Combining Self Training and Active Learning for Video Segmentation

Alireza Fathi[1]
afathi3@gatech.edu
Maria Florina Balcan[1]
ninamf@cc.gatech.edu
Xiaofeng Ren[2]
xiaofeng.ren@intel.com
James M. Rehg[1]
rehg@cc.gatech.edu

[1] College of Computing
Georgia Institute of Technology
[2] Intel Labs Seattle

Abstract

This work addresses the problem of segmenting an object of interest out of a video. We show that video object segmentation can be naturally cast as a semi-supervised learning problem and be efficiently solved using harmonic functions. We propose an incremental self-training approach by iteratively labeling the least uncertain frame and updating similarity metrics. Our self-training video segmentation produces superior results both qualitatively and quantitatively. Moreover, usage of harmonic functions naturally supports interactive segmentation. We suggest active learning methods for providing guidance to user on what to annotate in order to improve labeling efficiency. We present experimental results using a ground truth data set and a quantitative comparison to a representative object segmentation system.

1 Introduction

Video object segmentation is an important problem in video analysis, and it has many applications, including post-production, special effects, object recognition, object tracking, and video compression. In particular, the rapidly-growing numbers of videos which are available from the web represent an opportunity for new video applications and analysis methods.

A key motivation for this paper is the observation that video-based object tracking and interactive video object segmentation are closely-related problems. In both cases, the goal is to segment an object from the video with a minimum number of annotations provided by the user. In tracking systems, the user identifies the object in the first frame (e.g. by drawing a contour around it) and it is tracked automatically in the remaining frames [6, 10]. However, these methods can fail in cases of occlusion or due to a severe change in object appearance. In contrast, recent systems for interactive video object segmentation provide very fine-grained control via a well-designed interface, making it possible for a user to get pixel-perfect segmentations [4, 17, 26].

Our goal is to address both of these problems within the same framework. In a tracking context, our approach makes it easy to fix up an existing solution by carving the object labeled in the first frame through out the video. Applications to biotracking are a motivating

example, as there is a strong need for a general purpose tool for tracking a wide range of animals with different morphologies. In contrast to video post-production, in biotracking applications segmentations which are not pixel-perfect but which delineate the limbs of the animal (for example) can still be useful for animal behavior experiments.

Likewise, in the context of interactive video object cutout, our approach leverages constraints on video data and an active learning approach to minimize the number of annotations that must be supplied by the user. In case of biotracking, it is necessary to minimize the need for guidance by the user in order to have a useful tool for biologists. As a result our method aims at getting the most benefit from each user click. To achieve this goal, we develop an active learning method which chooses the most important frames to be labeled, and which guides the user in each frame about where to click.

Our framework is based on casting video segmentation as a semi-supervised learning problem with video-specific structures such as temporal coherence which makes the following contributions:

A framework for object cutout and interactive segmentation: In this work we present a framework which addresses video object cutout and has a natural extension to interactive segmentation. We use semi-supervised learning to propagate labels from known locations to unknown, and *uncertainty* is a key in a effective label propagation. Furthermore, uncertainty is incorporated into active learning to guide the user in annotating the video. Tsai et al [20] formalize tracking as a video object cutout where only the first frame was annotated. A limitation of their approach is that it is not clear how to make additional annotations effectively. We propose a new framework that addresses this problem.

Incremental self-training: We develop an iterative solution to semi-supervised video segmentation. At each step, we pick the least uncertain frame, fix all labels in the frame, and update system parameters (e.g. object appearance models). We show that incremental self-training is very effective in adapting to video content, outperforming standard semi-supervised learning and state-of-the-art tracking systems.

Intelligent user guidance: We develop a systematic way to provide intelligent guidance to users in an interactive setting. This is by selecting the most informative frames to be labeled by user, and guiding the user while labeling each frame.

2 Previous Work

Semi-supervised Learning: Semi-supervised learning is a very active sub-field in machine learning (detailed survey provided by Zhu [28]) with wide applications in natural language processing and bioinformatics. Examples in computer vision include image classification [13] and tracking [27]. For segmentation, [23] showed a few examples of interactive segmentation, and[22] applied multiple instance learning to semantic segmentation. Furthermore Badrinarayanan and others [3, 9] use an EM based semi-supervised learning algorithm to propagate labels from the start and end of a video. In comparison, we propose a self-training approach which significantly improves the results.

