UE3D), boundary recall distance (BRD), explained variation (EV), and label consistency (LC). These metrics evaluate the quality of supervoxel segmentation from multiple aspects and help us further understand the performance of a supervoxel method.

Given a video containing T frames, with the oversegmentation of k supervoxels, we denote the set of annotated groundtruth segments as $G = \{G_i\}_{i=1}^{m}$, and the supervoxel partition as: $C = \{C_j\}_{j=1}^{k}$. These metrics are defined as follows:

- 3D segmentation accuracy (SA3D). It measures the average fraction of groundtruth segments that is correctly covered by the supervoxels:

$$SA3D(C,G) = \frac{1}{m} \times \sum_{i=1}^{m} \frac{\sum_{\{C_j | V(C_j \cap G_i) \geq 0.5 V(C_j)\}} V(C_j \cap G_i)}{V(G_i)}, \quad (1)$$

where $V(\cdot)$ denotes the total number of voxels that are contained in the supervoxel or the groundtruth segment. Note that only when the overlap volume between supervoxel C_j and groundtruth segment G_i is more than half of the volume of C_j, C_j is considered as successfully detected G_i. The metric takes the average score from all groundtruth segments. We note that it imposes greater

penalty when supervoxels leak on smaller groundtruth segments.

- 3D undersegmentation error (UE3D). This metric measures the average fraction of the voxels that exceed the boundary of the 3D groundtruth segments:

$$UE3D(C,G) = \frac{1}{m} \times \sum_{i=1}^{m} \frac{\sum_{\{C_j | V(C_j \cap G_i) \neq 0\}} V(C_j) - V(G_i)}{V(G_i)}. \quad (2)$$

Similar to SA3D, UE3D also takes the average score from all groundtruth segments G. The low value of UE3D means low space-time leakage of supervoxels when overlapping groundtruth segments. The UE3D and SA3D are two complementary metrics to evaluate the segmentation accuracy of supervoxels.

- Boundary recall distance (BRD). This metric calculates the average distance from points on boundaries of groundtruth to the nearest points on boundaries of supervoxels frame by frame. It directly measures how well the groundtruth boundaries are successfully retrieved by the supervoxels. This metric is defined as:

$$BRD(C,G) = \frac{1}{\sum_t |B(G^t)|} \sum_{t=1}^{T} \sum_{i \in B(G^t)} \min_{j \in B(C^t)} d(i,j), \quad (3)$$

where $B(\cdot)$ returns the 2D boundary pixels of the supervoxel or the groundtruth segment, $d(\cdot)$ measures the Euclidean distance between two pixels, and $|\cdot|$ denotes the number of pixels contained in the 2D boundary.

- Explained variation (EV). This metric is used to measure the ability of supervoxels representing the whole video by the mean color of each supervoxel when considering the supervoxels as a compression method. It is defined as:

$$EV(C) = \frac{\sum_{j=1}^{k} (u_j - u)^2 |C_j|}{\sum_i (x_i - u)^2}, \quad (4)$$

where x_i is the color of voxel i, u is the average color of all voxels in the video, u_j is the average color of the supervoxel C_j, and $|C_j|$ is the voxel number in C_j. The larger value of EV denotes a better representation of the video by supervoxels.

- Label consistency (LC). This metric is proposed to evaluate the capability of supervoxels to track the parts of objects based on the annotated groundtruth flows. Suppose the vectorized groundtruth forward flow filed is denoted as $F = \{F^{t-1 \to t} | t = 2, ..., T\}$, and the $F^{t-1 \to t}(C_j)$ is the flow operator that projects pixels contained in C_j at frame $t-1$ to pixels at frame t. This metric is defined as:

$$LC(C,F) = \frac{\sum_{t=2}^{T} \sum_{j=1}^{k} |C_j^t \cap F^{t-1 \to t}(C_j)|}{\sum_{t=2}^{T} \sum_{j=1}^{k} |F^{t-1 \to t}(C_j)|}, \quad (5)$$

where C_j^t is the superpixel slice of C_j at frame t. We evaluate this metric on the MiddleburyFlow dataset [3].

