Unsupervised Semantic Scene Labeling for Streaming Data

Maggie Wigness and John G. Rogers III
U.S. Army Research Laboratory
maggie.b.wigness.civ@mail.mil, john.g.rogers59.civ@mail.mil

Abstract

We introduce an unsupervised semantic scene labeling approach that continuously learns and adapts semantic models discovered within a data stream. While closely related to unsupervised video segmentation, our algorithm is not designed to be an early video processing strategy that produces coherent over-segmentations, but instead, to directly learn higher-level semantic concepts. This is achieved with an ensemble-based approach, where each learner clusters data from a local window in the data stream. Overlapping local windows are processed and encoded in a graph structure to create a label mapping across windows and reconcile the labelings to reduce unsupervised learning noise. Additionally, we iteratively learn a merging threshold criteria from observed data similarities to automatically determine the number of learned labels without human provided parameters. Experiments show that our approach semantically labels video streams with a high degree of accuracy, and achieves a better balance of under and over-segmentation entropy than existing video segmentation algorithms given similar numbers of label outputs.

1. Introduction

Visual perception is critical in many applications that use scene semantics to help successfully perform tasks. Motivating examples include planning routes that avoid undesirable terrain and identifying goal landmarks in autonomous driving. Deep learning has helped push the state-of-the-art in visual classification [17, 30, 37] and semantic scene labeling [4, 7, 19] in recent years. These advances, in part, have been due to large sets of labeled data that required a tremendous amount of human annotation effort [5, 37].

However, the generalization of supervised learners across domains is still an open area of research. Even with millions of training images, the data distribution of a training set is unable to adequately represent all domains. Adapting requires additional training data, parameter tuning and/or re-training of at least part of the supervised learning system. This batch style training process inhibits on-line learning and discovery of novel concepts. In some real-world applications, the time latency introduced while humans label data to adapt visual classifiers has been addressed with semi-supervised [33] and self-supervised learning techniques [26, 27], but still require hours of labeling effort or are limited to binary classification tasks.

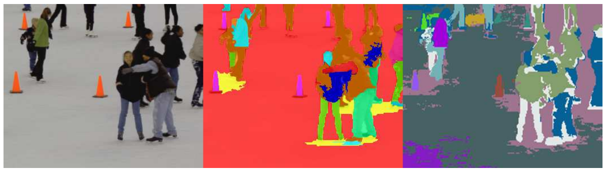


Figure 1. Comparison of segmentation output for our technique which directly models semantics without regard to locality, and hierarchical graph-based (GBH) segmentation which emphasizes segmentation coherency using strict locality rules. Our approach labels all four traffic cones consistently (column two), whereas GBH assigns unique labels to each cone (column three).

To further address the needs of these real-world applications, we introduce an unsupervised semantic scene labeling (USSL) technique. Similar to video segmentation, USSL localizes semantic concepts from a data stream without human intervention. However, most existing segmentation algorithms are used as an early pre-processing step to generate coherent, over-segmented regions in video [1, 18, 34]. In other words, the segmented output adheres to strict pixel connectivity in localized regions. Since our motivating applications seek semantic models for visual classification, USSL directly models the semantics in a scene without regard to locality. Figure 1 illustrates this difference, where the four traffic cones are assigned the same label by our USSL technique (second column), but assigned unique labels with a similar number of segments from the hierarchical graph-based segmentation [11] (column three).

Unsupervised learning can be error prone because of the lack of explicit guidance on which feature patterns to learn. Variations in illumination, perspective and occlusion are only some of the many challenges that impact feature patterns in data. To minimize variance of visual properties seen during processing, USSL learns on a local level, i.e., a small window of sequential frames from the data stream.

Learning is performed using agglomerative clustering with a merging threshold criteria unique to each window. Local models are learned for overlapping sliding windows to yield an ensemble of learners. The ensemble is encoded in a graph structure and used to reconcile unsupervised learning errors, and create a label mapping between the local windows to generate a global label set for the data stream.

We compare USSL with existing video segmentation algorithms to illustrate the uniqueness of directly modeling semantic concepts within the data stream, instead of simply outputting coherent connected pixels. Results show that USSL produces good label accuracy and better balances under and over-segmentation entropy. Further, our technique automatically determines the number of semantic labels to model since it iteratively learns a merging threshold from data similarities observed in previous frames of the stream.

2. Related Work

Semantic scene labeling has largely been addressed as a supervised learning problem. Semantic scene labeling of outdoor environments has been performed with CNNs [4, 7, 19], a forest of regression trees [23] and through combined learning of semantic classes and geometric classes [31]. He and Upcroft extend semantic scene labeling for 3-D environments [13], and techniques have been adapted for parsing and labeling cluttered indoor environments [4, 14]. The success of these supervised approaches comes at the high cost of collecting labeled training data for learning.

In many real-world applications, the ability to discover novel concepts on the fly or change domains with minimal visual perception interrupt is important. The autonomous robotics domain, in particular, has sought ways to learn semantic concepts with minimal human supervision. Techniques include evaluating structural change given environmental models [26], and pairing visual data with secondary data types such as contact sensor readings [15, 16], LiDAR [12, 27] or radar [21], which supply labels related to traversability automatically. However, these classifiers tend to only learn binary label models, e.g., *traversable*. These real-world applications and the need of multi-concept label sets are the primary motivation of our work.

