Evaluation of Super-Voxel Methods for Early Video Processing

Chenliang Xu and Jason J. Corso
Computer Science and Engineering, SUNY at Buffalo
{chenlian,jcorso}@buffalo.edu

Abstract

Supervoxel segmentation has strong potential to be incorporated into early video analysis as superpixel segmentation has in image analysis. However, there are many plausible supervoxel methods and little understanding as to when and where each is most appropriate. Indeed, we are not aware of a single comparative study on supervoxel segmentation. To that end, we study five supervoxel algorithms in the context of what we consider to be a good supervoxel: namely, spatiotemporal uniformity, object/region boundary detection, region compression and parsimony. For the evaluation we propose a comprehensive suite of 3D volumetric quality metrics to measure these desirable supervoxel characteristics. We use three benchmark video data sets with a variety of content-types and varying amounts of human annotations. Our findings have led us to conclusive evidence that the hierarchical graph-based and segmentation by weighted aggregation methods perform best and almost equally-well on nearly all the metrics and are the methods of choice given our proposed assumptions.

1. Introduction

Images have many pixels; videos have more. It has thus become standard practice to first preprocess images and videos into more tractable sets by either extraction of salient points [32] or oversegmentation into superpixels [31]. The preprocessing output data—salient points or superpixels—are more perceptually meaningful than raw pixels, which are merely a consequence of digital sampling [31]. However, the same practice does not entirely exist in video analysis. Although many methods do indeed initially extract salient points or dense trajectories, e.g., [20], few methods we are aware of rely on a supervoxel segmentation, which is the video analog to a superpixel segmentation. In fact, those papers that do preprocess video tend to rely on a per-frame superpixel segmentation, e.g., [21], or use a full-video segmentation, e.g., [15].

The basic position of this paper is that supervoxels have great potential in advancing video analysis methods, as superpixels have for image analysis. To that end, we perform a thorough comparative evaluation of five supervoxel methods; note that none of these methods had been proposed intrinsically as a supervoxel method, but each is either sufficiently general to serve as one or has been adapted to serve as one. The five methods we choose—segmentation by weighted aggregation (SWA) [6, 33, 34], graph-based (GB) [10], hierarchical graph-based (GBH) [15], mean shift [29], and Nyström normalized cuts [11, 12, 35]—broadly sample the methodology-space, and are intentionally selected to best analyze methods with differing qualities for supervoxel segmentation (see Figure 1 for examples). For example, both the SWA and the Nyström method use the normalized cut criterion as the underlying objective function, but SWA minimizes it hierarchically whereas Nyström does not. Similarly, there are two graph-based methods that optimize the same function, but one is subsequently hierarchical (GBH). We note a similar selection of segmentation methods have been used in the (2D) image boundary comparative study [1].

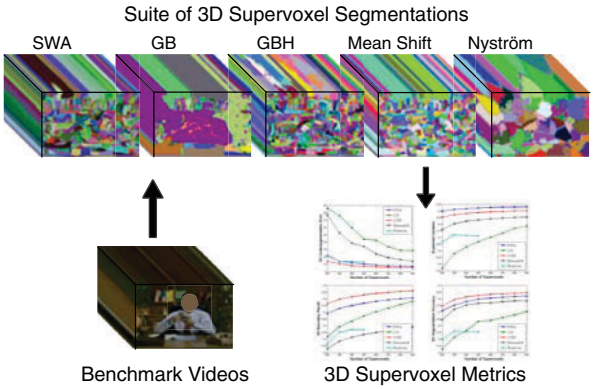

Figure 1. We comparatively evaluate five supervoxel methods on 3D volumetric metrics that measure various desirable characteristics of the supervoxels, e.g., boundary detection.

Our paper pits the five methods in an evaluation on a suite of 3D metrics designed to assess the methods on basic desiderata (Section 2.2), such as following object boundaries and spatiotemporal coherence. The specific metrics we use are 3D undersegmentation error, 3D segmentation

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

accuracy, 3D boundary recall, and explained variation. We use three complementary video data sets to facilitate the study (two of the three have hand-drawn object or region boundaries). Our evaluation yields conclusive evidence that two of the hierarchical methods (GBH and SWA) perform best and almost equally-well on nearly all the metrics (Nyström performs best on the 3D undersegmentation error) and are the methods of choice given our proposed metrics. Although GBH and SWA are quite distinct in formulation and may perform differently under other assumptions, we find a common feature among the two methods (and one that separates them from the other three) is the manner in which coarse level features are incorporated into the hierarchical computation. We thoroughly discuss comparative performance in Section 4 after presenting a theoretical background in Section 2 and a brief description of the methods in Section 3. Finally, to help facilitate the adoption of supervoxel methods in video, we make the developed source code—both the supervoxel methods and the benchmark metrics—and processed video results on the benchmark and major data sets available to the community.

