A Video Representation Using Temporal Superpixels

Jason Chang
CSAIL, MIT
jchang7@csail.mit.edu

Donglai Wei
CSAIL, MIT
donglai@csail.mit.edu

John W. Fisher III
CSAIL, MIT
fisher@csail.mit.edu

Abstract

We develop a generative probabilistic model for temporally consistent superpixels in video sequences. In contrast to supervoxel methods, object parts in different frames are tracked by the same temporal superpixel. We explicitly model flow between frames with a bilateral Gaussian process and use this information to propagate superpixels in an online fashion. We consider four novel metrics to quantify performance of a temporal superpixel representation and demonstrate superior performance when compared to supervoxel methods.

1. Introduction

Since their inception in the work of Ren and Malik [18], superpixels have become an important preprocessing step in many vision systems (e.g. [7, 8, 18]). Though many algorithms operate at the pixel level it is generally more efficient to process higher-level representations. For example, one can reduce the hundreds of thousands of pixels to hundreds (or thousands) of superpixels while still maintaining very accurate boundaries of objects.

Many video analysis applications begin by inferring temporal correspondences across frames. This can be done for a sparse number of locations with point trackers (e.g. [13]) or over the dense pixels with optical flow (e.g. [10]). For example, structure from motion typically tracks features to solve the correspondence of points within two frames and video segmentation algorithms (e.g. [8]) often use optical flow to relate segments between two frames. More recently, [5] and [19] use optical flow to obtain robust long-term point trajectories. By clustering these points based on motion, one can obtain a sparse video segmentation [4]. Furthermore, [16] develops a method to extend the sparse segmentation in a single frame to a dense segmentation with the help of superpixels.

Inspired by these previous methods, our primary focus is to develop a representation for videos that parallels the superpixel representation in images. We call these new elementary components, temporal superpixels (TSPs). Unlike

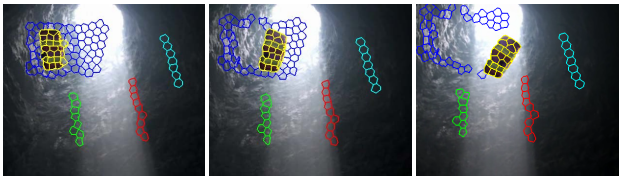


Figure 1: Example of TSPs obtained using our model. Notice that the same TSPs across the frames track the same points on the parachute and rock. While only a subset of TSPs are shown, each frame is entirely segmented. Coloring of TSPs is done by hand to illustrate correspondence.

[4], applying a motion-based clustering algorithm on TSPs instead of points directly yields a dense video segmentation without using [16].

In the seminal work of [18], Ren and Malik define a superpixel as a set of pixels that are "local, coherent, and which preserve most of the structure necessary for segmentation". We similarly define a TSP as a set of video pixels that are local in space and track the same part of an object across time. Consequently, intra-frame TSPs should represent a superpixel segmentation of the frame. Example TSPs are shown in Figure 1. We believe that the TSP representation bridges the gap between superpixels and videos.

A temporal superpixels are closely related to supervoxels. TSPs, however, are tailored to video data, whereas supervoxels are designed for 3D volumetric data. Supervoxel methods [24] process videos by treating the time dimension as a spatial dimension. While this may work well for actual volumetric data (e.g. in medical imaging), it does not represent videos well. For example, though one often assumes that each voxel in a volume is closely related to its 26 3D neighbors, the same relationship does not exist in videos with non-negligible motion. In fact, objects that are moving quickly may not even overlap in adjacent frames of a video.

Treating the temporal dimension differently is certainly not a new idea in the literature. Optical flow is commonly used as a measurement to aid object tracking or segmentation. For example, the work of [8] connects nodes in a 3D graph along flow vectors to produce oversegmentations of the video. As we discuss in Section 2, however, these over-

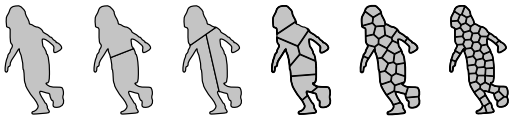


Figure 2: Example hierarchy of oversegmentations, ranging from object segmentation (left) to superpixels (right).

segmentations can violate the locality assumption of superpixels. Lastly, our work differs from [8] by explicitly modeling flow as a latent variable and inferring its value.

This work is also closely related to [27], which considers jointly estimating an oversegmentation and flow. The key differences are that [27] does not explicitly handle occlusions/disocclusions, enforces a very limiting temporal smoothness term, and must process an entire volume of data at once. While preliminary results in [27] look promising, we believe these three factors limit the method from performing well on different videos or long sequences.

