Flattening Supervoxel Hierarchies by the Uniform Entropy Slice

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

Abstract

Supervoxel hierarchies provide a rich multiscale decomposition of a given video suitable for subsequent processing in video analysis. The hierarchies are typically computed by an unsupervised process that is susceptible to under-segmentation at coarse levels and over-segmentation at fine levels, which make it a challenge to adopt the hierarchies for later use. In this paper, we propose the first method to overcome this limitation and flatten the hierarchy into a single segmentation. Our method, called the uniform entropy slice*, seeks a selection of supervoxels that balances the relative level of information in the selected supervoxels based on some post hoc feature criterion such as objectness. For example, with this criterion, in regions nearby objects, our method prefers finer supervoxels to capture the local details, but in regions away from any objects we prefer coarser supervoxels. We formulate the uniform entropy slice as a binary quadratic program and implement four different feature criteria, both unsupervised and supervised, to drive the flattening. Although we apply it only to supervoxel hierarchies in this paper, our method is generally applicable to segmentation tree hierarchies. Our experiments demonstrate both strong qualitative performance and superior quantitative performance to state of the art baselines on benchmark internet videos.*

1. Introduction

In recent years, segmentation has emerged as a plausible first step in early processing of unconstrained videos, without needing to make an assumption of a static background as earlier methods have [10]. For example, the key segments work [19] proposes a method to take frame-by-frame superpixel segmentations and automatically segment the dominant moving actor in the video with category independence. Recent works in video segmentation generate spatiotemporally coherent segmentations relatively efficiently by methods like point trajectory grouping [6, 15, 21], superpixel tracking [4, 29, 32], probabilistic methods [1, 7, 18], supervoxels by minimum spanning trees [16, 33, 34], or compositing multiple constituent segmentations [22, 26].

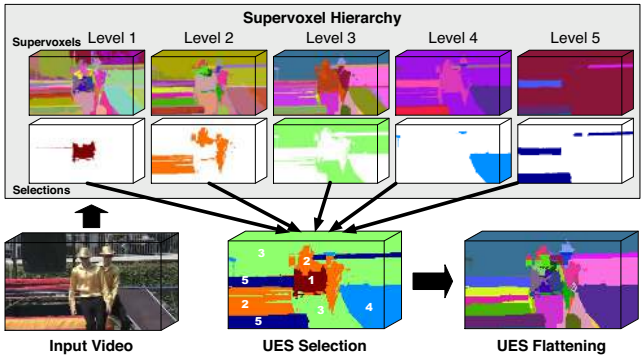


Figure 1. The uniform entropy slice (UES) selects supervoxels from multiple hierarchical levels in a principled way to balance the amount of information contributed by each selected supervoxel, according to some feature criterion (motion in this figure). UES Selection shows what levels are used and UES Flattening shows the final supervoxel output. Here, UES avoids over-segmentation of the background (present in Levels 1 and 2) and under-segmentation of the dancers (present in Levels 4 and 5); even just Level 3 joins the dancers' face with their shirts.

These advances in video segmentation have also been thoroughly evaluated. Leveraging contributions in image segmentation evaluation [3] and criteria for good video segmentation [11], we have proposed the LIBSVX benchmark [33], which implements a suite of six supervoxel algorithms and tests them in a set of evaluation metrics with three video datasets. Ultimately, it was determined that the two hierarchical agglomerative methods, Grundmann et al. [16] graph-based hierarchical method and Sharon et al. [27] segmentation by weighted aggregation, perform best overall due to the way in which multiscale region similarity was reevaluated as the hierarchy was generated.

Despite these advances, hierarchical video segmentation has not yet been actively adopted. The hierarchies contain a rich multiscale decomposition of the video, but we are unaware of a principled approach to make use of this rich information by *flattening* it to a single non-trivial segmentation. Trivial flattenings, by arbitrarily taking a level, would carry over intrinsic limitations of the bottom-up supervoxel methods, as Fig. 1 illustrates. For example, taking a low level would mean very many supervoxels (over-

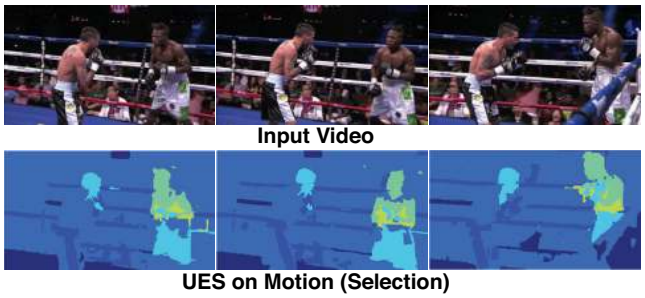


Figure 2. Example of supervoxel hierarchy selection by UES with a motion criterion on video *boxers*. The motion criterion drives the algorithm to select finer levels of the hierarchy (brighter regions on bottom row) on the dominant moving objects. The boxer on the right and the head of the boxer on the left are being selected from finer levels in the supervoxel hierarchy while the background segments are from coarser levels in the hierarchy. The boxer on the right (in an offensive posture) is moving much more than the boxer on the left.