2. Background

2.1. Superpixels

The term *superpixel* was coined by Ren and Malik [31] in their work on learning a binary classifier that can segment natural images. The main rationale behind superpixel oversegmentation is twofold: (1) pixels are not natural elements but merely a consequence of the discrete sampling of the digital images and (2) the number of pixels is very high making optimization over sophisticated models intractable. Ren and Malik [31] use the normalized cut algorithm [35] for extracting the superpixels, with contour and texture cues incorporated. Subsequently, many superpixel methods have been proposed [22, 23, 26, 40, 43] or adopted as such [5, 10, 41] and used for a variety of applications: e.g., human pose estimation [27], semantic pixel labeling [17, 37], 3D reconstruction from a single image [18] and multiple-hypothesis video segmentation [39] to name a few. Few superpixel methods have been developed to perform well on video frames, such as [8] who base the method on minimum cost paths but do not incorporate any temporal information.

2.2. What makes a good supervoxel method?

First, we define a *supervoxel*—the video analog to a superpixel. Concretely, given a 3D lattice Λ^3 (the voxels in the video), a supervoxel v is a subset of the lattice $v \subset \Lambda^3$ such that the union of all supervoxels comprises the lattice and they are pairwise disjoint: $\bigcup_i v_i = \Lambda^3 \wedge v_i \bigcap v_j = \varnothing \; \forall i, j$ pairs. Obviously, various image/video features may be computed on the supervoxels, such as color histograms and textons. In this initial definition, there is no mention of certain desiderata that one may expect, such as locality, coherence, and compactness. Rather than include them in mathematical terms, we next list terms of this sort as desirable characteristics of a *good* supervoxel method.

We define a good supervoxel method based jointly on criteria for good supervoxels, which follow closely from the criteria for good segments [31], and the actual cost of generating them (videos have an order of magnitude more pixels over which to compute). Later, in our experimental evaluation, we propose a suite of benchmark metrics designed to evaluate these criteria (Section 4.2).

Spatiotemporal Uniformity. The basic property of spatiotemporal uniformity, or *conservatism* [26], encourages compact and uniformly shaped supervoxels in space-time [22]. This property embodies many of the basic Gestalt principles—proximity, continuation, closure, and symmetry—and helps simplify computation in later stages [31]. Furthermore, Veksler et al. [40] show that for the case of superpixels, compact segments perform better than those varying in size on the higher level task of salient object segmentation. For temporal uniformity (called coherence in [15]), we expect a mid-range compactness to be most appropriate for supervoxels (bigger than, say, five frames and less than the whole video).

Spatiotemporal Boundaries and Preservation. The supervoxel boundaries should align with object/region boundaries when they are present and the supervoxel boundaries should be stable when they are not present; i.e., the set of supervoxel boundaries is a superset of object/region boundaries. Similarly, every supervoxel should overlap with only one object [23]. Furthermore, the supervoxel boundaries should encourage a high-degree of *explained variation* [26] in the resulting oversegmentation. If we consider the oversegmentation by supervoxels as a compression method in which each supervoxel region is represented by the mean color, we expect the distance between the compressed and original video to have been minimized.

Computation. The computation cost of the supervoxel method should reduce the overall computation time required for the entire application in which the supervoxels are being used.

Performance. The oversegmentation into supervoxels should not reduce the achievable performance of the application. Our evaluation will not directly evaluate this characteristic (because we study the more basic ones above).

Parsimony. The above properties should be maintained with as few supervoxels as possible [23].

3. Methods

We study five supervoxel methods—segmentation by weighted aggregation (SWA) [6, 33, 34], graph-based (GB) [10], hierarchical graph-based (GBH) [15], mean shift [29],

and Nyström normalized cuts [11, 12, 35]—that broadly sample the methodology-space among statistical and graph partitioning methods [1]. We have selected these five due to their respective traits and their inter-relationships: for example, Nyström and SWA both optimize the same normalized cut criterion. We describe the methods in some more detail below. We note that *many* other methods have been proposed in the computer vision literature for video segmentation, e.g., [2, 3, 14, 19, 24, 25, 39, 40, 41], but we do not cover them in any detail in this study. We also do not cover strictly temporal segmentation, e.g., [30].

Meanshift is a mode-seeking method, first proposed by Fukunaga and Hostetler [13]. Comaniciu and Meer [5] and Wang et al. [42] adapt the kernel to the local structure of the feature points, which is more computationally expensive but improves segmentation results. Original hierarchical mean shift in video [7, 28] improves the efficiency of (isotropic) mean-shift methods by using a streaming approach. The mean shift algorithm used in our paper is presented by Paris and Durand [29], who introduce Morse theory to interpret mean shift as a topological decomposition of the feature space into density modes. A hierarchical segmentation is created by using topological persistence. Their algorithm is more efficient than previous works especially on videos and large images.

