Efficient Video Segmentation using Parametric Graph Partitioning

Chen-Ping Yu[1] Hieu Le[1] Gregory Zelinsky[1,2] Dimitris Samaras[1]

Computer Science Department[1], Psychology Department[2]
Stony Brook University, Stony Brook NY 11790
{cheyu, hle, samaras}@cs.stonybrook.edu[1], gregory.zelinsky@stonybrook.edu[2]

Abstract

Video segmentation is the task of grouping similar pixels in the spatio-temporal domain, and has become an important preprocessing step for subsequent video analysis. Most video segmentation and supervoxel methods output a hierarchy of segmentations, but while this provides useful multiscale information, it also adds difficulty in selecting the appropriate level for a task. In this work, we propose an efficient and robust video segmentation framework based on parametric graph partitioning (PGP), a fast, almost parameter free graph partitioning method that identifies and removes between-cluster edges to form node clusters. Apart from its computational efficiency, PGP performs clustering of the spatio-temporal volume without requiring a pre-specified cluster number or bandwidth parameters, thus making video segmentation more practical to use in applications. The PGP framework also allows processing sub-volumes, which further improves performance, contrary to other streaming video segmentation methods where sub-volume processing reduces performance. We evaluate the PGP method using the SegTrack v2 and Chen Xiph.org datasets, and show that it outperforms related state-of-the-art algorithms in 3D segmentation metrics and running time.

1. Introduction

Image segmentation is a mature field, with the output of the segmentation methods being used for object detection and segmentation [24][12], semantic understanding [32], shadow detection [26], etc. It started with several seminal works that cluster pixels with similar intensity and/or color features into groups [8][20][6], and gradually moved into clustering superpixels that share similar low level features [27][31][5][14][19]. The increased importance of video analysis necessitates efficient video segmentation, due to the heavy processing cost of video data. Video segmentation methods in general attempt to cluster similar pixels together under a spatio-temporal setting, either by methods that generate a set of hierarchical segmentations (from very detailed to more coarse) [10][7][30], or by methods that group superpixels together to form spatio-temporal superpixels or supervoxels [2][15][25]. The latter methods are significantly more efficient computationally, albeit less accurate. In this paper, we propose a "superpixel grouping" method that improves the state-of-the-art by as much as 30%, and is approximately 20 times faster.

Hierarchical video segmentation methods have demonstrated excellent 3D (spatiotemporal) segmentation results on standard datasets such as Segtrack [23] and Chen's Xiph.org [4] using a variety of metrics [28]. However, their applicability in video processing pipelines remains limited, due to computational complexity and the difficulty of automatically selecting the appropriate hierarchical layer for particular applications. As an attempt to address these obstacles, the recently proposed Uniform Entropy Slice (UES) method ([29]) selects different supervoxels from the several hierarchical layers to form a single output segmentation, by balancing the amount of feature information of the selected supervoxels [29]. UES builds on top of hierarchical segmentation pre-processing techniques such as [10] and [7] to produce a single segmentation that is more practical for further use. This comes at the cost of increased computation and decreased performance in 3D quantitative performance metrics compared to [10] and [7]. To obviate the use of expensive hierarchical segmentation as pre-processing, we adopt the commonly used approach of grouping superpixels under a spatio-temporal setting with a novel and efficient clustering algorithm that performs parametric graph partitioning (PGP) on the spatio-temporal superpixel graph.

PGP is a graph partitioning method that performs clustering without needing the user to specify the number of clusters, or search-window parameters as in mean-shift [6]. The method optimizes a number of two-component parametric mixture models over the edge weights of a graph (one model per feature type). The edges are bi-partitioned into a within-cluster and a between-cluster group by performing infer-

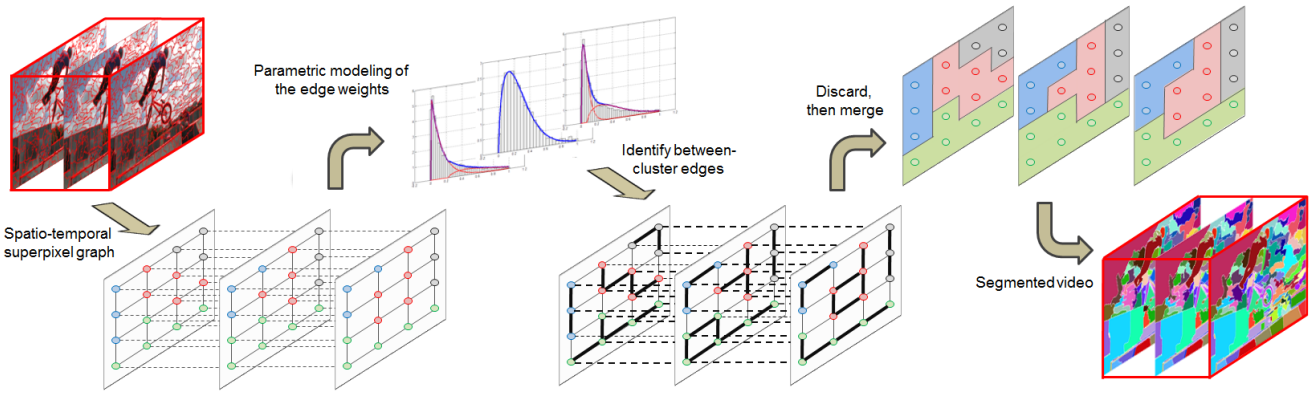

Figure 1. The overview of our proposed PGP method. First, a spatio-temporal superpixel graph is constructed from the video (dashed lines indicate temporal connections). Then, the edge weights are modeled by a mixture of Weibull distributions, computed separately for individual features. The mixture models generate information about the inter- and intra-cluster edges (intra-cluster edges in bold). The inter-cluster edges are discarded and each of the remaining groups of superpixels becomes a final video segment.

ence on these mixture models. Thus the between-cluster edges of the graph can be identified and removed, in order to create a variable amount of isolated clusters in an efficient manner (see also [33]). Each of these clusters corresponds to a 3D video segment. It is safe to assume that L_p-norm based similarity distance statistics in general are Weibull distributed [1]. Based on this observation, a previous method produced high quality single image partitioning results and segmented contiguous image regions well [33]. Using PGP has a number of advantages:

- PGP is computationally efficient; its run time is linear in the number of superpixels. This is especially suitable for processing big data such as video.
- PGP is a one-pass algorithm that produces a non-hierarchical result, which eliminates the need to select the appropriate hierarchical layer.

