Video Segmentation by Tracking Many Figure-Ground Segments

Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, James M. Rehg
School of Interactive Computing
Georgia Institute of Technology
{fli,ahumayun,rehg}@cc.gatech.edu, {cuponthetop,caihsiaoster}@gmail.com

Abstract

We propose an unsupervised video segmentation approach by simultaneously tracking multiple holistic figure-ground segments. Segment tracks are initialized from a pool of segment proposals generated from a figure-ground segmentation algorithm. Then, online non-local appearance models are trained incrementally for each track using a multi-output regularized least squares formulation. By using the same set of training examples for all segment tracks, a computational trick allows us to track hundreds of segment tracks efficiently, as well as perform optimal online updates in closed-form. Besides, a new composite statistical inference approach is proposed for refining the obtained segment tracks, which breaks down the initial segment proposals and recombines for better ones by utilizing high-order statistic estimates from the appearance model and enforcing temporal consistency. For evaluating the algorithm, a dataset, SegTrack v2, is collected with about 1,000 frames with pixel-level annotations. The proposed framework outperforms state-of-the-art approaches in the dataset, showing its efficiency and robustness to challenges in different video sequences.

1. Introduction

Image segmentation has received a considerable boost in recent years. Recent advances show that one can create a pool of several hundred overlapping figure-ground segment proposals so that for most objects in the scene, at least one segment in the pool covers $70-80\%$ of it. The figure-ground segments can be obtained optimally with a graph min-cut framework [17, 14, 11] and are invariant to internal edges. These advances have made unsupervised segmentation applicable to difficult high-level tasks, such as semantic segmentation [23, 2, 10] and video saliency reasoning [19].

We are interested in transfering such successes to the spatio-temporal video segmentation problem [15, 20, 6, 7, 31], which is instrumental for many high-level applications such as action recognition, depth and occlusion reasoning, as well as object tracking. A video sequence offers rich motion and depth cues, from where we could hope to automatically suppress temporally inconsistent segments and obtain good object proposals with fewer segments in the pool. However, a significant challenge is that local motion estimation methods (e.g. optical flow), although very good in general, do not work well on large motions [8] and in areas close to object boundaries, two important aspects for successful segmentation. Therefore, it is hard to strike correct temporal links without examining appearance information from a larger region [8], or temporal information over a longer period [7], both requiring extensive computations.

Inspired by the need of non-local information for successful video segmentation, this paper make an attempt to use non-local appearance cues more directly. We propose to solve unsupervised video segmentation by simultaneously tracking all segments from segment pools generated by figure-ground segmentation at each frame. Initially, one segment track is initialized from each unsupervised segment generated in the first frame. Then for each segment track, a persistent global appearance model is incrementally trained: at each frame, predictions from the models are used to find the segment that best matches each track, and those matching segments are then used to update the track models.

Such an approach – incremental learning with global appearance models – has been used in many tracking frameworks to increase robustness and prevent drift (e.g. [3]). However, a significant challenge of applying it to unsupervised multi-segment tracking is efficiency: it sounds formidable to simultaneously maintain and update hundreds of non-local appearance models. We address this issue by formulating multi-segment tracking as multi-output regularized least squares. We propose to train the appearance model for each track using *all* segments in the pools from all previous frames. At each frame, the target output is the overlap between a segment and the matching segment of the track at that frame. Different part segments have different target outputs based on their spatial overlap with the matching segment. Segments that do not overlap the matching segment would have a target of 0.

It may seem that we are making the problem harder by training models with many excessive examples. However, by equating the training examples in different models, we can exploit structures in least squares regression to make tracking highly efficient. We assume that all tracks start from the first frame, so that the training examples are the same for all the segment tracks. Then it turns out most of the costly operations for solving least squares need to be done only once, regardless of the number of segment tracks. This formulation allows us to track hundreds of segments with fairly low complexities in both time and space.

Our matching framework assumes that at least one good segment is present in most frames for each object. Moreover, we aggressively prune the segment tracks: at each frame, if multiple segment tracks match to the same segment, then only the track with the highest score is retained. Remarkably, our long-term appearance models are robust enough, so that under such strong assumptions and aggressive pruning, we are still able to cover most objects in the testing videos, while reducing the average number of tracks to 60 from about $1,200$ initial segments per frame.

Given the fully learnt appearance models, we adopt a recent composite statistical inference (CSI) approach [22] to refine the segments in the previous frames. This is important for complex objects where the initial segmentation is unlikely to be perfect, but a refinement can be made by exploiting the learnt appearance model. CSI breaks segment proposals into superpixels and recombines the superpixels by optimizing the likelihood of predictions on the segment proposals given by the appearance model. We propose two improvements to the framework: a speed-up by simplifying the formulation, and an approach to improve temporal consistency by specifying temporal links on matching (or partially matching) superpixels across consecutive frames. The CSI inference further improves the tracking results in terms of both accuracy and temporal consistency.

