2015 IEEE International Conference on Computer Vision

Multiple Hypothesis Tracking Revisited

Chanho Kim † Fuxin Li ‡† Arridhana Ciptadi † James M. Rehg †
† Georgia Institute of Technology ‡ Oregon State University

Abstract

This paper revisits the classical multiple hypotheses tracking (MHT) algorithm in a tracking-by-detection framework. The success of MHT largely depends on the ability to maintain a small list of potential hypotheses, which can be facilitated with the accurate object detectors that are currently available. We demonstrate that a classical MHT implementation from the 90's can come surprisingly close to the performance of state-of-the-art methods on standard benchmark datasets. In order to further utilize the strength of MHT in exploiting higher-order information, we introduce a method for training online appearance models for each track hypothesis. We show that appearance models can be learned efficiently via a regularized least squares framework, requiring only a few extra operations for each hypothesis branch. We obtain state-of-the-art results on popular tracking-by-detection datasets such as PETS and the recent MOT challenge.

1. Introduction

Multiple Hypotheses Tracking (MHT) is one of the earliest successful algorithms for visual tracking. Originally proposed in 1979 by Reid [36], it builds a tree of potential track hypotheses for each candidate target, thereby providing a systematic solution to the data association problem. The likelihood of each track is calculated and the most likely combination of tracks is selected. Importantly, MHT is ideally suited to exploiting higher-order information such as long-term motion and appearance models, since the entire track hypothesis can be considered when computing the likelihood.

MHT has been popular in the radar target tracking community [6]. However, in visual tracking problems, it is generally considered to be slow and memory intensive, requiring many pruning tricks to be practical. While there was considerable interest in MHT in the vision community during the 90s, for the past 15 years it has not been a mainstream approach for tracking, and rarely appears as a baseline in tracking evaluations. MHT is in essence a breadth-first search algorithm, hence its performance strongly depends on the ability to prune branches in the search tree quickly and reliably, in order to keep the number of track hypotheses manageable. In the early work on MHT for visual tracking [12], target detectors were unreliable and motion models had limited utility, leading to high combinatoric growth of the search space and the need for efficient pruning methods.

This paper argues that the MHT approach is well-suited to the current visual tracking context. Modern advances in tracking-by-detection and the development of effective feature representations for object appearance have created new opportunities for the MHT method. First, we demonstrate that a modern formulation of a standard motion-based MHT approach gives comparable performance to state-of-the-art methods on popular tracking datasets. Second, and more importantly, we show that MHT can easily exploit high-order appearance information which has been difficult to incorporate into other tracking frameworks based on unary and pairwise energies. We present a novel MHT method which incorporates long-term appearance modeling, using features from deep convolutional neural networks [20, 16]. The appearance models are trained online for each track hypothesis on all detections from the entire history of the track. We utilize online regularized least squares [25] to achieve high efficiency. In our formulation, *the computational cost of training the appearance models has little dependency on the number of hypothesis branches,* making it extremely suitable for the MHT approach.

Our experimental results demonstrate that our scoring function, which combines motion and appearance, is highly effective in pruning the hypothesis space efficiently and accurately. Using our trained appearance model, we are able to cut the effective number of branches in each frame to about 50% of all branches (Sec. 5.1). This enables us to make less restrictive assumptions on motion and explore a larger space of hypotheses. This also makes MHT less sensitive to parameter choices and heuristics (Fig. 3). Experiments on the PETS and the recent MOT challenge illustrate the state-of-the-art performance of our approach.

‡ This work was conducted while the 2nd author was at Georgia Tech.

1550-5499/15 $31.00 © 2015 IEEE
DOI 10.1109/ICCV.2015.533

2. Related Work

Network flow-based methods [35, 4, 45, 10] have recently become a standard approach to visual multi-target tracking due to their computational efficiency and optimality. In recent years, efficient inference algorithms to find the globally optimal solution [45, 4] or approximate solutions [35] have been introduced. However, the benefits of flow-based approaches come with a costly restriction: the cost function can only contain unary and pairwise terms. Pairwise costs are very restrictive in representing motion and appearance. In particular, it is difficult to represent even a linear motion model with those terms.

An alternative is to define pairwise costs between tracklets – short object tracks that can be computed reliably [26, 3, 18, 8]. Unfortunately the availability of reliable tracklets cannot be guaranteed, and any mistakes propagate to the final solution. In Brendel et al. [8], data association for tracklets is solved using the Maximum Weighted Independent Set (MWIS) method. We also adopt MWIS, but follow the classical formulation in [34] and focus on the incorporation of appearance modeling.

Collins [11] showed mathematically that the multidimensional assignment problem is a more complete representation of the multi-target tracking problem than the network flow formulation. Unlike network flow, there is no limitation in the form of the cost function, even though finding an exact solution to the multidimensional assignment problem is intractable.

Classical solutions to multidimensional assignment are MHT [36, 12, 17, 34] and Markov Chain Monte Carlo (MCMC) data association [19, 32]. While MCMC provides asymptotic guarantees, MHT has the potential to explore the solution space more thoroughly, but has traditionally been hindered by the exponential growth in the number of hypotheses and had to resort to aggressive pruning strategies, such as propagating only the M-best hypotheses [12]. We will show that this limitation can be addressed through discriminative appearance modeling.