Most similar to our approach is work done in label propagation and video segmentation. Label propagation is semi-supervised, where existing labels from a small set of images or video frames are propagated to other similar data. Jain and Grauman [6] introduced an active label propagation approach to obtain foreground/background masks for large image sets. Chen and Corso [3] learned weightings for motion and appearance models to propagate pixels labels throughout videos. Video segmentation assumes no a priori label information and techniques have used motion [2, 22], visual appearance features [9, 11] or a combination of both [29, 32] to partition out distinct concepts in the scene. There are two major drawbacks of these approaches. First, relying only on motion cues groups static background objects into a single class and does not capture complete scene context. Second, these techniques require the entire video to be loaded into memory for processing, which is incompatible with applications that provide a continuous stream of visual data. Stream-based alternatives to existing video segmentation models [11, 25] have been introduced, but show a significant degrade in performance relative to full video segmentation techniques [34]. Xu et al. [36] introduce a hierarchical streaming video segmentation approach, which processes non-overlapping sliding windows, and maintain label coherency throughout the stream by using segmentations and features from the previous sliding window when processing the current window.

The biggest difference between existing video segmentation approaches and our work in this paper is the desired output, and the process of determining the parameters to achieve this output. Existing video segmentation output is often highly over-segmented and not necessarily designed for applications seeking semantic models. Even techniques producing hierarchical output leave the hierarchical level selection to the user. Our work focuses on directly learning and modeling semantics without any human supervision.

3. Unsupervised Semantic Scene Labeling

We use unsupervised principles common to many segmentation algorithms, but seek a concise, i.e., minimally over-segmented, semantically labeled output similar to the goal of supervised semantic scene labelers. At a high level, our unsupervised semantic scene labeling (USSL) approach uses agglomerative clustering to iteratively create and adapt a set of semantic models as data flows in from the stream. Unlike many top performing segmentation algorithms and supervised semantic labelers, USSL learns the number of semantic labels without a priori specified parameters or knowledge of classes. This parameterless, bottom-up discovery allows our approach to easily model novel objects, terrain or other concepts throughout the stream.

While unsupervised learning has the advantage of not requiring labeled data, the lack of explicit direction regarding which feature patterns map to which semantics often results in noisy output. We use an ensemble-like approach and cluster over local windows to reduce visual variations seen over longer periods of time, which helps reduce some of the noise unsupervised learning may introduce. Like other streaming segmentation algorithms, local processing also avoids memory consumption issues. The ensemble results are encoded in a graph structure to generate a mapping to a global label set. Figure 2 illustrates the high level algorithm flow of USSL, and details of the approach are provided throughout the remainder of this section.

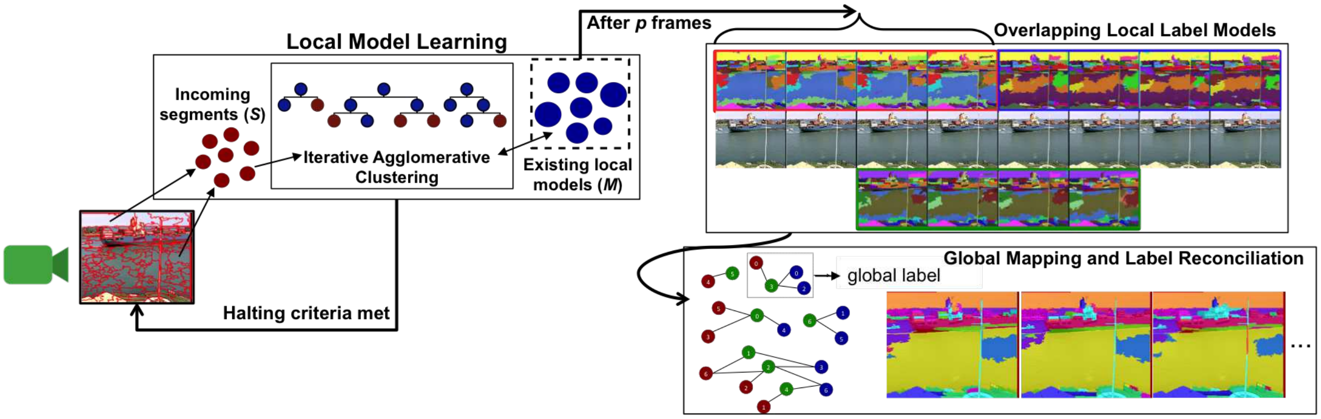

After p frames
Overlapping Local Label Models
Local Model Learning
Incoming segments (S)
Iterative Agglomerative Clustering
Existing local models (M)
Halting criteria met
Global Mapping and Label Reconciliation
global label
...

Figure 2. Overview of the unsupervised semantic scene labeling algorithm. The next image from the data stream is over-segmented, and segments are agglomeratively merged with existing models from previous frames in the stream. Overlapping local models are created for a window in the stream and these local label sets are mapped and reconciled using a graph encoding to generate a global label set.

3.1. Image Representation

Frames from the data stream enter the USSL system sequentially for processing. USSL performs scene segmentation starting with superpixels from incoming frames instead of individual pixels as superpixels provide more area to extract features important for semantic modeling. We use the graph based (GB) image segmentation [8] to generate the initial over-segmented superpixels that USSL cluster. The segmentation is run with parameters $\sigma = 0.5$, $K = 25$ and $min = 100$. The set of segments from an incoming frame is denoted as $S = \{s_1, s_2, \dots\}$.

Most image segmentation techniques rely on color and location features to identify coherent groupings of pixels. USSL uses additional features to help encode semantic information just as in many supervised approaches [23, 28]. Each s_i is represented by a LAB colorspace histogram comprised of 23 bins per channel, a 150 term codebook of SIFT descriptors [20], and Local Binary Patterns (LBP) [24] histograms created using 8 surrounding neighbors for neighborhoods of radii 1, 2 and 4. The three LAB channels, three LBP radii and the SIFT histograms are $L1$ normalized independently. These frame segments are then passed to the current instantiated local windows for processing.