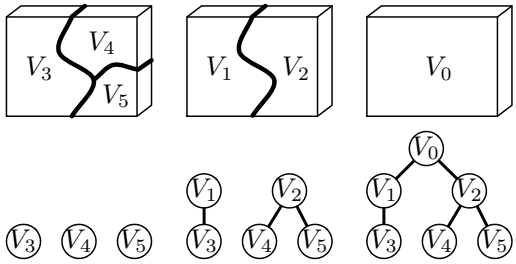


Figure 3. Illustration of the segmentation tree creation process. On the top of the figure, left, middle and right are bottom-up levels in a supervoxel hierarchy: T^1, T^2 and T^3 respectively. From left to middle, V_4 and V_5 are merged together, and V_3 remains itself as V_1 in the middle. From middle to right, V_1 and V_2 are merged together to a single top node V_0. The corresponding tree-graphs are in the bottom row.

segmentation), taking a high level would mean salient regions are missed (under-segmentation), but taking a middle level would over-segment in some regions but under-segment in others. We believe this is the key limitation to the adoption of supervoxels for early video analysis.

In this paper, we propose the first principled solution to overcome this key limitation of flattening a supervoxel hierarchy. Our emphasis is on video supervoxel hierarchies, but the core contribution is generally applicable to image and other segmentation hierarchies, given certain assumptions are met. Our approach includes a novel model—the uniform entropy slice (UES)—and a formulation for efficiently solving it via a binary quadratic program (QP). A *slice* through the hierarchy is a flattened supervoxel segmentation generally consisting of supervoxels from various levels of the hierarchy. The uniform entropy slice seeks to balance the amount of *information* in the selected supervoxels for a given feature criterion, such as motion, in which larger supervoxels from coarser-levels with less relative motion will be selected along with smaller supervoxels from finer-levels with more relative motion. Such a criterion enables us to pull out the most unique and dominant regions in a supervoxel hierarchy as shown in Figure 2.

The feature criterion, which drives the uniform entropy slice and hence the flattening, is independent of the supervoxel hierarchy itself. We explore four different cases for the feature criterion underlying the uniform entropy slice: motion, object-ness, human-ness, and car-ness. Motion is an unsupervised criterion that emphasizes relatively unique motion of segments in the flattened hierarchy; the other three are supervised criteria with object-ness based on the category independent measure [2] and human- and car-ness based on trained deformable parts models [13] from PASCAL VOC [12]. The variability of these underlying feature criteria and our ultimate findings demonstrate the high degree of versatility in the proposed method: indeed it can take any form of a subsequent criterion and apply it to a previously computed supervoxel hierarchy.

We have implemented and tested the uniform entropy slice on top of the state of the art graph-based segmentation (GBH) [16] and segmentation by weighted aggregation (SWA) [27] methods. Our quantitative comparison on the SegTrack [28] dataset using the LIBSVX benchmark [33] systematically finds that UES outperforms the natural baseline of selecting a single level from the hierarchy as well as the state of the art method, SAS [22], which combines multiple segmentations. Our qualitative results demonstrate numerous clear cases in which the flattened supervoxels are precisely what is expected for various feature criteria, like human-ness.

Our code as well as the two-actor videos are available as part of the LIBSVX 3.0 software library, which is downloadable at `http://www.cse.buffalo.edu/~jcorso/r/supervoxels/`.

2. Supervoxel Hierarchy Flattening Problem

Let $\mathcal{M}$ denote a given video and consider it as a mapping from the 3D lattice Λ^3 to the space of RGB colors. Each element of Λ^3 is a voxel. Based on some hierarchical supervoxel algorithm, consider an h level hierarchical oversegmentation of the video: $\mathcal{T} \doteq \{T^1, T^2, \ldots, T^h\}$ and V^i is the node set in supervoxel level T^i, $i \in [1, h]$. Individual nodes are denoted by subscripts V_s^i. The level superscript for V_s^i is dropped when the level is irrelevant. We let node V_0 be the root node of the supervoxel hierarchy $\mathcal{T}$, and V^1 is the set of leaf nodes in $\mathcal{T}$.

We consider only supervoxel hierarchies that are trees, i.e., each node has one and only one parent (other than the root) and each node has at least one child (other than the leaves). Figure 3 shows the general creation process of such a supervoxel tree; GBH generates such a tree. The

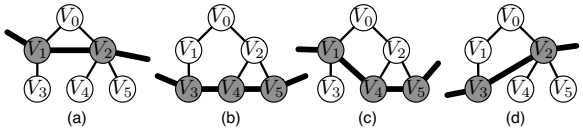


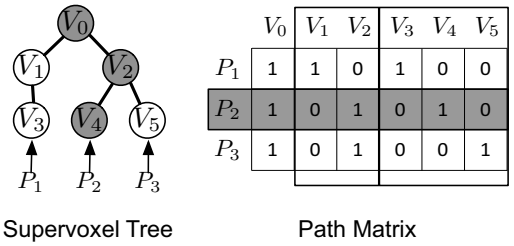
Figure 4. Slices in the example supervoxel tree. (a) - (d) list all 4 possible slices of the segmentation tree (excluding the root node). Each slice is highlighted as a thick black curve, and nodes on the slice are darkened.

algorithm initially builds a 26-connected voxel lattice over the whole video clip, then iteratively constructs a region graph over the obtained segmentation based on the minimum spanning tree merging criterion [14], and forms a bottom-up segmentation tree structure of the regions. The regions are described by their local texture histogram. The algorithm stops after a user-specified number of iterations. The algorithm tends to preserve the important region boundaries in the hierarchical merging process. We show results with both GBH and SWA, with a small modification of SWA to turn its general graph hierarchy into a tree.