Andriyenko [1] proposed a discrete-continuous optimization method to jointly solve trajectory estimation and data association. Trajectory estimation is solved by spline fitting and data association is solved via MRF inference. These two steps are alternated until convergence. Segal [37] proposed a related approach based on a message passing algorithm. These methods are similar to MHT in the sense that they directly optimize a global energy with no guarantees on solution quality. But in practice, MHT is more effective in identifying high quality solutions.

There have been a significant number of prior works that exploit appearance information to solve data association. In the network flow-based method, the pairwise terms can be weighted by offline trained appearance templates [38] or a simple distance metric between appearance features [45]. However, these methods have limited capability to model the complex appearance changes of a target. In [17], a simple fixed appearance model is incorporated into a standard MHT framework. In contrast, we show that MHT can be extended to include online learned discriminative appearance models for each track hypothesis.

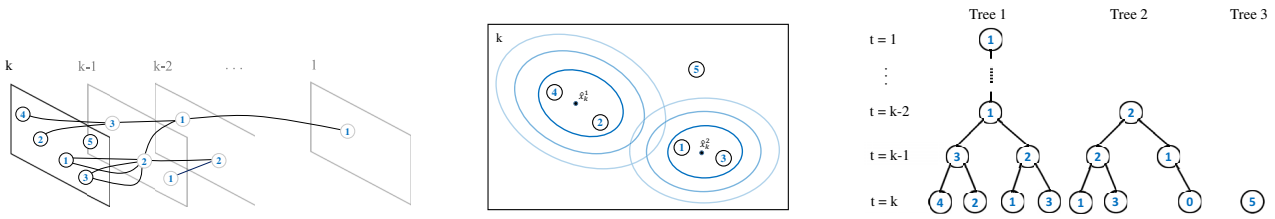
Online discriminative appearance modeling is a standard method for addressing appearance variation [39]. In tracklet association, several works [2, 42, 21, 22] train discriminative appearance models of tracklets in order to design a better affinity score function. However, these approaches still share the limitations of the tracklet approach. Other works [7, 40] train a classifier for each target and use the classification score for greedy data association or particle filtering. These methods only keep one online learned model for each target, while our method trains multiple online appearance models via multiple track hypotheses, which is more robust to model drift.

3. Multiple Hypotheses Tracking

We adopt a tracking-by-detection framework such that our observations are localized bounding boxes obtained from an object detection algorithm. Let k denote the most recent frame and M_k denote the number of object detections (i.e. observations) in that frame. For a given track, let i_k denote the observation which is selected at frame k, where $i_k \in \{0, 1, \ldots, M_k\}$. The observation sequence $i_1, i_2, \ldots, i_k$ then defines a **track hypothesis** over k frames. Note that the dummy assignment $i_t = 0$ represents the case of a missing observation (due to occlusion or a false negative).[1] Let the binary variable $z_{i_1 i_2 \ldots i_k}$ denote whether or not a track hypothesis is selected in the final solution. A **global hypothesis** is a set of track hypotheses that are not in conflict, i.e. that do not share any measurements at any time.

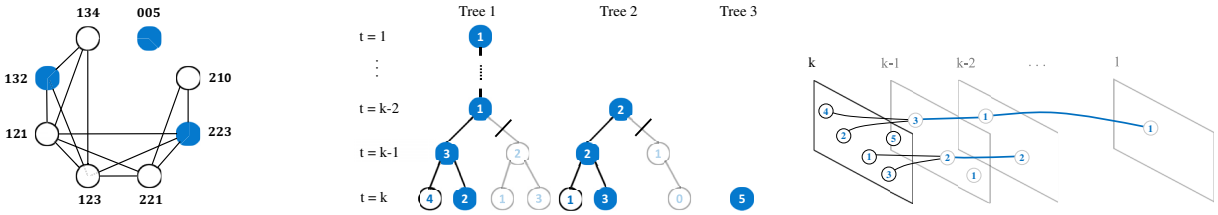
A key strategy in MHT is to delay data association decisions by keeping multiple hypotheses active until data association ambiguities are resolved. MHT maintains multiple track trees, and each tree represents all of the hypotheses that originate from a single observation (Fig. 1c). At each frame, the track trees are updated from observations and each track in the tree is scored. The best set of non-conflicting tracks (the best global hypothesis) can then be found by solving a maximum weighted independent set problem (Fig. 2a). Afterwards, branches that deviate too much from the global hypothesis are pruned from the trees, and the algorithm proceeds to the next frame. In the rest of this section, we will describe the approach in more detail.

[1] For notational convenience, observation sequences can be assumed to be padded with zeros so that all track hypotheses can be treated as fixed length sequences, despite their varying starting and ending times.



(a) Tracks in Video Frames (b) Gating (c) Track Trees

Figure 1. Illustration of MHT. (a) Track hypotheses after the gating test at time k. Only a subset of track hypotheses is visualized here for simplicity. (b) Example gating areas for two track hypotheses with different thresholds d_{th}. (c) The corresponding track trees. Each tree node is associated with an observation in (a).