This framework reflects our attempt to test the validity of using and tracking holistic figure-ground segment proposals for video segmentation. To extend it into a practical tracking algorithm, we would need to lift the assumption that all segment tracks start from the first frame. However, it is conceivable that our framework can serve as a tracklet generation model for general multi-target tracking over a long period. One can use the proposed framework for multiple intervals of several seconds and regard the generated segment tracks as tracklets. Then data association algorithms can be designed based on the appearance models, which could be more robust than matching frames independently. This is a direction we will explore in the future.

2. Related Work

Video segmentation has seen quite some interest in recent years. Many state-of-the-art approaches are based on the agglomerative clustering approach on supervoxels [16, 15, 37, 18]; spatial-temporal graph-cuts [36]; or tracking feature points or local regions [6, 34, 7]. The local tracking methods usually track non-overlapping feature points/superpixels and hence are different from our approach that tracks overlapping holistic segments. In the graph-based approaches, [12, 29, 31] uses higher-order Markov random fields or conditional random fields. [9, 32] uses variational approximations. Interactive segmentation approaches have also been proposed [4, 27, 38]. A few new approaches rely on multiple per-frame figure-ground segmentations: [20] utilizes motion saliency to detect the right segments to track, then run successive graph cuts on clips propagating from the most confident key segment. However, their model depends strongly on the success of the saliency which could fail when multiple adjacent objects are moving in the scene. [25] proposes a maximal weighted clique framework to optimally link segments in each frame, their mutual exclusion constraint allows only one segment to be selected in each frame, thus segments that partially match the segment tracks are not utilized.

Many state-of-the-art tracking algorithms track bounding boxes using global appearance features (e.g. [3]). Our segment tracking scheme however uses segment proposals which are better boundary-aligned than bounding boxes. For multi-target tracking, many approaches only use local or no appearance models for each individual track for efficiency considerations. A popular regime is tracking-by-detection (e.g. [26, 1]), where an object category appearance model (e.g. for all pedestrians) is applied to detect the desired object in all the frames of the video. Our approach trains a distinct global appearance model for each track, and is fully unsupervised thus do not utilize additional detectors to trim down the number of hypotheses.

Besides the video segmentation work that utilizes segment tracking [29, 34, 6], a great deal of research have been on segment tracking with active contours [13, 28, 5], which require a user-drawn region in the first frame. Our segment tracking does not have a requirement for user initialization.

3. Segment Pool Tracking by Online Multi-Output Linear Regression

Our Segment Pool Tracking (SPT) framework performs unsupervised video segmentation by the following steps, which will be described in detail in the subsequent sections:

- Generate a pool of segments for each frame via a multiple figure-ground segmentation algorithm (Sec. 3.1).
- Compute appearance features for each segment in all frames.
- Initialize a segment track for each segment in the first frame.
- Simultaneously learn the appearance models for all segment tracks by multi-output regression (Sec. 3.2).

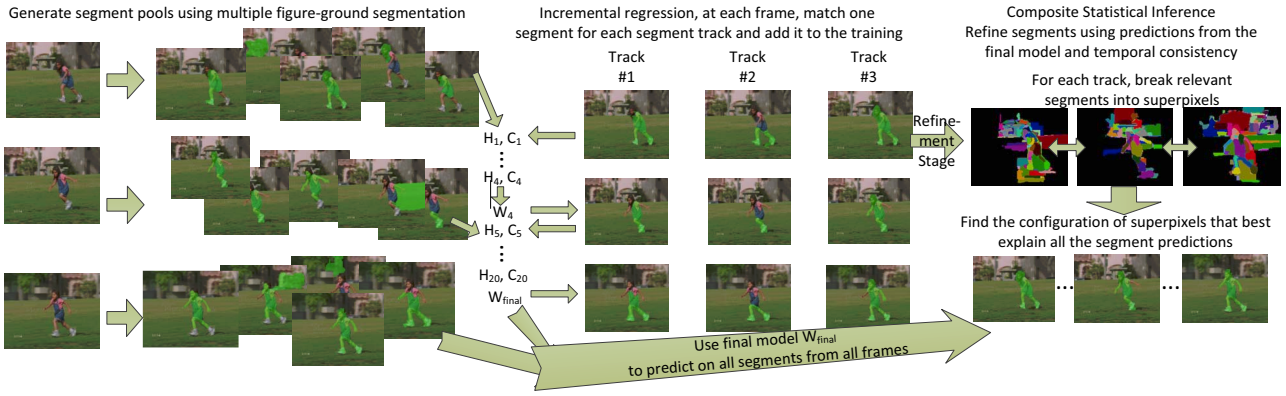

Figure 1. Workflow of the proposed video segmentation approach. In the first stage, a pool of figure-ground segment proposals is generated for each frame (left). Each segment from the first frame spawns a segment track, and the appearance models of all the tracks are learnt incrementally and simultaneously. At each frame, a segment that best matches each appearance model is found, and then all the segments are added to the training, with the target outputs decided by the overlap with the matching segments (middle). Finally, in order to refine the segments, the learned models are tested on all segments across all frames, then relevant regions for each segment track are broken into superpixels and an optimal configuration of the superpixels is found through a composite statistical inference (right).

- Match segments in the next frame to existing segment tracks with a greedy algorithm. Update appearance models with all the matched segments (Sec. 3.3). Start new tracks on unmatched segments if necessary.
- Iterate until the end of the video clip.
- For long enough segment tracks, perform composite statistical inference (Sec.4) to jointly refine segments in all frames.