Define a *tree slice* as a set of nodes from the hierarchy such that on each root-to-leaf path in the hierarchy, there is one and only one node in the slice set. Each such slice provides a plausible *flattened* hierarchy. If we combine all the nodes in the slice, then we can obtain a new composed segmentation of the original video from the hierarchical supervoxels. Fig. 4 shows example tree slices for the segmentation tree from the previous Fig. 3. The set of all tree slices includes both trivial (e.g., just nodes from one level) and non-trivial node selections. Note that we call this a tree slice rather than a *tree cut* to distinguish it from conventional use of the term *cut*, which generally indicates a set of edges and not nodes as we have in the slice.

More formally, consider a binary variable x_s for each node V_s in the tree $\mathcal{T}$. The binary variable x_s takes value 1 if node V_s is a part of the slice and value 0 otherwise. Denote the full set of these over the entire tree as $\mathbf{x}$. Any instance of $\mathbf{x}$ induces a selection of nodes in the tree $\mathcal{T}$, but not all instances of $\mathbf{x}$ are valid. For example, there are many instances of $\mathbf{x}$ that will select both a node and its ancestor. The trivial single-level selection is $\mathbf{x}(V^i) = 1$ and $\mathbf{x}(\mathcal{T} \setminus V^i) = 0$.

In a valid slice, each root-to-leaf path in the segmentation tree $\mathcal{T}$ has one and only node being selected. We formulate this constraint linearly. Let $\mathcal{P}$ denote a p by N binary matrix, where $p = |V^1|$ is the number of leaf nodes in $\mathcal{T}$, and $N = |\mathcal{T}|$ is the total number of nodes in $\mathcal{T}$. Each row of $\mathcal{P}$ encodes a root-to-leaf path by setting the corresponding columns for the nodes on the path as 1 and 0 otherwise. Such a matrix enumerates all possible root-to-leaf paths in $\mathcal{T}$. Fig. 5 shows the path matrix $\mathcal{P}$ for our example supervoxel tree from Fig. 3. There are three rows in the path matrix $\mathcal{P}$, which are the three root-to-leaf paths. The



	V_0	V_1	V_2	V_3	V_4	V_5
P_1	1	1	0	1	0	0
P_2	1	0	1	0	1	0
P_3	1	0	1	0	0	1

Figure 5. Supervoxel tree $\mathcal{T}$ and the corresponding path matrix $\mathcal{P}$. The path P_2 is highlighted to illustrate the path matrix $\mathcal{P}$ in which each row specifies a root-to-leaf path through the tree.

six columns of the path matrix $\mathcal{P}$ are the six nodes (including the root node V_0) in the segmentation tree $\mathcal{T}$. We use the path P_2 to illustrate the values on a row of $\mathcal{P}$—nodes $\{V_0, V_2, V_4\}$ are on path P_2.

For a given tree $\mathcal{T}$, we compute the path matrix $\mathcal{P}$ by a breadth-first search with complexity $O(ph)$. The size of $\mathcal{P}$ is tractable for typical videos: the number of rows is exactly the number of leaves in the tree, which is either the number of voxels or some number of supervoxels of the smallest scale maintained in the tree (in the case the full tree is not used); the number of columns is typically a factor of two on the number of rows due to the agglomerative nature of the supervoxel methods.

To ensure a tree slice is a valid, we have

$$\mathcal{P}\mathbf{x} = \mathbf{1}_p \ , \tag{1}$$

where $\mathbf{1}_p$ is an p-length vector of all ones. This linear constraint ensures that every root-to-leaf path (row of matrix $\mathcal{P}$) has one and only one node in the slice $\mathbf{x}$. If there is more than one node being selected in P_i, then $P_i\mathbf{x} > 1$. If there is no node being selected in P_i, then $P_i\mathbf{x} = 0$. The valid selection $\mathbf{x}$ is called a tree slice.

3. The Uniform Entropy Slice

The previous section presents the tree slice problem and a linear constraint to assess the validity of a slice; here we present a new model based on uniform entropy to quantify a slice. The intuitive idea behind the uniform entropy slice is that we want to select nodes in the tree that balance the information contribution to the overall slice. We are inspired by the Uniform Frequency Images work of Hunter and Cohen [17]. Their model is proposed for image compression; they automatically generate an invertible warping function that downshifts the image's highest spatial frequencies in exchange for upshifting some of its lowest spatial frequencies, producing a concentration of mid-range frequencies. In other words, the compressed image is able to focus more bits on the parts of the image that have a higher frequency signal than those with a lower frequency signal.