3.2. Local Model Learning

USSL learns semantic models by agglomeratively clustering data from local windows in the stream. We refer to the set of groups output by the clustering algorithm for a local window as local label models, $M = \{m_1, m_2, \dots\}$. Each local window, W, consists of p consecutive frames and M is constructed and adapted iteratively for each incoming frame. The Local Model Learning box in Figure 2 illustrates this iterative clustering flow. Segments in S from a new frame (shown as red circles) enter the system and are agglomeratively clustered with existing local models in M (learned from previous frames in W). Existing local models are shown as blue circles in the figure, whose different sizes denote that local models represent varying volumes of W.

Much of the novelty and contribution of USSL's local learning technique comes from how similarity is evaluated to define merge and halting criteria during agglomerative clustering. Specifically, USSL evaluates similarity between two models, m_i and m_j, with respect to each of the histogram feature types described in Section 3.1. We denote feature type r of model m as f^r_m. Formally, similarity with respect to feature r is

$$\rho(m_i, m_j, r) = \frac{1.0}{1.0 + \sqrt{(f^r_{m_i} - f^r_{m_j})^2}}, \qquad (1)$$

which yields values on the range of $[0.0, 1.0]$. Feature types are evaluated individually with the idea that models with high similarity across all appearance feature types are most likely to represent the same semantic concept. Thus, restricting merging to these models will reduce the noise introduced by unsupervised learning. However, not all features are relevant for all semantic classes so USSL also evaluates the overall linear combination of feature similarities:

$$\phi(m_i, m_j) = \sum_{\forall r \in R} \rho(m_i, m_j, r). \qquad (2)$$

Overall scores are on the range of $[0, |R|]$, where R is the set of feature types.

A merging threshold is learned for each feature similarity score, which are used to define the agglomerative clustering halting criteria. The halting criteria allows USSL to automatically determine the number of local label models in W without user defined parameters. USSL maintains a distribution of similarity history, H, that includes similarity scores computed between segments in S and their nearest neighbors (NN). Since S is composed of over-segmented superpixels, most NNs should share the same label. Thus, H models the expected similarities observed between models that USSL would want to merge.

The similarity history distributions are updated at each new frame by finding the NN of each s_i with respect to both S and M. The NN of s_i from S and M are defined as:

$$N_S = \underset{m_j \in S, m_j \neq s_i}{\arg\max} \ \phi(s_i, m_j) \tag{3}$$

$$N_M = \underset{m_j \in M}{\arg\max} \, \phi(s_i, m_j). \tag{4}$$

The set of neighboring pairs for all of S is

$$N_t = \{N_S, N_M \,|\, \forall s_i \in S\}, \tag{5}$$

and the observed similarities to be added to H is

$$H_r = [\rho(m_i, m_j, r) \,|\, \forall (m_i, m_j) \in N_i] \tag{6}$$

$$H_o = \ [\phi(m_i, m_j) \,|\, \forall (m_i, m_j) \in N_i], \tag{7}$$

where similarity history is maintained per feature r and the overall combination of similarities. For all experiments in this paper, we compute the 3-NN with respect to S and M, when constructing H for USSL.

H is used to model the expected similarity between models that represent the same semantic concept. To account for noise in the unsupervised NN modeling, per-feature merging thresholds are defined by the mean and standard deviation of the H_r distribution,

$$\alpha_r = \mu_r - \sigma_r. \tag{8}$$

This definition of α_r models the observed similarities as a Gaussian distribution and assumes that the left tail representing one standard deviation below the mean are outlier values. While ideally models representing the same semantic concept would have high similarity across all feature types, USSL also accounts for feature irrelevance with a secondary merging threshold defined as the mean of distribution H_o, i.e., $\alpha_o = \mu_o$. This threshold is set higher with respect to the distribution statistics to ensure that most feature types are significantly strong to compensate for a feature similarity that falls below the α_r threshold.

Using these thresholds, a sign weighting β is defined for the similarity between models m_i and m_j, such that

$$\beta = \begin{cases} -1 & \phi(m_i, m_j) < \alpha_o \\ -1 & \rho(m_i, m_j, r) < \alpha_r \\ 1 & otherwise. \end{cases} \tag{9}$$

The β weight is applied to the overall similarity scores between models and indicates which model pairs meet the merging threshold criteria. When no pairs obtain a positive similarity score, agglomerative clustering algorithm halts.

Existing unsupervised segmentation algorithms use adjacency context to select which pixels to merge, which enforces pixel connectivity in the segmentation output. Adjacency is valuable information for USSL as well since any $s_i \in S$ in the same relative location are likely to be over-segmented regions of the same semantic label. An adjacency matrix, A, is maintained for models in M, where models m_i and m_j are adjacent, i.e., $A[m_i][m_j] = 1$, if they have adjacent pixels in the same frame or pixels with the same coordinates in adjacent frames (i.e., adjacent in time). Since our goal is to model semantic concepts directly, without regard to locality, a set of randomly selected non-adjacent models, T, are also evaluated as potential merging options. This allows the semantic models to grow fast locally since all adjacencies are evaluated, but also expand when a good non-adjacent merge is found. The model pair resulting in the greatest similarity score,

$$l^* = \max_{\forall m_i \in M, m_j \in A[m_i] \cup T} \beta * \phi(m_i, m_j), \tag{10}$$

is selected as the next merge. If l^* is negative, no model pairs met the merging threshold criteria and the agglomerative clustering halts. The system then begins to process the next frame. Experiments in this paper choose two random non-adjacent models for each m_i comparison.