Active Learning: Active learning is another growing sub-field in learning (detailed survey provided by Settles [19]). Zhu et. al [30] present an active-learning approach based on risk minimization of unlabeled nodes using harmonic functions. In this work, we demonstrate that uncertainty leads to a better solution than risk minimization for video segmentation. Other examples of active learning include classifying video [25], object categorization [14] and image co-segmentation [5]. Kohli and Torr [15] introduced a method for estimating uncertainty for graph-cut through an expensive post-processing which can be used for

active learning. Instead, we use a continuous formulation using harmonic functions which computes soft labels and naturally provides uncertainty.

Object tracking: Significant progress has been made recently on object tracking by various approaches where [2, 6, 10, 11] are a few representatives. Many of these trackers can produce object segmentations [6, 10, 18, 20]. Ren and Malik [18] showed that segmentation helped avoid drifting on long sports videos. Bibby and Reid [6] demonstrated the adaptation of target models and integration of multiple terms to track a wide range of targets. Chockalingam et. al [10] developed a level-set based system which tracks the target by dividing it into multiple regions combining spatially constrained appearance model and motion estimation. Tsai et al [20] developed a multi-label MRF model for offline tracking solved with Fast-PD. Our use of self-training in tracking is novel and yields promising results comparing to the state of the art [10, 20]. In addition we propose an interactive segmentation framework.

Interactive Segmentation: There is a huge literature on interactive image segmentation. The most relevant to our work is that of Grady's [12], which uses random walks for label prediction in images. For interactive video segmentation, a popular toolkit is LabelMe Video [26], where the user uses polygons to delineate objects. More complicated interactive trackers [4, 24] show segmentation results frame by frame, and let the user fix errors. These works are focused on pixel-perfect segmentations and user interface designs. In comparison, our goal is to intelligently guide the user to where to annotate and thus minimize user effort for each specific video. We propose a method that achieves a close to perfect segmentation after getting a few annotations from the user.

3 Video Segmentation Framework

The goal of this work is to address object tracking and interactive video segmentation in a unified framework. We want to enable robust extraction of objects from challenging videos (with large changes in shape, appearance and motion) using minimal user input. One application is biotracking: video segmentations desired in biotracking may not need to be pixel-perfect, but it has to handle a wide range of animals with very different morphologies, and it has to do with minimal effort from non-expert users.

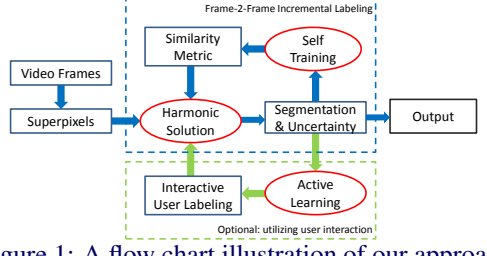


Figure 1: A flow chart illustration of our approach.

Figure 1 gives the flow chart for our approach. We formulate video segmentation as a semi-supervised learning problem on a graph of super-pixels. Harmonic functions provide an efficient solution to the graph labeling problem and produce soft labels, which we use to measure labeling uncertainties at both the super-pixel and the frame level. We then iteratively choose the least uncertain (or most certain) frame in the video, discretize the soft labels, and apply self-training to update similarity metrics and the affinity graph. We will demonstrate that incremental self-training significantly improves segmentation accuracy in comparison to standard baselines and state of the art segmentation-based tracking methods.

Our approach has a natural extension to interactive segmentation. We can present the

current segmentation to the user and ask for more input. To minimize user effort, we devise an effective scheme to intelligently suggest the user which frame and which super-pixel to label. Our scheme is empirically validated through simulation.

In the rest of this section we describe the semi-supervised learning formulation for video segmentation, its harmonic solution, and the underlying graph structures.

3.1 Semi-supervised learning and segmentation

Video object segmentation can be naturally viewed as a semi-supervised learning problem over a graph. In our case we treat super-pixels as data points, and our goal is to propagate the information from labeled ones to others. In a semi-supervised learning setting, there are $n = l + u$ data points x_i (l labeled and u unlabeled points, typically $u \gg l$), and the goal is to label the unlabeled points. One can formulate this problem on a graph $G = (V, E)$ with nodes $V = L + U$ and edges E, where L are the labeled nodes and U the unlabeled ones. In our approach, we represent connectivities in this graph by a $n \times n$ symmetric matrix W, where w_{ij} represents the similarity or affinity between two nodes x_i and x_j:

$$w_{ij} = exp(-g_{ij}^T \Sigma g_{ij}) \tag{1}$$

where $g_{ij} = x_i - x_j$ and Σ is the inverse covariance matrix. We assume Σ is diagonal and contains hyperparameters to scale the elements of g. In Section 4.1, we present a method for learning the weights in Σ.