Our ability to simultaneously track hundreds of segments comes from the adoption of the regression-to-overlap framework and casting the problem as multi-output regularized least squares regression. The regression-to-overlap framework was proposed in [23] as an approach to utilize imperfect segments in training. Instead of a positive/negative binary classification target, the object model learns to predict a real-valued target, which is 1 for perfect training segments, 0 for negative segments, and a real value in $(0, 1)$ for imperfect segments based on the amount of their overlap with the ground truth. In this way, object parts are effectively utilized in the training without compromising the convexity of the optimization.

Importantly, with the overlap as targets, different segment tracks can now train on the same set of training examples. This is efficiently utilized in our least-squares regression framework, where the input is encoded in the sample covariance matrix, and the output encoded in an input-output correlation matrix. The computation and factorization of the sample covariance is invariant to the number of tracks, and is usually more costly than the input-output correlation. In consequence, adding more segment tracks adds very little to the training/testing time, unless the number of tracks exceed the feature dimension. By storing and updating only these two matrices, we can learn the optimal appearance models of all the segment tracks simultaneously.

3.1. Figure-Ground Segmentation with Spatial-Temporal Boundaries

A pool of figure-ground segments is generated for each frame by a parametric min-cut [17] figure-ground segmentation algorithm such as [11, 14]. For an image I, let x_i be pixel labels in I and $x_i \sim x_j$ denote x_i is adjacent to x_j in a 4 or 8-way pixel neighborhood. The figure-ground segmentation problem can be casted as energy minimization:

$$\min_{x_i \in \{0,1\}} \sum_{x_i} (a_i + \lambda b_i) x_i + \sum_{x_i \sim x_j} \mathbb{I}(x_i \neq x_j) E(x_i, x_j) \quad (1)$$

where pixels with $x_i = 1$ are considered as the foreground object and pixels with $x_i = 0$ are considered background, a_i and b_i are algorithm-specific, $\mathbb{I}$ is the indicator function, and $E(x_i, x_j)$ reflects the edge strength between x_i and x_j. The λ parameter incurs a penalty on the number of foreground pixels (with $x_i = 1$), thus implicitly controls the size of the foreground object. By varying λ, a spectrum of segments ranging from a few pixels to the whole image can be generated. The segments are invariant to internal edges since their sizes are controlled by λ and pairwise losses are only counted at the boundaries (when $x_i \neq x_j$). Compounded with a grid-based enumeration of foreground seed pixels, such a figure-ground segmentation approach can generate several hundreds of segments per image that covers full objects and parts within a consistent framework.

To incorporate spatial-temporal boundaries into the segmentation algorithm, we feed optical flow as an image to an edge detection algorithm [21]. In order to create more diversity, the resulting boundaries are fed to the segmentation algorithm as $E(x_i, x_j)$ in 3 different ways: image boundaries only, flow boundaries only, and a 50%-50% linear combination between image boundaries and flow boundaries (Fig.2). The resulting segment pool contains all the segments generated from the three boundary types.

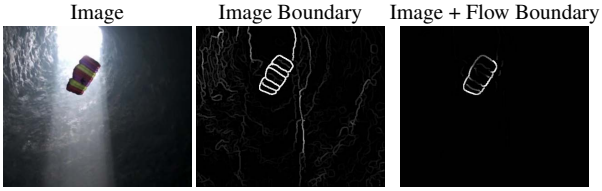

Figure 2. For some images, motion boundaries lead to easier segmentation for regions with distinct color and uniform motion. However sometimes motion boundaries are unreliable, therefore we enumerate segments generated from different types of boundaries (image, flow, and image+flow) in the pool.

3.2. Appearance Models by Multi-Output Regularized Least-Squares

For each segment, we extract two feature vectors from the foreground and background of the segment, respectively. In principle any feature can be used in the framework. In this paper we choose to use bag-of-words on Color SIFT [33], since color is a reliable feature for tracking. We did not use shape or motion features, since shape can deform across frames and motion can also vary significantly. In order to exploit nonlinear kernels in a linear framework, we adopt random Fourier (RF) feature mappings on the exponential χ^2 kernel [35, 24]. Under this, one can view the linear regressor as similar to a kernel regressor $f(x) = \sum_i \alpha_i K(x, x_i)$ with the exponential χ^2 kernel. Such similarity kernels on bag-of-word features are natural overlap predictors. Suppose one segment overlaps 50% with the other segment, then they likely share about 50% similar SIFT feature points. After bag-of-words and kernel mapping, their similarity should be around 50% – a good estimator of their overlap. Since RF is an approximate mapping to the kernel space, our linear regressors in the RF mapping space should be good overlap predictors also.