3.3. Global Mapping and Label Reconciliation

We leverage an ensemble-like approach to create a global semantic labeling of the data stream from the local labeled windows. Specifically, a new local window W_i is created every $\frac{p}{2}$ frames so each local modeling processes a set of overlapping frames as its neighboring windows, W_{i-1} and W_{i+1}. The right half of Figure 2 illustrates the local window overlap. Three local windows outlined in red, green and blue can be seen above or below their p frames in the data stream. Region colors in the images represent a modeled semantic concept for that local window. Notice that concepts are over-learned at the local level, i.e., many colors map to the same ground truth concept. USSL uses the frame overlap between adjacent windows to map across local labelings, reconcile labeling errors and minimize over-learning to generate a global label set.

The idea behind this ensemble is as follows. Let M_1 and M_2 be sets of over-learned label models for the same window W. Assume that all $m_i \in M_1, m_j \in M_2$ represent models consisting of pixels from exactly one ground truth class, and that M_1 and M_2 do not have any identical label models, i.e., $m_i \neq m_j \, \forall m_i, m_j \in M_1, M_2$. Label models from M_1 and M_2 can be easily mapped with a graph-based encoding to generate labeled output with less over-segmentation. Let the graph, $G = (V, E)$, be constructed such that each $m_i, m_j \in V$, and $e(m_i, m_j) \in E$ if m_i and m_j have at least one common pixel in their modeled data.

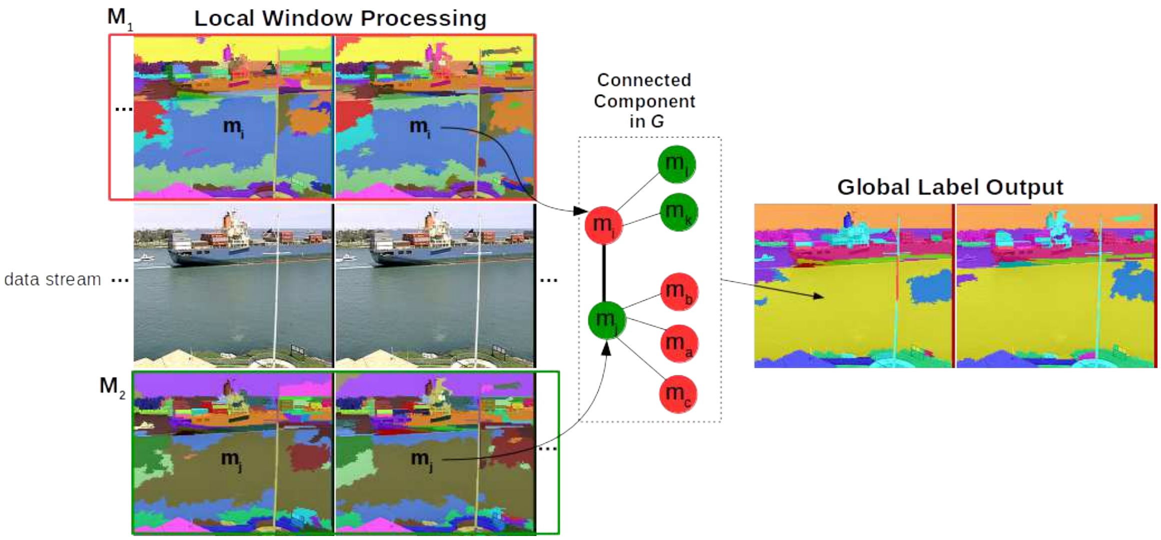


Figure 3. Illustration of mapping from a local model ensemble to a global semantic set. Local models that assign a label to the same pixels, e.g., $m_i \in M_1$ and $m_j \in M_2$, are encoded as edge connected vertices in a graph. Encoding all pixels overlaps in the ensemble generates connected components in the graph that represent models for the global label set.

Given this encoding, connected components in G represent label models from the ensemble of M_1 and M_2, and produce a label set with no worse over-segmentation than any of the two local models given our assumptions.

In practice, the ensemble of models produce connected components that can drastically reduce over-segmentation seen locally. Figure 3 illustrates a graph-based encoding and global label output produced from two overlapping windows using USSL on the xiph.org *container* video [3]. Local windows M_1 and M_2 are outlined in red and green, respectively. In this illustration, $p = 4$ and we only show the two frames where M_1 and M_2 overlap. We focus on local models $m_i \in M_1$ and $m_j \in M_2$, which both include pixels representing water in the video. Notice that with respect to pixel overlap, $m_i \cap m_j > 0$, which is encoded by $e(m_i, m_j)$ in G and shown in the illustration with the thickest graph edge. However, $m_i \cap m_j \neq m_i \cup m_j$ meaning m_i and m_j also overlap with other $m \in M_1, M_2$. These edges are also encoded in G, and a connected component representing the union of pixels from these overlapping models is formed. This aggregation of slightly different learned models in M_1 and M_2 results in a global labeling with less over-segmentation as indicated in the global output with larger areas of the water assigned the same "yellow" label.

