Minimizing Human Effort in Interactive Tracking by Incremental Learning of Model Parameters

Arridhana Ciptadi and James M. Rehg
School of Interactive Computing
Georgia Institute of Technology
`arridhana,rehg@gatech.edu`

Abstract

We address the problem of minimizing human effort in interactive tracking by learning sequence-specific model parameters. Determining the optimal model parameters for each sequence is a critical problem in tracking. We demonstrate that by using the optimal model parameters for each sequence we can achieve high precision tracking results with significantly less effort. We leverage the sequential nature of interactive tracking to formulate an efficient method for learning model parameters through a maximum margin framework. By using our method we are able to save $\sim 60 - 90\%$ *of human effort to achieve high precision on two datasets: the VIRAT dataset and an Infant-Mother Interaction dataset.*

1. Introduction

The past decade has seen an explosive growth of video data. The ability to easily annotate/track objects in videos has the potential for tremendous impact across multiple application domains. For example, in computer vision annotated video data can be used as an extremely valuable source of information for the training and evaluation of object detectors (video provides continuous view of how an object's appearance might change due to viewpoint effects). In sports, video-based analytics is becoming increasingly popular (*e.g.* the Italian company Deltatre employed 96 people to pour over multiple video footage for live player tracking during the 2014 World Cup). In behavioral science, video has been used to assist the coding of children's behavior (*e.g.* for studying infant attachment [1], typical development [12] and autism [8]).

The problem of object tracking in video has a long history in computer vision. The tracking problem is challenging because of the often dramatic appearance variations in the object being tracked (*e.g.*, due to lighting and viewpoint change) and occlusions. As a result, fully-automated high precision object tracking remains an open problem. Note that getting accurate object tracks is important in many applications. For example, biologists who use video to monitor the movement of animals care about accurately tracking these animals at all times. Errors in tracking are unacceptable since they can contaminate the research findings. To obtain practically useful accurate tracking, several *interactive approaches* have been pursued (*e.g.*, LabelMe Video [22] and the crowdsourcing method of Vondrick *et al*. [18]). Unfortunately, most existing interactive tracking approaches are not optimized for human effort. However, minimizing human annotation effort is extremely important in practice since video can be prohibitively expensive to label (*e.g.*, twenty six hours of surveillance video cost tens of thousands of dollars to annotate despite using a state-of-the-art annotation system [10]).

In this paper, we propose an interactive tracking system that is designed to minimize the amount of human effort required to obtain high precision tracking results. We achieve this by leveraging user annotations for incrementally learning *instance specific* model parameters within the tracking cost function. This is in contrast to the common practice of hand-tuning the model parameters on a training set and applying the same fixed parameters on any new testing data. This approach is both time consuming (due to hand-tuning) and gives suboptimal accuracy on individual tracking instances. We cast the problem of learning the optimal model parameters as the problem of learning a structured prediction model in a maximum margin framework. Our key insight is that the incremental nature of an interactive tracking process is particularly well-suited for *efficient* maximum margin learning of model parameters. We show that our approach significantly outperforms the current best practice of using hand-tuned model parameters on two datasets: the VIRAT challenge dataset and the Infant-Mother Interaction dataset recently introduced in [11]. The main contribution of this paper is an annotation-driven maximum margin framework for efficiently learning instance-specific model parameters.

2. Related Work

Early work in interactive tracking focused on creating a system that can quickly incorporate a new annotation given by the user during the interactive stage to refine the tracking result [3, 21]. The goal was to enable the user to quickly evaluate the quality of the tracking result and decide whether additional annotation is necessary. To achieve this, Buchanan and Fitzgibbon [3] combined an efficient data structure based on K-D tree and a dynamic programming approach for interactive feature tracking. The K-D tree allows for fast lookup of patches with similar appearance, while dynamic programming provides an efficient solution for inferring the trajectory of the tracking target. Wei *et al*. [21] used the dynamic programming approach proposed by Buchanan and Fitzgibbon and combined it with object detection to build an interactive object tracking system. The basic idea is that given some initial annotations, an interactive tracking system should be able to anticipate likely object locations in a given frame by performing object detection (with a conservative threshold). This allows the system to more quickly respond to the user's input during the interactive stage to perform object trajectory optimization.

Another line of work in interactive tracking focuses on interpolation strategies. Wei and Chai [20] propose a weighted template model (based on color histogram) for interpolating object appearance. The idea is that the appearance of the target object in all frames can be adequately described by a linear combination of the appearance of the object in the annotated frames. The LabelMe video work by Yuen *et al*. [22] presents a strategy for interpolating the location of the target object in between keyframes by using homography-preserving linear interpolation. Using linear interpolation to infer an object trajectory is an efficient alternative to the dynamic programming approach presented in [3, 21], but it assumes that annotations are performed densely such that the object moves linearly between the annotated frames. To achieve good tracking results by using linear interpolation, Vondrick *et al*. [18] estimated that on average 1 out of every 5 frames would need to be annotated.