Figure 1 gives an overview of our method. In summary our main contributions are:

1. A novel general graph partitioning algorithm, that requires the differences between the feature vectors to be correlated and non-identically distributed. Our features satisfy this requirement (Section 3.1).
2. An efficient and robust video segmentation method that outputs a single segmentation map. We use PGP to partition the spatio-temporal superpixel graph. PGP models parametrically the similarity distance statistics for the features of interest. In this paper they are: intensity, hue, ab (from the Lab colorspace), motion (optic flow), and gradient orientation. We integrate these cues with an adaptive feature weighting scheme.
3. An extensive evaluation and comparison with the previous state-of-the-art method[29] on two widely used benchmark video datasets. Our video segmentation method improves the state-of-the-art significantly in 7 out of 8 total evaluation metrics, using less than 1/20th of the computation time and memory.

We describe the details and our adaptation of the PGP method in Section 2, followed by a series of experiments and quantitative evaluations with related work using SegTrack v2 [15] and Chen's Xiph.org datasets [3] in Section 3. Finally, possible improvements and future work for our proposed work is discussed in Section 4.

2. Segmentation via superpixel clustering

In image segmentation, it is common to build a graph where the nodes represent superpixels and the edges connect adjacent superpixels, weighted by the similarity distance of some feature [26][27]. Then an algorithm partitions this graph to obtain a set of disjoint superpixel groups as the resulting segmentation. For video segmentation, pixel-based methods tend to perform better than superpixel segmentation methods ([10][7]) because superpixels segmented on individual frames can be temporally unstable. However, since superpixel image segmentation methods are more computationally efficient compared to working with pixels, they are sometimes the only viable choice [11][18]. The most prominent methods for grouping superpixels are spectral clustering [11][9] and agglomerative clustering [18]. However, spectral clustering requires a quadratic increase in storage and computation as data instances grow, which is particulatly costly for videos, while the computational efficiency of agglomerative clustering comes at the cost of reduced segmentation performance.

We propose a novel clustering method for video segmentation via superpixel grouping, called parametric graph partitioning (PGP). The algorithm attempts to label and remove between-cluster edges of a graph by fitting a two-

component Weibull mixture model (WMM) over the distribution of similarity distances between the connected nodes (i.e. the edge weights) of the entire graph. The value at the cross-over point (critical value) between the two components represents the split between the intra- and inter-cluster edge weights. The edge weights that are higher than the critical value are removed and identified as the inter-cluster edges. An edge of a graph can only be labeled as inter- or intra-cluster, hence the algorithm fixes the number of mixture components at two.

In the next section, we describe the PGP algorithm in detail, including the motion features used for video segmentation, our adaptive feature weighting scheme, and branch reduction method as parts of an integrated framework.

2.1. Spatio-temporal superpixel graph

Given a graph $G = (V, E)$, an edge $e_{u,v} \in E$ connects two neighboring nodes (superpixels) $u, v \in V$. Let x_i be the weight of the i^{th} edge e_i of the graph, the task is to assign a binary label to e_i by an indicator function $y_i = I(x_i)$, such that $y_i = 1$ if e_i is an intra-cluster edge that should be retained, or $y_i = 0$ if e_i is a inter-cluster edge that should be removed from the graph. For a given feature $f \in \mathbf{F}$, x_i is the similarity distance between the feature histograms of nodes (superpixels) v_a and v_b connected by the i^{th} edge e_i such that $x_i = D(v_a, v_b|f)$, and we denote $\mathbf{x}$ as similarity distances of different features.

Neighboring nodes in a spatio-temporal graph are defined as nodes that are spatially or temporally adjacent to one another, where temporal adjacency in our framework is defined differently depending on whether the motion feature (described in Section 3.2) is used: if the motion feature is used, the temporal neighbors of v_a are nodes located within a $n \times n$ window on the next temporal frame, where the center of the window is specified by the mean motion vector of v_a; if the motion feature is not used, then temporal adjacency is defined by a $4n \times 4n$ window directly on the next temporal frame using the centroid of v_a as the center of the window.

2.2. Parametric graphic partitioning

Since the edges can only be either intra-cluster or inter-cluster, the distribution of the edge weights $\mathbf{x}$ computed from a given feature f is therefore composed of the two respective populations, where the lower values are more likely to be intra-cluster distances and the higher values are more likely to belong to the inter-cluster group. Given a single feature, there is one ideal critical value that separates $\mathbf{x}$ into these two components. Finding this critical value would solve the edge labeling problem. To find the critical value, one can naively perform k-means with $k = 2$ or fit a two-component Gaussian mixture model over the distribution of $\mathbf{x}$ (assuming the similarity distances are Normally distributed). However, the underlying structure of the similarity distance is unknown, so these assumptions are potentially wrong. However, it has been shown [1] that, if an L_p-norm based distance (e.g. Earth Mover's Distance [33]) is used as the similarity distance metric for the feature histograms, that L_p distance follows a Weibull distribution if the differences between the two feature vectors to be compared are correlated and non-identically distributed. We show that our features satisfy the above assumptions in Section 3.1. It is therefore theoretically plausible to find the critical value by fitting a 2-component Weibull mixture model over the distribution of L_p distance statistics, and retain the cross-point of the two components as the critical value for graph partitioning. The Weibull mixture model (WMM) has the general form:

$$\mathcal{W}^K(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k \phi_k(\mathbf{x}; \theta_k) \tag{1}$$

$$\phi(x|\alpha, \beta, c) = \frac{\beta}{\alpha}\left(\frac{x-c}{\alpha}\right)^{\beta-1} e^{-\left(\frac{x-c}{\alpha}\right)^{\beta}} \tag{2}$$