G is constructed on-line, where vertices and edges are added after each W_i is processed. Unsupervised local models undoubtedly are noisy, so the raw connected component output includes any of this noise encoded in G. To reconcile some of the label noise from the ensemble of unsupervised learners, edges that provide minimal pixel overlap evidence are cut. An edge weight, w_e, is set to the number of intersecting pixels between its vertices. Each edge contributes a fraction of the total edge weight associated with one of its vertices. This fraction of weight is used to represent the label correspondence evidence. Let edge $e(v_i, v_j)$ link $v_i \in M_i$ and $v_j \in M_j$, then the evidence score relative to v_i is computed as

$$\epsilon_{v_i}(e(v_i, v_j)) = \frac{w_{e(v_i,v_j)}}{\sum_{\forall e_i \in E_{W_j}} w_{e_i}}, \tag{11}$$

where E_{W_j} is the set of edges connecting v_i and a label model from W_j (one of its adjacent, overlapping sliding windows). If $\epsilon(e, v_i) < \tau$ or $\epsilon(e, v_j) < \tau$ then the edge is cut. For our experiments, $\tau = 0.5$. Any connected component remaining in G believed to contain enough pixel information to adequately model a semantic concept (covering at least $.05\%$ of the video volume) is used to represent a global label model. Connected components that are too small are merged into their NN global model so every pixel in the stream has a global label in the final output.

4. Evaluation

We compare USSL to two segmentation algorithms implemented in the LIBSVX library [35]. The hierarchical graph-based (GBH) segmentation algorithm [11] was shown by Xu and Corso to be the most successful super-voxel segmentation in their comparison of five algorithms [34]. GBH is a hierarchical extension of GB segmentation [8]. The output is a set of hierarchical levels, where levels closer to the root contain fewer and coarser-grained segments. GBH was not originally designed to work on streaming data, and requires every frame to be loaded into memory simultaneously for processing. Stream GBH [36]

Table 1. Number of labels in the ground truth, and those found by USSL, GBH and S-GBH. GBH and S-GBH output is from the hierarchy level with a similar number of labels as USSL.

	$\|T\|$	# Segments : $\|S\|$			Hierarchy Level	
Video	GT	USSL	GBH	S-GBH	GBH	S-GBH
bus	10	17	19	25	20	12
container	7	24	24	31	19	12
garden	4	17	17	18	21	14
ice	4	11	15	11	19	14
soccer	6	19	20	24	18	13
stefan	5	13	15	12	19	15

(S-GBH) was introduced as an extension to GBH to process data streams of infinite length. Only a subset of frames from the stream are loaded into memory at any given time. In this respect, S-GBH is very similar to USSL. S-GBH also outputs a hierarchical set of segmentations. During evaluation, some metrics are computed across the entire hierarchical output of GBH and S-GBH, but we focus on comparisons using the hierarchical segmentation level that produces a similar number of labels as USSL.

In the remainder of this section we summarize the videos and segment output of USSL, GBH and S-GBH. We use three quantitative metrics to evaluate the segmentation/labeling traits of the compared methods. For these measures we define S as the set of segments or labels produced by an algorithm, where S_j indexes the jth label. T is the set of ground truth segments and V is the entire video. We denote the volume of a particular segment using the notation $|S_j|$ and the total video volume as $|V|$.

4.1. Dataset Overview

For quantitative evaluation of USSL, we use the dataset from Chen et al. [3], which comprises a subset of the xiph.org videos. Each video has been annotated with pixel-wise labels from 24 semantic classes (the same classes defined in the MSRC object dataset [28]). Although eight videos have ground truth labels, we only use the six (*bus, container, garden, ice, soccer* and *stefan*) that have a 50% majority of their pixels labeled. The average video length of this dataset is about 80 frames.

While GBH and S-GBH vary a set of parameters to produce a hierarchy of labeled output, USSL aims to directly discover the number of semantic labels in the data stream and produces a single labeled output. Table 1 summarizes the number of labels in the ground truth (GT), and output by USSL, GBH and S-GBH. We select labeled output of GBH and S-GBH from the hierarchical level (also shown in the table) that most closely matches the number of labels discovered by USSL. Notice that the hierarchical level selections for GBH and S-GBH are consistent across the videos. This suggests that USSL has learned a segmentation that maps to a particular range of parameters used in the hierarchical approaches.

Table 2. Comparison of average per-class and overall pixel-wise labeling accuracy for USSL and GBH variants.

	AVG-ACC			OVERALL-ACC		
Video	USSL	GBH	S-GBH	USSL	GBH	S-GBH
bus	0.294	**0.314**	0.137	0.401	**0.647**	0.370
container	0.613	0.491	**0.641**	**0.907**	0.786	0.855
garden	**0.638**	0.627	0.418	0.686	**0.689**	0.438
ice	**0.628**	0.524	0.534	**0.941**	0.898	0.870
soccer	**0.446**	0.426	0.438	**0.910**	0.876	0.892
stefan	0.544	**0.571**	0.541	0.841	**0.878**	0.837
Average	**0.527**	0.492	0.452	0.781	**0.796**	0.710

4.2. 3D Segmentation Accuracy

Xu and Corso [34] use 3D segmentation accuracy to compare several supervoxel segmentation techniques. Segmentation accuracy for T_i is defined as the fraction of pixels correctly classified by segments in S. Specifically, all $S_j \in S$ with a majority of pixels that overlap T_i make up $\bar{S}$. The total intersection of $S_j \in \bar{S}$ with T_i makes up the correctly classified fraction. Formally,

$$ACC(T_i) = \frac{\sum_{\bar{S}} |V_{T_i} \cap V_{\bar{S}_j}|}{|V_{T_i}|}. \qquad (12)$$