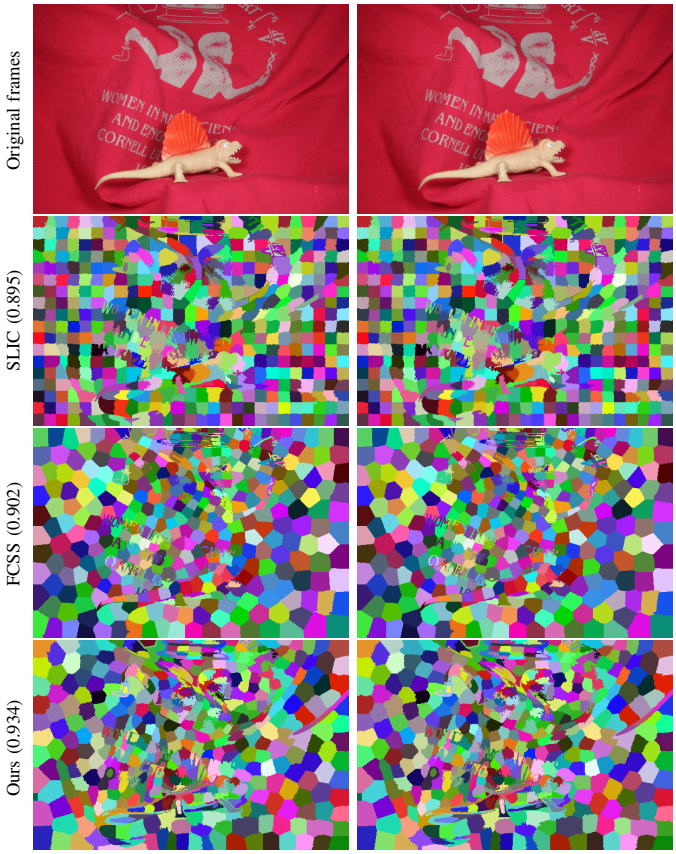


Fig. 4: Comparison with other seed-based methods on label consistency with LC values.

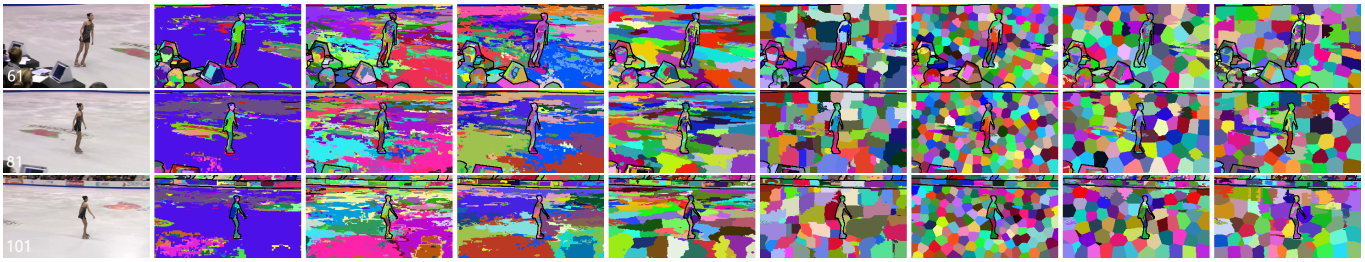
These basic metrics evaluate the quality of supervoxels from different aspects. SA3D, UE3D and BRD are important metrics for the supervoxel quality compared with the groundtruth segments. As discussed in Xu and Corso's survey [19], it is very difficult for a segmentation method to achieve best scores on all metrics except the perfect partition. In addition, the current methods are not efficient enough on the memory and time cost. Our method achieves a good balance between quality and efficiency of supervoxels, which is the main contribution of our work.

B. Experimental Results

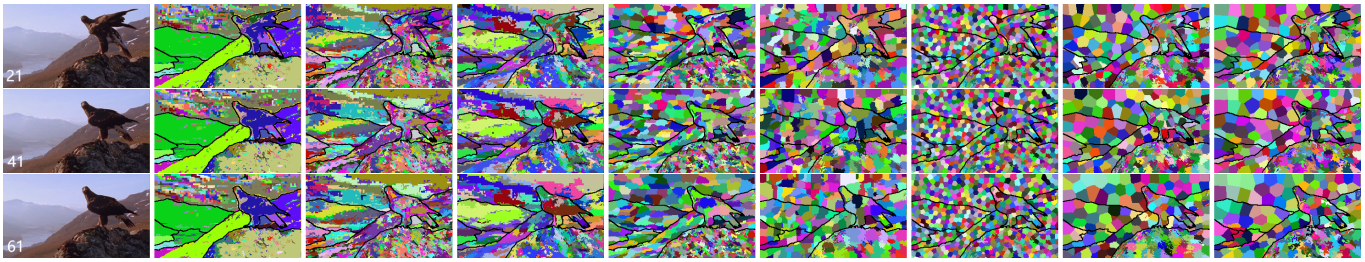
We show more experimental results to analyze the performance of different anisotropic supervoxels. First, we illustrate the performance of thirteen methods on the BuffaloXiph [6], BVDS [11], [18] and CamVid [4] datasets, we also give an example to show the comparison of label consistency on MiddleburyFlow [3] dataset. Finally, we show the results of supervoxel segmentation of some example videos. It can be seen that our algorithm can generate good quality segmentation.