The labeling problem in G can be solved using min-cut [7]. Zhu et. al [29] propose a framework which relaxes the min-cut objective function and leads to a simple algorithm with interesting behaviors. They look for a harmonic function $f : V \to \mathbb{R}$ that is real-valued on the unlabeled data U, but constrained to be $f_L \in \{0,1\}$ on the labeled data L. The intuition behind the harmonic solution is that it returns the probabilities of starting from unlabeled nodes and arriving at nodes with a particular label by randomly walking in the graph.

The harmonic solution f can be computed in polynomial time by simple matrix operations [29]. It is possible to represent the combinatorial laplacian $\Delta = D - W$ as four blocks separating labeled and unlabeled nodes:

$$\Delta = \begin{bmatrix} \Delta_{ll} & \Delta_{lu} \\ \Delta_{ul} & \Delta_{uu} \end{bmatrix}$$

and similarly split f into labeled (f_l) and unlabeled (f_u) parts. The harmonic solution subject to $f|_L = f_l$ is

$$f_u = \Delta_{uu}^{-1} \Delta_{ul} f_l \tag{2}$$

The formulation above considers only pairwise edges. It is easy to incorporate priors or unary edges: add an auxiliary node for each class, and connect it to all the nodes using priors as edge weights.

The harmonic solution has several key advantages for video segmentation comparing to mincut. It provides a real-valued solution where uncertainty (and related quantities such as risk) can be easily computed and optimized, crucial to our approach. Furthermore, examples in [12] and our experiments show that it tends to be less sensitive to noise and weak boundaries.

3.2 Graph construction for video segmentation

The graph G we use for video object segmentation has three types of edges: (1) spatial, (2) temporal and (3) prior edges. We model spatial coherencies using appearance and boundary saliency, and model temporal coherency using appearance and optical flow.

Spatial edges. Spatial edges connect the adjacent super-pixels inside each frame. Super-pixels have different sizes and shapes. We represent each super-pixel by its color and texture histograms. Texture descriptors [21] are computed for each pixel and quantized into the 256 nearest k-means centers. Similarly, color descriptors for each pixel are quantized into 128 k-means bins. The distance g^s_{ij} between a pair of super-pixels is the difference between their color and texture histograms. We concatenate g^s_{ij} with the saliency of the shared contour between i and j as in [1], obtaining a 385 dimensional feature vector. We compress these vectors to 64 dimensions using PCA. We multiply the spatial weights $w^s_{ij} = exp(-(g^s_{ij})^T \Sigma^s g^s_{ij})$ by the length of the shared contour to compensate for size differences between super-pixels.

Temporal edges. We compute dense optical flow between adjacent frames using [8]. A super-pixel i is linked with a super-pixel j in the adjacent frame if optical flow connects some of their underlying pixels. Similar to the spatial edges, we use color and texture histograms to compute the distance g^t_{ij} (384 dimensional) between i and j and compress the acquired vectors to 64 dimensions using PCA. We observe that the appearance-based distance g^t_{ij} can compensate for errors in motion estimation. We multiply $w^t_{ij} = exp(-(g^t_{ij})^T \Sigma^t g^t_{ij})$ by the number of corresponding pixels between the super-pixels as given by the flow field. In addition to optical flow, we also compute sparse SIFT features[16] correspondence. If we find SIFT correspondences between two super-pixels (i.e. the number of inliers above a threshold), we increment w_{ij} with a large constant.

Prior edges. The spatial and temporal edges capture the similarity between adjacent super-pixels. To add priors for super-pixels belonging to each of the K classes, we add K labeled nodes, one for each label class, to the graph and connect all the unlabeled nodes to them. We learn the weights for these connections as $w^k_i = exp(-(g_i)^T \Sigma^k g_i)$, where g_i contains the color and texture histograms for super-pixel i.