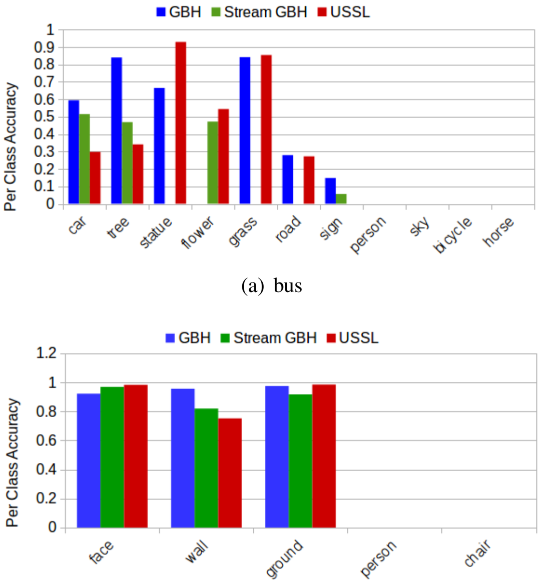
For each experiment we present the average accuracy of all T_i, which gives equal weight to all ground truth classes, and the overall accuracy which depicts the total number of correctly classified pixels:

$$AVG\text{-}ACC(V) = \frac{1}{t} \sum_{i=0}^{t} ACC(T_i) \qquad (13)$$

$$OVERALL\text{-}ACC(V) = \frac{|V_{T_i}| ACC(T_i)}{\sum |V_{T_i}|} \qquad (14)$$

Table 2 shows the average per class and overall accuracy achieved by the techniques with respect to the ground truth. USSL outperforms S-GBH in all but one accuracy measure (AVG-ACC for *container*), and most performance improvements are significantly large. Further, USSL performs comparably to GBH in terms of accuracy, yielding better average per class and overall accuracy in half of the videos. USSL's worst performance is seen on the *bus* video, which significantly drags down the dataset average. Omitting the *bus* video, USSL overall accuracy is better than GBH, outperforming it 0.857 to 0.825.

GBH does produce higher average and overall accuracy than USSL on some videos, but this does not correspond to being outperformed on every class in the video. Figure 4 shows the accuracy breakdown per class for these two videos, *bus* and *stefan*. Classes are shown in decreasing order of pixel frequency. The low accuracy achieved by USSL on the *car* and *tree* classes in the *bus* video (Figure 4(a)) account for most of its poor overall performance. However, USSL identifies the next four classes just as well or better



(a) bus

(b) stefan

Figure 4. Breakdown of the per-class classification accuracy for the *bus* and *stefan* video clips.

than GBH. Similarly, USSL achieves better segmentation for the *face* and *ground* classes in the *stefan* video.

4.3. Over-Segmentation and Under-Segmentation Entropy

Gong and Shi [10] define two conditional entropy measures to evaluate general image segmentation. The over-segmentation and under-segmentation conditional entropies show the trade-off between fine and coarser-grained segmentations, respectively. These measures are evaluated by overlaying the ground truth and segmentation output on top of one another. The set of probabilities used in the conditional entropy measures can be found by determining the volume of labels in T and S:

$$P(T=i) = \frac{|V_{T,i}|}{|V|} \tag{15}$$

$$P(S=j) = \frac{|V_{S,j}|}{|V|} \tag{16}$$

$$P(T=i, S=j) = \frac{|V_{T,i} \cap V_{S,j}|}{|V|} \tag{17}$$

Given these definitions, over-segmentation entropy (OSE) is found by overlaying S onto T

$$\mathcal{H}\{S|T\} = -\sum_{S,T} P(T,S) \log P(S|T), \tag{18}$$

where a more consistent mapping, i.e., there is one dominant S_j overlaid on T_i, produces a lower entropy measure.

Similarly, under-segmentation entropy (USE) is found by overlaying T onto S

$$\mathcal{H}\{T|S\} = -\sum_{S,T} P(T,S) \log P(T|S). \tag{19}$$

Figure 5 shows the balance between over-segmentation and under-segmentation entropy of each technique. Curves are drawn for each of the 30 hierarchical levels produced by GBH and S-GBH, but the level from Table 1 for each technique is outlined in red. Overall, the subfigures show a similar performance trend as was seen in the accuracy comparisons. USSL achieves lower conditional entropy measures than S-GBH for all but one video, and performs similarly or better than GBH in many of the videos.