Graph-based. Felzenszwalb and Huttenlocher [10] propose a graph-based algorithm for image segmentation; it is arguably the most popular superpixel segmentation method. Their algorithm runs in time nearly linear in the number of image pixels, which makes it suitable for extension to spatiotemporal segmentation. Initially, each pixel, as a node, is placed in its own region R, connected with 8 neighbors. Edge weights measure the dissimilarity between nodes (e.g. color differences). They define the internal difference of a region $Int(R)$ as the largest edge weight in the minimum spanning tree of R. Traversing the edges in a non-decreasing weight order, the regions R_i and R_j incident to the edge are merged if the current edge weight is less than the relaxed minimum internal difference of the two regions:

$$\min(Int(R_i) + \tau(R_i), Int(R_j) + \tau(R_j)) \ , \qquad (1)$$

where $\tau(R) = k/|R|$ is used to trigger the algorithm and gradually makes it converge. k is a scale parameter that reflects the preferred region size. The algorithm also has an option to enforce a minimum region size by iteratively merging low-cost edges until all regions contain the minimum size of pixels. We have adapted the algorithm for video segmentation by building a graph over the spatiotemporal volume, in which voxels are nodes connected with 26 neighbors in 3D space-time. One challenge in using this algorithm is the selection of an appropriate k for a given video, which the hierarchical extension (next) overcomes.

Hierarchical graph-based video segmentation algorithm is proposed by Grundmann et al. [15]. Their algorithm builds on an oversegmentation of the above spatiotemporal graph-based segmentation. It then iteratively constructs a region graph over the obtained segmentation, and forms a bottom-up hierarchical tree structure of the region (segmentation) graphs. Regions are described by local Lab histograms. At each step of the hierarchy, the edge weights are set to be the χ^2 distance between the Lab histograms of the connected two regions. They apply the same technique as above [10] to merge regions. Each time they scale the minimum region size as well as k by a constant factor s. Their algorithm not only preserves the important region borders generated by the oversegmention, but also allows a selection of the desired segmentation level, which is much better than directly manipulating k to control region size.

Nyström. Normalized Cuts [35] as a graph partitioning criterion has been widely used in image segmentation. A multiple eigenvector version of normalized cuts is presented in [11]. Given a pairwise affinity matrix W, they compute the eigenvectors V and eigenvalues Λ of the system

$$(D^{-1/2}WD^{-1/2})V = V\Lambda \ , \qquad (2)$$

where D is a diagonal matrix with entries $D_{ii} = \sum_j W_{ij}$. Each voxel is embedded in a low-dimensional Euclidean space according to the largest several eigenvectors. The k-means algorithm is then be used to do the final partitioning. To make it feasible to apply to the spatiotemporal video volume, Fowlkes et al. [12] use the Nyström approximation to solve the above eigenproblem. Their paper demonstrates segmentation on relatively low-resolution, short videos (e.g., $120 \times 120 \times 5$) and randomly samples points from the first, middle, and last frames.

SWA is an alternative approach to optimizing the normalized cut criterion [6, 33, 34] that computes a hierarchy of sequentially coarser segmentations. The method uses an algebraic multigrid solver to compute the hierarchy efficiently. It recursively coarsens the initial graph by selecting a subset of nodes such that each node on the fine level is strongly coupled to one on the coarse level. The algorithm is nearly linear in the number of input voxels, and produces a hierarchy of segmentations, which motivates its extension to a supervoxel method.

4. Experiments and Discussion

4.1. Experiment Setup and Data Sets

We compare the above five methods as fairly as possible. Each method is only allowed to use two feature cues: location (x, y, t) and color space. However, each method has its own tunable parameters; we have tuned these parameters strictly to achieve a certain desired number of supervoxels for each video in our study. In our experiments, GB is set

based on per-video and per-supervoxel-number; Meanshift is set based on per-video; Nyström, GBH, and SWA are set based on per-data-set. Following the above setup, we have computed the segmentation results for each video but with a distribution of supervoxel numbers varying from less than 200 to more than 900. To facilitate comparison of the methods for each data set, we use linear interpolation to estimate each methods' metric outputs densely. In order to run all the experiments in reasonable time and memory, the input video resolution is scaled to 240×160.