In our case for supervoxel hierarchies, one can relate finding the best slice in a hierarchy to such a compression

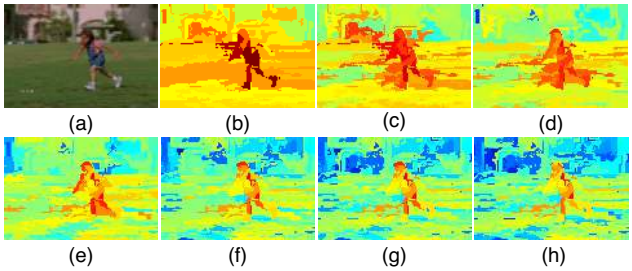


Figure 6. Example hierarchy node entropy for the motion feature criterion. (a) is the raw video *girl* from SegTrack, (b: coarse) – (h: fine) are node entropy from various levels in the hierarchy. The entropy color from dark blue to dark red maps entropy changing from low to high (using the *jet* colormap in Matlab). Notice how the entropy of the girls limbs is relatively higher than that of the background for corresponding hierarchy levels.

problem; we want a slice that is able to retain the greatest amount of information relative to the number of selected supervoxels: select bigger supervoxels from coarse levels when there is little information content and conversely, select smaller supervoxels from fine levels when there is high information content.

Information content is specified relative to a certain feature criterion, such as motion or human-ness. We specify four such feature criteria later in Sec. 3.2 and experiment with them in Sec. 4. For the current discussion, assume we have a feature criterion $\mathcal{F}(\cdot)$ that maps a node V_s to a discrete distribution over the feature range. For example, consider an unsupervised motion feature criterion in which we want the slice to focus on regions of the video that are moving uniquely relative to the rest of the video—e.g., a girl running leftward while the camera slowly pans as in Fig. 6. In this case, we compute optical flow over the video and then compute a bivariate discrete distribution over a set of flow magnitudes and flow directions for $\mathcal{F}$.

The information content of each node V_s in the hierarchy is computed by the entropy over $\mathcal{F}(\cdot)$:

$$E(V_s) \doteq -\sum_{\gamma} P_{\mathcal{F}(V_s)}(\gamma) \log P_{\mathcal{F}(V_s)}(\gamma) \ , \qquad (2)$$

with γ varying over the bivariate discrete feature range.

We next propose the uniform entropy objective, which considers the node information content according to Eq. 2 and seeks a tree slice that balances the overall information content of the selected nodes. Again, consider a valid tree slice $\mathbf{x}$ which is a vector of binary variables with one x_s for each node V_s in the hierarchy taking value 1 if the node is on the slice and 0 otherwise. The uniform entropy objective hence seeks a valid tree slice that minimizes the difference in entropy of selected nodes:

$$\mathbf{x}^* = \arg\min \sum_{V_s, V_t \in \mathcal{T}} |E(V_s) - E(V_t)| x_s x_t \ . \qquad (3)$$

where the minimization is over valid tree slices $\mathbf{x}$.

The intuition behind the uniform entropy objective is twofold. First, in a common case, the entropy of a supervoxel in coarser levels of the hierarchy drops down when the segment breaks up into smaller pieces at finer levels. Again consider Fig. 6, which shows the node entropy for a motion feature criterion on the video *girl* from the SegTrack dataset [28]. It is clear that the node entropy generally decreases from coarser to finer levels, and those informative supervoxels (the girl in this case) have overall more motion entropy than the background. It is hence plausible the slice will select nodes around the girl at finer levels to match similar motion entropies to the background at coarser levels in the hierarchy. Second, regions of the video that are salient for the specified feature criterion tend to have higher entropy than non-salient regions because of articulation and variability of the features near the salient region boundaries. Hence, when selecting the supervoxels, our goal is to preserve the detail in the segmentation of these salient regions and less so in the non-salient regions.

3.1. Uniform Entropy Slice as a Binary QP

Directly minimizing Eq. 3 is complex because it requires enumerating all valid tree slices and includes a degenerate minimum which selects the root node only. We instead reformulate the objective as the following binary quadratic program, which we call the *uniform entropy slice*.

$$\begin{aligned} \text{minimize} \quad & \sum_s \alpha_s x_s + \sigma \sum_{s,t} \beta_{s,t} x_s x_t \qquad (4) \\ \text{subject to} \quad & \mathcal{P}\mathbf{x} = \mathbf{1}_p \\ & \mathbf{x} = \{0,1\}^N \end{aligned}$$

where α_s forms a vector with length equal to N, $\beta_{s,t}$ is an entry in an N by N matrix, and σ controls the balance between the two terms. Note the $\mathcal{P}\mathbf{x} = \mathbf{1}_p$ slice validity constraint from Eq. 1. Furthermore, note that there is no explicit notion of neighborhood in the uniform entropy slice, but $\beta_{s,t}$ can be specified based on the neighborhood structure in the tree.

The linear term makes the slice prefer simpler segmentations when possible, i.e., prefer coarser levels in the hierarchy rather than finer levels in the hierarchy. The following is the unary potential we set:

$$\alpha_s = |V^i| \quad \text{if } V_s \in V^i \ , \qquad (5)$$

where $|V^i|$ means the total number of supervoxels in ith level of the tree. In typical supervoxel trees, there is a quadratic relationship between $|V^i|$ and $|V^{i+1}|$ due to algorithm construction.