where $\theta_k = (\alpha_k, \beta_k, c_k)$ is the parameter vector for the k^{th} mixture component, and ϕ denotes the three-parameter Weibull p.d.f. with the scale (α), shape (β), and location (c) parameter, and mixing parameter π such that $\sum_k \pi_k = 1$. In this case, the two-component WMM contains a 6-parameter vector $\theta = (\alpha_1, \beta_1, \alpha_2, \beta_2, c_2, \pi)$ that yields the following complete form:

$$\begin{aligned}\mathcal{W}^2(\mathbf{x}|\theta) = \pi&\left(\frac{\beta_1}{\alpha_1}\left(\frac{\mathbf{x}}{\alpha_1}\right)^{\beta_1-1} e^{-\left(\frac{\mathbf{x}}{\alpha_1}\right)^{\beta_1}}\right.\\ &\left.+ (1-\pi)\left(\frac{\beta_2}{\alpha_2}\left(\frac{\mathbf{x}-c_2}{\alpha_2}\right)^{\beta_2-1} e^{-\left(\frac{\mathbf{x}-c_2}{\alpha_2}\right)^{\beta_2}}\right)\right.\end{aligned} \tag{3}$$

To optimize the above mixture model, we estimate the parameters using both maximum likelihood estimation (MLE) and Nonlinear least squares (NLS) and compare the results in Tables 1 and 2. The log-likelihood function of $\mathcal{W}^2(\mathbf{x}; \theta)$ is given by:

$$\begin{aligned}\ln \mathcal{L}(\theta; \mathbf{x}) = \sum_{n=1}^{N} \ln\{&\pi\left(\frac{\beta_1}{\alpha_1}\left(\frac{x_n}{\alpha_1}\right)^{\beta_1-1} e^{-\left(\frac{x_n}{\alpha_1}\right)^{\beta_1}}\right.\\ &\left.+ (1-\pi)\left(\frac{\beta_2}{\alpha_2}\left(\frac{x_n-c_2}{\alpha_2}\right)^{\beta_2-1} e^{-\left(\frac{x_n-c_2}{\alpha_2}\right)^{\beta_2}}\right)\right\}\end{aligned} \tag{4}$$

We adopt the Nelder-Mead algorithm as a derivative-free optimization method of MLE [17] due to the complexity of the likelihood function. With NLS, we approximate $\mathbf{x}$ with histograms where the appropriate bin-width is adaptively computed by $l = 2(IQR)n^{-1/3}$, where IQR is the interquartile range of $\mathbf{x}$ with sample size n [13]. Then, NLS

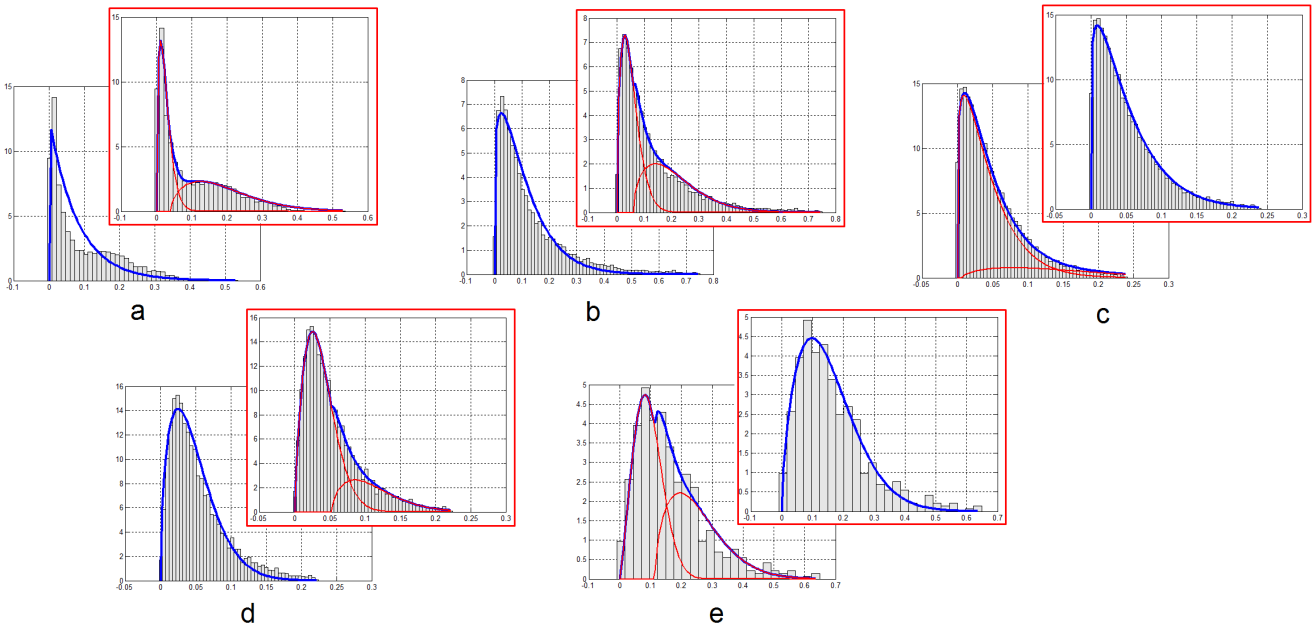

Figure 2. The nonlinear least-squares fits of Weibull Mixture Models (one and two components) on the Earth Mover's Distance statistics of the five features with model selection done for the BMX-1 video. The blue lines are the posterior probability and the red lines are the probability of individual mixture components. The models in the red boxes are the selected ones by AIC. a: intensity, b: hue, c: ab, d: motion, and e: gradient orientation.

optimizes the parameters of Eq. 3 by treating the height of each bin as a curve fitting problem; the least squares minimizer is computed by applying the trust-region method [21].

Both optimization methods require a well chosen initial guess parameter vector θ' to avoid local minima. The initial guesses are intuitive in our graph framework. Given a node v_i and its set of adjacent neighbors $\mathbf{v}_i^N$, the neighbor that is most similar to v_i is most likely to belong to the same cluster as v_i. We therefore fit a single Weibull over the minimum neighbor distance of all superpixels as the initial guess of the first mixture component. For the initial guess of the second mixture component, we extract the edge weights that are more than some percentile of $\mathbf{x}$, where $p = 0.6$ was found to be a good point in our experiments.