We believe that this new type of representation has utility in many computer vision tasks. Various algorithms (e.g. [23]) run off-the-shelf superpixel algorithms independently on frames of a video, producing superpixels that are unrelated across time. Following this, complex methods are utilized to solve for correspondences between two independently segmented frames. By using a slightly richer representation, a TSP representation can mitigate the complexity of many segmentation and tracking algorithms. Additionally, to our knowledge, there has yet to be any generative, probabilistic models to represent superpixels. Many commonly used superpixel algorithms (e.g. [7, 15, 18, 26]) formulate the superpixel problem on an affinity graph and solve the spectral clustering problem with graph cuts. Other algorithms (e.g. [1, 11]) formulate the problem as an energy minimization problem and perform gradient descent to find a solution. As we shall see, the proposed generative model of TSPs in videos reduces to a generative superpixel model in a single frame.

In the sequel, we develop a model of TSPs. In contrast to supervoxel methods ([25] being a notable exception) the proposed TSP approach infers superpixels using only past and current frames and thus scales linearly with video length. We consider novel metrics that evaluate desired traits of TSPs, and quantitatively show that our method outperforms the supervoxel methods presented in [24].

2. TSPs vs. Oversegmentations

We begin with a more precise description of a temporal superpixel. Superpixel representations reduce the number of variables by orders of magnitude while often maintaining the most salient features of an image. For example, an image can be approximated by setting each superpixel to a constant color. We desire to have the same representative power with TSPs. For example, in videos, one may want to approximate the flow between frames with a constant translation for each superpixel, as is done in [27]. Intra-frame TSPs should therefore form a superpixel segmentation. Additionally, as is commonly enforced with superpixels, the spatial support of a TSP in any frame should be a single connected component.

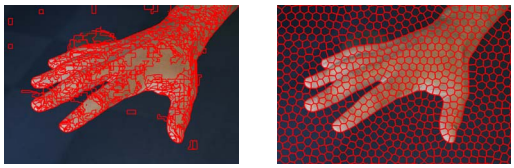


Figure 3: An example of using [8] (left) versus the TSP method (right). Both algorithms have approximately 700 segments. Notice that the majority of the background on the left is grouped into a single segment whereas the TSPs evenly divide up the image into local segments.

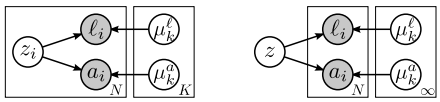
As shown in Figure 2, there are many valid oversegmentations for a given image. While the terms "oversegmentation" and "superpixel segmentation" are commonly used interchangeably, we suggest that they are quite different. A superpixel segmentation is an oversegmentation that preserves the salient features of a pixel-based representation. Figure 3 shows an example of the finest oversegmentation produced using [8] compared with a TSP representation at the same granularity. This example illustrates the difference between oversegmentations and superpixels. Describing the motion of each small superpixel as a translation may be a suitable approximation, but doing so for the entire background may introduce much larger errors.

3. Initializing TSPs

In this section, we discuss the inference of TSPs for the first frame. Because intra-frame TSPs are superpixels, we will often refer to these terms interchangeably. Given the vast literature on superpixel methods, we focus on those that most closely relate to the proposed method. The recent work of [1] presents an extremely fast superpixel algorithm called Simple Linear Iterative Clustering (SLIC). As shown by the authors, SLIC rivals other state-of-the-art superpixel techniques in preserving boundaries on the Berkeley Segmentation Dataset [14] while achieving orders of magnitude in speed gains. After briefly reviewing SLIC, we present the TSP model for a single frame and extend it to multiple frames in Section 4.

3.1. Simple Linear Iterative Clustering

SLIC assigns a 5-dimensional feature vector to each pixel, composed of x- and y-location, and the three components of the *Lab* colorspace. The algorithm consists of two steps: (1) perform k-means clustering for a fixed number



(a) SLIC (b) TSP (single frame)

Figure 4: Graphical models for SLIC [1] and TSPs.

of iterations; and (2) eliminate single-pixel superpixels and enforce that each superpixel is a single 4-connected region. While not explicitly stated in [1], the first step of SLIC can be formulated as a Gaussian mixture model with the graphical model in Figure 4a, where z_i is the associated cluster label assigned to pixel i, a_i is the *Lab* color, ℓ_i is the location, and μ_k^a and μ_k^ℓ are the mean parameters for cluster k. We note that the second step of SLIC is not easily represented as inference in a graphical model.

3.2. A Generative Model for Superpixels

Utilizing digital topology concepts, we formulate a similar model without the need for post-processing. Specifically, if a single pixel changes, the topology of the binary object changes if and only if that pixel is not a *simple point*. Checking if a pixel is a simple point can be done in constant time [3], allowing efficient incorporation into inference procedures [9]. If a binary object does not change in topology, the connectivity also does not change, making it directly applicable to superpixels. Using these concepts, we restrict the distribution over superpixel labels such that each unique label must be a single 4-connected region. Any configuration that does not satisfy this topology is assigned zero probability. Further details can be found in the supplement.