(a) MWIS (b) N-scan Pruning (c) Remaining Track Hypotheses

Figure 2. (a) An undirected graph for the example of Fig. 1 in which each track hypothesis is a node and an edge connects two tracks that are conflicting. The observations for each hypothesis in the last three frames are indicated. An example of the Maximum Weighted Independent Set (MWIS) is highlighted in blue. (b) An N-scan pruning example ($N = 2$). The branches in blue contain the global hypothesis at frame k. Pruning at $t = k - 2$ removes all branches that are far from the global hypothesis. (c) Track hypotheses after the pruning. The trajectories in blue represent the finalized measurement associations.

3.1. Track Tree Construction and Updating

A track tree encapsulates multiple hypotheses starting from a single observation. At each frame, a new track tree is constructed for each observation, representing the possibility that this observation corresponds to a new object entering the scene.

Previously existing track trees are also updated with observations from the current frame. Each track hypothesis is extended by appending new observations located within its gating area as its children, with each new observation spawning a separate branch. We also always spawn a separate branch with a dummy observation, in order to account for missing detection.

3.2. Gating

Based on the motion estimates, a gating area is predicted for each track hypothesis which specifies where the next observation of the track is expected to appear.

Let $\mathbf{x}_k^l$ be the random variable that represents the likely location of the l^{th} track at time k. The variable $\mathbf{x}_k^l$ is assumed to be normally distributed with mean $\hat{\mathbf{x}}_k^l$ and covariance Σ_k^l determined by Kalman filtering. The decision whether to update a particular trajectory with a new observation i_k is made based on the Mahalanobis distance d^2 between the observation location $\mathbf{y}_{i_k}$ and the predicted location $\hat{\mathbf{x}}_k^l$:

$$d^2 = (\hat{\mathbf{x}}_k^l - \mathbf{y}_{i_k})^\top (\Sigma_k^l)^{-1} (\hat{\mathbf{x}}_k^l - \mathbf{y}_{i_k}) \leq d_{\text{th}}. \quad (1)$$

The distance threshold d_{th} determines the size of the gating area (see Fig. 1b).

3.3. Track Scoring

Each track hypothesis is associated with a track score. The l^{th} track's score at frame k is defined as follows:

$$S^l(k) = w_{\text{mot}} S^l_{\text{mot}}(k) + w_{\text{app}} S^l_{\text{app}}(k) \quad (2)$$

where $S^l_{\text{mot}}(k)$ and $S^l_{\text{app}}(k)$ are the motion and appearance scores, and w_{mot} and w_{app} are the weights that control the contribution of the location measurement $\mathbf{y}_{i_k}$ and the appearance measurement X_{i_k} to the track score, respectively.

Following the original formulation [6], we use the log likelihood ratio (LLR) between the target hypothesis and the null hypothesis as the motion score. The target hypothesis assumes that the sequence of observations comes from the same target, and the null hypothesis assumes that the sequence of observations comes from the background. Then

the l^{th} track's motion score at time k is defined as:

$$S^l_{\text{mot}}(k) = \ln \frac{p(\mathbf{y}_{i_{1:k}}|i_{1:k} \subseteq T_l)}{p(\mathbf{y}_{i_{1:k}}|i_{1:k} \subseteq \phi)} \tag{3}$$

where we use the notation $i_{1:k}$ for the sequence of observations $i_1, i_2, ..., i_k$. We denote by $i_{1:k} \subseteq T_l$ the target hypothesis that the observation sequence comes from the l^{th} track and we denote the null hypothesis by $i_{1:k} \subseteq \phi$. The likelihood factorizes as:

$$\frac{p(\mathbf{y}_{i_{1:k}}|i_{1:k} \subseteq T_l)}{p(\mathbf{y}_{i_{1:k}}|i_{1:k} \subseteq \phi)} = \frac{\prod_{t=1}^{k} p(\mathbf{y}_{i_t}|\mathbf{y}_{i_{1:t-1}}, i_{1:t} \subseteq T_l)}{\prod_{t=1}^{k} p(\mathbf{y}_{i_t}|i_t \subseteq \phi)} \tag{4}$$

where we assume that measurements are conditionally independent under the null hypothesis.

The likelihood for each location measurement at time t under the target hypothesis is assumed to be Gaussian. The mean $\hat{\mathbf{x}}^l_t$ and the covariance Σ^l_t are estimated by a Kalman filter for the measurements $\mathbf{y}_{i_{1:t-1}}$. The likelihood under the null hypothesis is assumed to be uniform. The factored likelihood terms at time t are then written as:

$$\begin{aligned} p(\mathbf{y}_{i_t}|\mathbf{y}_{i_{1:t-1}}, i_{1:t} \subseteq T_l) &= \mathcal{N}(\mathbf{y}_{i_t}; \hat{\mathbf{x}}^l_t, \Sigma^l_t), \\ p(\mathbf{y}_{i_t}|i_t \subseteq \phi) &= 1/V \end{aligned} \tag{5}$$

where V is the measurement space [6, 12], which is the image area or the area of the ground plane for $2.5D$ tracking.