However, even with the above setup, the Nyström method is still not scalable as the number of supervoxels or the length of video increases. Recall that the Nyström method can be viewed as 3 steps: (1) building the affinity matrix; (2) Nyström computing and; (3) k-means clustering. Sampling too many points makes the Nyström method require too much memory, while sampling too few gives unstable and low performance. Meanwhile, the ordinary k-means clustering algorithm is sufficient for a video segmentation with few clusters, but a more efficient clustering method is expected in a supervoxel method. In our experiment, we generate results using Nyström method with less than 500 supervoxels.

Implementation. We have independently implemented all methods except for the Meanshift method, for which we use source code provided on the authors' website (`http://people.csail.mit.edu/sparis/#code`). The SWA implementation is based on our earlier 3D-SWA work in the medical imaging domain [6]. The complete supervoxel library, benchmarking code, and documentation is available for download at `http://www.cse.buffalo.edu/~jcorso/r/supervoxels/`. Various supervoxel results on major data sets in the community (including the three in this paper) are also available at this location to allow for easy adoption of the supervoxel results by the community.

Data sets. We use three video data sets for our experimental purposes, with varying characteristics. The first data set is *SegTrack* from Tsai et al. [38] and provides a set of human-labeled single-foreground objects with the videos stratified according to difficulty on color, motion and shape. SegTrack has six videos, an average of 41 frames-per-video (fpv), a minimum of 21 fpv and a maximum of 71 fpv.

The second data set is from Chen et al. [4] and is a subset of the well-known `xiph.org` videos that have been supplemented with a 24-class semantic pixel labeling set (the same classes from the MSRC object-segmentation data set [36]). The eight videos in this set are densely labeled with semantic pixels and have an average 85 fpv, minimum 69 fpv and maximum 86 fpv. This data set allows us to evaluate the supervoxel methods against human perception.

The third data set is from Grundman et al. [15] that comprises 15 videos of varying characteristics, but predominantly with a small number of *actors* in the shot. In order to run all the five supervoxel segmentation methods, we restrict the videos to a maximum of 100 frames (they have an average of 86 fpv and a minimum of 31 fpv). Unlike the other two data sets, this data set does not have a ground-truth segmentation, which inspires us to further explore the human independent metrics.

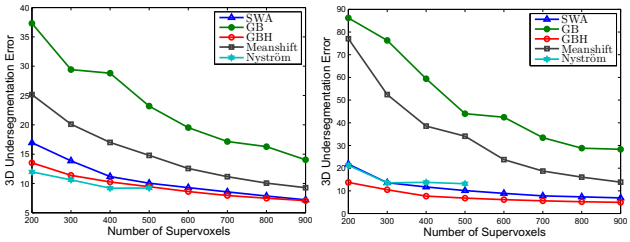

Figure 2. 3D undersegmentation error vs. number of supervoxels. Left: the results on SegTrack data set. Right: the results on Chen's data set.

4.2. Metrics and Baselines

Rather than evaluating the supervoxel methods on a particular application, as [16] does for superpixels and image segmentation, we directly consider all of the base traits described in Section 2.2 at a fundamental level. We believe these more basic evaluations have a greater potential to inform the community than those potential evaluations on a particular application.

We note that some quantitative superpixel evaluation metrics have been recently used in [22, 23, 26, 40, 43], but all of them are frame-based 2D image metrics, which are not suitable in our supervoxel measures. We extend those most appropriate to validate our desiderata in 3D space-time. Given a ground-truth segmentation into segments $g_1, g_2, ..., g_m$, and a video segmentation into supervoxels $s_1, s_2, ..., s_n$, here we propose a suite of the volumetric video-based 3D metrics.

4.2.1 3D Undersegmentation Error (3D UE)

This metric measures what fraction of voxels exceed the volume boundary of the ground-truth segment when mapping the supervoxels onto it.

$$UE(g_i) = \frac{\left[\sum_{\{s_j | s_j \cap g_i \neq \emptyset\}} \text{Vol}(s_j)\right] - \text{Vol}(g_i)}{\text{Vol}(g_i)} \quad (3)$$

gives the 3D UE for a single ground-truth segment g_i in the video, where Vol is the segment volume. We take the average across all ground-truth segments in the video, giving equal weight to all ground-truth segments.