Additionally, the proposed method incorporates a penalty on the number of superpixels. As we show in Section 4, new superpixels will have to be created to explain new objects and disocclusions. If new superpixels can be created, the optimal configuration in the absence of such a penalty is the set of single-pixel superpixels, each with a mean centered at the data of that pixel. We place a simple geometric distribution on the number of superpixels, resulting in the following distribution over labels

$$p(z) \propto \hat{\alpha}^K \mathbb{1}\{T(z)\}, \tag{1}$$

where $\hat{\alpha}$ is a hyper-parameter controlling coarseness, K is the number of unique values in z, and $\mathbb{1}\{T(z)\}$ is one iff z is a valid topology. To avoid confusion, we note that $\hat{\alpha}^K$ is proportional to the typical geometric distribution, $(1-\alpha')^K\alpha'$, for some α'. Finally, we model the cluster means as being drawn from a uniform distribution

$$p(\mu_k^\ell) = {}^1\!/_N \quad , \quad p(\mu_k^a) = ({}^1\!/_{256})^3, \tag{2}$$

where N is the number of pixels, and each color channel is assumed to only take on values in $[0, 255]$. The Gaussian observation likelihood can then be expressed as

$$p(x_i|z_i = k, \mu) = \prod\nolimits_d \mathcal{N}(x_{i,d}\ ;\ \mu_{k,d}\ ,\ \sigma_d^2) \tag{3}$$

where $x_i = [\ell_i, a_i]$ is the 5-dimensional feature vector, d indexes a dimension, and σ_d^2 are hyper-parameters. The corresponding graphical model is shown in Figure 4b.

3.3. Inference

We now derive an optimization method from the generative model. The main departure from traditional Gaussian mixtures is the joint distribution over z. We could alternate between label and parameter optimization similar to k-means. However, as shown in the supplement, joint optimization of labels and parameters finds better extrema. We optimize the hidden variables by proposing a set of label "moves" and choose the optimal parameter associated with proposal. By only accepting moves that increase the probability, we are guaranteed to find a local optimum within our search space. The joint log likelihood over all random variables can be expressed as

$$\begin{aligned}\mathcal{L}(z) &\overset{C}{=} \log\Big[p(z)\prod\nolimits_k\prod\nolimits_d p(\mu_{k,d})\prod\nolimits_i p(x_{i,d}|z_i,\mu_d)\Big] \\ &\overset{T}{=} \alpha K + \sum\nolimits_k\sum\nolimits_d \log p(x_{\mathcal{I}_k,d}, \mu_{k,d}),\end{aligned} \tag{4}$$

where $\overset{C}{=}$ denotes equality up to an additive constant, $\overset{T}{=}$ assumes that the following configuration of z is a valid topology, some constants have been combined to form α, and $\mathcal{I}_k$ denotes the set of pixels in superpixel k. Note that the log likelihood is only defined as a function of z because, for each z, the optimal parameters, μ, will be implicitly found. We denote the following relevant superpixel statistics

$$t_{k,d} = \sum\nolimits_{i\in\mathcal{I}_k} x_{d,i}\ ,\ T_{k,d} = \sum\nolimits_{i\in\mathcal{I}_k} x_{d,i}^2. \tag{5}$$

Although not explicit, these statistics are functions of z through $\mathcal{I}_k$. The uniform prior in Equation 2 yields empirical means as the optimal parameters. Denoting $N_k = |\mathcal{I}_k|$, the optimal parameters can be expressed as

$$\hat{\mu}_{k,d}(z) = {}^{t_{k,d}}\!/_{N_k}. \tag{6}$$

We define the observation log likelihood for superpixel k as

$$\mathcal{L}_n(x_{\mathcal{I}_k,d}) \triangleq \log p(x_{\mathcal{I}_k,d}, \hat{\mu}_{k,d}) \tag{7}$$

$$\overset{C}{=} -N_k \log\sigma_d + \tfrac{t_{k,d}^2 - N_k T_{k,d}}{2N_k\sigma_d^2} \tag{8}$$

The log likelihood of Equation 4 can then be expressed as

$$\mathcal{L}(z) \overset{C,T}{=} \alpha K - \sum\nolimits_k\sum\nolimits_d \mathcal{L}_n(x_{\mathcal{I}_k,d}). \tag{9}$$

Additional details can be found in the supplement. We now describe the three types of proposed label moves to change z: local moves, merge moves, and split moves. If any of proposed moves increase the log likelihood, we make the move to a more optimal configuration. The algorithm converges when all proposed moves are rejected.

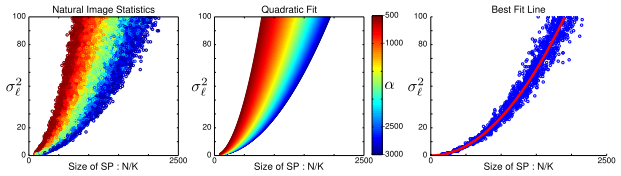


Figure 5: Fitting hyper-parameters to natural images.

Local Moves: change the label of a single pixel. Because of the topology constraints, only pixels bordering another superpixel can change. A random border pixel i is chosen, and among all the labels that preserve the topology constraint, we find the labeling that corresponds to the highest joint likelihood.

Merge Moves: combine two superpixels by observing that merging two neighboring 4-connected regions still results in a 4-connected region. A random superpixel is chosen, and the largest likelihood for merging with any of the neighboring superpixel is found. The merge is accepted if it increases the likelihood.