2.3. Model selection and edge labeling

While the edge weights $\mathbf{x}$ are composed of the two populations (intra- and inter-cluster), there are several situations when the distribution of $\mathbf{x}$ appears more uni-modal than bi-modal (as per our assumption), such as when there are very few inter-cluster edges, or more frequently when the differences between the two populations are less pronounced and more tapered. Therefore, we also fit a single Weibull model over $\mathbf{x}$ in addition to the two-component WMM, and select the appropriate model via the Akaike Information Criterion (AIC) (Figure 2). AIC suits our two-population assumption because it penalizes the more complex (two-component) model less. Standard AIC is used with MLE. For NLS, we use the corrected AIC (AICc) with residual sum of squares (RSS) because the sample size is smaller (generally if $n/k \leq 40$, where n is the number of histogram bins, and k is number of model parameters):

$$AICc = n\ln(\frac{RSS}{n}) + 2k + \frac{2k(k+1)}{n-k-1} \tag{5}$$

When the two-component model is selected such that $AIC(\mathcal{W}^2) \leq AIC(\mathcal{W}^1)$, the critical value γ is the cross-point between the two components. Otherwise γ is set at a given sample percentile parameter τ, computed by the inverse c.d.f. of $\mathcal{W}^1$:

$$\begin{aligned} &\text{if} \quad \text{AIC}(\mathcal{W}^2) \leq \text{AIC}(\mathcal{W}^1) \\ &\qquad \gamma = x, \quad \text{s.t.} \;\; \pi_1\phi_1(x|\theta_1) = \pi_2\phi_2(x|\theta_2) \\ &\text{otherwise} \\ &\qquad \gamma = -\alpha_1(\ln(1-\tau))^{1/\beta_1} \end{aligned}$$

After obtaining γ, we combine the n features to compute the label for x_i as the weighted-sum over the scaled $\mathbf{x}$:

$$y_i = I(\sum_{k=1}^{n} w_i\sigma(x_i,\gamma) < 0 \mid f_k) \tag{6}$$

where $\sigma(x_i,\gamma)$ is a scaling function that linearly scales $[\min(\mathbf{x}),\gamma]$ to $[-1,0]$, and $[\gamma,\max(\mathbf{x})]$ to $[0,1]$. $\sigma(x_i,\gamma)$ is a piecewise function with γ being the breakpoint:

$$\sigma(x_i, \gamma) = \begin{cases} (x_i - \gamma)/(\gamma - \min(\mathbf{x})) & \text{if } x_i < \gamma \\ (x_i - \gamma)/(\max(\mathbf{x}) - \gamma) & \text{if } x_i \geq \gamma \end{cases} \quad (7)$$

Notice that when there is only one feature being considered, Eqs. 6 and 7 combine into a simple threshold function and γ becomes a threshold that partitions the graph such that $y_i = I(x_i >= \gamma)$.

When multiple features are considered, we expect that in different parts of the image, different features will be most prominent. For instance if two neighboring superpixels both undergo significant motions, their mean motion feature value will be higher than most other superpixel pairs, indicating that motion similarity is of higher importance when combined with the other features in Eq. 6. Hence, for each such superpixel pair, we want to promote the weight w_i of the most prominent feature by computing w_i adaptively. This adaptive scheme comes from the intuition that the importance of a feature depends on how much of it is present. w_i is the mean feature value of v_a and v_b (connected by e_i), normalized by the maximum feature value for the entire video: $w_i = avg((f_k|v_a), (f_k|v_b))/\max(f_k)$. Specifically, the saturation value is used to measure how much color is present, while the intensity feature weight is 1 minus the weight of the color feature.

As a final step before labeling the edges, we compute the minimum spanning tree (MST) of the fully connected graph G before making the cuts. This step has been shown to reduce the under-segmentation of the graph by removing cycles and retaining only the most essential edges of the graph [33]. The MST is computed over the product of the edge weights for the n features, further multiplied by the distance $\mathbf{d}$ between the centroids of the neighboring superpixels, to ensure that closer neighbors are more likely to be under the same cluster. E' is the list of edges from the MST:

$$E' = MST(\mathbf{d} \prod_{k=1}^{n} \mathbf{x}_k \mid f_k) \quad (8)$$

Edge labeling is performed over E' according to Eq. 6.

2.4. Branch reductions

After removing the between-cluster edges that are identified by PGP, we obtain a set of disjoint superpixel clusters which are merged into separate video segments. However, a single frame slice of a spatio-temporal segment may result in several non-contiguous regions (branches) on that single frame. Although the branches may be the result of minor occlusion, they are undesirable in most cases and care should be taken to address this issue [18]. Therefore, we separate the branches of the spatio-temporal segments by post-processing using a greedy algorithm: we iterate through every spatio-temporal segment and check if it produces non-contiguous regions in any single video frame. If found, the algorithm picks the two largest regions on that frame and removes the edge with the highest weight along the path between the two regions. This step is repeated until all branches of the given spatio-temporal segment are removed, or a pre-defined number of iterations is reached . Figure 3 illustrates an example of this process.

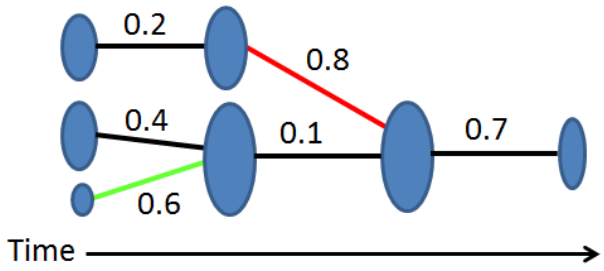

Figure 3. Example process of our greedy branch reduction method. Given the nodes (representing superpixels), the red edge is removed first, followed by the green edge.

3. Experiments and results

We evaluate our PGP video segmentation method quantitatively using two video datasets: Segtrack V2 [15] and Chen's Xiph.org [4]. The recently proposed Segtrack V2 is an extended version of the Segtrack V1 dataset [23], where the number of videos increased from 6 to 14. Videos vary in length and each has multiple objects and pixel-level ground-truth. Chen's dataset is a 8-video subset of the Xiph.org videos that contains semantic pixel-level labeling of multiple classes as ground-truth.