The quadratic term implements the uniform entropy objective

$$\beta_{s,t} = |E(V_s) - E(V_t)||V_s||V_t| \qquad (6)$$

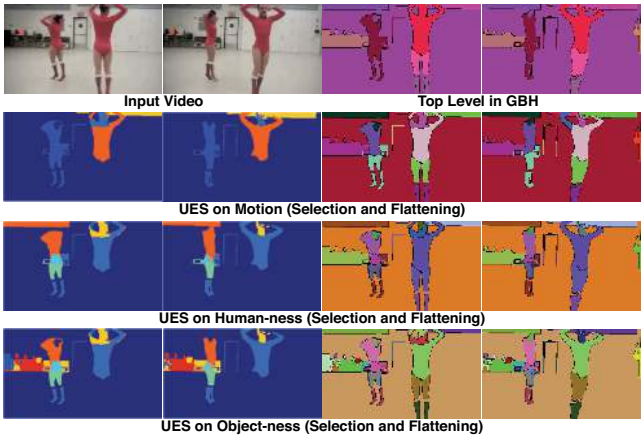


Figure 7. Different feature criteria focus on different parts of the video *dancers*. Here, the motion feature focuses mostly on the dominant man in front and some attention to the woman in the back. On the other hand, the human-ness criterion focuses on both dancers, while the object-ness also focuses on the chairs in the back. All these feature criteria try to avoid under-segmentation of interesting objects as shown in the top level in GBH (the woman merged with the door and bench in the back), and maintain a uniform clean background. In the UES Selection images (left two columns), the dark red to dark blue means the finer levels to coarser levels in the supervoxel hierarchy tree.

where $|V_s|$ and $|V_t|$ denote the volume of the supervoxels V_s and V_t respectively. Although nodes in the coarser levels of the tree have relatively higher entropy than nodes in the finer levels, the number of coarser level nodes is dramatically less than those in the finer levels. By adding the volume factors, we push the selection down the hierarchy unless a uniform motion entropy has already been achieved. Indeed this behavior has generally been observed in our quantitative and qualitative experiments. See, for example, the level of the hierarchy selection for the video *girl* in Fig. 9 in Sec. 4.

We solve the QP using a standard solver (IBM CPLEX), but note that other approaches to solving it are plausible, such as spectral relaxation [20].

3.2. Feature Criteria

The uniform entropy slice operates directly on a supervoxel hierarchy that was computed by an unsupervised method such as GBH. However, the feature criteria, which drive the tree slice optimization, provide a doorway to apply situation-specific guidance post hoc. To illustrate this versatility, we describe four such feature criteria that span the spectrum of unsupervised to class-specific supervised. Each of these have been implemented and used in our experiments (Sec. 4). In Figure 7, we show the different feature criteria on one video and observe how different slices are computed with criterion-specific foci of attention.

Unsupervised: Motion. The motion feature criterion has been discussed as an example earlier and we hence do not describe it in detail here. For computing the feature we use the Liu [23] optical flow method and compute flow on each frame of the video. For the map $\mathcal{F}$ we discretize the range to four magnitude bins and eight angular bins.

Supervised, Category-Independent: Object-ness. This demonstrates a general category-independent, object-ness feature, as it is common in problems like video object segmentation [19]. We sample 1000 windows per frame using [2] according to their probability of containing an object. Then we convert this measure to per-pixel probabilities by summing the object-ness score over all windows covering a pixel, and normalizing the result over the video, which is similar to [30]. We use six quantization levels for $\mathcal{F}$.

Supervised: Human-ness and Car-ness. The last two feature criteria are class-specific and demonstrate further post hoc flattening goals. We use the state of the art deformable part based model [13] with previously trained PASCAL VOC detectors to compute car-ness and human-ness. We use a low detection threshold to get more detection bounding boxes for each frame. Then, similar to object-ness, we count the per-pixel detection hits to obtain a detection hit map for each frame. We again set six quantization levels for $\mathcal{F}$.

4. Experiments

We evaluate the uniform entropy slice (UES) both quantitatively (Sec. 4.1) and qualitatively (Sec. 4.2) on various benchmark and new, challenging unconstrained videos. To explore the generality of UES, we apply it to supervoxel hierarchies generated by two different methods, GBH [16] as implemented in [33] and SWA [27] as implemented in [9]. For GBH, we construct a supervoxel tree directly from its output supervoxel hierarchy, since the method itself generates a tree structure. For SWA, we simplify the supervoxel hierarchy, which is a general directed acyclic graph, to a tree structure by taking the most dominant parent for each child node and denote this variant of SWA as SWA^T.

The most important parameter in UES is the ratio σ between the linear and quadratic terms. However, we have observed that, in practice, the relative hierarchy selection of supervoxels is not very sensitive to it. We L-1 normalize both of these terms and in our quantitative experiments, we empirically set $\sigma = 10$ for all the videos.

4.1. Quantitative Evaluation

Benchmark and Dataset. We use the recently published supervoxel benchmark LIBSVX [33] to evaluate the UES with GBH and SWA^T methods. The benchmark provides six supervoxel methods and a set of supervoxel evaluation metrics. We use the SegTrack dataset from Tsai et al. [28], which provides a set of human-labeled single-foreground objects with six videos stratified according to difficulty on color, motion and shape.