Split Moves: split a single superpixel into two. A split is constructed by running k-means on a random superpixel followed by enforcing connectivity similar to SLIC. The split is accepted if it increases the likelihood.

3.4. Hyper-parameter Selection

There are three hyper-parameters to set in the generative model: $\alpha, \sigma_\ell^2, \sigma_a^2$. As the desired level of coarseness may be driven by external factors, we require one user-specified parameter, M, capturing the approximate number of desired superpixels. We now discuss how to set the other parameters automatically.

The ratio between σ_a^2 and σ_ℓ^2 determines the regularity of superpixel shapes. As superpixels get larger, they will also have to be more irregularly shaped to capture boundaries. We found empirically that $\sigma_a^2 = M\sigma_\ell^2/2$ produces nicely-shaped superpixels. The model selection term, α, and the location variance, σ_ℓ^2, are related to the size of the resulting superpixels and are learned based on natural image statistics. The inference algorithm was run on randomly sized patches from the Berkeley Dataset [14] for particular values of α and σ_ℓ^2. The resulting mean superpixel area (N/K) is shown in Figure 5. Here, K refers to the number of superpixels produced by the algorithm, and M is the number of desired superpixels. The best quadratic fit relating the parameters is then found to be

$$\sigma_\ell^2 = -\left[N/M\right]^2 \div (12.5123\alpha). \tag{10}$$

The variances can then be automatically set based on the desired number of superpixels and some arbitrary α. Example superpixel segmentation results are shown in Figure 6.

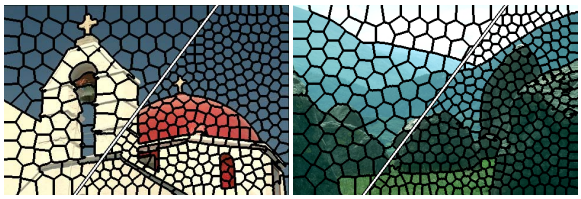


Figure 6: Example superpixels at two granularities of images from the Berkeley Segmentation Dataset [14].

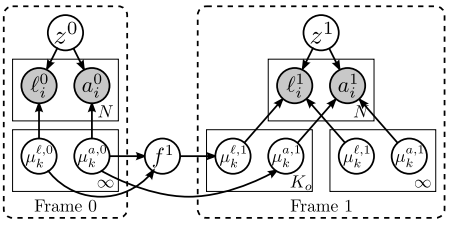


Figure 7: Graphical model for TSPs.

4. Temporal Consistency

In this section, we extend the superpixel model to develop a temporal superpixel representation. Figure 7 depicts the graphical model used for TSPs, which imposes dynamics on the learned parameters of each superpixel.

4.1. Temporal Dynamics

Due to object motion and varying illumination, the parameters of each TSP evolve over time. The appearance means are chosen to evolve independently with

$$p(\mu_k^{a,t} \mid \mu_k^{a,t-1}) = \mathcal{N}(\mu_k^{a,t}\ ;\ \mu_k^{a,t-1},\ \delta_a^2 \mathrm{I}), \tag{11}$$

where the superscript t indicates the frame. The mean locations of the superpixels evolve in a more complex fashion. Because objects move somewhat smoothly, we make the assumption that superpixels that are close in location and that look alike should move similarly. We capture this notion with a Gaussian process (GP), f^t, modeled with

$$p(f^t|\mu^{t-1}) = \mathcal{N}(f^t\ ;\ \mu^{\ell,t-1}\ ,\ \Sigma(\mu^{t-1})), \tag{12}$$

where an element in the covariance matrix is defined as

$$\Sigma_{k,j}(\mu^{t-1}) = \prod\nolimits_d h(\mu_{k,d}^{t-1}, \mu_{j,d}^{t-1}), \tag{13}$$

and $h(\cdot)$ is the squared exponential kernel (c.f. [17]).

A single GP is not often used to model flow fields. One reason is that the prior on flow fields must be able to accommodate both smoothness within objects and discontinuities across objects. For example, using an L2 penalty on neighbor differences in the Horn-Schunck optical flow formulation [10] is related to a GP with a precision that has a 4-connected neighbor sparsity pattern. These types

Figure 8: Samples from the Gaussian process flow. From left to right: superpixel segmentation; GP with 4-connected neighbor precision; GP with location kernel; GP with bilateral kernel; mapping between flow vectors and colors.

of GPs, where the kernel depends only on location, are often too smooth to model dense flows. By incorporating the appearance in the covariance kernel, we are able to model both smoothness and discontinuities that are consistent with flows. We call this covariance kernel the *bilateral kernel*, as it is similar to the bilateral filter [21]. Exemplary samples drawn from the sparse precision GP, location-only GP, and the bilateral GP are shown in Figure 8.