A further line of work in interactive tracking focuses on frame selection strategies to minimize the number of annotations that a user will need to perform to obtain good tracking results. Vondrick and Ramanan [19] propose an active learning framework for interactive tracking. They present an approach for deciding which frame to present based on the expected change in the tracking result if the user were to annotate that frame (similar to the popular maximum expected gradient length (EGL) algorithm for active learning [13]). In the video segmentation domain, Fathi *et al*. [7] present an active learning approach based on using frame uncertainty to decide which frame to annotate. Their approach is based on the assumption that the frame with the highest uncertainty estimate is the one that will be the most informative for segmentation purposes. Vijayanarasimhan and Grauman [17] present a frame selection method for video segmentation based on expected label propagation error. In contrast to these works, our focus is not on the selection of the best frame for the user to annotate. Rather, our goal is to utilize the annotation information more effectively for the task of interactive tracking. Our approach exploits the sequential nature of an interactive tracking process for online *incremental* learning of a structured model.

3. Object Tracking

In this section we describe our framework for estimating object track in a video. We first give a description of the object representation technique that we use (Section 3.1). We then present the formulation for estimating the object trajectory given a set of observations (Section 3.2). Finally, we describe an efficient approach to optimize the object trajectory in Section 3.3.

3.1. Object Representation

We represent an object by using a joint histogram of oriented gradients (HOG) [6] and 3D color histogram: $x = [HOG\ RGB]$.^{T} HOG has been shown to achieve good results in many tasks that require compact object representation [9, 6]. The same observation has been made for the color histogram [4, 5].

To model the global appearance of the object, we use a discriminative approach. For each annotated frame, we use the annotated bounding box and some perturbed version of it as the positive instances and extract a large number of negative bounding boxes that do not overlap (or have very minimal overlap) with the annotation. To learn the object model, we use the positive and negative instances to train a linear SVM. In every frame we detect K object candidates using the learned model (we use a very conservative value of $K = 500$ to avoid false negatives).

3.2. Tracking Model

Our task is to track an object in an image sequence of length T frames. An object track is a set of T object locations $Y = \{y_t\}_{t=1...T}$. With each y_t is associated x_t, our object representation based on HOG and color histogram. The set of all x_t is denoted as X.

A track is initialized by bounding box annotations l_i made by the user in a set of *keyframes*. Note that the user could select only a single keyframe. The annotations are represented by their locations $L = \{l_i\}_{i \in N}$, with $1 \leq i \leq T$ and $|N| \leq T$. Under this model, a tracking algorithm can be described as a method that takes L as an input and returns Y, the trajectory of the object for the entire image sequence.

Given the description above, we now define the cost function that serves as a measure of the track quality:

$$\begin{aligned} E(Y;w) &= \sum_t e(y_t;w) &(1)\\ e(y_t;w) &= [w_1 w_2 w_3] \begin{bmatrix} d(x_t) \\ s_{app}(x_t, x_{t-1}) \\ s_{mot}(y_t, y_{t-1}) \end{bmatrix} &(2) \end{aligned}$$

where $d(\cdot)$ is the cost of deviating from the global appearance model of the object (we use the SVM score), $s_{app}(\cdot)$ is the appearance smoothness cost, and $s_{mot}(\cdot)$ is the cost of deviating from the location predicted by optical flow. The contribution of $d(\cdot)$, $s_{app}(\cdot)$, and $s_{mot}(\cdot)$ to the overall cost is described by the parameters of the cost function: $w = [w_1, w_2, w_3]$. Note that the value of these parameters significantly impacts the tracking performance for a given video (see Section 4).

In this formulation, the tracking problem is reduced to finding the trajectory Y that minimizes the cost function $E(Y;w)$. In addition, we also have to ensure that the hard constraints of $y_i = l_i$ for all $i \in N$ are satisfied. In order to be robust to occlusion, we augment Y with an occlusion flag to reduce the penalty when an object undergoes occlusion.

3.3. Tracking Optimization

The task is to find the best track Y that minimizes the cost function described in Equation 2 subject to the constraints $y_i = l_i$ for all $i \in N$. If we assume there are K candidate locations for the object in each frame, a naive approach to finding the best track would take $\mathcal{O}\binom{K}{T}$ time. Fortunately, this problem exhibits optimal substructure that lends itself to an efficient dynamic programming (DP) solution (interested reader can refer to [2, 3] for more details).