Figure 2 shows the dependency of 3D UE on the number of supervoxels. GBH, SWA and Nyström have much

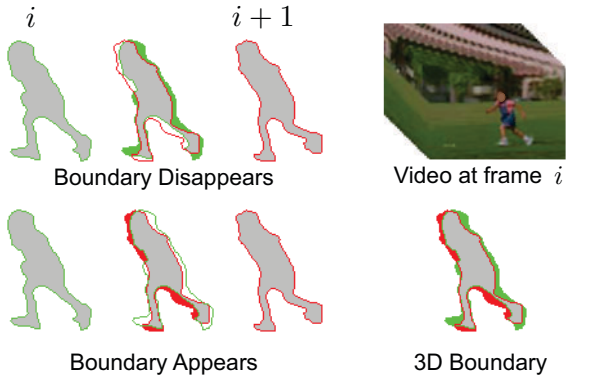

Figure 3. A visual explanation of the distinct nature of 3D boundaries in video (please view in color). We overlap each frame to compose a volumetric video, the green colored area which is a part of girl in frame i should not be counted as a part of girl in frame $i+1$, similarly the red area which is a part of girl in frame $i+1$ should not be counted as a part of girl in frame i. The lower right graph shows the 3D boundary along the time axis (imagine you are looking through the paper).

better performance than Meanshift and GB with fewer supervoxels. When the number of supervoxels is more than 500, GBH and SWA almost have the same competitive performance on both data sets. As the number of supervoxels increases, Meanshift quickly converges, while GB is much slower. Figure 8 shows SWA, GBH, and Nyström generate more spatiotemporally uniform supervoxels than the other two.

4.2.2 3D Boundary Recall (3D BR)

In the ideal case, one can imagine the 3D boundary as the shape boundary of a 3D object, composed by surfaces. However, given a video, the 3D boundary face is not that smooth, since videos are actually discrete and voxels are rectangular cubes. Therefore, the 3D boundary should not only capture the 2D within-frame boundary but also the between-frame boundary. Figure 3 shows a between-frame boundary concept by using the ground-truth segment as an example. It follows the concept that we have proposed in Section 2.2. The 3D boundary recall metric measures the spatiotemporal boundary detection: for each segment in the ground-truth and supervoxel segmentations, we extract the within-frame and between-frame boundaries and measure recall using the standard formula (not included for space).

Figure 4 shows the dependency of 3D BR on the number of supervoxels. GBH and SWA again perform best in 3D BR. GB quickly converges toward GBH as the number of supervoxels increases, since it serves as a preprocessing step of GBH.

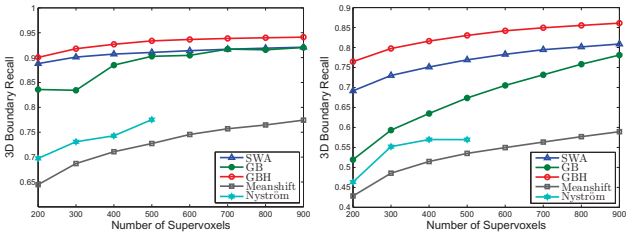

Figure 4. 3D boundary recall vs. number of supervoxels. Left: the results on SegTrack data set. Right: the results on Chen's data set.

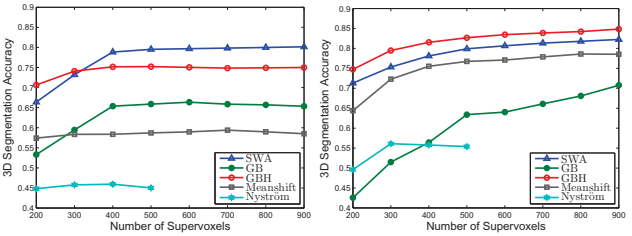

Figure 5. 3D segmentation accuracy vs. number of supervoxels. Left: the results on SegTrack data set. Right: the results on Chen's data set.

4.2.3 3D Segmentation Accuracy (3D ACCU)

This metric measures what fraction of a ground-truth segment is correctly classified by the supervoxels: each supervoxel should overlap with only one object/segment as a desired property in Section 2.2. For the ground-truth segment g_i, we assign a binary label to each supervoxel s_j according to the majority part of s_j that resides inside or outside of g_i. Then we have a set of correctly labeled supervoxels $\overline{s_1}, \overline{s_2}, ..., \overline{s_k}$. We define the 3D SA for g_i with the fraction

$$ACCU(g_i) = \frac{\sum_{j=1}^{k} \text{Vol}(\overline{s_j} \cap g_i)}{\text{Vol}(g_i)} \quad . \tag{4}$$

To evaluate the overall segmentation quality, we also take the average of the above fraction across all ground-truth segments in the video.

Figure 5 shows the dependency of 3D ACCU on the number of supervoxels. GBH and SWA perform again better than the other three methods on both data sets. Meanshift performs better on Chen's data set than on SegTrack because Chen's data has full-scene segmentation that includes large relatively homogeneous background segments (like sky) whereas the SegTrack data only segments out a single foreground object.