The features $\mathbf{f}$ that we use in this work are i) intensity (256-bin 1D histogram), ii) hue of the HSV colorspace (77×77 2D histogram), iii) the color component (AB) of the LAB colorspace (20×20 2D histogram), iv) motion optic flow (50×50 2D histogram), and v) gradient orientation (360-bin 1D histogram). Earth Mover's Distance (EMD) is used for all features. We generate superpixels using [16]. For the temporal neighbors' $n \times n$ search window size, we empirically set n to be 2.5% of the (video width + video height)/2; and cap the branch reduction iteration at 100 per spatio-temporal segment.

3.1. Feature properties

In order for the L_p distance statistics to follow a Weibull distribution, the compared feature differences must be correlated but non-identically distributed random variables, as mentioned in Section 2.2. We follow [1] to test the 5 features used in this paper.

In summary, for each feature, we take its feature histogram from a randomly selected reference superpixel s

	SegTrack V2																			
	3D Accuracy (AC)					3D Under-segmentation Error (UE)					3D Boundary Recall (BR)					3D Boundary Precision (BP)				
Video-obj	MLE+	NLS-	NLS+	[29]a	[29]b	MLE+	NLS-	NLS+	[29]a	[29]b	MLE+	NLS-	NLS+	[29]a	[29]b	MLE+	NLS-	NLS+	[29]a	[29]b
Bird of P.	93.2	96.3	93.5	**97.6**	95.3	3.1	2.9	3.0	**1.6**	2.6	85.4	91.7	87.8	**94.8**	91.6	10	6.7	**8.2**	6.9	5.8
Birdfall	67.3	**71.3**	70.7	65.3	53.8	81.5	50.9	44.4	28.8	**23.2**	83.6	84.7	83.8	**88.5**	82.1	**1.5**	0.8	1.1	0.8	0.9
BMX-1	**95.6**	95.0	**95.6**	90.3	65.6	7.1	7.1	6.9	**6.3**	7.7	97.3	97.4	97.5	**97.7**	93.6	4.7	4.2	4.5	**5.1**	3.6
BMX-2	78.2	78.9	**79.0**	44.3	27.1	9.4	**8.4**	10.0	11.7	16.8	90.6	91.5	91.0	**92.6**	88.1	4.3	3.9	4.1	**4.7**	3.3
Cheetah-1	72.8	**74.3**	73.9	0	39.4	9.0	**6.5**	9.5	47.4	34.1	93.4	**97.2**	94.7	63.8	75.3	1.1	1.1	1.1	1.4	**1.6**
Cheetah-2	69.9	**70.1**	69.7	0	12.0	8.7	**6.9**	8.9	54.5	34.4	98.4	**98.5**	98.4	76.8	75.3	1.5	1.4	1.5	**2.2**	2.0
Drift-1	92.4	92.2	**92.6**	83.0	75.6	6.7	**3.9**	6.0	3.5	7.1	92.9	93.6	93.5	**94.9**	90.6	**1.2**	1.0	1.1	1.0	**1.2**
Drift-2	91.9	**93.2**	92.8	84.9	56.8	7.5	**4.1**	7.3	4.2	10.0	91.3	**92.7**	91.6	87.6	82.9	**0.9**	0.8	**0.9**	0.7	0.8
Frog	14.3	33.5	25.6	n/a	**62.4**	16.5	15.8	15.7	n/a	**13.1**	29.5	55.1	44.6	n/a	**81.4**	**13.2**	8.7	7.1	n/a	1.7
Girl	87.2	**88.4**	87.9	65.5	63.5	9.9	10.3	**9.3**	12.3	13.5	90.3	91.7	**92.0**	75.6	83.2	4.3	**5.2**	4.1	**5.2**	4.6
Hbird-1	64.5	64.1	**69.5**	55.9	0	9.1	20.1	**8.8**	7.8	14.5	76.3	80.7	**86.8**	79.8	35.0	2.8	5.8	3.2	**6.3**	3.3
Hbird-2	78.4	67.5	**81.9**	70.6	0	**7.8**	11.2	7.9	8.0	13.4	90.4	83.3	**95.5**	92.7	86.0	5.0	9.0	5.3	**11**	12
Monkey	88.8	87.5	**90.7**	87.5	0	6.7	**3.5**	5.7	11.7	19.6	91.8	**94.3**	93.6	92.0	64.0	1.7	1.5	1.6	1.7	**3.4**
Mdog-1	86.7	**88.2**	88.1	41.7	79.9	12.6	**11.0**	11.6	41.4	43.2	94.8	**95.7**	**95.7**	94.3	91.0	1.6	1.5	1.5	2.1	**3.0**
Mdog-2	56.9	65.0	**66.7**	43.2	0	8.0	8.1	**6.5**	27.9	22.0	84.5	**88.5**	**90.0**	78.5	44.0	1.0	1.0	1.0	**1.3**	1.0
Parachute	**93.5**	93.4	93.4	90.9	89.3	22.0	**7.2**	19.9	18.5	38.6	94.9	**97.3**	96.0	95.7	87.4	1.3	0.7	1.1	**1.5**	10
Penguin-1	96.2	**96.4**	94.6	94.8	85.0	3.4	3.3	3.3	2.2	**1.8**	49.3	48.8	49.3	**77.3**	65.5	1.1	0.9	1.1	**1.4**	0.9
Penguin-2	96.4	96.5	**96.6**	85.1	93.1	4.7	4.6	4.7	3.3	**2.1**	69.6	**74.9**	71.7	66.8	75.3	**1.6**	1.5	**1.6**	1.3	1.1
Penguin-3	**96.2**	95.6	96.1	87.6	83.7	4.0	3.8	4.0	2.9	**2.5**	70.7	**74.2**	71.6	54.2	72.7	**1.6**	1.4	**1.6**	1.0	1.0
Penguin-4	96.0	95.8	**96.1**	83.8	82.8	3.9	3.9	3.8	**2.0**	2.4	73.4	**75.4**	74.0	45.7	56.7	**1.4**	1.2	**1.4**	0.7	0.7
Penguin-5	87.9	**89.3**	**89.3**	81.8	72.3	8.2	9.7	8.1	4.6	**4.1**	72.9	**76.0**	74.9	59.2	54.0	**1.2**	1.1	**1.2**	0.8	0.6
Penguin-6	96.7	**98.8**	97.0	85.7	86.8	4.1	4.1	4.0	**2.2**	2.5	61.1	47.1	66.2	**66.4**	63.3	1.2	0.8	**1.3**	1.1	0.8
Soldier	88.5	87.1	**89.5**	65.3	67.2	10.0	**4.9**	**4.9**	8.3	10.8	91.6	**94.4**	94.3	88.4	86.3	**3.2**	1.9	2.2	1.9	2.5
Worm	90.9	**92.4**	91.2	n/a	0	23.0	**20.3**	21.3	n/a	32.8	87.1	**90.3**	90.0	n/a	69.2	**1.9**	1.5	1.6	n/a	1.8
Average	82.5	83.8	**84.2**	68.4	53.8	12.0	**9.7**	9.8	14.2	15.5	81.7	84.0	**84.4**	80.2	74.8	**2.9**	2.6	2.5	2.7	2.8
Median	88.7	88.9	**89.6**	82.4	64.6	8.1	**7.0**	7.6	7.9	13.3	88.7	**90.9**	90.5	83.7	78.4	1.6	1.5	1.5	1.5	**1.8**