Let K_t be the set of object candidates at frame t. Let y_t^k be the k-th candidate location of the object at frame t. Let $C_t(y_t^k)$ be the cumulative cost of the track up until y_t^k, if y_t^k is picked as a part of the object track. We can compute $C_t(y_t^k)$ for all $t \in T, k \in K_t$ in $\mathcal{O}(TK^2)$ by using forward recursion:

$$\begin{aligned} C_0(y_0^k) &= w_1 d(x_0^k) \\ C_t(y_t^k) &= w_1 d(x_t^k) + \min_{j \in K_{t-1}} P_{t-1}^j(y_t^k) \\ P_{t-1}^j(y_t^k) &= C_{t-1}(y_{t-1}^j) + w_2 s_{app}(x_t^k, x_{t-1}^j) \\ &\quad + w_3 s_{mot}(y_t^k, y_{t-1}^j) \end{aligned} \qquad (3)$$

To obtain the best track, we can store the pointer to the match in the previous frame (Eq. 4) and backtrack from the location with the lowest cost in the last frame in T.

$$M_t^k(y_t^k) = \arg\min_{j \in K_{t-1}} P_{t-1}^j(y_t^k) \qquad (4)$$

To ensure that the track satisfies the hard constraints $y_i = l_i$ for all $i \in N$, we simply set $d(x_t^k) = -\infty$ for all of the manually annotated locations l_i. Similar to [3], to account for occlusion we augment the set of object candidates in each frame with an occlusion state ($[y_t]_{occ} = 1$ means the object is occluded), effectively modifying the cost function into the following:

$$E(Y;w) = \sum_t \begin{cases} e(y_t;w) & [y_t]_{occ} = 0 \\ \lambda_o & [y_t]_{occ} = 1, [y_{t-1}]_{occ} = 0 \\ \lambda_r & [y_t]_{occ} = 1, [y_{t-1}]_{occ} = 1 \\ \lambda_v & [y_t]_{occ} = 0, [y_{t-1}]_{occ} = 1 \end{cases} \qquad (5)$$

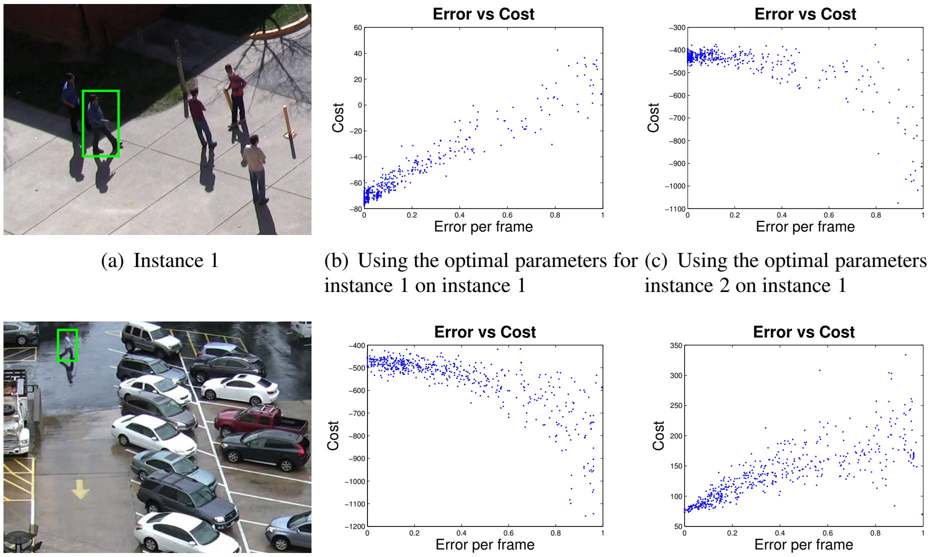
We set $\lambda_v = \lambda_o$, and $\lambda_r = 0.4\lambda_o$, so there is only one parameter to choose a value for.

This optimization method is very efficient. It takes less than 2 seconds to compute the globally optimal solution for $T = 1000$ and $K = 500$. That means that for every new annotation that a user has made, he/she can immediately observe how it affects the tracking result. This is a very desirable property for an interactive system. Note that this formulation has been been used in a number of interactive tracking work [3, 21, 19]. Thus, our approach to improve the cost function (Sec. 4) applies more broadly.

4. Instance Specific Tracking Model

An important question that needs to be addressed is how do we *weight* the contributions of the different parts of the cost function. In other words, how do we select the appropriate values for $w = [w_1, w_2, w_3]$ in Equation 2? Currently, a popular solution for this parameter selection task is hand-tuning: the parameters that minimize the average training error are identified and used for all new testing videos. There are three problems with this approach: 1) There is no single value that is optimal for all of the possible testing videos. This is a major problem from the perspective of highly accurate tracking in which every video is important, as by minimizing the average error we accept the possibility of large error on specific video clips; 2) It can be very time consuming to exhaustively search for the best parameter value; and 3) Adding new components to the cost function requires substantial additional work. For example, if we want to incorporate an additional way to model global appearance into the cost function, we have to redo the parameter search step.