4.2.4 Explained Variation

Explained Variation is proposed in [26] as a human-independent metric—in other words, it is not susceptible to

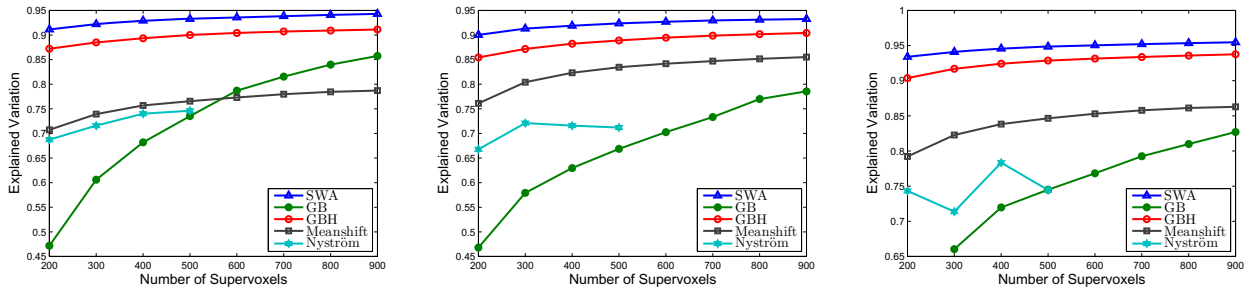

Figure 6. Explained variation metric (R^2) vs. number of supervoxels on SegTrack, Chen and Gatech data sets, respectively.

variation in annotator perception that would result in differences in the human annotations, unlike the other proposed metrics. It considers the supervoxels as a compression method of a video described in Section 2.2.

$$R^2 = \frac{\sum_i (\mu_i - \mu)^2}{\sum_i (x_i - \mu)^2} \tag{5}$$

sums over i voxels where x_i is the actual voxel value, μ is the global voxel mean and μ_i is the mean value of the voxels assigned to the supervoxel that contains x_i. Erdum et al. [9] observe a correlation between explained variation and the human-dependent metrics for a specific object tracking task; our results, which we discuss next, show a similar trend, substantiating our prior use of the metrics based on the human annotations.

Figure 6 shows the dependency of explained variation on the number of supervoxels. SWA and GBH perform better and more stably than the others, even with a relatively low number of supervoxels. The performance of GB increases dramatically and converges quickly as the number of supervoxels increases. The performance of Nyström on these three data sets further demonstrates our claim in Section 4.1 that the method is sensitive to the actual point sampling density.

4.2.5 Computational Cost

Our operating workstation is a Dual Quad-core Intel Xeon CPU E5620 @ 2.4 GHz, 16Gb RAM running Linux. Figure 7 presents the average running time of each method over all three data sets. Over all five methods, GB is the most efficient in time and memory usage. Its running time for one video does not significantly change as the number of supervoxels increases. Meanshift is the second most efficient method. Interestingly, neither Meanshift nor GB performs best in any of the quality measures—there is an obvious trade-off between the computational cost of the methods and the quality of their output (in terms of our metrics). The two slowest methods, GBH and SWA, consistently perform best in our quality metrics. Despite the faster running time, the memory consumption of SWA is nearly five times that of GBH (yet it still fits in 16Gb of memory).

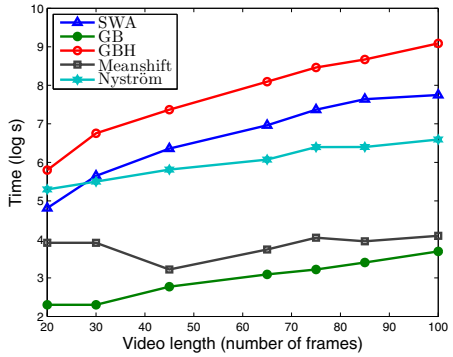

Figure 7. Comparison of the running time among all five methods with parameters set to output about 500 supervoxels, averaged over all three data sets. The comparison is not exactly apples to apples: the SWA and GBH methods output all layers of the hierarchy and the Nyström method is in Matlab whereas the others are in C++. Nevertheless, the trend expounds the trade-off between computational expense and quality of supervoxel output (i.e., the GBH and SWA methods consistently perform best in our metrics and have the longest running time).

5. Discussion and Conclusion

We have presented a thorough evaluation of five supervoxel methods on four 3D volumetric performance metrics designed to evaluate supervoxel desiderata. Samples from the data sets segmented under all five methods are shown in Figure 8. We have selected videos of different qualities to show in this figure. These visual results convey the overall findings we have observed in the quantitative experiments. Namely, two of the hierarchical methods (GBH and SWA) perform better than the others at preserving object boundaries. The Nyström supervoxels are the most compact and regular in shape and the SWA supervoxels observe a sim-