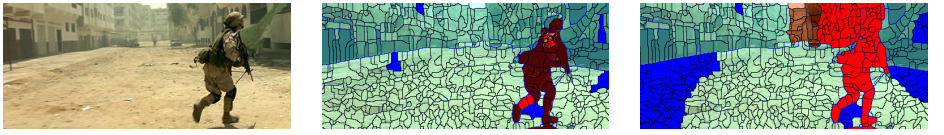
We find that spatial and temporal edges are sufficient in many cases to produce good results. In some cases (e.g. the penguin sequence in SegTrack dataset [10, 20]), putting a significant weight on the object appearance model can hurt segmentation accuracy, because the object looks similar to the background. On the other hand, for some videos with inaccurate optical flow, it is necessary to have prior knowledge about object appearance.

The final edge weight W is a linear combination of the individual weights $W = W^s + \alpha W^t + \beta(W^0 + \cdots + W^{K-1})$. Our system learns and updates the similarity metrics Σ^s, Σ^t, $\Sigma^0, ..., \Sigma^{K-1}$ iteratively in the self-training process, thereby automatically adapting the features to the particular sequence.

4 Incremental Labeling using Self-Training

In the previous section, we presented a semi-supervised learning formulation of video segmentation and showed how to construct the graph over super-pixels in all video frames. Directly solving for the unlabeled nodes over this graph, through either mincut or the harmonic solution, does not work well (see Table 1). This is partly because it is impossible to find a set of parameters (in particular feature weights Σ) that work for all videos, and partly because objects change appearance over time and temporal links between frames can be noisy for large motions.

To address these issues, we develop an incremental approach where we iteratively label frames and apply self-training to adapt model parameters. At each step, we compute the harmonic solution, choose the least uncertain frame and fix (discretize) the labels in that frame. Afterwards, we use linear regression to update and balance the feature weights Σ.



(a) (b) (c)

Figure 2: Self-training fails on a per-superpixel basis, showing the need for frame-level inference. (b) Segmentation based on harmonic solution (user labels in red and blue). (c) Superpixel based self-training fails after 100 iterations. Bright red and blue show labels fixed after self-training. Per-frame self-training works well on this sequence (see more examples in Fig 4).

4.1 Per-frame self-training

Self-training is a commonly used semi-supervised learning technique [28] which trains a classifier on the labeled data and applies it on the unlabeled data. In our studies, we find that self-training, when used in the context of spatial-temporal coherencies in a video graph, can lead to much better results than direct graph-based solution of mincut.

What is special about videos? Videos consist of a sequence of frames ordered in time. Video frames are closely coupled within themselves, and the associations between frames are "Markov". The object appearance in a frame is usually closest to its appearance in adjacent frames. As a result, conditioned on the labels in a frame, the labels in adjacent frames can be predicted with a high certainty.

We leverage the special structures in videos to assign labels iteratively. At each step, we compute the uncertainty of each unlabeled frame, and select the one with least uncertainty. This frame is often the one adjacent to the last labeled frame. We fix (discretize through thresholding) the labels in this new frame given the harmonic solution. After adding these nodes in the new frame to the set of labeled nodes L, we update the spatial, temporal and unary similarity metrics (Σ^s, Σ^t and $\Sigma^0, \ldots, \Sigma^{K-1}$) using the labels f_L. We use the new metrics to update the adjacency matrix W and the Laplacian Δ, estimate the new soft labels f_u, fix the labels for the next least uncertain frame, and iterate. In each iteration the final labels for a new frame are fixed, and the algorithm iterates until all frames are labeled.

The frame uncertainty (entropy) H of a frame $\mathcal{F}$ is calculated by adding the uncertainty of all super-pixels belonging to $\mathcal{F}$ weighted by their size:

$$H(\mathcal{F}) = \sum_{i \in \mathcal{F}} H(f_i)\mathcal{S}(i) \tag{3}$$

where $\mathcal{S}(i)$ is the area of super-pixel i and $H(f_i)$ is its entropy:

$$H(f_i) = \sum_{k=0}^{K-1} -log(f(i,k))f(i,k) \tag{4}$$

A plausible alternative, which ignores the video structures, is to iteratively apply thresholding and self-training, but at the super-pixel level (i.e. fixing the label for the least uncertain super-pixel at each step) instead of the frame level. This strategy often results in poor segmentations; an example can be found in Fig 2.