To illustrate the problem of using a single set of weights for all videos, consider two instances of a basic tracking task illustrated in Figure 1: tracking a person in the parking lot with other people around (instance 1) and without (instance 2). We sample a number of object trajectories that are close to the groundtruth trajectory and we compute the cost (according to (2)) for each of these trajectories with



(a) Instance 1

(b) Using the optimal parameters for instance 1 on instance 1

(c) Using the optimal parameters for instance 2 on instance 1

(d) Instance 2

(e) Using the optimal parameter for instance 1 on instance 2

(f) Using the optimal parameter for instance 2 on instance 1

Figure 1. Error vs cost for two different sets of parameter values. We sample a number of trajectories that are close to the groundtruth, and we plot the error for each of these trajectories under two different parameter settings. Note that the optimal parameter value for one instance can result in a bad model for the other instance in the sense that low cost is assigned to the trajectories that in fact have high error. In these scatter plots the ideal distribution is a line with a slope of 1, reflecting a cost function which matches the error relative to groundtruth.

two different weight values, that correspond to the optimal weights for instances 1 and 2 (these values are computed by using our approach presented in Section 4.1). In Figure 1(e) we can see that an optimal set of weights for instance 1 results in a very bad model for instance 2 (and vice versa) where the trajectories that have more error actually have less cost. Note that even though in both instances we are tracking people, the *context* is different. In video 1 there are other objects with similar appearance in the scene (other people), in video 2 there are no objects present with similar appearance. Ideal weight parameters should reflect this difference in the nature of the tracking problem. Indeed, our approach is able to learn that very little weight should be assigned to the global appearance in instance 1 (since there are other people in the scene with very similar appearance) and instead the motion should be emphasized. In the subsequent sections we present our approach to incrementally learning the optimal value of the weight parameters *for each object trajectory in an interactive setting*.

4.1. Learning The Optimal Weights

In tracking optimization, the goal is to find a trajectory that has the lowest cost. The underlying assumption is that the groundtruth object trajectory has the lowest cost compared to all other possible trajectories. Therefore, by optimizing the cost function, we can obtain the groundtruth trajectory. Let Y^{gt} be the groundtruth trajectory. We can express this property mathematically as follow:

$$E(Y;w) > E(Y^{gt};w) \qquad \forall Y \neq Y^{gt} \tag{6}$$

We have discussed in the previous section how the choice of w plays a critical role in determining the validity of the above assumption. If this assumption is violated, then optimizing the cost function is a fool's errand because it does not reflect the quality of the trajectory. In interactive tracking, this translates into the user having to provide substantial manual annotations to correct for tracking mistakes, which are inevitable since the costs are wrong. This is extremely wasteful given that a better choice of w could greatly alleviate this problem.

Our goal is to find the optimal value for the weight parameter w for each tracking instance such that the groundtruth configuration has the lowest cost. The inequalities in (6) can have infinitely many solutions (*e.g.* a simple scaling of w will not change the inequality since our cost function is linear in w). A common trick to resolve this type of issue is to frame the problem as a maximum margin learning problem where the task is to find w that will maximize the margin between the groundtruth trajectory and all other trajectories:

$$\begin{gathered} \min \tfrac{1}{2}||w||^2 \\ E(Y;w) - E(Y^{gt};w) \geq 1 \qquad \forall Y \neq Y^{gt} \end{gathered} \tag{7}$$

Due to the modeling limitation of the cost function and noise in the data, the above program may not have any solution. To address this issue we add the slack variables ξ_n. Thus we allow the individual margins to be smaller, but this is discouraged by adding the slack variables into the objective.

$$\begin{gathered} \min \tfrac{1}{2}||w||^2 + \tfrac{C}{N}\sum_n \xi_n \\ E(Y;w) - E(Y^{gt};w) \geq 1 - \xi_n \qquad \forall Y \neq Y^{gt} \end{gathered} \tag{8}$$

The program described above assigns unit margin to all of the trajectories that are not groundtruth (0-1 loss). While this should work well in an ideal scenario, if there is noise in the data the algorithm might produce suboptimal results since the optimization enforces the same margin on all of the trajectories (*i.e.* the same weight is assigned to all of the trajectories). The algorithm will be more likely to produce the desired result if we can instead use a better loss measure. This is the essence of the maximum margin learning approach to structured prediction [15], which we would adopt.