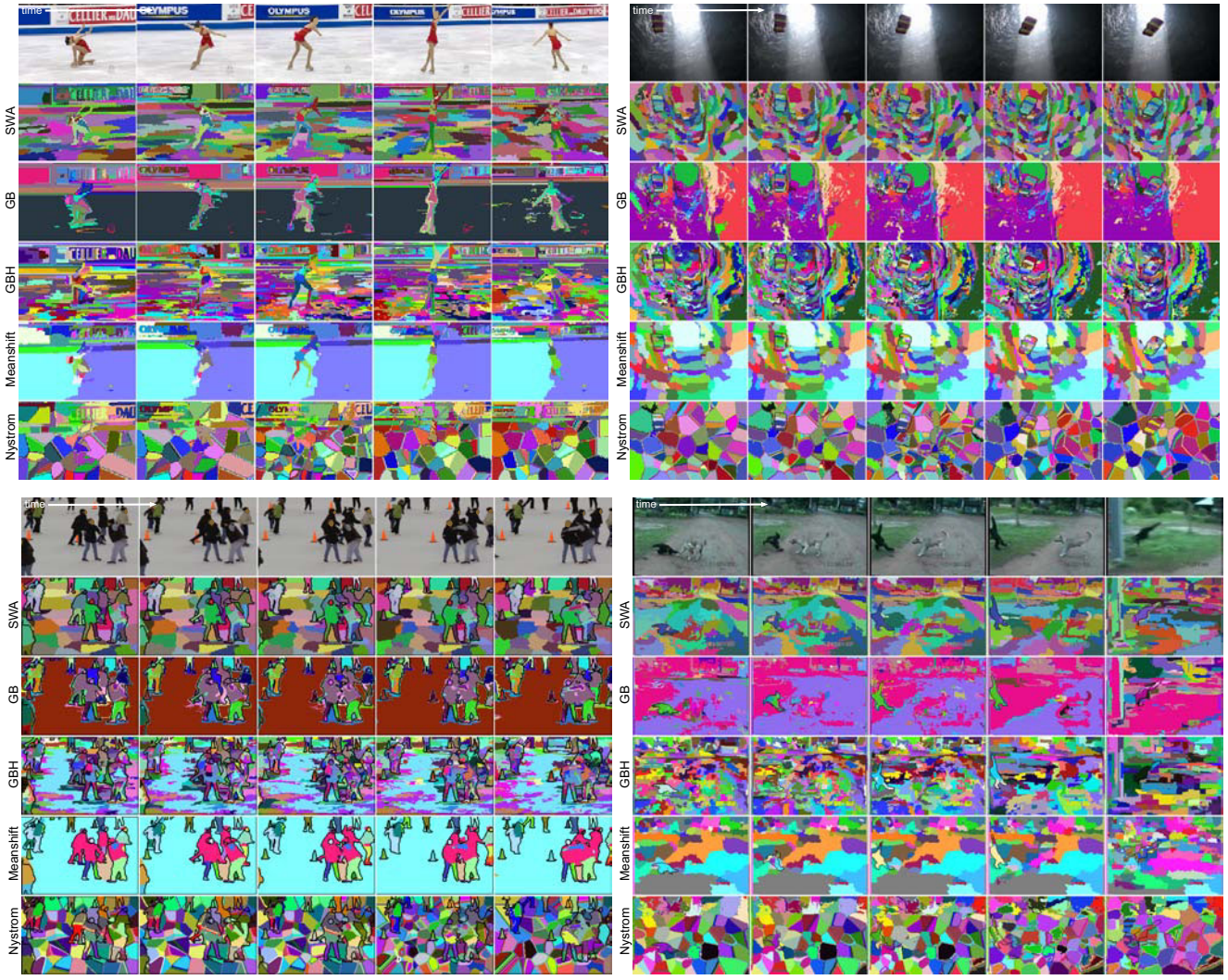

Figure 8. Visual comparative results of the five methods on four videos (with roughly 500 supervoxels); top-left is from Gatech, bottom-left is from Chen, and right-two are from SegTrack. A black line is drawn to represent human-drawn boundaries in those videos that have been annotated; note for the SegTrack data set only one object has a boundary and for the Chen data set many regions have boundaries. Each supervoxel is rendered with its distinct color and these are maintained over time. Faces have been redacted for presentation (the original videos were processed when computing the segmentations). We recommend viewing these images zoomed on an electronic display.

ilar compactness but seem to adapt to object boundaries better (recall that SWA and Nyström are both normalized cut solvers). It seems evident that the main distinction behind the better performance of GBH and SWA is the way in which they both compute the hierarchical segmentation. Although the details differ, the common feature among the two methods is that during the hierarchical computation, coarse-level aggregate features replace or modulate fine-level individual features. None of the other three approaches use any coarse-level features.