Comparison on other datasets: In Fig. 3, we illustrate the performance of different methods on the BuffaloXiph [6], BVDS [11], [18] and CamVid [4] datasets. Due to the high memory and time cost, we did not run the NCut method on BVDS and CamVid datasets. In the evaluation, our method

(a) Supervoxels on frame #11, #16 and #21



(b) Supervoxels on frame #61, #81 and #101



(c) Supervoxels on frame #21, #41 and #61

Fig. 5: Supervoxels obtained by GB [8], GBH [12], streamGBH [20], SWA [7], [15], [16], MeanShift [14], NCut [9], TSP [5], FCSS [22] and ours. All methods generate approximately 1000 supervoxels. TSP, FCSS and ours produce regular supervoxels. Our algorithm generates results with much faster speed and achieves good balance on evaluation metrics.

achieves good performance on UE3D, BRD and EV. The performance of our method on SA3D metric under 2000 supervoxels is average. As we have discussed in the main paper, it is mainly because the JFA's parallel mechanism does not allow a seed to adjust its search range based on its surrounding content richness. Our seed initialization strategy only slightly adjusts the uniform distribution of seeds, resulting in low performance on segmentation accuracy with small number of supervoxels. As the number of supervoxels increases, the segmentation accuracy is greatly improved and catches up with the advanced methods. The running time of our method is not affected by the number of supervoxels. Meanwhile, our method has high performance on UE3D, EV and BRD metrics, which demonstrates from other aspects that our method generates video segmentation with good quality.

Comparison example on label consistency metric: Our method is a Lloyd-like optimization algorithm. Different with other seed-based methods, such as SLIC, qd-CSS and FCSS, our method designs the anisotropic distance metric for every seed based on the motion information around it. In terms of the label consistency (LC) performance, our algorithm outperforms other seed-based methods by a significant margin. As shown in Fig. 4, the LC value of our method is 0.934 with about 500 supervoxels. Our method is able to detect more details, and keeps the label more consistent with the aid of motion fields.

Supervoxel segmentation on videos: In order to better compare the results of different algorithms, we show the supervoxels of three videos compared with the state-of-the-art methods in Fig. 5. We can see that our algorithm is able to detect more details of objects than other methods.

REFERENCES

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels. Technical report, 2010.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.
[3] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31, 2011.
[4] 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*, pages 44–57. Springer, 2008.
[5] Jason Chang, Donglai Wei, and John W Fisher. A video representation using temporal superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2058, 2013.
[6] Albert YC Chen and Jason J Corso. Propagating multi-class pixel labels throughout video frames. In *2010 Western New York Image Processing Workshop*, pages 14–17. IEEE, 2010.
[7] Jason J Corso, Eitan Sharon, Shishir Dube, Suzie El-Saden, Usha Sinha, and Alan Yuille. Efficient multilevel brain tumor segmentation with integrated bayesian model classification. *IEEE transactions on medical imaging*, 27(5):629–640, 2008.
[8] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.
[9] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrom method. *IEEE transactions on pattern analysis and machine intelligence*, 26(2):214–225, 2004.
[10] Charless Fowlkes, Serge Belongie, and Jitendra Malik. Efficient spatiotemporal grouping using the nystrom method. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.
[11] Fabio Galasso, Naveen Shankar Nagaraja, Tatiana Jimenez Cardenas, Thomas Brox, and Bernt Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3527–3534, 2013.
[12] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *2010 ieee computer society conference on computer vision and pattern recognition*, pages 2141–2148. IEEE, 2010.
[13] 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*, pages 2192–2199, 2013.
[14] Sylvain Paris and Frédo Durand. A topological approach to hierarchical segmentation using mean shift. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
[15] Eitan Sharon, Achi Brandt, and Ronen Basri. Fast multiscale image segmentation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 70–77. IEEE, 2000.
[16] Eitan Sharon, Meirav Galun, Dahlia Sharon, Ronen Basri, and Achi Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):810, 2006.
[17] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000.
[18] Patrik Sundberg, Thomas Brox, Michael Maire, Pablo Arbeláez, and Jitendra Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR 2011*, pages 2233–2240. IEEE, 2011.
[19] 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.
[20] Chenliang Xu, Caiming Xiong, and Jason J Corso. Streaming hierarchical video segmentation. In *European Conference on Computer Vision*, pages 626–639. Springer, 2012.
[21] Zipeng Ye, Ran Yi, Minjing Yu, Yong-Jin Liu, and Ying He. Fast computation of content-sensitive superpixels and supervoxels using q-distances. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3770–3779, 2019.
[22] R Yi, Z Ye, W Zhao, M Yu, YK Lai, and YJ Liu. Feature-aware uniform tessellations on video manifold for content-sensitive supervoxels. *IEEE transactions on pattern analysis and machine intelligence*, 2020.