In tracking, we can measure loss by using the Hamming distance between a trajectory and the groundtruth trajectory $\Delta(Y, Y^{gt})$. In this sense, we can view the problem of learning the optimal weight parameter in tracking as an instance of maximum margin structured prediction learning. By using the Hamming distance as our loss measure, the constraints in (8) now become:

$$E(Y;w) - E(Y^{gt};w) \geq \Delta(Y, Y^{gt}) - \xi_n \quad \forall Y \neq Y^{gt} \tag{9}$$

The above constraint means that we desire larger margin for the trajectories that are further from the groundtruth. Or in other words, this loss-scaled margin means that the trajectories that are further from the groundtruth should have higher cost than the trajectories that are closer (smaller margin). This is certainly a very desirable property for a cost function. Unfortunately, it is not feasible to solve the above program due to two factors: 1) we do not know the groundtruth trajectory Y^{gt}; and 2) there are an exponential number of constraints (K^T assuming there are K object candidates in every frame in a T-frame long video sequence).

During the interactive tracking process, a user incrementally adds one annnotation at a time. As a result of this, we obtain a series of trajectory estimates $Y^1, Y^2, \ldots Y^N$ (assuming the user has made N annotations) where Y^{i+1} is likely to be closer to the groundtruth than Y^i. Our insight is that we can exploit this process to incrementally learn w. So instead of using the groundtruth trajectory (which we do not have) as the positive instance for max margin learning, *we can use the current best estimate of the trajectory as the positive instance* and perform the optimization over a much smaller set of constraints that correspond to the other previously estimated trajectories that we have obtained during the interactive tracking process. So for every new annotation a user has made, we can estimate the parameter value that will make the most recent trajectory estimate have the lowest cost. This process is aligned with our original formulation where we desire parameters that will make the cost function assign lower cost to the trajectory that is closer to the groundtruth (*i.e.* the latest trajectory estimate Y^N) compared to the trajectories that are further from the groundtruth (*i.e.* other previously obtained trajectories $Y^1, Y^2, \ldots Y^{N-1}$). We can implement this as the following optimization:

$$\begin{gathered} \min \tfrac{1}{2}||w||^2 + \tfrac{C}{N}\sum_{i=1}^{N} \xi_i \\ E(Y^i;w) - E(Y^N;w) \geq \Delta(Y^i, Y^N) - \xi_i \quad {\scriptstyle i=1\ldots N-1} \\ w_j \geq 0 \qquad \forall w_j \in w \end{gathered} \tag{10}$$

By solving the above program after every annotation, we are guaranteed to have w that assigns the lowest cost to the latest trajectory estimate (within some slack tolerance). Note that if the user annotated the whole video sequence ($N = T$), the above program reduces to the original formulation in Equation 9, but with a much smaller set of constraints.

To account for the fact that we now optimize over a significantly smaller set of constraints compared to the original formulation in (9), we add an additional set of constraints to enforce every single element of w to be nonnegative. This is a subtle but important addition since this set of constraints serve as a way to represent the trajectories that are far from the groundtruth in the optimization. Many of the high loss trajectories will have high values of $d(\cdot)$, $s_{app}(\cdot)$ or $s_{mot}(\cdot)$. Consider for example a trajectory that jumps from one corner of the image to a different corner in successive frames. This trajectory will have a very high $s_{mot}(\cdot)$ (similar examples can be drawn for the other two components of the cost function, $d(\cdot)$ and $s_{app}(\cdot)$). Since our constraint set consists of only trajectories that are close to the groundtruth, it will most likely not contain examples of those high-loss trajectories. Because of this, there is a possibility that we obtain a negative w which can result in the high-loss trajectories (which are not represented in the constraint set) to obtain the lowest cost. Adding the nonnegativity constraint for w alleviates this problem.

To illustrate the result of our incremental learning of w, let's revisit our earlier example of tracking a person in the presence of other people (Fig. 1(a)). Due to the existence of similar looking objects in the scene (other people), we know that intuitively the global appearance component should carry less weight in the overall cost function. Our incremental weight learning approach is able to quickly learn this context information (see Table 1). Also note how given the same set of annotations, the w that we learn incrementally results in a better cost function for the problem (which is reflected by the lower error rate).