Table 1. Quantitative evaluation on the SegTrack v2 dataset, [29]a is UES+SWA, and [29]b is UES+GBH. For the 3D under-segmentation metric, the lower the error the better. For all the other metrics, the higher the score, the better. Best values are shown in **bold**. For the variants of our methods, the letters stand for the optimization method used, and +/- indicates the use of motion feature. All of the reported variants were based on 300 initial superpixels per frame at a 1/4 sub-volume processing mode. This table shows that our method's averages outperformed [29] in all 4 metrics, while our medians came out on top in 3. The n/a entry indicates that the method failed to converge to a result, the cases here were due to memory overload.

	Chen Xiph.org																			
	3D Accuracy (AC)					3D Under-segmentation Error (UE)					3D Boundary Recall (BR)					3D Boundary Precision (BP)				
Video	MLE+	NLS-	NLS+	[29]a	[29]b	MLE+	NLS-	NLS+	[29]a	[29]b	MLE+	NLS-	NLS+	[29]a	[29]b	MLE+	NLS-	NLS+	[29]a	[29]b
Bus	61.7	68.2	**69.3**	55.5	8.1	37.6	**6.8**	37.0	33.2	656	73.8	80.7	74.6	**84.5**	29.1	**41.5**	38.4	39.2	35.5	55.5
Container	82.2	89.8	**90.2**	89.2	77.3	11.6	3.6	7.5	**1.8**	4.5	58.6	**71.8**	69.4	64.7	51.5	**14.9**	8.3	**14.9**	9.9	11.6
Garden	82.7	**83.8**	83.3	**83.8**	63.1	1.8	1.9	1.8	**1.6**	3.3	69.9	73.6	70.6	**76.1**	40.0	13.9	13.4	13.9	12.1	**25.3**
Ice	**89.6**	87.6	87.5	79.4	46.6	29.3	26.0	27.1	**16.6**	69.1	83.3	**84.3**	83.0	80.6	50.3	33.8	31.4	32.4	36.1	**43.8**
Paris	49.5	**52.0**	50.9	47.6	2.0	14.7	**12.4**	14.3	19.2	40.2	46.4	**53.0**	50.8	44.8	34.1	4.7	4.4	4.6	3.8	**5.3**
Salesman	71.1	**82.2**	72.5	64.6	0	4.0	4.4	4.1	**3.4**	12.7	38.0	**51.0**	38.0	40.0	1.0	6.4	**7.1**	6.6	5.7	0.7
Soccer	72.9	**79.4**	78.3	70.9	26.4	46.6	32.7	34.0	**17.0**	145	73.3	**76.1**	74.9	75.2	58.8	16.7	14.4	16.3	16.3	**29.2**
Stefan	74.5	84.3	**85.4**	65.5	64.3	8.2	6.6	**6.3**	19.0	13.4	62.7	**82.2**	81.0	72.1	63.9	12.7	11.1	11.5	13.2	**13.4**
Avgerage	73.0	**78.4**	77.2	69.6	36.0	19.2	**11.8**	16.5	14.0	107	63.2	**71.6**	67.8	67.2	41.1	18.1	16.1	17.4	16.6	**23.1**
Median	73.7	**83.0**	80.1	68,2	36.5	13.2	**6.7**	10.9	16.8	26.8	66.3	**74.8**	72.6	73.7	45.2	14.4	12.3	14.4	12.7	**19.4**

Table 2. Quantitative evaluation on the Chen Xiph.org dataset. All of the settings are exactly the same as Table 1. For this dataset, both our average and median values outperformed [29] in 3 out of 4 metrics.

and 100 other randomly selected superpixels T, and compute the difference $\Delta_i = |s_i - T_i|^p$, at all histogram bins i. Then, we compute the Wilcoxon rank sum test between the difference values Δ_i and Δ_j of all possible pairs of indices, $i \neq j$. We set the confidence level at 0.05, and resample s and T 500 times per video, for all 5 features. As a result, we found that over 98% of the feature differences for all 5 features for both video datasets are non-identically distributed. This procedure is repeated for testing whether the feature differences are correlated, the second required assumption for satisfying the Weibull distribution property of L_p-norm based distance statistics. In this case we used Pearson's correlation, and we again found that over 98% of all feature differences for all 5 features are significantly correlated. This is not surprising as [1] showed that even for hand-designed-features such as SIFT, SPIN, and GLOH over 85% of the feature differences satisfy both conditions.

3.2. Quantitative evaluations

In the following, we compare our methods with the state-of-the-art related method on two datasets, and discuss the effects of the motion feature, number of superpixels extracted in pre-processing, sub-volume processing, the percentile parameter τ, and provide run time analysis.