The appearance track score is defined as:

$$S^l_{\text{app}}(k) = \ln \frac{p(X_{i_{1:k}}|i_{1:k} \subseteq T_l)}{p(X_{i_{1:k}}|i_{1:k} \subseteq \phi)} = \ln \frac{p(i_{1:k} \subseteq T_l|X_{i_{1:k}})}{p(i_{1:k} \subseteq \phi|X_{i_{1:k}})} \tag{6}$$

where we obtain the posterior LLR under the assumption of equal priors. The posterior ratio factorizes as:

$$\frac{p(i_{1:k} \subseteq T_l|X_{i_{1:k}})}{p(i_{1:k} \subseteq \phi|X_{i_{1:k}})} = \frac{\prod_{t=1}^{k} p(i_t \subseteq T_l|i_{1:t-1} \subseteq T_l, X_{i_{1:t}})}{\prod_{t=1}^{k} p(i_t \subseteq \phi|X_{i_t})} \tag{7}$$

where we utilize $\{i_{1:k} \subseteq T_l\} = \bigcup_{t=1}^{k} \{i_t \subseteq T_l\}$ for the factorization. We assume that $i_t \subseteq T_l$ is conditionally independent of future measurements $X_{i_{t+1:k}}$ and the $i_t \subseteq \phi$ hypotheses are independent given the current measurement X_{i_t}.

Each term in the factored posterior comes from the online learned classifier (Sec. 4) at time t. Given prior observations $i_{1:t-1}$, we define the posterior of the event that observation i_t is in the l^{th} track as:

$$p(i_t \subseteq T_l|i_{1:t-1} \subseteq T_l, X_{i_{1:t}}) = \frac{e^{F(X_{i_t})}}{e^{F(X_{i_t})} + e^{-F(X_{i_t})}} \tag{8}$$

where $F(\cdot)$ is the classification score for the appearance features X_{i_t} and the classifier weights are learned from $X_{i_{1:t-1}}$. We utilize the constant probability c_1 for the posterior of the background (null) hypothesis.

$$p(i_t \subseteq \phi|X_{i_t}) = c_1 \tag{9}$$

The track score expresses whether a track hypothesis is more likely to be a true target ($S^l(k) > 0$) or false alarm ($S^l(k) < 0$). The score can be computed recursively [6]:

$$S^l(k) = \quad S^l(k-1) + \Delta S^l(k), \tag{10}$$

$$\Delta S^l(k) = \begin{cases} \ln \frac{1-P_D}{1-P_{FA}} \approx \ln(1-P_D), & \text{if } i_k = 0 \\ w_{\text{mot}} \Delta S^l_{\text{mot}}(k) + w_{\text{app}} \Delta S^l_{\text{app}}(k), & \text{otherwise} \end{cases} \tag{11}$$

where P_D and P_{FA} (assumed to be very small) are the probabilities of detection and false alarm, respectively. $\Delta S^l_{\text{mot}}(k)$ and $\Delta S^l_{\text{app}}(k)$ are the increments of the track motion score and the track appearance score at time k and are calculated using Eqs. (5), (8), and (9) as:

$$\begin{aligned} \Delta S^l_{\text{mot}}(k) &= \ln \frac{V}{2\pi} - \frac{1}{2} \ln |\Sigma^l_k| - \frac{d^2}{2}, \\ \Delta S^l_{\text{app}}(k) &= -\ln \left(1 + e^{-2F(X_{i_k})}\right) - \ln c_1. \end{aligned} \tag{12}$$

The score update continues as long as the track hypothesis is updated with detections. A track hypothesis which is assigned dummy observations for N_{miss} consecutive frames is deleted from the hypothesis space.

3.4. Global Hypothesis Formation

Given the set of trees that contains all trajectory hypotheses for all targets, we want to determine the most likely combination of object tracks at frame k. This can be formulated as the following k-dimensional assignment problem:

$$\begin{aligned} &\max_{\mathbf{z}} \sum_{i_1=0}^{M_1} \sum_{i_2=0}^{M_2} \cdots \sum_{i_k=0}^{M_k} s_{i_1 i_2 \ldots i_k} z_{i_1 i_2 \ldots i_k} \\ &\text{subject to} \sum_{i_1=0}^{M_1} \cdots \sum_{i_{u-1}=0}^{M_{u-1}} \sum_{i_{u+1}=0}^{M_{u+1}} \cdots \sum_{i_k=0}^{M_k} z_{i_1 i_2 \ldots i_u \ldots i_k} = 1 \\ &\text{for} \quad i_u = 1,2,...,M_u \quad \text{and} \quad u = 1,2,...,k \end{aligned} \tag{13}$$

where we have one constraint for each observation i_u, which ensures that it is assigned to a unique track. Each track is associated with its binary variable $z_{i_1 i_2 \ldots i_k}$ and track score $s_{i_1 i_2 \ldots i_k}$ which is calculated by Eq. (2). Thus, the objective function in Eq. (13) represents the total score of the tracks in the global hypothesis. This optimization problem is known to be NP-hard when k is greater than 2.