Denote a video as a sequence of T image frames $I_1, \ldots, I_T$. The segments for the t-th frame are denoted as $A_{t1}, \ldots, A_{tn_t}$, where n_t is the number of segments at the t-th frame. For segment A_{ti}, denote its feature vector after d-dimensional RF mapping as X_{ti}, and denote $\mathbf{X}_t = [X_{t1}^\top, \ldots, X_{t,n_t}^\top]^\top$ the $n_t \times d$ feature matrix with each row being a feature vector. Suppose there are n segment tracks, each is represented at frame t by a matching segment $A_t^j, j = 1, \ldots, n$. An $n_t \times n$ overlap matrix $\mathbf{V}_t$ is computed between all segments in the frame and the matching segments. Throughout the tracking process, we maintain a covariance matrix $\mathbf{H}_t = \sum_{i=1}^t \mathbf{X}_i^\top \mathbf{X}_i$, and a correlation matrix $\mathbf{C}_t = \sum_{i=1}^t \mathbf{X}_i^\top \mathbf{V}_i$, which are initialized at 0 for the first frame and updated incrementally. Suppose we already have $\mathbf{H}_{t-1}$ and $\mathbf{C}_{t-1}$ then at time t, they are updated as:

$$\mathbf{H}_t = \mathbf{H}_{t-1} + \mathbf{X}_t^\top \mathbf{X}_t, \mathbf{C}_t = \mathbf{C}_{t-1} + \mathbf{X}_t^\top \mathbf{V}_t \qquad (2)$$

Then a regression weight matrix $\mathbf{W}_t$ is obtained by solving the regularized multi-output least-squares problem:

$$\min_{\mathbf{W}_t} \sum_{i=1}^t \|\mathbf{X}_i \mathbf{W}_t - \mathbf{V}_i\|_F^2 + \lambda \|\mathbf{W_t}\|_F^2 \qquad (3)$$

The solution is given by the linear system:

$$(\mathbf{H}_t + \lambda \mathbf{I}) \mathbf{W}_t = \mathbf{C}_t \qquad (4)$$

which is solved via Cholesky decomposition. $\mathbf{W_t}$ is now the learned model for all the segment tracks. Given a new segment $A_{t+1,i}$ in frame $t+1$, $X_{t+1,i}\mathbf{W_t}$ predicts its overlap with hypothetical ground truth segments corresponding to the objects represented by all the segment tracks.

The time complexity of this procedure is $O(nd^2 + d^3)$. Most of the time, we have at most several hundred tracks while thousands of feature dimensions, hence the time complexity is often dominated by $O(d^3)$, the Cholesky decomposition time. One can also use conjugate gradient methods with warm-starting, which could bring the time complexity down to $O(nd^2)$. The space complexity of maintaining the two matrices is $O(d^2 + dn)$.

3.3. Greedy Matching

To match segments and eliminate redundant segment tracks rapidly, a greedy matching algorithm is proposed to extend segment tracks to new frames. Suppose we have segment tracks represented by the weight matrix $\mathbf{W}_t$ and need to find matching segments for all tracks in frame I_{t+1}. The predicted overlap at time $t+1$ is computed as $\hat{\mathbf{V}}_{t+1} = \mathbf{X}_{t+1}\mathbf{W}_t$. Suppose $\hat{\mathbf{V}}_{t+1} = [\hat{V}_1, \ldots, \hat{V}_n]$ where $\hat{V}_j$ is the prediction vector for segment track T_j on all the segments in frame $t+1$.

For each segment track T_j, we first threshold with crude motion cues (e.g. centroid displacement) to determine which segments might be a possible match. Among all segments that satisfy the motion threshold, we find $k = \arg\max \hat{V}_j$ so that A_{tk} is the segment with the best predicted overlap $s_{jk} = \max \hat{V}_j$ for the track T_j. If the same segment A_j is matched to multiple tracks, then only the track with the highest score $\arg\max_j s_{jk}$ is retained (Fig. 3). This simple greedy procedure serves as a non-maximum suppression (NMS) process to reduce the number of segment tracks. It is not optimal, but in preliminary experiments have performed similar or better than a more costly Hungarian matching algorithm. Intuitively, because the appearance model incorporates appearances from all previous frames, matching errors in one frame can be corrected in the subsequent frames since they do not drift the model much. Importantly, greedy matching retains an order of magnitude fewer segment tracks than Hungarian because of the NMS effect. Improvements can be made, such as using priors to make longer tracks harder to be killed, we will pursue these improvements in an extended version.

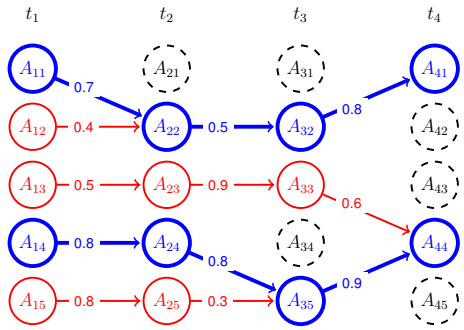

Figure 3. Greedy matching of segment tracks. The number on the edges is the predicted overlap score. At each frame, NMS is performed among the tracks that match to the same segment and the low-scoring tracks (red) are stopped.

After matching, each surviving segment track is updated with a new segment at time $t+1$. Then, all the segments at time $t+1$ are added to the training set, with the target output computed as the overlaps between the segment and the matching segment of each segment track. Then the corresponding matrices are updated as in (2). We start a new track for each segment that has not been matched to any track. In the experiments we assume all the objects are present from frame 1, therefore we only start segment tracks in the first 5 frames to strike a balance between speed and robustness to missing segmentations in the first few frames.