While the bilateral GP is able to model flow discontinuities, the prior does not fit the movement of a deformable object composed of a single color. We therefore model the location means as i.i.d. perturbations from the smooth flow

$$p(\mu_k^{\ell,t}|f^t) = \mathcal{N}(\mu_k^{\ell,t}\ ;\ f^t, \delta_\ell^2 \mathrm{I}). \tag{14}$$

4.2. New, Old, and Dead Superpixels

Due to camera motion, occlusions, and disocclusions, we must also allow old superpixels to disappear and new superpixels to emerge. We define a dead superpixel as one that existed in the previous frame but no longer exists in the current frame. Let K_o denote the number of "old" TSPs that existed in the previous frame and did not die and K_n denote the number of "new" TSPs that have appeared in the frame. In the first frame, each TSP was treated as a new superpixel with the corresponding log likelihood of Equation 9. In subsequent frames, the label distribution is updated to

$$p(z) \propto \hat{\alpha}^{K_n} \hat{\beta}^{K_o} \mathbb{1}\{T(z)\}, \tag{15}$$

where $\hat{\beta}$ is the geometric distribution parameter for old TSPs. We note that setting $\hat{\beta}$ larger than $\hat{\alpha}$ encourages using old TSPs versus generating new TSPs.

In the previous section, we showed that α influences the size of the superpixels. Because $\hat{\beta}$ plays the same role for old TSPs that $\hat{\alpha}$ plays for new ones, higher $\hat{\beta}$'s correspond to favoring smaller superpixels. Consequently, we try to separate the size of a superpixel from the tradeoff between using an old or new superpixel. This is accomplished by introducing an area term into the label distribution:

$$p(z) \propto \hat{\alpha}^{K_n} \hat{\beta}^{K_o} \mathbb{1}\{T(z)\} \prod\nolimits_k \mathcal{N}(N_k; \tfrac{N}{M}, \sigma_M^2), \tag{16}$$

where σ_M^2 controls the variability of the area of each TSP.

4.3. Inference

Inference in the dynamic case is similar to the static case. While joint inference over all the hidden variables is complicated by the GP flow, f^t, an iterative approach is straightforward. We find empirically that results are sensitive to the initialization of the flow. However, initializing with optical flow using [12] significantly improves results. Conditioned on f, optimizing z can be done in a similar fashion as before. Then, conditioned on z, estimating f is a GP regression (c.f. [17]) on the old TSPs, $o = \{1, \cdots, K_o\}$:

$$f^t = \Sigma(\Sigma + \delta_\ell^2 \mathrm{I})^{-1}(\mu_o^{\ell,t} - \mu_o^{\ell,t-1}) + \mu_o^{\ell,t-1}. \tag{17}$$

As before, we propose different label moves with corresponding optimal parameters, and accept the move if it increases the likelihood. In the case of old TSPs, the prior on the parameters is no longer uniform, and the form of the optimal values change. Due to Gaussian conjugacy, the optimal parameters can still be found in closed form.

For convenience, we denote $\theta_{k,d}^t$ as the mean of the mean parameter. In the location case, $\theta_{k,d}^t \triangleq f_k^t$, and in the appearance case, $\theta_{k,d}^t \triangleq \mu_{k,d}^{t-1}$. For the following likelihood

$$p(x_{\mathcal{I}_k,d}, \mu_{k,d}^t | \theta_{k,d}^t) = p(\mu_{k,d}^t|\theta_{k,d}^t) \prod_{i \in \mathcal{I}_k} p(x_{d,i}|\mu_{k,d}^t), \tag{18}$$

we show in the supplement that the optimal parameter is

$$\hat{\mu}_{k,d}^t = \tfrac{\theta_{k,d}^t \sigma_d^2 + t_{k,d}\delta_d^2}{N_k \delta_d^2 + \sigma_d^2}. \tag{19}$$

Using the optimal mean, the observation log likelihood for old TSPs becomes (up to a constant)

$$\mathcal{L}_o(x_{\mathcal{I}_k,d}) \triangleq \log p(x_{\mathcal{I}_k,d}, \hat{\mu}_{k,d}^t | \theta_{k,d}^t) \overset{C}{=} \tag{20}$$
$$-\log\left[\delta_d \sigma_d^{N_k}\sqrt{2\pi}\right] + \tfrac{t_{k,d}^2\delta_d^2 + 2t_{k,d}\theta_{k,d}\sigma_d^2 - N_k\theta_{k,d}^2\sigma_d^2}{2\sigma_d^2(N_K\delta_d^2+\sigma_d^2)} - \tfrac{T_{k,d}}{2\sigma_d^2}.$$

The time superscripts have been omitted to avoid confusion. The resulting joint log likelihood can then be expressed as

$$\mathcal{L}(z) \overset{C,T}{=} \beta K_o + \alpha K_n - \sum\nolimits_k \tfrac{(N_k - \frac{N}{M})^2}{2\sigma_M^2} \sum\nolimits_d \mathcal{L}_{s_k}(x_{\mathcal{I}_k,d}), \tag{21}$$

where the state of a TSP, $s_k \in \{o, n\}$, indicates if it is old or new, and selects an expression from Equation 8 or 20.