4.2 Metric learning and weight balancing

The role of the inverse covariance matrices Σ is to assign weights to the features in g, such as color vs texture, to capture intra similarities and inter differences between object and background super-pixels. A perfect metric, both for spatial or temporal edges, has the following property

Sequence	GraphCut	GraphCut + Self Training	Ours	[20]	[10]	Average Object Size	Number of Frames
Parachute	254	253	251	502	**235**	3683	51
Girl	4121	1616	**1206**	1755	1304	8160	21
Monkey-dog	1312	3727	598	683	**563**	1440	71
Penguin	19569	19569	**1367**	6627	1705	20028	42
Bird-fall	454	766	342	454	**252**	495	30
Cheetah	1961	898	**711**	1217	1142	1584	29
Soldier	3344	2484	**1368**	2984	2228	6321	32
Monkey-water	1306	1266	**1009**	4142	2814	6011	31

Table 1: We compare our tracking results with [10, 20] using the average number of errors (pixels) per frame. Our results outperforms these methods in most of the sequences, and achieve marginal results to [20] in the rest. We further compare our method to graph-cut and graph-cut with self-training given the same set of parameters.

$$w_{ij} = e^{-g_{ij}^T \Sigma g_{ij}} = \begin{cases} 1 & if\ f_i = f_j \\ 0 & if\ f_i \neq f_j \end{cases} \tag{5}$$

where $f_i, f_j \in \{0, \cdots, K-1\}$ are the labels. It means we want the similarity weight between two super-pixels belonging to the same object to be high. Assuming that Σ is diagonal, it is more convenient to write $w_{ij} = exp(-\sum_{d=1}^{m} \sigma(d) g_{ij}(d)^2)$ where $\sigma(d)$ is the d-th diagonal element of Σ, and $g_{ij}(d)$ is the d-th element of the feature difference $g_{ij} = x_i - x_j$.

It is very hard, if not impossible, to find a set of weights $\sigma(d)$ that can work for a variety of videos. On the other hand, once the system has seen a specific video and acquired some labels, it can search for a custom metric Σ which satisfies the property of Eq 5 over the known labels. That is, we want $\Sigma_{d=1}^{m} \sigma(d) g_{ij}(d)^2$ to be 0 for the case of $w_{ij} = 1$ and ∞ for $w_{ij} = 0$. We use linear regression to estimate the $\sigma(d)$'s and find that regression works well with a limited amount of training data. Similarly, for unary metrics Σ^k, if super-pixel i in the labeled data belongs to class k, we force w_i^k to be 1 and 0 otherwise, and the metrics $\Sigma^0, \cdots, \Sigma^{K-1}$ are also learned through linear regression.

5 Assisted Interactive Segmentation

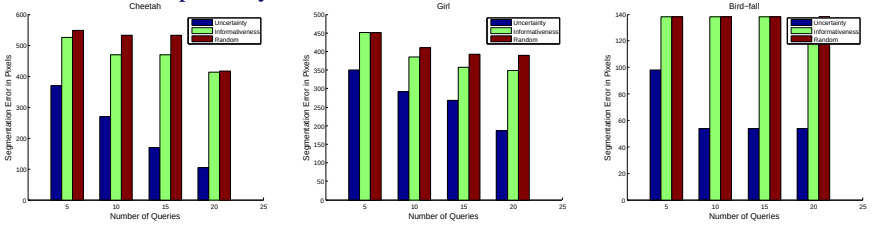
In this section we introduce an efficient interactive framework which achieves a close to perfect video segmentation with the minimum amount of user input. Our interactive segmentation method consists of two steps. In the first step our algorithm selects the most informative frame from the set of unlabeled frames and asks the user to label it. Then in the second step, our algorithm guides the user in annotating the frame by iteratively suggesting the next best super-pixel to be labeled.

5.1 Selecting the most informative frame

Previous interactive segmentation algorithms[4, 17] start from the first frame of the video and ask the user to annotate frames sequentially one after the other. This is necessary for acquiring a pixel-perfect segmentation. However, our goal is to reach a close to perfect video segmentation with as few user inputs as possible. As a result we intend to incorporate the user input for only a few frames instead of getting the user to annotate every single frame.

Method	Parachute	Girl	Monkey-dog	Penguin	Bird-fall	Cheetah	Soldier	Monkey
Most Uncertain	115	366	520	296	4	343	538	410
Random	122	412	344	329	4	428	769	513
Sequential	132	464	530	1171	45	445	1011	864
Initial Error	145	595	536	1291	45	497	1035	887

Table 2: The results of asking the user to annotate the most uncertain frames versus annotating the frames sequentially and randomly. Our method often achieve a better segmentation error given the same amount of user effort. In all cases we simulate annotating 5 frames. We compute the error based on the area of difference between groundtruth super-pixel labels and the labels computed by different schemes.