4. Refinement using Composite Statistical Inference

It would be too optimistic to assume that a perfect segment is always present in the initial segment pool. Besides, the greedy matching procedure can also generate suboptimal results. Therefore, in this section we propose an approach to refine the segments in each frame given the learned segment tracks. We choose to perform this refinement as a second stage, so that we operate on fewer tracks after spurious ones have been filtered out.

Composite statistical inference (CSI) [22] is a recent inference approach designed to perform inference using predictions on segment statistics. A crucial difference from previous approaches is that CSI focuses on statistics estimates, instead of probability estimates. While inference on high-order probability terms often requires computing multi-dimensional integrals, CSI models the one-dimensional error distributions of many region statistics computed on different segments and hence avoid intractable computations. In CSI, superpixels are obtained from *multiple intersections* on the candidate segments, defined as the crudest superpixel partition of the image, so that each superpixel either completely belongs to a segment or stays completely outside. Then, real-valued superpixel statistics are defined so that the segment statistics are *computable* from them. This means that there exist a formula to compute the segment statistics given the superpixel statistics. With these links, one can maximize the composite likelihood of the noisy predictions on segment statistics to recover the unknown superpixel statistics. Finally, the most likely segment is obtained from the superpixel statistics.

While [22] deals with semantic segmentation, this paper extends the CSI approach to segment tracking. There are two main differences. First, there are many false positive predictions in semantic segmentation because of the need to generalize to new objects. A binary latent variable indicating true positive/false positive was created for each prediction in [22], leading to a complicated inference. In tracking this issue is not as severe, therefore our algorithm does not have latent variables and is much faster. Second, we introduce temporal consistency terms to connect superpixels in adjacent frames, which leads to segment tracks that deform more smoothly over time.

Formally, suppose an object is tracked over time $t = 1, \ldots, T$, with F_t being the ground truth segment at frame t. Suppose each frame I_t is separated into superpixels $S_{t1}, \ldots, S_{tm}$ by multiple intersections, and let $S_{tk} \in A_{ti}$ denote that the superpixel S_{tk} is completely inside A_{ti}, and $S_{tk} \notin A_{ti}$ otherwise. Then define superpixel statistic θ_{tk} as the percentage of pixels in S_{tk} that belongs to object F_t. The overlap statistic $V(A_{ti}, F_t)$ can be computed as:

$$\begin{aligned} V_\theta(A_{ti}, F_t) &= \frac{|F_t \cap A_{ti}|}{|F_t \cup A_{ti}|} \\ &= \frac{\sum_{S_{tk} \in A_{ti}} \theta_{tk} |S_{tk}|}{\sum_{S_{tk} \in A_{ti}} |S_{tk}| + \sum_{S_{tk} \notin A_{ti}} \theta_{tk} |S_{tk}|} \end{aligned} \tag{5}$$

where $|S_{tk}|$ is the number of pixels within S_{tk}. Note that V_θ is computable from θ without the knowledge of F_t. Now suppose the final appearance model for one segment track is W_T, we use it to predict $\hat{V}_{ti} = X_{ti} W_T$ for all segments A_{ti} in all frames $t = 1, \ldots, T, i = 1, \ldots n_t$. We model the regression error distribution $P(\hat{V}|V)$ as a simple Gaussian with mean V, the unknown ground truth overlap. Now we are ready to find the best θ by maximizing the composite likelihood of observing the predicted values $\hat{V}_{ti}$:

$$\begin{aligned} &\min_\theta \sum_{t=1}^{T} \sum_{i=1}^{n_t} \left(V_\theta(A_{ti}, F_t) - \hat{V}_{ti} \right)^2 \\ &\text{s.t. } 0 \le \theta_{tk} \le 1, k = 1, \ldots, n \end{aligned} \tag{6}$$

In order to find a unique solution and enforce temporal consistency, we add two regularizations, an L_2 regularization to θ: $\sum_k |S_{tk}| \theta_{tk}^2$ as in [22], and a temporal regularization based on optical flow. It is hard to obtain a 1-1 mapping of superpixels in video sequences, therefore we utilize many-to-many mappings. Suppose w_{kj}^f percent of the pixels in superpixel S_{tk} is mapped to superpixel $S_{t+1,j}$ by forward optical flow, then our temporal consistency term has the form:

$$\sum_{t=1}^{T-1} \sum_{k=1}^{n_t} (S_{tk} - \sum_j w_{kj}^f S_{t+1,j})^2$$

which maps S_{tk} partially to many superpixels in the next frame with different proportions and specifies a joint constraint term. Likewise, we can define w^b_{kj} as the percent of pixels in superpixel S_{tk} that are mapped to $S_{t-1,j}$ and add a backward consistency term:

$$\sum_{t=2}^{T}\sum_{k=1}^{n_t}(S_{tk} - \sum_j w^b_{kj} S_{t-1,j})^2$$

Putting everything together, we solve the joint optimization problem on the entire segment track:

$$\begin{aligned}\min_{\theta} \quad & \sum_{t=1}^{T}\sum_{i=1}^{n_t}\left(V_\theta(A_{ti}, F_t) - \hat{V_{ti}}\right)^2 + \lambda_1 \sum_{t=1}^{T}\sum_k |S_{tk}|\theta^2_{tk} \\ & + \lambda_2 \sum_{t=1}^{T-1}\sum_{k=1}^{n_t}(S_{tk} - \sum_j w^f_{kj} S_{t+1,j})^2 \\ & + \lambda_2 \sum_{t=2}^{T}\sum_{k=1}^{n_t}(S_{tk} - \sum_j w^b_{kj} S_{t-1,j})^2 \\ \text{s.t.} \quad & 0 \le \theta_{tk} \le 1, k = 1, \ldots, n \end{aligned} \tag{7}$$