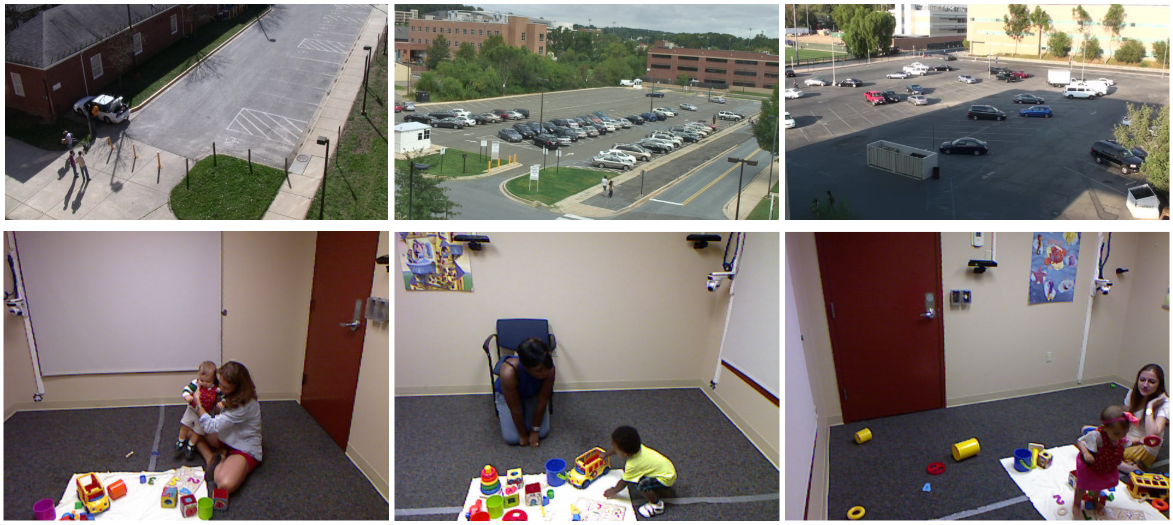


Figure 2. Dataset used for experiments: VIRAT dataset (top row) and Infant-Mother Interaction dataset (bottom row).

N **annotations**	w_1	w_2	w_3	**Error/frame**
1	0.33	0.33	0.33	0.5100
2	0.18	0.46	0.36	0.3800
3	0.08	0.40	0.52	0.0733
4	0.03	0.37	0.60	0.0367

Table 1. Incremental learning of w. This table illustrates the effect of our incremental learning of the cost function parameters. We annotate a 300-frame long sequence at 4 uniformly-spaced locations, and we perform trajectory estimation given those annotations with 4 different w values (the starting w and w that is learned incrementally after annotations 2, 3 and 4). Note that our approach is able to learn to place less and less weight on the global appearance cost (w_1) since there are many similar-looking objects in the scene (Fig. 1(a)).

4.2. Improving The Objective

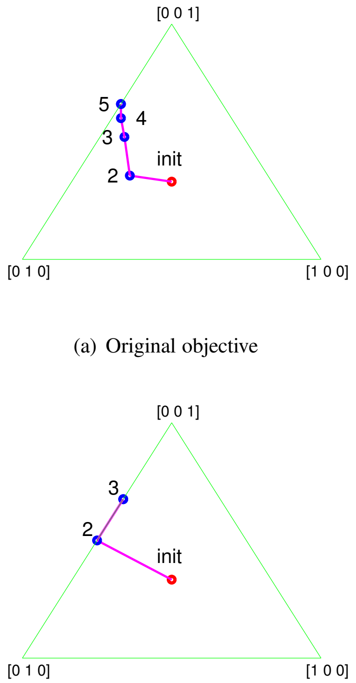
A potential problem with the loss-scaled constraint in Equation 10 is that the algorithm may give a suboptimal solution since it focuses on the constraints with high loss. Since we scale the margin by the loss, a w that gives Y^N the lowest energy (which is our goal) may not be selected in the optimization if there are any high-loss constraints that do not have a large enough margin. This means that the earlier trajectory estimates (which are the constraints that have high loss) can potentially overwhelm the ultimate objective which is finding a w that gives the most recent trajectory estimate Y^N the lowest cost. In order to compensate for this, we can add directly to the objective the difference between the cost of the two latest trajectory estimates, given a w parameter ($E(Y^N; w) - E(Y^{N-1}; w)$). This can be interpreted as putting more emphasis for the algorithm to search for the solution that maximizes the separation between the two data points that are closest to the decision boundary. This acts as a counter-weight to the high loss constraints. The final objective then becomes the following:

$$\min \frac{1}{2}||w||^2 + \frac{C_1}{N}\sum_{i=1}^{N}\xi_i + C_2(E(Y^N; w) - E(Y^{N-1}; w)) \tag{11}$$

This formulation is similar to Szummer *et al.* [14] and Tsochantaridis *et al.* [16] but is adapted to our sequential formulation.

To illustrate the effect of the new objective on the parameter learning process, we consider once more the interactive tracking task in Figure 1(a) (tracking a person in the presence of other people). We start with $w = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ and perform interactive tracking on the sequence by doing annotation one frame at a time. We use the same annotation schedule (same set of frames with the same annotation ordering) and compare the convergence behavior of the two objectives. Starting from the initial annotation, after each subsequent annotation we compute the optimal w according to the two objectives. We normalize the w to sum to 1 and plot its value on a simplex (note that normalizing w does not change the inequality constraints in (10)). Figure 3 illustrates the convergence behavior of the two objectives.