To optimize Equation 21, we use the three moves described in Section 3.3 and an additional "switch" move. The split move is slightly modified to accommodate the difference in new and old TSPs. For less than 1000 superpixels per frame, inference takes a few seconds for the first frame and tens of seconds for subsequent frames.

Split Moves: split superpixel k' using the same k-means-based partitioning algorithm as before. Assuming that the proposed partition is $\mathcal{I}_k = \{\mathcal{I}_{k1} \cup \mathcal{I}_{k2}\}$, with $\mathcal{I}_{k1} \cap \mathcal{I}_{k2} = \emptyset$,

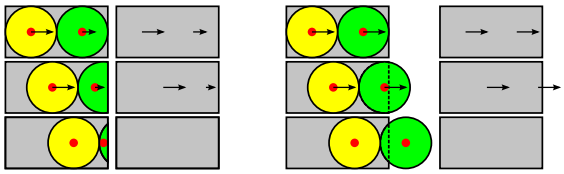


Figure 9: Example TSPs with (left) and without (right) representing support outside of the image domain. Corresponding vector difference in means is shown for each case.

we consider multiple possible labelings. We define $\mathcal{K}$ as the set of labels that we can split into:

$$\mathcal{K} = \{k \; ; \; N_k = 0, s_k = o\} \cup k_{\text{new}}, \tag{22}$$

consisting of all the dead TSPs and a possible new one. For each value $k \in \mathcal{K}$, we propose two possible labels:

$$\{z_{\mathcal{I}_{k1}} = k', z_{\mathcal{I}_{k2}} = k\} \text{ or } \{z_{\mathcal{I}_{k1}} = k, z_{\mathcal{I}_{k2}} = k'\}. \tag{23}$$

The optimal proposal is chosen from this set. Note that while $\mathcal{K}$ could represent all the TSPs that have died in all previous frames, for computational purposes, we only include those that existed in the previous frame.

Switch Moves: allow new TSPs to link to dead TSPs. A random TSP, k', is chosen and possibly relabeled to a dead $k \in \mathcal{K}$. If the optimal relabeling has a higher likelihood than the current configuration, it is accepted.

4.4. Boundary Effects

Here, we consider the role image boundaries play in representing TSPs. In particular, if the support of a superpixel is not represented outside of the image domain, the data, ℓ_i, favors choosing means within the boundaries. Consider Figure 9, which illustrates the tracking of two superpixels that are moving to the right at the same speed. Because the green superpixel is moving out of the image, the empirical location mean does not move correctly, causing errors in the flow estimates and the optimal parameter estimates. Consequently, we represent the full support of any superpixel that contains a pixel in the image domain. Details of the modified formulation accounting for image boundaries are explained in the supplement.

5. Experiments

In this section, we compare our temporal superpixel method to the supervoxel methods described in [24] and [25]. We introduce new metrics that measure the essence of what a TSP represents. Parameter values for all videos and experiments were fixed excluding M, which indicates the desired number of superpixels per frame. Specific parameter choices can be found in our publicly available source code: http://people.csail.mit.edu/jchang7/.

5.1. Proposed Metrics

In the work of [24], three metrics were presented to evaluate super-voxel methods against ground truth object labels. We consider a set of additional metrics aimed to capture the following aspects of a good model: object segmentation consistency, 2D boundary accuracy, intra-frame spatial locality, inter-frame temporal extent, and inter-frame label consistency. For a more detailed description of how the metrics are calculated, please refer to the supplement.

Object Segmentation Consistency. Each supervoxel should only belong to one object. The 3D undersegmentation error (UE) and the 3D segmentation accuracy (ACC) of [24] are aimed at measuring this trait. UE captures the fraction of pixels that bleed past the boundary of a ground truth segment and ACC captures the fraction of a ground-truth segment that is correctly classified by the supervoxels. The metrics are averaged across the ground truth segments.

2D Boundary Accuracy. Each superpixel should accurately capture object boundaries. While [24] introduces the 3D boundary recall (BR3), we find that BR3 captures a mixture of information between 2D boundary recall and object segmentation consistency. As such, we consider 2D boundary recall averaged across frames. The typical boundary recall metric finds the percent of ground truth boundaries that are also declared superpixel boundaries. However, this metric is not robust to small localization errors. The superpixel boundaries are often dilated to reconcile the localization problem, but this causes the error to depend on the amount of dilation. We introduce the 2D boundary recall distance (BRD) metric, related to the metric of [6], as the average *distance* between points on the ground truth boundary to a declared super pixel boundary averaged across frames. The distance between boundaries exactly measures localization errors and is also more easily interpretable.

Intra-Frame Spatial Locality. As shown in Figure 3, a good representation should be local in nature. As superpixels get larger, they lose their representative power. Consequently, assuming a perfect ACC, UE, and BRD, assigning equally-sized superpixels corresponds to the best representation. We therefore introduce the size variation (SZV) metric which considers the average standard deviation of superpixel sizes across all frames.