Following [34], the task of finding the most likely set of tracks can be formulated as a Maximum Weighted Independent Set (MWIS) problem. This problem was shown in [34] to be equivalent to the multidimensional assignment

problem (13) in the context of MHT. An undirected graph $G = (V, E)$ is constructed by assigning each track hypothesis T_l to a graph vertex $x_l \in V$ (see Fig. 2a). Note that the number of track hypotheses needs to be controlled by track pruning (Sec. 3.5) at every frame in order to avoid the exponential growth of the graph size. Each vertex has a weight w_l that corresponds to its track score $S^l(k)$. An edge $(l, j) \in E$ connects two vertices x_l and x_j if the two tracks cannot co-exist due to shared observations at any frame. An independent set is a set of vertices with no edges in common. Thus, finding the maximum weight independent set is equivalent to finding the set of compatible tracks that maximizes the total track score. This leads to the following discrete optimization problem:

$$\begin{aligned} &\max_{\mathbf{x}} \sum_{l} w_l x_l \\ &\text{s.t. } x_l + x_j \leq 1, \quad \forall (l, j) \in E, \quad x_l \in \{0, 1\}. \end{aligned} \tag{14}$$

We utilize either an exact algorithm [33] or an approximate algorithm [9] to solve the MWIS optimization problem, depending on its hardness (as determined by the number of nodes and the graph density).

3.5. Track Tree Pruning

Pruning is an essential step for MHT due to the exponential increase in the number of track hypotheses over time. We adopt the standard N-scan pruning approach. First, we identify the tree branches that contain the object tracks within the global hypothesis obtained from Eq. (14). Then for each of the selected branches, we trace back to the node at frame $k - N$ and prune the subtrees that diverge from the selected branch at that node (see Fig. 2b). In other words, we consolidate the data association decisions for old observations up to frame $k-(N-1)$. The underlying assumption is that the ambiguities in data association for frames 1 to $k - N$ can be resolved after looking ahead for a window of N frames [12]. A larger N implies a larger window hence the solution can be more accurate, but makes the running time longer. After pruning, track trees that do not contain any track in the global hypothesis will be deleted.

Besides N-scan pruning, we also prune track trees that have grown too large. If at any specific time the number of branches in a track tree is more than a threshold B_{th}, then we prune the track tree to retain only the top B_{th} branches based on its track score.

When we use MHT-DAM (see Table 1), the appearance model enables us to perform additional branch pruning. This enables us to explore a larger gating area without increasing the number of track hypotheses significantly. Specifically, we set $\Delta S_{\text{app}}(t) = -\infty$, preventing the tree from spawning a branch for observation i_t, when its appearance score $F(X_{i_t}) < c_2$. These are the only pruning mechanisms in our MHT implementation.

4. Online Appearance Modeling

Since the data association problem is ill-posed, different sets of kinematically plausible trajectories always exist. Thus, many methods make strong assumptions on the motion model, such as linear motion or constant velocity [37, 44, 10]. However, such motion constraints are frequently invalid and can lead to poor solutions. For example, the camera can move or the target of interest may also suddenly change its direction and velocity. Thus, motion-based constraints are not very robust.

When target appearances are distinctive, taking the appearance information into account is essential to improve the accuracy of the tracking algorithm. We adopt the multi-output regularized least squares framework [25] for learning appearance models of targets in the scene. As an online learning scheme, it is less susceptible to drifting than local appearance matching, because multiple appearances from many frames are taken into account.

We first review the Multi-output Regularized Least Squares (MORLS) framework and then explain how this framework fits into MHT.

4.1. Multi-output Regularized Least Squares

Multiple linear regressors are trained and updated simultaneously in multi-output regularized least squares. At frame k, the weight vectors for the linear regressors are represented by a $d \times n$ weight matrix $\mathbf{W}_k$ where d is the feature dimension and n is the number of regressors being trained. Let $\mathbf{X}_k = [X_{k,1}|X_{k,2}|...|X_{k,n_k}]^\top$ be a $n_k \times d$ input matrix where n_k is the number of feature vectors (i.e. detections), and $X_{k,i}$ represents the appearance features from the i-th training example at time k. Let $\mathbf{V}_k = [V_{k,1}|V_{k,2}|...|V_{k,n}]$ denote a $n_k \times n$ response matrix where $V_{k,i}$ is a $n_k \times 1$ response vector for the i^{th} regressor at time k. When a new input matrix $\mathbf{X}_{k+1}$ is received, the response matrix $\hat{\mathbf{V}}_{k+1}$ for the new input can be predicted by $\mathbf{X}_{k+1}\mathbf{W}_k$.

The weight matrix $\mathbf{W}_k$ is learned at time k. Given all the training examples $(\mathbf{X}_i, \mathbf{V}_i)$ for $1 \leq i \leq k$, the weight matrix can be obtained as:

$$\min_{\mathbf{W}_k} \sum_{t=1}^{k} \|\mathbf{X}_i\mathbf{W}_k - \mathbf{V}_i\|_F^2 + \lambda \|\mathbf{W}_k\|_F^2 \tag{15}$$

where $\|\cdot\|_F$ is the Frobenius norm. The optimal solution is given by the following system of linear equations:

$$(\mathbf{H}_k + \lambda \mathbf{I})\mathbf{W}_k = \mathbf{C}_k \tag{16}$$

where $\mathbf{H}_k = \sum_{t=1}^{k} \mathbf{X}_t^\top \mathbf{X}_t$ is the covariance matrix, and $\mathbf{C}_k = \sum_{t=1}^{k} \mathbf{X}_t^\top \mathbf{V}_t$ is the correlation matrix.