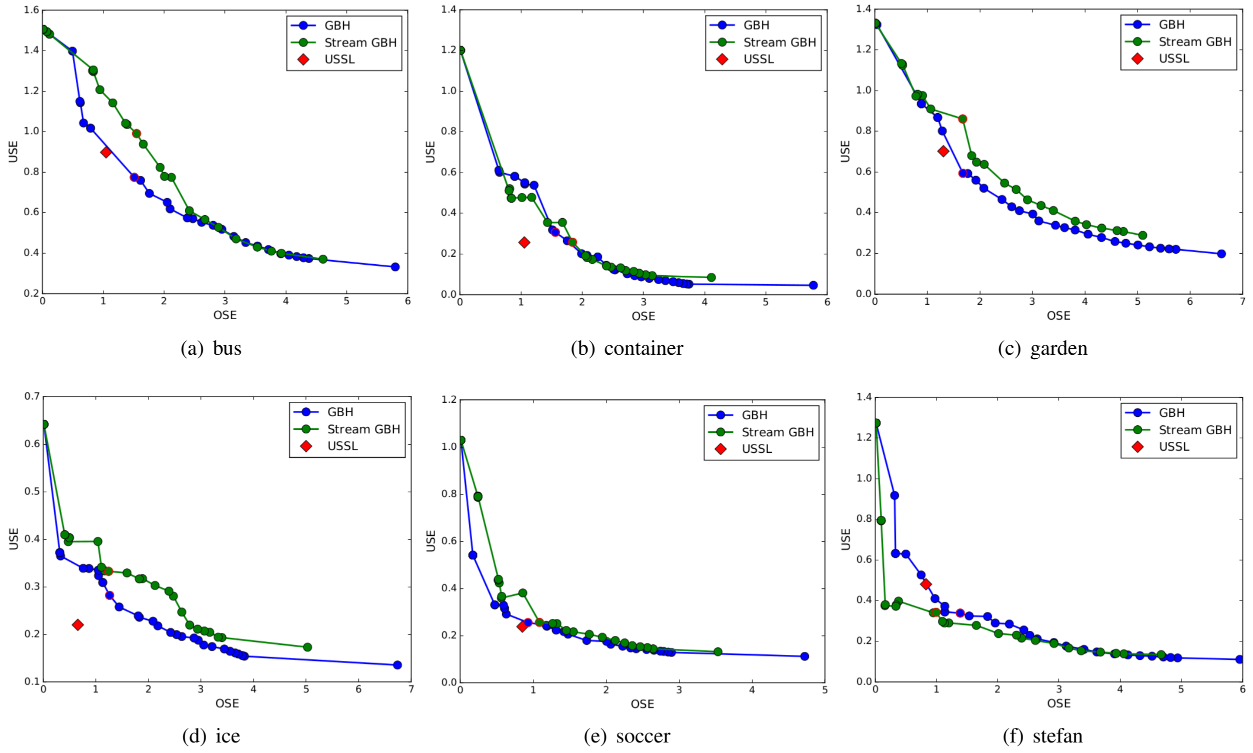
4.4. Qualitative Comparison

Figure 6 shows label output on frame 40 of the videos. This qualitative comparison shows a smoother labeling output by USSL. That is, USSL displays less over-segmentation, yielding larger areas of correctly labeled regions. These qualitative results also show many examples of USSL assigning the same semantic concept to disconnected pixels in the video. In addition to the cone example discussed earlier (seen in column four), disconnected segments are found for humans in the *soccer* and *stefan* videos and areas of water (colored in blue) in the *container* video.

The qualitative images also reiterate that USSL had a number of challenges with the *bus* video. We hypothesize this is due to high occlusion of objects and the reflective and transparent properties of windows in the scene. The fence occludes the vehicles and creates noisy features for these regions even though it in fact is not part of these objects. Similarly, features from other objects may be associated with the vehicles in the scene because they are visible through the transparent window or can be seen in the reflection of the glass. Thus, USSL incorrectly assigns the same label to most of the trees and vehicles.

5. Conclusion

We introduced a variation on video segmentation that focuses on directly learning higher-level semantic concepts in streaming data without human annotation. Our unsupervised semantic labeling approach analyzes underlying patterns in local windows of a video stream, while avoiding strict locality modeling to ensure disconnected regions of the same semantic are modeled together. By over-learning locally, noise introduced by unsupervised learning can be minimized, and remaining errors can be reconciled using an ensemble of local learners encoded in a graph-based structure. This approach balances under and over-segmentation entropy better than existing video segmentation algorithms, while automatically determining the number of semantic labels without human provided parameters.



(a) bus

(b) container

(c) garden

(d) ice

(e) soccer

(f) stefan

Figure 5. Comparison of the under-segmentation versus over-segmentation entropies for the six xiph.org videos. All 30 levels of the hierarchical output for GBH and S-GBH are plotted to compare against the single output of USSL. The GBH and S-GBH hierarchical levels that produce about the same number of semantic labels as USSL (also listed in Table 1) is outlined in red.

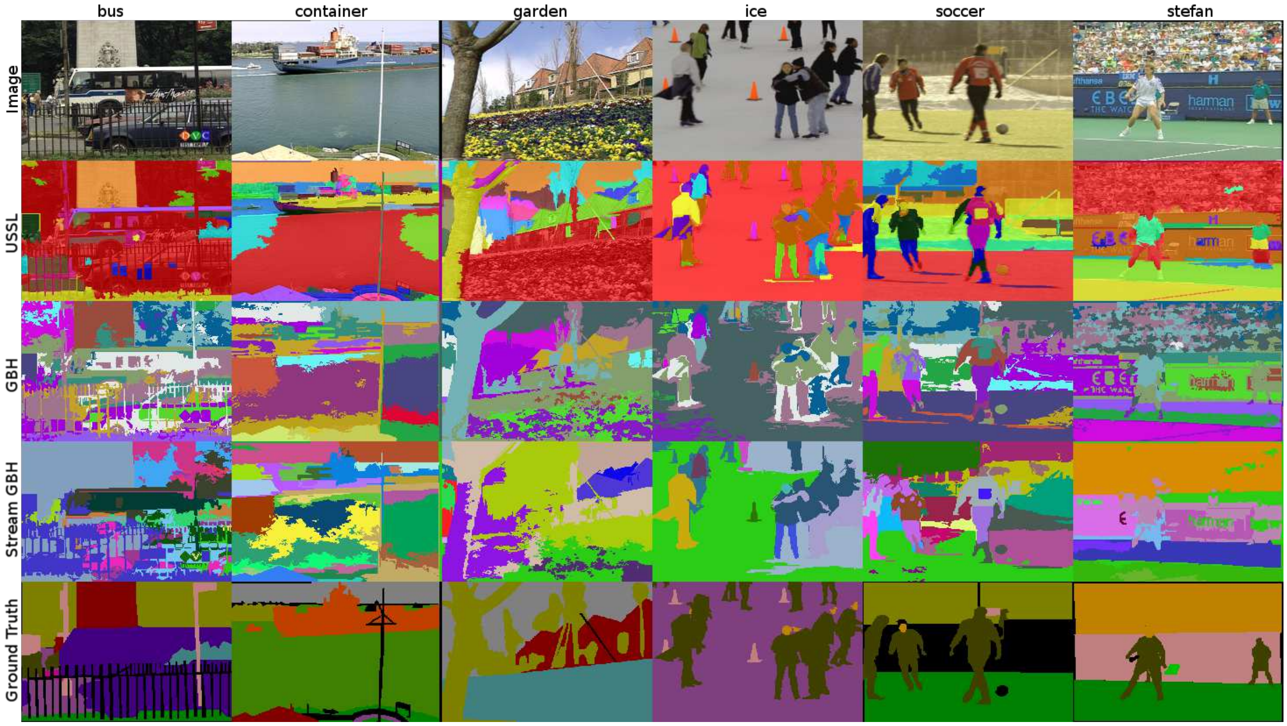


Figure 6. Qualitative comparison of output on the xiph.org dataset. The output is from frame 40 of each video, which roughly corresponds to the middle frame of each video clip.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012. 1