(a) (b) (c)

Figure 3: We show average number of erroneous pixels in a random frame, after t iterations of interactively labeling super-pixels for 3 representetive sequences. We have compared the average segmentation error per frame given each method after $5, 10, 15, 20$ queries.

This means that we have to seek a smart way of selecting those few frames in order to acquire the most useful information from the user.

Here we propose an active learning strategy based on selecting the most uncertain frame $\mathcal{F}$ at each step and we show this method outperforms random or sequential frame selection approaches. Frame uncertainty is computed in the same way as in self-training (described in Eq 3). After the user is satisfied and finished with labeling a frame $\mathcal{F}$, the harmonic solution is updated to incorporate the new information, and the procedure is repeated. We compare the user effort required in this scheme against the approach of asking the user to annotate the frames sequentially or randomly.

5.2 Selecting the most informative super-pixel

When labeling each frame, our algorithm iteratively suggests the next superpixel to be labeled by the user. The user is asked to perform a click at each iteration: left click for object and right click for background. We explore different strategies for selecting the next superpixel. We first evaluate a standard active learning technique based on risk minimization. Zhu et. al [30] compute the expected risk as $\sum_{k=1}^{K} f(i,k)\hat{\mathcal{R}}(f^{+x(x_i,k)})$, where $\hat{\mathcal{R}}(f^{+x(x_i,k)})$ is the risk after adding node i with label k. The risk is defined and estimated in [30] as the generalization error of Bayes classifier. We can suggest to the user the superpixel which minimizes the risk. Empirically, we find that risk minimization does not produce good suggestions, likely because the risk estimation is often inaccurate.

The second strategy we consider is querying the superpixel with the highest uncertainty. The most uncertain nodes are always located near the boundary between classes in the current labeling. We found this method more effective for our problem. The reason is that once the algorithm reaches a close to accurate segmentation, uncertainty based method queries super-pixels at the ambiguous areas on object boundary and refines the object mask. We

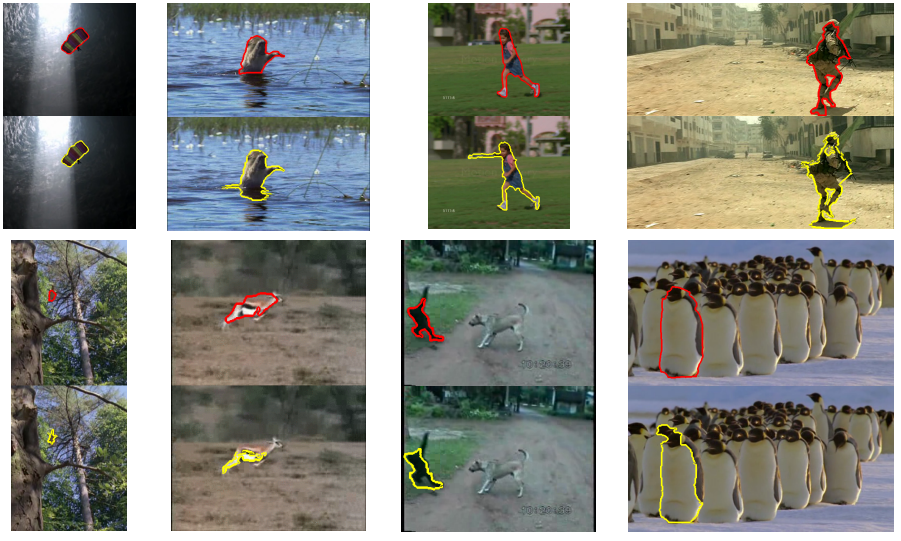


Figure 4: We qualitatively compare our results with [20]. Our results are shown with red contours, and the results from [20] with yellow contours. The sequences in the first row are *parachute*, *monkey-water*, *girl*, *soldier* and the ones in second row are *bird-fall*, *cheetah*, *monkey-dog* and *penguin*. The contours produced by our algorithm are smoother, and the segmentations are usually more accurate.

compare these strategies with randomly selecting the next super-pixel.