The model is online because at any given time only $\mathbf{H}_k$ and $\mathbf{C}_k$ need to be stored and updated. $\mathbf{H}_k$ and $\mathbf{C}_k$ can be updated recursively via:

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \mathbf{X}_{k+1}^\top \mathbf{X}_{k+1}, \tag{17}$$

$$\mathbf{C}_{k+1} = \mathbf{C}_k + \mathbf{X}_{k+1}^\top \mathbf{V}_{k+1} \tag{18}$$

which only requires the inputs and responses at time $k+1$.

4.2. Application of MORLS to MHT

We utilize each detected bounding box as a training example. Appearance features from all detection boxes at time k form the input matrix $\mathbf{X}_k$. Each tree branch (track hypothesis) is paired with a regressor which is trained with the detections from the time when the track tree was born to the current time k. Detections from the entire history of the track hypothesis serve as positive examples and all other detections serve as negative examples. The response for the positive example is 1, and the responses for the negative examples are set to -1. Note that a classification loss function (e.g. hinge loss) will be more suitable for this problem, but then the benefits of efficient updates and an analytic globally optimal solution would be lost.

The online nature of the least squares framework makes it efficient to update multiple regressors as the track tree is extended over time. Starting from one appearance model at the root node, different appearance models will be generated as the track tree spawns different branches. $\mathbf{H}$ and $\mathbf{C}$ in the current tree layer (corresponding to the current frame) are copied into the next tree layer (next frame), and then updates according to Eqs. (17) and (18) are performed for all of the tree branches in the next tree layer. Suppose we have $\mathbf{H}_{k-1}$ and $\mathbf{C}_{k-1}$ and are branching into n branches at time k. Note that the update of $\mathbf{H}_k$ only depends on $\mathbf{X}_k$ and is done once, no matter how many branches are spawned at time k. $\mathbf{C}_k$ depends on both $\mathbf{X}_k$ and $\mathbf{V}_k$. Hence, for each new tree branch i, one matrix-vector multiplication $\mathbf{X}_k^\top V_{k,i}$ needs to be performed. The total time complexity for computing $\mathbf{X}_k^\top \mathbf{V}_k = [\mathbf{X}_k^\top V_{k,1} | \mathbf{X}_k^\top V_{k,2} | ... | \mathbf{X}_k^\top V_{k,n}]$ is then $O(dnn_k)$ which is linear in both the number of tree branches n and the number of detections n_k.

The most time-consuming operation in training the model is updating and decomposing $\mathbf{H}$ in solving Eq. (16). This operation is shared among all the track trees that start at the same frame and is independent of the branches on the track trees. Thus, one can easily spawn many branches in each track tree with minimal additional computation required for appearance updating. This property is unique to tree-based MHT, where all the branches have the same ancestry. If one is training long-term appearance models using other global methods such as [31] and [32], then such computational benefits disappear, and the appearance model would need to be fully updated for each target separately, which would incur substantial computational cost.

As for the appearance features, we utilize the convolutional neural network features trained on the ImageNet+PASCAL VOC dataset in [16]. We follow the protocol in [16] to extract the 4096-dimensional feature for each detection box. For better time and space complexity, a principal component analysis (PCA) is then performed to reduce the dimensionality of the features. In the experiments we take the first 256 principal components.

5. Experiments

In this section we first present several experiments that show the benefits of online appearance modeling on MHT. We use 11 MOT Challenge [24] training sequences and 5 PETS 2009 [14] sequences for these experiments. These sequences cover different difficulty levels of the tracking problem. In addition to these experimental results, we also report the performance of our method on the MOT Challenge and PETS benchmarks for quantitative comparison with other tracking methods.

For performance evaluation, we follow the current evaluation protocols for visual multi-target tracking. The protocols include the multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) [5]. MOTA is a score which combines false positives, false negatives and identity switches (IDS) of the output trajectories. MOTP measures how well the trajectories are aligned with the ground truth trajectories in terms of the average distance between them. In addition to these metrics, the number of mostly tracked targets (MT), mostly lost targets (ML), track fragmentations (FM), and IDS are also reported. Detailed descriptions about these metrics can be found in [30].

Table 1 shows the default parameter setting for all of the experiments in this section. In the table, our baseline method that only uses motion information is denoted as MHT. This is a basic version of the MHT method described in Section 3 using only the motion score $S_{\text{mot}}(k)$. Our novel extension of MHT that incorporates online discriminative appearance modeling is denoted as MHT-DAM.