Notice that even though both objectives essentially converged to the same value (both learned to place no weight on the global appearance due to the presence of similar looking objects in the scene), the improved objective found the optimal parameter value much more quickly than the original objective, converging after only 3 annotations instead of 5. Our hypothesis is that the additional term in the objective allows the algorithm to quickly converge to the optimal solution by admitting a solution that does not provide enough margin to the high loss constraint (in this case, constraint induced by the first trajectory estimate Y^1). We look at the value of the slack variable ξ_1 after annotation 3 to confirm



(a) Original objective

(b) Improved objective

Figure 3. Convergence behavior of the original objective (10) and the improved objective (11) on the tracking instance in Fig. 1(a) (tracking an object in the presence of similar looking objects). The green simplex is the solution space. Red dot is the starting value of w and the blue dots are the value of w after each annotation. Note how the improved objective converged quicker to the optimal solution.

our hypothesis, and indeed the value of this variable in the new objective is higher than that in the original objective. This confirms our idea that the additional term in the new objective can serve as a balancing term to the high loss constraints, allowing the algorithm to focus more on the solution that maximizes the separation between data points that are closest to the decision boundary.

5. Experiments

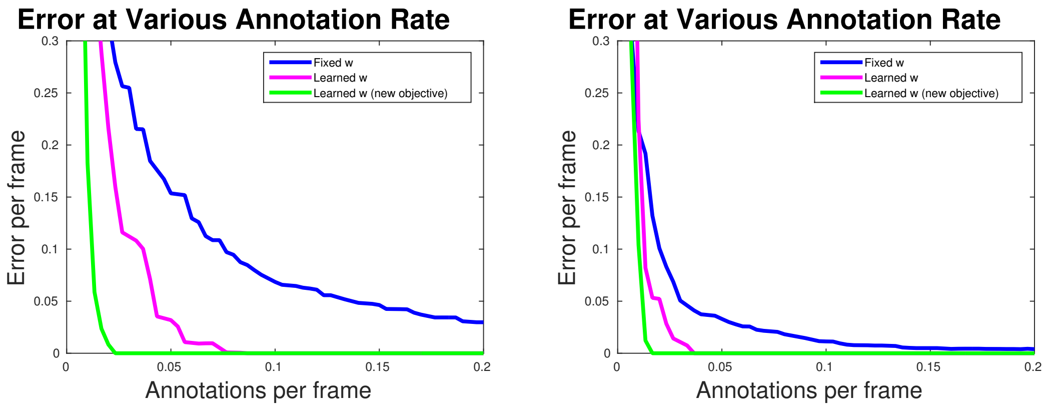
To demonstrate the advantage of our instance specific max-margin tracking parameter learning approach, we perform experiments on two datasets: 1) the VIRAT challenge dataset [10]; and 2) an Infant-Mother Interaction dataset recently introduced in [11]. The VIRAT dataset consists of over 300 surveillance videos captured in various locations. Our task in this dataset is to track moving people and cars (in VIRAT there are a lot of objects that are completely stationary, which are trivial to track). The Infant-Mother Interaction dataset consists of 15 videos of a dyadic interaction between an infant and a mother in a room during a behavioral study called The Strange Situation [1]. This dataset serves as an important practical application for interactive tracking since being able to obtain high precision tracks of the people in the scene has a tremendous amount of utility for quantifying the behavior of individuals. The task in this dataset is to track the heads of the people in the scene. A representative set of frames from the two datasets can be seen in Figure 2.

We compare our incremental weight learning approach against the traditional fixed-weight approach (hand-tuned to each of the datasets). We use MATLAB's quadprog as our quadratic program solver. We use HOG with bin size 8 and 3D color histogram with bin size 216 for our object representation. For each interactive tracking task, we employ a uniform annotation frame selection strategy where in each step, the tracker requests an annotation on a frame that will make the temporal distribution of the annotations as uniform as possible (*i.e.* the first annotation will be requested from the middle frame of the video, the second from the middle of the second half of the video, and so forth).

We measure tracking error based on how well the tracker is able to estimate the groundtruth annotations. For every frame, an object track is considered to be correct if its intersection over union with the groundtruth is at least 0.3 (a similar metric is used in [19]). We quantify tracking error by the error-per-frame metric (*i.e.* an error of 0.01 means that for every 100 frames there is 1 frame where the IoU is less than 0.3). To quantify human effort, we use the annotations-per-frame metric (an annotations-per-frame of 0.1 means that a user annotated 10 frames out of 100). For an interactive tracking system, the goal is to obtain high precision tracking results with *as few annotations* as possible. To capture this, for each experiment we report the number of annotations-per-frame that is required from the user to achieve a certain error-per-frame target (we report results on high precision target error ranging from 0 to 0.05).