This is a smooth optimization problem with bound constraints. There is usually only $10 - 200$ superpixels per frame thus a few thousand optimization variables overall. We use an LBFGS-B algorithm in the `minConf` package[1] to solve it. The segment with the best predicted overlap in each frame is used as a natural initialization for θ.

After obtaining θ, we adopt the following procedure in [22] in order to output the optimal segment for each frame given θ:

- Sort all θ in descending order. Initialize $C = 0$.
- From the start of the sorted list, include superpixel into the final segment one-by-one and compute the overlap V of the current segment using formula (5) from θ.
- Stop when $V > \frac{\theta_j}{1-\theta_j}$, and output the segment with superpixels 1 to $j-1$ in the sorted list.

5. Experiments

5.1. SegTrack v2

For a more comprehensive evaluation of video segmentation algorithms, we introduce an updated version of the SegTrack dataset [31], called SegTrack v2. SegTrack v2 reflects two major enhancements from the original SegTrack. First, in SegTrack, only one moving object is annotated for each sequence. However, many videos have more than one object of interest. We provide additional annotations of objects for three sequences `Monkeydog`, `Cheetah` and `Penguin` in the SegTrack dataset. In addition, we introduce 8 new sequences: `Bird of paradise`, `BMX`, `Drifting car`, `Hummingbird`, `Monkey`, `Frog`, `Worm`, and `Soldier`. These 8 new sequences add a total of 11 new objects and 732 annotated frames, which leads to a total of 14 sequences with 24 objects over 947 annotated frames in SegTrack v2. Although the size is still modest, the sequences are carefully chosen in order to present different challenges to segmentation algorithms, such as motion blur (`BMX`), appearance change (`Bird of paradise`, `Drifting car`), complex deformation (`Worm`, `Hummingbird`), slow-motion (`Frog`), occlusion (`Penguin`, `Cheetah`, `Frog`) and multiple adjacent/interacting objects (`BMX`, `Drifting Car`, `Penguin`, `Cheetah`). An algorithm needs to adapt to all these challenges to get a good overall score.

For the performance metric, we find the pixel error metric in [31] misleading. Under the pixel error metric, small objects naturally have a smaller error and larger objects have larger errors. The metric also cannot distinguish between the case where all the error pixels reside along the boundary because of a 3-pixel boundary deviation, and the case where there exists an erroneous region that is visually more intrusive. For sake of comparison, we still report the pixel error metric on the original SegTrack dataset but we are not reporting SegTrack v2 results with these metrics. Instead, we adopt the intersection-over-union overlap metric that is commonly used in image segmentation. We will also present the results with some additional metrics in an extended version.

5.2. Experiment Setup and Results

The experiments are all performed on a 3.2GHz Intel i7-3930K machine with 6 cores. The algorithms are programmed in MATLAB with some time-consuming functions implemented in C++. The CPMC algorithm [11] is used to compute the segment proposals. Parameters for SPT and CSI inference are fixed across all sequences, with the SPT regularization parameter $\lambda = 80$ (Eq. 3), CSI parameters $\lambda_1 = \lambda_2 = 0.3$ (Eq. 7). $3,000$ RF features are used for each feature descriptor (foreground/background), leading to a total of $6,000$ dimensions. The RF for exponential χ^2 kernel is approximated with an analytic approximation proposed in [24] using 5 dimensions for each input dimension and a exponential kernel width of 1.5. The optical flow is computed using the Classic+NL algorithm [30].

We present results with and without CSI refinement, respectively. In the result tables, SPT refers to the online segment tracking algorithm presented in Section 3 without refinement. SPT+CSI refers to the results obtained by CSI refinement of the SPT segment tracks. Besides, Table 3 presents the computational time of various stages in SPT and CSI. The timing results are obtained on the first frame of the sequence `Drifting car` which has the highest resolution among all sequences (640x360).