6 Experiments

We provide results both on object tracking and on interactive video segmentation. We perform our experiments on 8 challenging sequences with groundtruth segmentation [10, 20]: parachute, girl, monkey-dog, penguin, bird-fall, cheetah, soldier and monkey-water. The first six are the videos used in SegTrack dataset [20] in order to fairly compare our work with previous methods. The length of these sequences are between 21 to 71 frames. In Table 1 we show that our approach produces better segmentations than [10], and also compares favorably to [20]. We further show qualitative comparisons in Figure 4.

As a baseline, in Table 1 we also compare to a standard graphcut solution and graph-cut combined with our self-training method using the same adjacency graph. In the results of standard graph-cut, the object mask either shrinks or expands over time. The self-training method is meant to solve this issue by iteratively segmenting frames one after the other. The combination of graph-cut and self-training produces better results than graph-cut alone, however still the shrinking bias in graph-cut is an issue. We show that the combination of self-training and harmonic solution produces the best result.

For interactive segmentation, we compare different strategies for frame selection in Table 2, and those for superpixel selection in Figure 3. To compare different frame selection strategies we simulate the user behavior using the ground-truth segmentation. In each iteration we select the next frame based on the scheme criteria (the most uncertain frame, random, sequential), use the ground truth to label it, and recalculate the segmentation. We empirically compare the three methods in Table 2.

We compare super-pixel selection strategies in Fig 3. We start with the first frame labeled,

and simulate different strategies using the ground-truth labels instead of getting input from an actual user. Selecting the next super-pixel based on uncertainty outperforms both the risk minimization proposed in [30] and random selection. The reason is that given the first frame mask, the object is segmented throughout the video with a high accuracy, and the uncertainty based strategy queries super-pixels at the object boundary and can improve the result faster.

7 Conclusion

We have proposed a semi-supervised learning approach to video segmentation using harmonic functions. We show that an incremental self-training approach, iteratively labeling the least uncertain frame and updating similarity metrics, outperforms the state of the art on challenging video benchmarks. Our approach naturally extends to an interactive setting, where active learning can guide a user to most informative locations, minimizing user effort required in video object segmentation.

8 Acknowledgment

Portions of this research were supported in part by NSF Award 0916687 and ARO MURI award 58144-NS-MUR.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: an empirical evaluation. In *CVPR*, 2009.

[2] S. Avidan. Ensemble tracking. In *CVPR*, 2005.

[3] V. Badrinarayanan, F. Galasso, and R. Cipolla. Label propagation in video sequences. In *CVPR*, 2010.

[4] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. In *SIGGRAPH*, 2009.

[5] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010.

[6] C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *ECCV*, 2008.

[7] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, 2001.

[8] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[9] I. Budvytis, V. Badrinarayanan, and R. Cipolla. Label propagation in complex video sequences using semi-supervised learning. In *BMVC*, 2010.

[10] P. Chockalingam, N. Pradeep, and S. T. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, 2009.

[11] R. Collins, Y. Liu, and M. Leordeanu. On-line selection of discriminative tracking features. In *PAMI*, 2005.

[12] L. Grady. Random walks for image segmentation. In *PAMI*, 2006.

[13] M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal semi-supervised learning for image classification. In *CVPR*, pages 902–909. IEEE, 2010.

[14] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *ICCV*, 2007.

[15] P. Kohli and P. Torr. Measuring uncertainty in graph cut solutions. In *ECCV*, 2006.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2): 91–110, 2004.

[17] B. L. Price, B. Morse, and S. Cohen. Livecut: learning-based interactive video segmentation by evaluation of multiple propagated cues. In *CVPR*, 2010.

[18] X. Ren and J. Malik. Tracking as repeated figure-ground segmentation. In *CVPR*, 2007.

[19] B. Settles. Active learning literature survey. In *CS Tech Report 1648, University of Wisconsin-Madison*, 2009.

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

[21] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. In *IJCV*, 2005.

[22] A. Vezhnevets and J. M. Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *CVPR*, 2010.

[23] F. Wang, C. Zhang, H. C. Shen, and J. Wang. Semi-supervised classification using linear neighborhood propagation. In *CVPR*, 2006.

[24] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. In *ACM Trans. Graph.*, 2005.

[25] R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multi-class active learning. In *ICCV*, 2003.

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

[27] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof. On-line semi-supervised multiple-instance boosting. In *CVPR*, 2010.

[28] X. Zhu. Semi-supervised learning literature survey. *University of Wisconsin-Madison*, 2008.

[29] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

[30] X. Zhu, Z. Ghahramani, and J. Lafferty. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML workshop*, 2003.