	N-scan	B_{th}	N_{miss}	P_D	d_{th}	$w_{\text{mot}}, w_{\text{app}}$	c_1, c_2
MHT-DAM	5	100	15	0.9	12	0.1, 0.9	0.3, −0.8
MHT	5	100	15	0.9	6	1.0, 0.0	

Table 1. Parameter Setting

5.1. Pruning Effectiveness

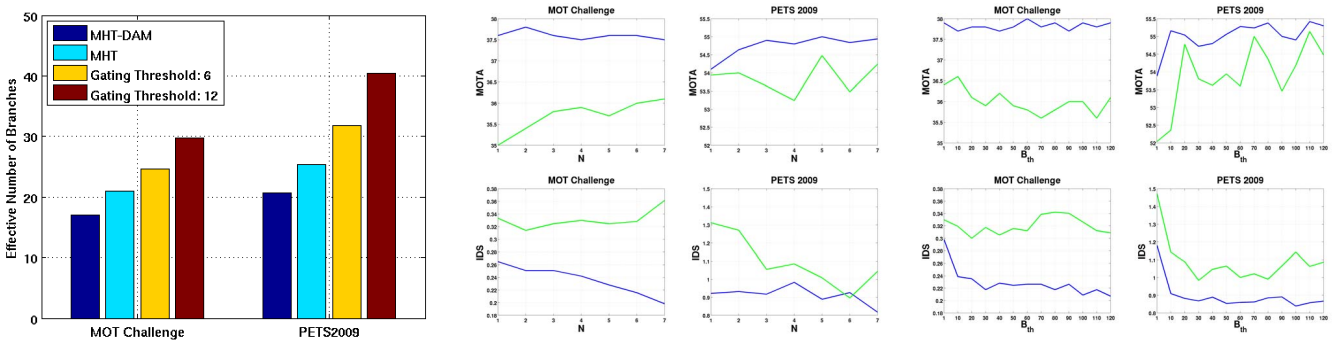
As we explained earlier, pruning is central to the success of MHT. It is preferable to have a discriminative score function so that more branches can be pruned early and reliably. A measure to quantify this notion is the entropy:

$$H(B_k) = -\sum_v p(B_k = v) \ln p(B_k = v) \tag{19}$$

where $p(B_k = v)$ is the probability of selecting v^{th} tree branch at time k for a given track tree and defined as:

$$p(B_k = v) = \frac{e^{\Delta S^v(k)}}{\sum_v e^{\Delta S^v(k)}}. \tag{20}$$

For the normalization, we take all the branches at time k from the same target tree.



(a) (b) (c)

Figure 3. (a) Average effective number of branches per track tree for different pruning mechanisms. MHT-DAM uses a gating threshold $d_{\text{th}} = 12$ and MHT uses a gating threshold $d_{\text{th}} = 6$. Even with a larger gating area, the appearance model for MHT-DAM is capable of significantly reducing the number of branches. (b) Sensitivity analysis for N-scan parameter N. (c) Sensitivity analysis for the maximum number of branches B_{th}. The **Blue** lines are the results from MHT-DAM and the **Green** lines are the results from MHT. The first row shows the MOTA score (higher is better) and the second row shows the number of ID switches (averaged per target, lower is better) over different pruning parameters.

Table 2. Results from $2D$ MOT 2015 Challenge (accessed on $9/25/2015$)

Method	MOTA	MOTP	FAF	MT	ML	FP	FN	IDS	FM	Hz
MHT-DAM	**32.4**	**71.8**	1.6	**16.0**%	**43.8**%	9,064	**32,060**	**435**	826	0.7
MHT	**29.2**	**71.7**	1.7	**12.1**%	53.3%	9,598	**33,467**	**476**	**781**	0.8
LP_SSVM [41]	25.2	**71.7**	**1.4**	5.8%	53.0%	8,369	36,932	646	849	**41.3**
ELP [27]	25.0	71.2	**1.3**	7.5%	**43.8**%	**7,345**	37,344	1,396	1,804	5.7
MotiCon [23]	23.1	70.9	1.8	4.7%	52.0%	10,404	35,844	1,018	1,061	1.4
SegTrack [28]	22.5	**71.7**	**1.4**	5.8%	63.9%	**7,890**	39,020	697	**737**	0.2
CEM [29]	19.3	70.7	2.5	8.5%	46.5%	14,180	34,591	813	1,023	1.1
RMOT [43]	18.6	69.6	2.2	5.3%	53.3%	12,473	36,835	684	1,282	7.9
SMOT [13]	18.2	71.2	1.5	2.8%	54.8%	8,780	40,310	1,148	2,132	2.7
TBD [15]	15.9	70.9	2.6	6.4%	47.9%	14,943	34,777	1,939	1,963	0.7
TC_ODAL [2]	15.1	70.5	2.2	3.2%	55.8%	12,970	38,538	637	1,716	1.7
DP_NMS [35]	14.5	70.8	2.3	6.0%	**40.8**%	13,171	34,814	4,537	3,090	**444.8**

With the entropy, we can define the effective number of the branches N_{eff} within each track tree as:

$$N_{\text{eff}} = e^{H(B_k)}. \tag{21}$$

When all the branches in the target tree have the same probability (i.e. when the features are not discriminative), N_{eff} is equal to the actual number of branches, which means one would need to explore all the possibilities. In the opposite case where a certain branch has the probability of 1, N_{eff} is 1 and it is only necessary to examine a single branch.