Video	SWA^T Flattening												GBH Flattening											
	3D ACCU			3D UE			3D BR			3D BP			3D ACCU			3D UE			3D BR			3D BP		
	BASE	SAS	UES	BASE	SAS	UES	BASE	SAS	UES	BASE	SAS	UES	BASE	SAS	UES	BASE	SAS	UES	BASE	SAS	UES	BASE	SAS	UES
birdfall2	9.0	0.0	**69.7**	36.8	38.3	**26.5**	82.1	81.9	**84.9**	0.66	0.65	**0.70**	1.8	0.0	**53.8**	26.9	27.1	**23.2**	74.3	74.0	**82.1**	0.83	0.83	**0.94**
cheetah	0.0	0.0	0.0	47.4	47.4	47.4	65.7	65.7	65.7	1.93	1.93	1.93	30.2	30.2	**39.4**	**31.7**	32.4	34.1	78.3	**79.3**	75.3	1.42	1.43	**1.60**
girl	**56.4**	55.9	56.1	7.8	8.2	**5.9**	56.6	56.5	**57.7**	3.36	**3.39**	3.31	41.9	**45.6**	41.9	11.2	**11.1**	13.7	54.4	54.1	**58.1**	2.90	2.91	**3.94**
monkeydog	0.0	0.0	0.0	52.0	52.2	**51.9**	84.9	**86.8**	86.7	3.32	3.12	**3.35**	71.9	**79.9**	**79.9**	37.1	**36.6**	43.2	90.7	90.9	**91.0**	2.55	2.47	**2.95**
parachute	83.7	85.5	**90.3**	23.6	24.4	**22.3**	93.2	93.0	**94.6**	1.66	1.69	**1.72**	89.4	89.4	89.4	38.6	38.6	38.6	87.4	87.4	87.4	10.0	10.0	10.0
penguin	**94.7**	94.4	94.4	**1.8**	1.9	**1.8**	**73.7**	72.3	71.0	1.36	**1.37**	1.27	84.7	83.1	**85.0**	2.2	1.9	1.8	**66.7**	65.4	65.5	**1.10**	0.96	0.88
AVERAGE	40.6	39.3	**51.8**	28.2	28.7	**26.0**	76.0	76.0	**76.8**	**2.05**	2.03	**2.05**	53.3	54.7	**64.9**	**24.6**	**24.6**	25.8	75.3	75.2	**76.6**	3.14	3.11	**3.39**

Table 1. Quantitative comparison of UES against the other two baseline methods on SegTrack dataset. We evaluate on two different hierarchical supervoxel methods: SWA^T and GBH. The leading scores of each metric per video are in bold font.

Baseline Methods. We compare with two baseline methods. The first is a simple trivial slice that takes a single level from the hierarchy, which we denote as "Base" in Table 1. Another method is a video extension of Segmentation by Aggregating Superpixels (SAS) [22], which composites multiple segmentations together based on bipartite graph matching. It achieves state-of-the-art performance on the image Berkeley Segmentation Database [25]. To the best of our knowledge, we are not aware of other video supervoxel selection algorithms. The number of supervoxels from the input hierarchy varies from less than 10 to about 800. For fair comparison, we feed SAS and UES with the unsupervised motion feature only. The scores in Table 1 are generated for the same number of supervoxels for all three methods per video. The scores of "Base" are generated by linear interpolation of nearby levels as in [33].

3D Segmentation Accuracy measures the average percentage area of the ground-truth segments being correctly segmented by the supervoxels. 3D Undersegmentation Error measures the fraction of voxels that go beyond the boundary of the ground-truth when mapping the supervoxels onto it. Along with 3D Boundary Recall, we add 3D Boundary Precision as a new metric. Overall, the proposed UES achieves better performance for both SWA^T and GBH supervoxel hierarchies than the other two baseline methods, and in some cases, such as 3D ACCU the improvement is significant for both methods. We note that neither the baseline one level selection nor the SAS can correctly segment the video "birdfall2" with only a small number of supervoxels. In some cases, such as the video "cheetah" using SWA^T, the scores are frequently the same for the three methods; this is a failure case of the overall supervoxel hierarchies, which we have observed to have little variation in supervoxels covered on the object at multiple levels in the hierarchy.

4.2. Qualitative Evaluation

UES on Motion. Figure 8 is an example video showing that UES can help avoid foreground under-segmentation and background over-segmentation. UES selects the coarse levels of the hierarchy for the background when doing so does not lead to foreground segments leaking, as in GBH.