[2] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *Proceedings of the European conference on computer vision*, pages 282–295. Springer, 2010. 2

[3] A. Y. Chen and J. J. Corso. Propagating multi-class pixel labels throughout video frames. In *Proceedings of the Western New York Image Processing Workshop*, pages 14–17. IEEE, 2010. 2, 5, 6

[4] C. Couprie, C. Farabet, L. Najman, and Y. Lecun. Convolutional nets and watershed cuts for real-time semantic labeling of rgbd videos. *The Journal of Machine Learning Research*, 15(1):3489–3511, 2014. 1, 2

[5] J. Deng, W. Dong, L.-J. Socher, Richard Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 2009. 1

[6] S. Dutt Jain and K. Grauman. Active image segmentation propagation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2864–2873. IEEE, 2016. 2

[7] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013. 1, 2

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

[9] C. Fowlkes, S. Belongie, and J. Malik. Efficient spatiotemporal grouping using the nystrom method. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–231. IEEE, 2001. 2

[10] H. Gong and J. Shi. Conditional entropies as over-segmentation and under-segmentation metrics for multi-part image segmentation. Technical Report MS-CIS-11-17, University of Pennsylvania, 2011. 7

[11] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2141–2148. IEEE, 2010. 1, 2, 5

[12] M. Häselich, M. Arends, N. Wojke, F. Neuhaus, and D. Paulus. Probabilistic terrain classification in unstructured environments. *Robotics and Autonomous Systems*, 61(10):1051–1059, 2013. 2

[13] H. He and B. Upcroft. Nonparametric semantic segmentation for 3d street scenes. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 3697–3703. IEEE, 2013. 2

[14] S. Hickson, S. Birchfield, I. Essa, and H. Christensen. Efficient hierarchical graph-based segmentation of rgbd videos. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 344–351. IEEE, 2014. 2

[15] M. Hoffmann, K. Štěpánová, and M. Reinstein. The effect of motor action and different sensory modalities on terrain classification in a quadruped robot running with multiple gaits. *Robotics and Autonomous Systems*, 62(12):1790–1798, 2014. 2

[16] D. Kim, J. Sun, S. M. Oh, J. M. Rehg, and A. F. Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In *Proceedings of the International Conference on Robotics and Automation*, pages 518–525. IEEE, 2006. 2

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[18] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2097–2104. IEEE, 2011. 1

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3431–3440. IEEE, 2015. 1, 2

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3

[21] A. Milella, G. Reina, and J. Underwood. A self-learning framework for statistical ground classification using radar and monocular vision. *Journal of Field Robotics*, 32(1):20–41, 2015. 2

[22] Q. Mo and B. A. Draper. Semi-nonnegative matrix factorization for motion segmentation with missing data. In *Proceedings of the European Conference on Computer Vision*, pages 402–415. Springer, 2012. 2

[23] D. Munoz, J. A. Bagnell, and M. Hebert. Stacked hierarchical labeling. In *Proceedings of the European Conference on Computer Vision*, pages 57–70. Springer, 2010. 2, 3

[24] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996. 3

[25] S. Paris. Edge-preserving smoothing and mean-shift segmentation of video streams. *Proceedings of the European Conference on Computer Vision*, pages 460–473, 2008. 2

[26] P. Ross, A. English, D. Ball, B. Upcroft, and P. Corke. Online novelty-based visual obstacle detection for field robotics. In *Proceedings of the International Conference on Robotics and Automation*, pages 3935–3940. IEEE, 2015. 1, 2

[27] M. Shneier, T. Chang, T. Hong, W. Shackleford, R. Bostelman, and J. S. Albus. Learning traversability models for autonomous mobile vehicles. *Autonomous Robots*, 24(1):69–86, 2008. 1, 2

[28] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 1–15. Springer, 2006. 3, 6

[29] H. S. Sokeh and S. Gould. Towards unsupervised semantic segmentation of street scenes from motion cues. In *Proceedings of the Conference on Image and Vision Computing New Zealand*, pages 232–237. ACM, 2012. 2

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of Computer Vision and Pattern Recognition*. IEEE, June 2015. 1

[31] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *Proceedings of the Europen Conference on Computer Vision*, pages 352–365. Springer, 2010. 2

[32] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3899–3908, 2016. 2

[33] M. Wigness, J. G. Rogers, L. E. Navarro-Serment, A. Suppe, and B. A. Draper. Reducing adaptation latency for multi-concept visual perception in outdoor environments. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 2784–2791. IEEE, 2016. 1

[34] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1202–1209. IEEE, 2012. 1, 2, 5, 6

[35] C. Xu and J. J. Corso. LIBSVX: A supervoxel library and benchmark for early video processing. *International Journal of Computer Vision*, 119:272–290, 2016. 5

[36] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 626–639. Springer, 2012. 2, 5

[37] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 1