Inter-Frame Temporal Extent. A good TSP representation should contain TSPs that track objects for long periods of time. The authors of [25] introduce the mean duration time (MDT) metric which computes the average number of frames a supervoxel exists in. Longer videos can inherently have higher MDT, complicating the comparison of results across videos of different lengths. We therefore introduce the temporal extent (TEX) metric which normalizes the MDT by the total number of frames in the video.

Inter-Frame Label Consistency. Finally, we introduce the label consistency (LC) which measures how well superpixels track parts of objects. Superpixel labels at frame $t-1$ are propagated using annotated ground truth flow and compared to the superpixel labels at frame t. LC counts the average number of pixels that agree between the inferred superpixels and the ones propagated via the flow.

5.2. Algorithm Comparison

Using these metrics, we compare our TSP algorithm to the top two supervoxel methods of [24] (GBH [8] and SWA [20]) and the streaming version of GBH developed in [25]. We use the GBH implementation provided by [24] which does not use optical flow since the original algorithm does not produce superpixel segmentations (as shown in Figure 3). Unlike our algorithm, GBH and SWA exploit future data by processing the entire video sequence at once. Streaming GBH considers only looking at k frames in the future. For a fair comparison with TSPs, we consider Streaming GBH with $k = 1$. We note that it is difficult to tune the parameters of the other algorithms to sweep the range of desired superpixels. In contrast, our algorithm only required changing M, the desired number of superpixels per frame.

Videos that are longer in length or that contain a lot of background motion should contain more supervoxels than short, static scenes. Therefore, unlike [24] which plots the metrics against the number of supervoxels, we plot the metrics against the average number of superpixels per frame. Quantitative results are shown in Figure 10. We evaluate the algorithms using the videos from [12] and [22] which have ground truth object segmentation. For the LC, we use the videos from [2] and [12] since ground truth flow is needed.

TSPs perform slightly better in UE, comparable in ACC, and worse in BRD. The BRD value for TSPs indicates that the average distance between a ground truth boundary and a superpixel boundary is approximately 1 pixel. This error is often within the tolerance of most applications. TSPs perform better in the final three metrics, indicating that TSPs extend farther in time, vary less in size, and maintain better label consistency than supervoxels.

A visual comparison between TSP, GBH, and SWA is shown in Figure 11. After obtaining the full superpixel segmentation, we manually color a subset of superpixels that existed in the first frame and visualize their extent in time by looking at subsequent frames. TSP track superpixels correctly through all frames, while GBH and SWA lose the tracks as time progresses and exhibit drifting effects (shown in blue). Additional TSP results are shown in Figure 12.

6. Conclusion

In this paper, we have presented a low-level video representation called the temporal superpixel. While related

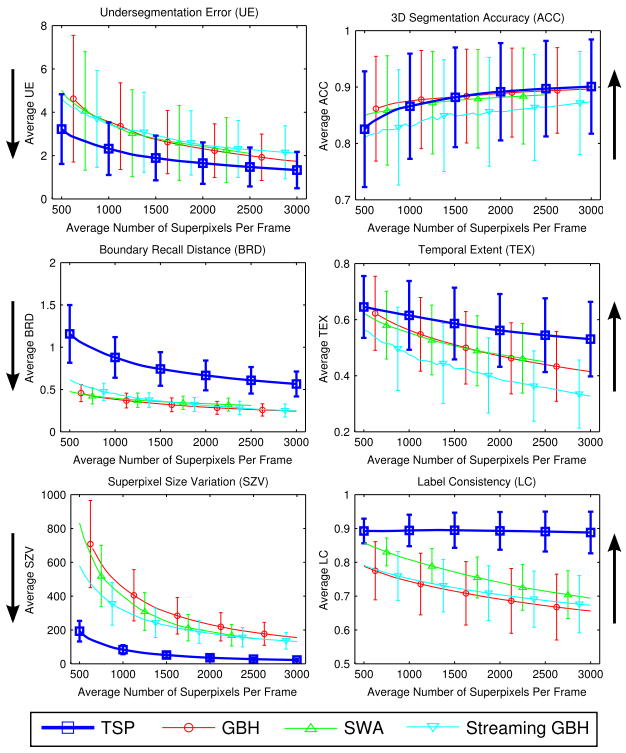
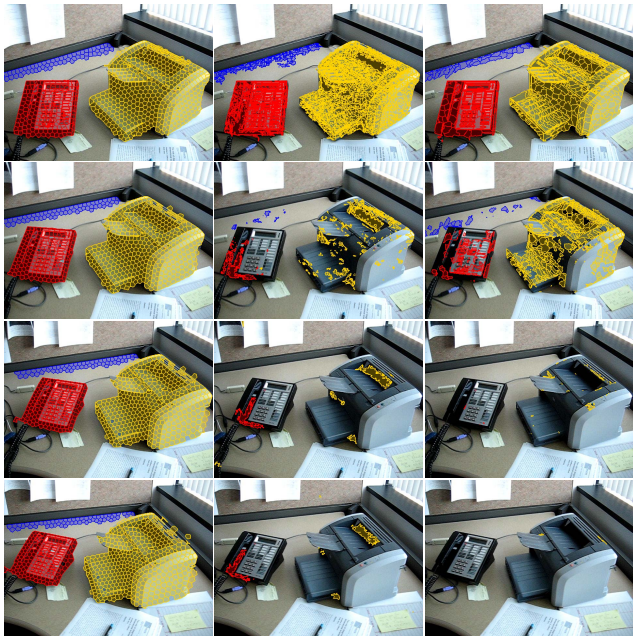


Figure 10: Quantitative evaluation of TSPs, GBH, SWA, and Streaming GBH. Arrows beside plots indicate direction of better performance. Error bars indicate one stddev.