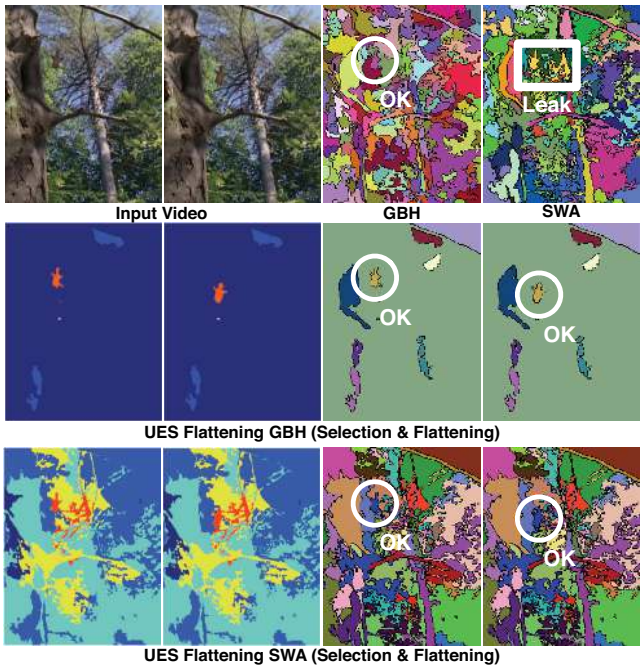


Figure 8. UES helps avoid foreground under-segmentation and background over-segmentation on video *birdfall2*. GBH and SWA on the top row show the middle levels from each hierarchy. A white circle means the bird has no segmentation leak, whereas a white rectangle means a segmentation leak with the surrounding tree branches. Here, we use the motion criterion.

Similarly, UES pushes the foreground and the corresponding leaking parts of the video down to the finer levels of the SWA hierarchy, while it still keeps the other background regions in the coarser levels of hierarchy.

UES vs. Baselines. In Figure 9, the girl is running leftward, and the camera is also slowly moving leftward. The position of the girl in the video does not change much, but the pose changes drastically. The articulated pose generates more motion entropy over time than the surroundings do, which also allows UES to focus on the girl, as shown on the right half of the figure with both motion and object-ness criteria. In contrast, a simple selection of a middle level from GBH gives a quite fragmented background. If we take a coarser level of the hierarchy, then the girl is merged too much with the grass in background. SAS does merge the supervoxels,

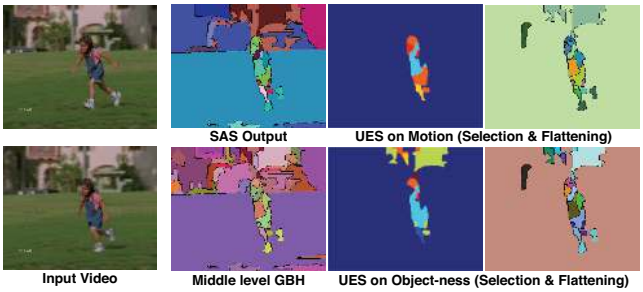


Figure 9. Comparison of UES against baseline methods on video *girl* from SegTrack. UES on Motion and SAS (based on motion) have identical number of supervoxels in their final outputs. We also show a simple selection of the middle level from GBH as well as UES on Object-ness for comparison.

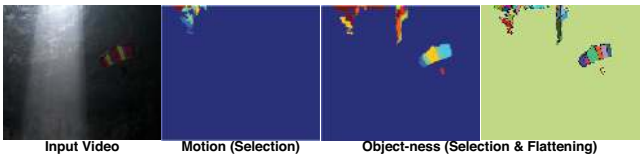


Figure 10. UES on Object-ness selects the *parachute* segments and the human, while UES on Motion fails.

but it lacks a focus on selection.

Object-ness vs. Motion. Sometimes, the motion criterion fails when the rigid objects have same motion as the camera in a video, or in a video with chaotic motion. The object-ness can better handle the above situations. We show an example in Figure 10, where the motion completely fails to select the rigid object *parachute*, because the motion of it is uniform over the video (from left to right) with the camera. However, with the object-ness criteria, the algorithm can easily select it from the lower levels in the hierarchy. The supervoxels in the top part of the object-ness selection image may seem to be errors, but indeed, these are expected: the parachute moves from left to right across the light and these selected supervoxels touch it at an earlier frame when it was passing by.

Human-ness and Car-ness. Recall that Figure 7 shows an example of how different feature criteria drive the algorithm to focus on different parts of a video. The top level hierarchy in GBH mistakes the woman in the left with the door and bench in the background. With the motion criterion, UES selects the man in the front from a finer level than the woman in the back, since the man is the most dynamically moving part of the video. Interestingly, the human-ness focuses on the two dancers while the object-ness not only focuses on the two dancers but also on the chairs in the back. Figures 11 and 12 further demonstrate examples of the supervised feature criteria in comparison to the motion criterion; in both cases the unsupervised motion criterion slices as well as the trained feature criterion suggesting the unsupervised measure may be as useful as the trained ones, at least in cases of relatively static backgrounds.

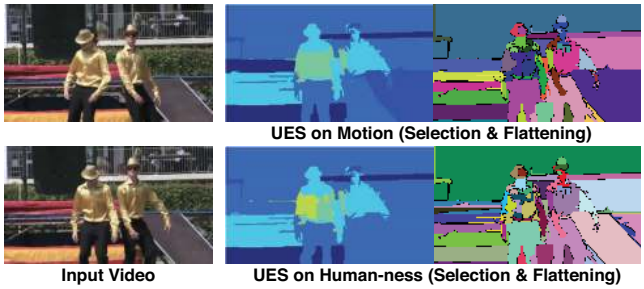


Figure 11. UES on Motion and Human-ness on video *danceduo*.