Among all segment tracks returned by a video segmenta-

[1] http://www.di.ens.fr/~mschmidt/Software/minConf.html

tion algorithm, we report the performance on the best track w.r.t. each ground truth object. This is common for unsupervised segmentation algorithms, as the desired objects in each task can vary, and the goal for unsupervised segmentation is to obtain a small set of object hypotheses that always contain all the desired objects. We compare against several different approaches in SegTrack (Table 1) and SegTrack v2 (Table 2). The main competitors are the key segments approach by Lee and Grauman [20] which uses multiple segment proposals followed by an spatial-temporal graph-cut, and Grundmann et al. [15] which is based on hierarchical agglomerative aggregation of superpixels. The algorithm in [20] is designed to only capture the most prominent object in the sequence, but due to its inherent randomness, we are able to obtain multiple objects by running the algorithm many times with different random seeds. The algorithm of Grundmann et al. creates a hierarchy of segment tracks, and we report the score of the best segment track among all levels. In addition, we adapt a recent tracking-by-detection approach [26] to our segment tracking problem (represented as Pairwise ([26]) in Table 2), in order to make a comparison between our long-term appearance models and tracking based on pairwise appearance similarities. The approach is based on dynamic programming on an adjacency graph. In our adoption we put 0 as the unary term (since we do not have detectors) and the similarity computed by the exponential χ^2 kernel on our feature descriptors as the pairwise terms connecting segments in adjacent frames. We also explored the built-in non-maximum suppression option [26], but found that it decreases performance in our case and therefore did not include it in the result.

Sequence/Object	SPT	SPT + CSI	[21]	[15]	MVPD	CPMC Best
Girl	1573	1564	1785	5777	**1304**	1164
Birdfall	**188**	242	288	305	252	199
Parachute	339	328	**201**	1202	235	242
Cheetah–Deer	983	1156	**905**	1219	1142	599
Monkeydog–Monkey	558	**483**	521	493	563	322
Penguin–#1	5026	5116	136285	2116	**1705**	3146

Table 1. Comparisons on SegTrack v1 using the error pixel metric. MVPD is an algorithm from [31] with a user-drawn initialization in the first frame. CPMC Best represents the average score for the best CPMC segment in each frame, respectively. CPMC Best is thus the theoretical upper bound (lower bound under this error metric) for the performance of SPT. However, it is possible that SPT+CSI can be better than CPMC Best since recombination of segments have been performed.

From the results one can see that SPT is consistent across all sequences and CSI is able to improve over its results in many sequences. SPT is able to reduce the over $1,000$ segments in each frame from CPMC down to about 60 segment tracks while still capturing most of the objects. This confirms that the temporal cues available in video can make unsupervised segmentation much stronger. The algorithm of [20] has impressive performance and outperforms SPT+CSI

Sequence/Object	SPT	SPT +CSI	Pairwise ([26])	[20]	[15]	CPMC Best
Mean per object	62.7	65.9	55.4	45.3	51.8	78.6
Mean per sequence	68.0	71.2	58.6	57.3	50.8	81.5
Girl	**89.1**	**89.2**	83.4	87.7	31.9	93.5
Birdfall	62.0	**62.5**	47.8	49.0	57.4	72.2
Parachute	93.2	93.4	91.3	**96.3**	69.1	95.5
Cheetah–Deer	40.1	37.3	18.3	**44.5**	18.8	67.0
Cheetah–Cheetah	**41.3**	40.9	22.2	11.7	24.4	66.6
Monkeydog–Monkey	58.8	71.3	24.1	**74.3**	68.3	83.0
Monkeydog–Dog	17.4	18.9	16.5	4.9	18.8	44.6
Penguin–#1	51.4	51.5	59.3	12.6	**72.0**	75.8
Penguin–#2	73.2	76.5	79.1	11.3	**80.7**	90.4
Penguin–#3	69.6	75.2	**75.6**	11.3	75.2	85.4
Penguin–#4	57.6	57.8	47.1	7.7	**80.6**	67.6
Penguin–#5	63.4	**66.7**	45.8	4.2	62.7	68.1
Penguin–#6	48.6	50.2	56.7	8.5	**75.5**	76.6
Drifting Car–#1	73.8	**74.8**	65.4	63.7	55.2	82.1
Drifting Car–#2	58.4	**60.6**	59.8	30.1	27.2	75.3
Hummingbird–#1	45.4	**54.4**	35.0	46.3	13.7	70.0
Hummingbird–#2	65.2	72.3	65.8	**74.0**	25.2	82.2
Frog	65.8	**72.3**	69.0	0	67.1	87.1
Worm	75.6	82.8	59.5	**84.4**	34.7	89.8
Soldier	83.0	**83.8**	50.7	66.6	66.5	84.3
Monkey	84.1	**84.8**	70.9	79.0	61.9	88.3
Bird of Paradise	88.2	**94.0**	81.1	92.2	86.8	94.7
BMX–Person	75.1	85.4	74.5	**87.4**	39.2	86.9
BMX–Bike	24.6	24.9	30.9	**38.6**	32.5	58.5
Avg. Number of Tracks	60.0	60.0	702.8	10.6	336.6	1219.3

Table 2. Overlap of the best segment from each algorithm on SegTrack v2. The **Mean per object** score is an average of the overlaps on all 24 objects. The **Mean per sequence** score is an average of mean overlaps on all 14 sequences, hence the 6 `Penguin` objects would not have an out-of-proportion impact on the average. See the text for more result analyses.

in some sequences with fairly small number of tracks, but fails in some others, mainly due to either multiple adjacent moving objects, or very slow movement in the case of `Frog`. Since their algorithm is dependent on motion-based saliency, the slow motion in `Frog` makes their algorithm unable to output any segment. The algorithm of [15] works well when the number of distinct colors inside the object of interest is not too large (`Penguin`, `Bird of Paradise`, `Monkeydog`), but fails otherwise because of wrong choices made in local superpixel aggregations. The pairwise tracking approach [26] works well when the appearance is consistent, but fails for highly deformable objects such as `Worm` and `Monkeydog--Monkey`.