In this paper, we have explicitly studied the general supervoxel desiderata and avoided any direct application for scope. The obvious question to ask is how well will the findings on these general metrics translate to application specific ones, such as tracking and activity recognition. A related additional point that we have not studied is the application-specific trade-off between quality of the output and the run-time of the method used to generate it. For example, in real-time streaming applications, it may be that GB or Meanshift strikes the appropriate balance. We plan to study these important questions in future work.

Acknowledgements This work was partially supported by the National Science Foundation CAREER grant (IIS-0845282), the Army Research Office (W911NF-11-1-0090), the DARPA Mind's Eye program (W911NF-10-2-0062), and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20069

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/NBC, DARPA, ARO, NSF or the U.S. Government.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011.

[2] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *ICCV*, 2009.

[3] I. Budvytis, V. Badrinarayanan, and R. Cipolla. Semi-supervised video segmentation using tree structured graphical models. In *CVPR*, 2011.

[4] A. Y. C. Chen and J. J. Corso. Propagating multi-class pixel labels throughout video frames. In *Proc. of Western NY Image Proc. Workshop*, 2010.

[5] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *TPAMI*, 24(5):603–619, 2002.

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

[7] D. DeMenthon and R. Megret. Spatio-Temporal Segmentation of Video by Hierarchical Mean Shift Analysis. In *Statistical Methods in Video Proc. Workshop, Image and Vision Computer*, 2002.

[8] F. Drucker and J. MacCormick. Fast superpixels for video analysis. In *IEEE WMVC*, 2009.

[9] C. Erdem, B. Sankur, and A. Tekalp. Performance measures for video object segmentation and tracking. *TIP*, 13(7):937 –951, 2004.

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *IJCV*, 59(2):167–181, 2004.

[11] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *TPAMI*, 26:2004, 2004.

[12] C. Fowlkes, S. Belongie, and J. Malik. Efficient spatiotemporal grouping using the Nyström method. In *CVPR*, pages 231–238, 2001.

[13] K. Fukunaga and L. D. Hostetler. The estimation of a density function with applications in pattern recognition. *TIT*, 21:32–40, 1975.

[14] H. Greenspan, J. Goldberger, and A. Mayer. Probabilistic Space-Time Video Modeling via Piecewise GMM. *TPAMI*, 26(3):384–397, 2004.

[15] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.

[16] A. Hanbury. How do superpixels affect image segmentation? In *Proc. of Iberoamerican Cong. on Pattern Recognition*, 2008.

[17] X. He, R. S. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV*, 2006.

[18] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM TOG*, 24(3):577–584, 2005.

[19] S. Khan and M. Shah. Object Based Segmentation of Video Using Color, Motion, and Spatial Information. In *CVPR*, volume 2, pages 746–751, 2001.

[20] I. Laptev. On space-time interest points. *IJCV*, 2005.

[21] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011.

[22] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *TPAMI*, 31(12):2290–2297, 2009.

[23] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *CVPR*, 2011.

[24] S. Liu, G. Dong, C. H. Yan, and S. H. Ong. Video segmentation: Propagation, validation and aggregation of a preceding graph. In *CVPR*, 2008.

[25] R. Megret and D. DeMenthon. A Survey of Spatio-Temporal Grouping Techniques. Technical report, UMD, 2002.

[26] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. In *CVPR*, 2008.

[27] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, volume 2, pages 326–333, 2004.

[28] S. Paris. Edge-preserving smoothing and mean-shift segmentation of video streams. In *ECCV*, 2008.

[29] S. Paris and F. Durand. A topological approach to hierarchical segmentation using mean shift. In *CVPR*, 2007.

[30] N. V. Patel and I. K. Sethi. Video shot detection and characterization for video databases. *PR*, 30(4):583 – 592, 1997.

[31] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, volume 1, pages 10–17, 2003.

[32] C. Schmid and R. Mohr. Local Grayvalue Invariants for Image Retrieval. *TPAMI*, 19(5):530–535, 1997.

[33] E. Sharon, A. Brandt, and R. Basri. Fast Multiscale Image Segmentation. In *CVPR*, volume I, pages 70–77, 2000.

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

[35] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *TPAMI*, 22(8):888–905, 2000.

[36] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(2):2–23, 2009.

[37] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

[38] D. Tsai, M. Flagg, and J. M. Rehg. Motion coherent tracking with multi-label mrf optimization. In *BMVC*, 2010.

[39] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, 2010.

[40] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and supervoxels in an energy optimization framework. In *ECCV*, 2010.

[41] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *TPAMI*, 13(6):583–598, 1991.

[42] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and Video Segmentation by Anisotropic Kernel Mean Shift. In *ECCV*, volume 2, pages 238–249, 2004.

[43] G. Zeng, P. Wang, J. Wang, R. Gan, and H. Zha. Structure-sensitive superpixels via geodesic distance. In *ICCV*, 2011.