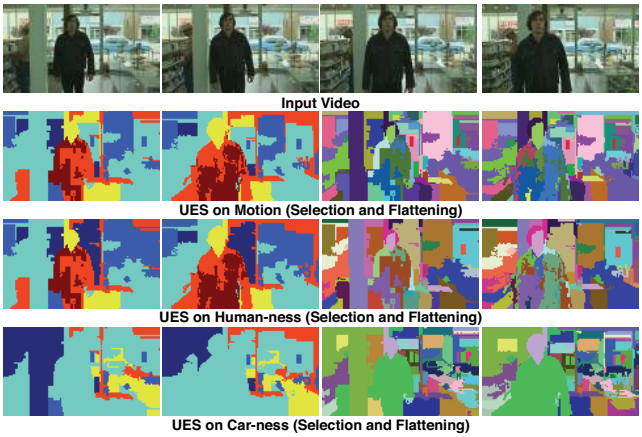


Figure 12. UES on Motion, Human-ness and Car-ness on video *nocountryforoldmen* from [16]. For Motion and Human-ness, the moving man is selected from the finer levels, while most others are from coarser levels. For car-ness, the car and nearby regions are selected from finer levels. The red selection around the window is to avoid leaks.

5. Discussion and Conclusion

Summary. Supervoxel segmentation has gained potential as a first step in early video processing due to recent advances in hierarchical methods [16], streaming methods [34] and related evaluations [33]. However, the high-performing methods generate a hierarchy of supervoxels that often renders the user with more questions than at the outset due to the intrinsic limitations of unsupervised grouping. We have proposed the first principled method to flatten the hierarchy, called the uniform entropy slice (UES). Our method seeks to balance the level of information across the selected supervoxels: choose bigger supervoxels in *uninteresting* regions of the video and smaller ones in *interesting* regions of the video. A post hoc feature criterion is used to drive this information selection, and is independent of the original supervoxel process. Our experiments demonstrate strong qualitative and quantitative performance.

Generality. Although our paper has strictly discussed video supervoxel hierarchies thus far, the proposed method is general and can directly be applied to other segmentation hierarchies, such as those on images [27] or even a hierarchical clustering on top of existing trajectories [6, 31], so

long as two assumptions are met. First, the hierarchy must be a tree (or adequately transformed into one as we did for SWA in this paper). Second, a feature criterion can be defined to drive the slice.

Implications to Related Video Problems. The proposed uniform entropy slice makes it plausible to provide an initial supervoxel map for further processing in problems like video object segmentation. In particular, every video object segmentation method we are aware of [19, 24, 35] begins with an over-segmentation (typically frame-level superpixels) and extracts a single moving foreground object. We expect our flattened output to provide a strong input for such methods as the community moves from single to multiple objects. Second, our method of using any feature criterion is more general than the existing strictly object-ness criterion that has thus far been used in video object segmentation. And, this has strong implications as the community begins to consider semantic video segmentation on unconstrained videos, which is a relatively new problem in video that has thus far focused on constrained videos [5, 8].

Limitations. The feature criterion is independent of the supervoxel method. In some respects, this fact is a clear benefit of the method, but it can also be considered a limitation: there is no guarantee that the uniform entropy slice is the optimal supervoxel segmentation for a given video and feature criterion. In other words, since the supervoxel hierarchy is computed independent of the feature criterion, its segments may not coincide with the natural ones for a given criterion. Our experiments demonstrate that for typical feature criteria this limitation is not critical, but further work is needed to better understand the induced error for a feature criterion-hierarchical supervoxel method pair.

Future Work. In the future, we plan to extend UES into a streaming setting to handle longer videos [34]. A key hurdle to overcome will be the tractability of the subsequent NP-hard quadratic program; we plan to pursue adequate approximations in this streaming case.

Acknowledgments. We are grateful to Zhenguo Li and Shih-Fu Chang for early discussions and for providing the code to SAS. This work was partially supported by NSF CAREER IIS-0845282, ARO YIP W911NF-11-1-0090 and DARPA Mind's Eye W911NF-10-2-0062.

References

[1] C. Aeschliman, J. Park, and A. C. Kak. A probabilistic framework for joint segmentation and tracking. In *CVPR*, 2010.

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 2011.

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

[5] G. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: a high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[6] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010.

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

[8] A. Y. C. Chen and J. J. Corso. Temporally consistent multi-class video-object segmentation with the video graph-shifts algorithm. In *WMVC*, 2011.

[9] 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. *TMI*, 27(5):629–640, 2008.

[10] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *CVPR*, 2006.

[11] C. Erdem, B. Sankur, and A. Tekalp. Performance measures for video object segmentation and tracking. *TIP*, 2004.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 2010.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010.

[14] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[15] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012.

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

[17] A. Hunter and J. D. Cohen. Uniform frequency images: adding geometry to images to produce space-efficient textures. In *Visualization*, 2000.

[18] J. Lee, S. Kwak, B. Han, and S. Choi. Online video segmentation by bayesian split-merge clustering. In *ECCV*, 2012.

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

[20] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005.

[21] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR*, 2011.

[22] Z. Li, X.-M. Wu, and S.-F. Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *CVPR*, 2012.

[23] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[24] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012.

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[26] J. Pont-Tuset and F. Marques. Supervised assessment of segmentation hierarchies. In *ECCV*, 2012.

[27] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 2006.

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

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

[30] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised semantic segmentation with a multi-image model. In *ICCV*, 2011.

[31] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[32] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, 2011.

[33] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, 2012.

[34] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012.

[35] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013.