The results for the VIRAT dataset can be seen in Figure 4(a). Our approach is able to outperform the fixed weight approach by a large margin. For example, on average by learning the weight parameter during the annotation process using our method, we are able to achieve 0.04 error tracking results using only 0.017 annotations-per-frame, compared to the 0.17 annotations-per-frame that is required by the fixed weight approach. This is an improvement of 90% which means that by using our approach, we can annotate this dataset to the same desired accuracy with only 10% of the effort. This can potentially translate to a saving in the order of tens of thousands of dollars for a dataset this size. Also note that the improved objective that we propose gives a considerable improvement over the standard maximum margin objective.

Similar to the VIRAT dataset, our approach is able to significantly improve the annotation efficiency in the Infant-



(a) Results on VIRAT dataset

(b) Results on Infant-Mother Interaction dataset

Figure 4. Results on VIRAT and the Infant-Mother Interaction dataset. y-axis is the error rate, x-axis is the annotations rate. Our approach is able to significantly outperform the standard practice of using fixed w.

Mother Interaction dataset (see Figure 4(b)). For the target error rate of 0.04, our approach is able to achieve the same tracking accuracy with only 32.5% of the human effort (going from 0.04 annotations-per-frame to 0.013). Note that the Infant-Mother Interaction dataset represents the ideal dataset for the hand-tuned fixed weight approach since on the surface there seems to be minimal variations in the scene (there is only one type of target object and all of the videos are captured in the same room). However, even in this setup our approach is still able to provide a large improvement. This means that even on videos captured from similar scene with the same type of target object, there is always a significant variability in the individual tracking instances. Note that as is the case in VIRAT, the proposed new objective gives the best results.

6. Conclusion

We have presented an approach to address a critical problem in tracking: determining the parameter value of the cost function. We leverage the sequential nature of interactive tracking to formulate an efficient approach for learning instance specific model parameters through a maximum margin framework. We have demonstrated that by using our approach we can save human effort for annotation by $\sim 60-90\%$ to achieve high precision tracking results, a significant improvement in efficiency compared to the existing approach.

Acknowledgments

The authors would like to thank Dr. Daniel Messinger and The Early Play and Development Laboratory at the University of Miami for providing the videos used in the Infant-Mother Interaction Dataset. Portions of this work were supported in part by NSF Expedition Award number 1029679 and the Intel Science and Technology Center in Embedded Computing.

References

[1] M. D. S. Ainsworth, M. C. Blehar, E. Waters, and S. Wall. *Patterns of attachment: A psychological study of the strange situation.* Lawrence Erlbaum, 1978. 1, 7

[2] R. E. Bellman and S. E. Dreyfus. Applied dynamic programming. 1962. 3

[3] A. Buchanan and A. Fitzgibbon. Interactive feature tracking using kd trees and dynamic programming. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2006. 2, 3

[4] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005. 2

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002. 2

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005. 2

[7] A. Fathi, M.-F. Balcan, X. Ren, and J. M. Rehg. Combining self training and active learning for video segmentation. In *British Machine Vision Conference (BMVC)*, 2011. 2

[8] C. Lord, M. Rutter, P. DiLavore, S. Risi, and K. Gotham. *ADOS: Autism Diagnostic Observation Schedule*. Western Psychological Services, 2008. 1

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2

[10] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011. 1, 7

[11] E. B. Prince, K. Martin, D. Gangi, R. Jia, D. Messinger, A. Ciptadi, A. Rozga, and J. M. Rehg. Automated measurement of dyadic interactions predicts expert ratings of attach-

ment in the strange situation. *Association for Psychological Science Annual Convention*, 2015. 1, 7

[12] J. M. Rehg et al. Decoding children's social behavior. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013. 1

[13] B. Settles. Active learning literature survey. *Computer Sciences Technical Report 1648, University of Wisconsin Madison*, 2009. 2

[14] M. Szummer, P. Kohli, and D. Hoiem. Learning crfs using graph cuts. In *European Conference on Computer Vision (ECCV)*, pages 582–595. Springer, 2008. 6

[15] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: A large margin approach. In *Proceedings of the 22nd International Conference on Machine learning*. ACM, 2005. 5

[16] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning Research*, pages 1453–1484, 2005. 6

[17] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. In *European Conference on Computer Vision (ECCV)*. Springer, 2012. 2

[18] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013. 1, 2

[19] C. Vondrick and D. Ramanan. Video annotation and tracking with active learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2011. 2, 3, 7

[20] X. K. Wei and J. Chai. Interactive tracking of 2d generic objects with spacetime optimization. In *European Conference on Computer Vision (ECCV)*. Springer, 2008. 2

[21] Y. Wei, J. Sun, X. Tang, and H.-Y. Shum. Interactive offline tracking for color objects. In *International Conference on Computer Vision (ICCV)*. IEEE, 2007. 2, 3

[22] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *International Conference on Computer Vision (ICCV)*. IEEE, 2009. 1, 2