Method Comparisons. Table 1 and 2 shows our quantitative results using the metrics proposed by [28] and com-

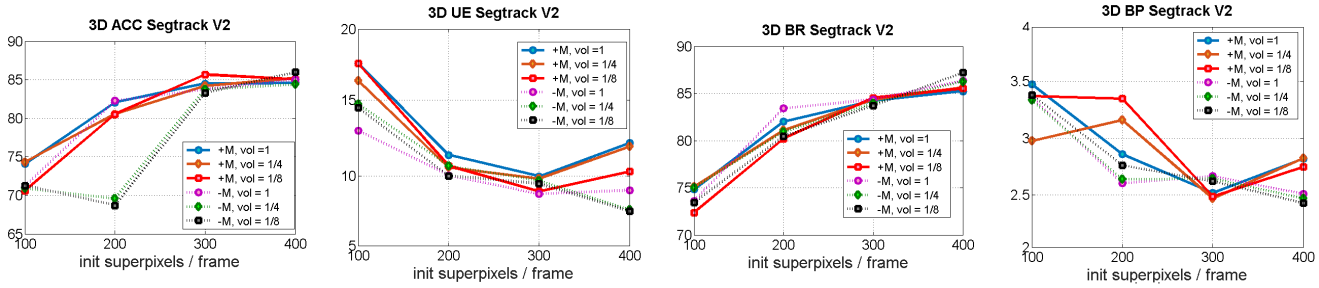

Figure 4. The plots that show the effects of varying different conditions (initial number of superpixels, size of the sub-volume processing, and with or without motion), for Segtrack V2 dataset: +/-M means with or without motion feature, vol stands for the sub-volume's size as a portion of the original video. These plots suggest that there is no clear indication of the benefit in using the motion feature (dotted lines) within the PGP framework, and that sub-volume processing performs approximately the same as processing the full volume.

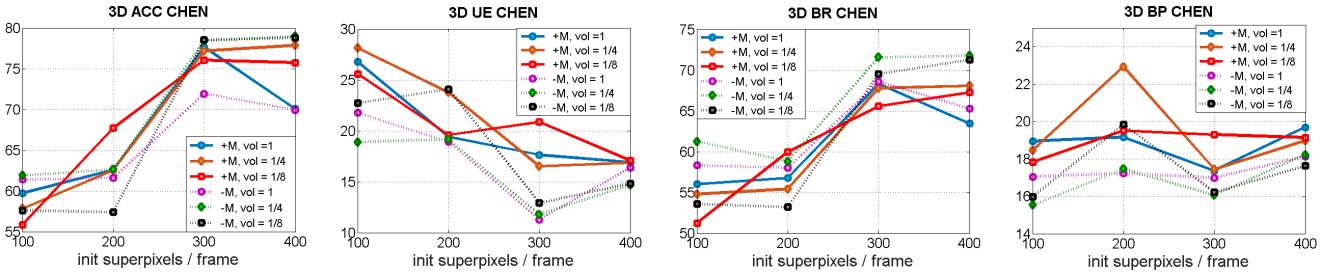

Figure 5. The plots show the effects of varying different conditions for Chen's dataset. Sub-volume processing seems to slightly outperform full video mode, and better performance tends to be associated with the exclusion of motion feature.

pared to the Uniform Entropy Slice (UES) method [29]. Like our method, UES also aims to produce just one single segmentation output, by automatically selecting and combining the appropriate supervoxels from the multiple layers of segmentations using SWA [7] or GBH [10]. We test several variants of our method: MLE+ for MLE optimized results, NLS+ and NLS- for optimization done using NLS, where +/- refers to the inclusion of motion feature or not, respectively. Due to space constraints, we report only one variant of MLE, since in our experiments MLE performed slightly but consistently worse than NLS optimization. The table shows that all the variants of our method outperform considerably both UES variants in all categories other than 3D BP in Chen's dataset. Their higher 3D BP is likely due to the significant overall under-segmentation of UES-GBH which heavily raises precision values.

Motion feature. When the motion feature is used, our algorithm uses the optical flow vectors for a more refined search of the temporal neighbors. Similar to the other features, we model a WMM over the similarity distance statistics based on motion feature histograms, and find the critical point γ that indicates the point of dissimilarity, which defines the inter-cluster values. In particular, motion feature similarity is considered only by spatial neighbors. The temporal neighbors just use motion vectors to specify the location of the search window. If the motion feature is not used, we search for temporal neighbors within a pre-defined window. We extract the motion feature using [22], but any optical flow extraction algorithm can be used. As both tables 1 and 2, and Figures 4 and 5 show, the results of using motion features within our framework are mixed, as motion seems to have improved the results for the SegTrack v2 dataset but not for Chen's dataset. Furthermore, not using motion lowers run time because optical flow extraction methods can be time consuming. Furthermore, Figure 6 shows that example results with (2nd row) and without (3rd row) motion are qualitatively similar.

Initial superpixel resolution. While our method produces a single segmentation result, the effects of varying the initial superpixel resolution are worth investigating. We evaluate our method with 100, 200, 300, and 400 initial superpixels per frame. Figures 4 and 5 show that a low number of superpixels tends to cause under-segmented results, most likely because the initial segmentation is less precise. The plots in Figures 4 and 5 also indicate that, after a point, increasing the superpixel resolution does not further improve results. Our method is robust to different initial superpixel resolutions as long as the number of superpixels is enough to produce a good initial over-segmentation (i.e. 300 per frame for the tested datasets). Figure 6 shows examples of the results that started with 300 (2nd and 3rd row) and 100 (4th row, somewhat under-segmented result) initial superpixels per frame.

Sub-volume processing. So far, we have been describing how our method processes the edge weights from the entire video. However, it can also be used to fit WMMs over sub-volumes and find the 'local critical values' that best describe the feature similarities at certain shots. The streaming GBH method [30] processes videos in chunks for efficiency, but loses information when optimizing only a group of frames at a time. In contrast, our PGP framework benefits from making the right divisions into sub-volumes for the WMM to optimize locally, as the feature similarities are more specific within a shot boundary. An example would be a change in activity: a triathlete is bike riding for the first 10 seconds of the video, followed by the swimming part of the contest for the next 10 seconds. Optimizing the entire video would effectively scramble the similarity distance values from both shots and result in a γ that is non-specific. However, if the video is processed at the two shots separately, the PGP method would obtain more specific, shot-appropriate γ's. Figures 4 and 5 show that optimizing at the entire video (vol = 1) is not always optimal, and better performance is achived when sub-dividing the volume for processing. Although we used a fixed set of subdivisions: 1, 1/4, and 1/8, processing smaller, appropriate sub-volumes is still beneficial. This would also allow for parallel video processing, where the sub-volumes can be optimized separately without performance cost.