TSP GBH SWA

Figure 11: Visual comparison of algorithms for some supervoxels that existed in the first frame. Each row is an algorithm, and each column is a subsequent frame. Colors are added manually to aid in in the visualization.

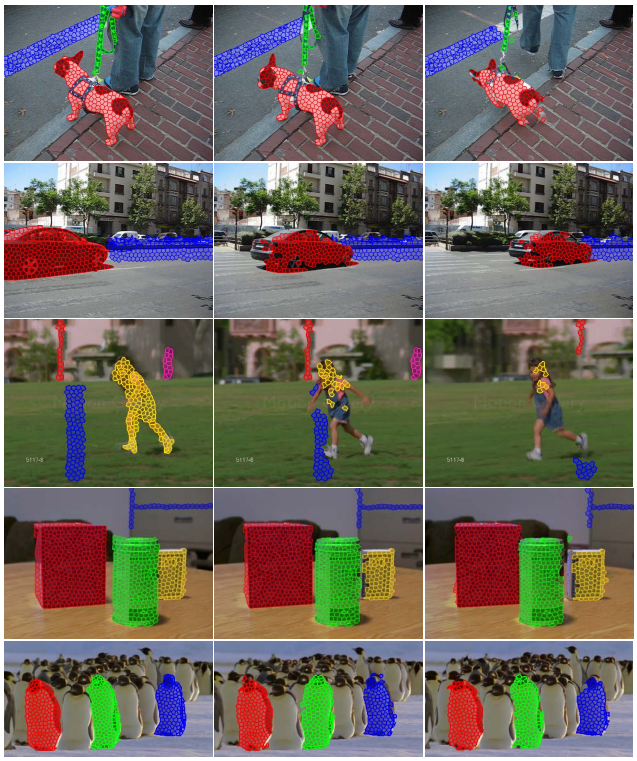


Figure 12: Results from the TSP algorithm. Colors are added manually to aid in in the visualization.

to the volumetric voxel, a TSP inherently treats the temporal dimension differently to accommodate videos. We have shown quantitatively that TSP representations outperform supervoxel methods. We encourage others to consider TSPs as an intermediate representation and hope that our public code facilitates that consideration.

Acknowledgements. J.C. and D.W. were partially supported by the Office of Naval Research Multidisciplinary Research Initiative (MURI) program, awards N00014-11-1-0688 and N00014-09-1-1051, respectively. J.F. was partially supported by the Defense Advanced Research Projects Agency, award FA8650-11-1-7154.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, 2012. 2, 3

[2] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *ICCV*, 2007. 7

[3] G. Bertrand. Simple points, topological numbers and geodesic neighborhoods in cubic grids. *Pattern Recogn. Lett.*, 1994. 3

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. *ECCV*, 2010. 1

[5] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *PAMI*, 2011. 1

[6] V. Chalana and Y. Kim. A methodology for evaluation of boundary detection algorithms on medical emages. *IEEE Trans. on Medical Imaging*, 1997. 6

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004. 1, 2

[8] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph based video segmentation. *CVPR*, 2010. 1, 2, 7

[9] X. Han, C. Xu, and J. L. Prince. A topology preserving level set method for geometric deformable models. *PAMI*, 2003. 3

[10] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981. 1, 4

[11] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *PAMI*, 2009. 2

[12] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss. Human-assisted motion annotation. *CVPR*, 2008. 5, 7

[13] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Int. Joint Conference on AI*, 1981. 1

[14] D. R. 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. *ICCV*, 2001. 2, 4

[15] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. *CVPR*, 2004. 2

[16] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. *ICCV*, 2011. 1

[17] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. 4, 5

[18] X. Ren and J. Malik. Learning a classification model for segmentation. *CVPR*, 2003. 1, 2

[19] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *CVPR*, 2006. 1

[20] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. *CVPR*, 2000. 7

[21] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *ICCV*, 1998. 5

[22] D. Tsai, M. Flagg, and J. M.Rehg. Motion coherent tracking with multi-label mrf optimization. *BMVC*, 2010. 7

[23] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. *ECCV*, 2010. 2

[24] C. Xu and J. Corso. Evaluation of super-voxel methods for early video processing. *CVPR*, 2012. 1, 2, 6, 7

[25] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. *ECCV*, 2012. 2, 6, 7

[26] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. *CVPR*, 2004. 2

[27] C. Zitnick, N. Jojic, and S. B. Kang. Consistent segmentation for optical flow estimation. *ICCV*, 2005. 2