6. Conclusion

In this paper we present a new unsupervised video segmentation approach by tracking a pool of holistic, figure-ground segments on each frame, generated by a multiple figure-ground segmentation algorithm. Long-term appearance models are learnt using a regression-to-overlap framework on many segment tracks initialized from all the segment proposals in the pool. By using the same training examples for many segment tracks, we are able to track

Stage	Time
Preprocessing:	
Segment Pool Generation	up to 3.5 mins
Feature Computation	29.3 secs
Tracking:	
Overlap Computation	0.75 secs
Training Models	1.46 secs
Testing and Matching	0.09 secs
Refinement:	
CSI Inference	0.56 secs (per track)

Table 3. Breakdown of the per-frame computation time for the first frame in the `Drifting Car` video. The segment generation and feature computation steps are still very slow at the moment, which we aim to improve in future work. However, the tracking framework is relatively fast once it has the features. Note in frames other than the first one, training and testing time scales linearly with the number of frames a segment can start on, but overlap computation takes shorter after pruning the targets.

hundreds of segments efficiently by exploiting structures in least squares regression. The learnt long-term appearance models are robust to partial occlusion, drift, appearance changes, and in the experiments are able to perform consistently well over many video sequences posing different challenges. Besides, an algorithm based on composite statistical inference is proposed to refine the segment tracks using the learnt appearance models as high-order potentials, and shown to be efficient while able to improve the appearance and temporal consistency in many sequences.

Acknowledgements: This work was supported in part by NSF project IIS-1016772 and ARO MURI W911NF-11-1-0046.

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008. 2

[2] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012. 1

[3] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 33:1619–1632, 2011. 1, 2

[4] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *IJCV*, 82:113–132, 2009. 2

[5] C. Bibby and I. Reid. Real-time tracking of multiple occluding objects using level sets. In *CVPR*, pages 1307–1314, 2010. 2

[6] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *ICCV*, pages 833 –840, 2009. 1, 2

[7] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 1, 2

[8] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *PAMI*, 33(3):500–513, 2011. 1

[9] I. Budvytis, V. Badrinarayanan, and R. Cipolla. Semi-supervised video segmentation using tree structured graphical models. In *CVPR*, pages 2257 –2264, 2011. 2

[10] J. Carreira, F. Li, and C. Sminchisescu. Object Recognition by Sequential Figure-Ground Ranking. *IJCV*, 2012. 1

[11] J. Carreira and C. Sminchisescu. CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts. *PAMI*, 2012. 1, 3, 6

[12] H.-T. Cheng and N. Ahuja. Exploiting nonlocal spatiotemporal structure for video segmentation. In *CVPR*, 2012. 2

[13] D. Cremers. Dynamical statistical shape priors for level set-based tracking. *PAMI*, 28(8):1262 –1273, 2006. 2

[14] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, pages 575–588, 2010. 1, 3

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

[16] Y. Huang, Q. Liu, and D. Metaxas. Video object segmentation by hypergraph cut. In *CVPR*, 2009. 2

[17] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *ICCV*, 2007. 1, 3

[18] J. Lee, S. Kwak, B. Han, and S. Choi. Online video segmentation by bayesian split-merge clustering. In *ECCV*, pages 856–869. 2012. 2

[19] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012. 1

[20] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011. 1, 2, 7

[21] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Efficient closed-form solution to generalized boundary detection. In *ECCV*, 2012. 3

[22] F. Li, J. Carreira, G. Lebanon, and C. Sminchisescu. Composite statistical inference for semantic segmentation. In *CVPR*, 2013. 2, 5, 6

[23] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *CVPR*, 2010. 1, 3

[24] F. Li, G. Lebanon, and C. Sminchisescu. A linear approximation to the chi2 kernel with geometric convergence. Technical report, arXiv:1206.4074, 2013. 4, 6

[25] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012. 2

[26] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 2, 7

[27] B. Price, B. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, 2009. 2

[28] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi. Tracking deforming objects using particle filtering for geometric active contours. *PAMI*, 29(8):1470–1475, 2007. 2

[29] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, 2007. 2

[30] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 6

[31] D. Tsai, M. Flagg, and J. M.Rehg. Motion coherent tracking with multi-label mrf optimization. In *BMVC*, 2010. 1, 2, 6

[32] M. Unger, M. Werlberger, T. Pock, and H. Bischof. Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In *CVPR*, 2012. 2

[33] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *PAMI*, 9:1582–1596, 2010. 4

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

[35] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *PAMI*, 34, 2012. 4

[36] T. Wang and J. Collomosse. Probabilistic motion diffusion of labeling priors for coherent video segmentation. *Multimedia, IEEE Transactions on*, 14(2):389 –400, april 2012. 2

[37] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012. 2

[38] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *ICCV*, 2009. 2