Method parameter τ. Our proposed PGP video segmentation method has only one parameter value τ, used when a single Weibull model is selected by the model selection, to obtain the critical value γ at the τ percentile of the fitted Weibull. We have observed that the two-component WMM were selected by AIC in the vast majority of the cases; hence the selection of τ has a minimal effect on the overall accuracy. Indeed we varied τ from 0.5 to 0.9 and the resulting accuracies did not vary more than 1%. We have uniformly set τ at 0.6 throughout the presented evaluations.

Run time analysis. We conducted our experiments on a Xeon X3470 at 2.93Ghz with 32 Gb of memory. All experiments used a single core. Superpixels take roughly 1 second per frame, and our method on average takes about 170 seconds for an 85-frame video after superpixel extraction for the results of Table 2. This is more than 20 times faster than [29]'s processing time of around 4000 seconds; our combined end-to-end run time of 250 seconds on average is again 20 times faster than the total run time of 4700 - 6600 seconds of [29], which include the expensive GBH and SWA methods. Furthermore, our run time is on par with the leading streaming video segmentation method [30] although our current implementation is offline and not optimized for parallel processing.

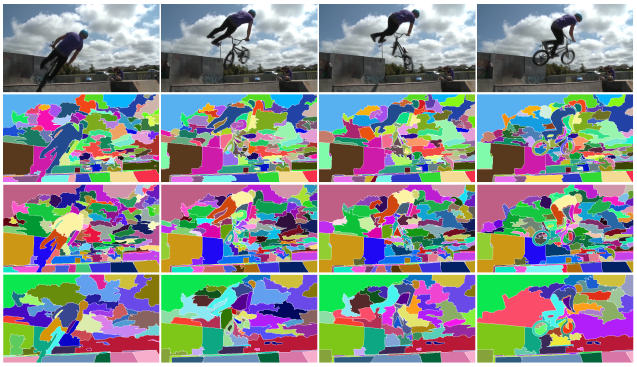

Figure 6. Example outputs for video BMX from SegTrack v2. From left to right: frame 1, 10, 20, and 30; Top to bottom: original frame, our results of 300 superpixels with motion (NLS+M in Table 1, 300 superixels without motion (NLS-M), and 100 superpixels without motion. Additional results can be found in the supplementary material.

4. Conclusion

We have proposed a fast and robust video segmentation method under PGP, a novel parametric graph partitioning framework. Our framework groups superpixels by modeling two-component mixtures of Weibull distributions over the edge weights, that permit low computational cost and robust inference on the parametric model (theoretically known to be the underlying structure of the L_p-norm based similarity distance statistics). We conducted extensive quantitative evaluations on the recently proposed SegTrack v2 and the well-known Chen Xiph.org dataset, and shown that our method significantly outperforms the related state-of-the-art method in most 3D metrics. We have also shown that our run time is on par with the state-of-the-art streaming methods, with the potential of parallelizing the bulk of the processing. Our framework is versatile and can be further improved with our sub-volume processing scheme. As a next step, we plan to investigate the application of shot-boundary techniques to explore the optimal sub-volume division that would further improve our method's performance.

5. Acknowledgments

This work was partially supported by NSF IIS-1161876, IIS-1111047, IIS-1161876, and the Subsample project from the DIGITEO In institute, France. We thank Matvey Genkin for his useful comments.

References

[1] G. J. Burghouts, A. W. M. Smeulders, and J.-M. Geusebroek. The distribution family of similarity distances. In *NIPS*, 2007.

[2] J. Chang, D. Wei, and J. Fisher. A video representation using temporal superpixels. In *CVPR*, 2013.

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

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

[5] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *ICCV*, 2011.

[6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI*, 2002.

[7] 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*, 2008.

[8] P. F. Felzenszwalb and H. D. P. Efficient graph-based image segmentation. In *ICCV*, 2004.

[9] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *ACCV*, 2012.

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

[11] Y. Huang, Q. Liu, and D. Metaxas.] video object segmentation by hypergraph cut. In *CVPR*, 2009.

[12] A. Ion, J. Carreira, and C. Sminchisescu. Image segmentation by figure-ground composition into maximal cliques. In *ICCV*, 2011.

[13] A. J. Izenman. Recent developments in nonparametric density estimation. *Journal of the American Statistical Association*, 1991.

[14] S. Kim, S. Nowozin, P. Kohli, and C. D. Yoo. Higher-order correlation clustering for image segmentation. In *NIPS*, 2011.

[15] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.

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

[17] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 1965.

[18] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, 2014.

[19] S. R. Rao, H. Mobahi, A. Y. Yang, S. Sastry, and Y. Ma. Natural image segmentation with adaptive texture and boundary encoding. In *ACCV*, 2009.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 2000.

[21] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 1983.

[22] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.

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

[24] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.

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

[26] T. Y. Vicente, C. Yu, and D. Samaras. Single image shadow detection using multiple cues in a supermodular mrf. In *BMVC*, 2013.

[27] J. Wang, Y. Jia, X.-S. Hua, C. Zhang, and L. Quan. Normalized tree partitioning for image segmentation. In *CVPR*, 2008.

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

[29] C. Xu, S. Whitt, and J. J. Corso. Flattening supervoxel hierarchies by the uniform entropy slice. In *ICCV*, 2013.

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

[31] A. Y. Yang, J. Wright, Y.Ma, and S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *CVIU*, 2008.

[32] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: joint object detection scene classification and semantic segmentation. In *CVPR*, 2012.

[33] C.-P. Yu, W.-Y. Hua, D. Samaras, and G. Zelinsky. Modeling clutter perception using parametric proto-object partitioning. In *NIPS*, 2013.