Fig. 3a shows the number of effective branches for different pruning mechanisms. For this experiment, we set the default gating threshold d_{th} to 12. The highest bar (dark red) in each PETS sequence in Fig. 3a shows the average number of tree branches generated per frame with the default gating parameter. A smaller gating area ($d_{\text{th}} = 6$) (yellow bar) only reduces the number of branches by a small amount but might prune out fast-moving hypotheses. Combined with the Kalman filter motion model, the reduction is more significant (cyan bar), but the algorithm still retains more than half of the effective branches compared to the full set with $d_{\text{th}} = 12$.

Incorporating the appearance likelihood significantly reduces the effective number of branches. In both the MOT Challenge and PETS sequences, the average effective number of branches in a tree becomes $\sim$50% of the total number of branches. And this is achieved without lowering the size of the gating area, thereby retaining fast-moving targets. This shows that long-term appearance modeling significantly reduces the ambiguities in data association, which makes MHT search more effective and efficient.

Analysis of Pruning Parameters. MHT was known to be sensitive to its parameter settings [32]. In this section, we perform a sensitivity analysis of MHT with respect to its pruning parameters and demonstrate that our appearance model helps to alleviate this parameter dependency.

In our MHT implementation, there are two MHT pruning parameters. One is the N-scan pruning parameter N, the other is the maximum number of tree branches B_{th}. We tested MHT using 7 different values for N and 13 different values for B_{th}. We assessed the number of errors in terms of the MOTA score and identity switches (IDS).

Fig. 3b shows the results from this analysis over different

N-scan parameters. We fix the maximum number of tree branches to 300, a large enough number so that very few branches are pruned when N is large. The results show that motion-based MHT is negatively affected when the N-scan parameter is small, while MHT-DAM is much less sensitive to the parameter change. This demonstrates that appearance features are more effective than motion features in reducing the number of look-ahead frames that are required to resolve data association ambiguities. This is intuitive, since many targets are capable of fast movement over a short time scale, while appearance typically changes more slowly.

Fig. 3c illustrates the change in the MOTA and IDS scores when the maximum number of branches varies from 1 to 120. We fix the N-scan pruning parameter to 5 which is the setting for all other experiments in the paper. Note that appearance modeling is particularly helpful in preventing identity switches.

5.2. Benchmark Comparison

We test our method on the MOT Challenge benchmark and the PETS 2009 sequences. The MOT benchmark contains 11 training and 11 testing sequences. Users tune their algorithms on the training sequences and then submit the results on the testing sequences to the evaluation server. This benchmark is of larger scale and includes more variations than the PETS benchmark. Table 2 shows our results on the benchmark where MHT-DAM outperforms the best previously published method by more than 7% on MOTA. In addition, 16.0% of the tracks are mostly tracked, as compared to the next competitor at 8.5%. We also achieved the lowest number of ID switches by a large margin. This shows the robustness of MHT-DAM over a large variety of videos under different conditions. Also note that because MOT is significantly more difficult than the PETS dataset, the appearance model becomes more important to the performance.

Table 3 demonstrates the performance of MHT and MHT-DAM on the PETS sequences compared to one of the state-of-the-art tracking algorithms [31]. For a fair comparison, the detection inputs, ground truth annotations, and evaluation script provided by [31] were used. Our basic MHT implementation already achieves a better or comparable result in comparison to [31] for most PETS sequences and metrics. Cox's method is also surprisingly close in performance to [31] with ∼6% lower MOTA on average with the exception of the S2L2 sequence where it is ∼20% lower. However, considering that Cox's MHT implementation was done almost 20 years ago, and that it can run in real time due to the efficient implementation (40 FPS on average for PETS), the results from Cox's method are impressive. After adding appearance modeling to MHT, our algorithm MHT-DAM makes fewer ID switches and has higher MOTA and MOTP scores in comparison to previous methods.

6. Conclusion

Multiple Hypothesis Tracking solves the multidimensional assignment problem through an efficient breadth-first search process centered around the construction and pruning of hypothesis trees. Although it has been a workhorse method for multi-target tracking in general, it has largely fallen out-of-favor for visual tracking. Recent advances in object detection have provided an opportunity to rehabilitate the MHT method. Our results demonstrate that a modern formulation of a standard MHT approach can achieve comparable performance to several state-of-the-art methods on reference datasets. Moreover, an implementation of MHT by Cox [12] from the 1990s comes surprisingly close to state-of-the-art performance on 4 out of 5 PETS sequences. We have further demonstrated that the MHT framework can be extended to include on-line learned appearance models, resulting in substantial performance gains. The software and evaluation results are available from our project website.[2]

Acknowledgments: This work was supported in part by the Simons Foundation award 288028, NSF Expedition award 1029679 and NSF IIS award 1320348.

Table 3. Tracking Results on the PETS benchmark

Sequence	Method	MOTA	MOTP	MT	ML	FM	IDS
S2L1	MHT-DAM	**92.6%**	**79.1%**	**18**	0	**12**	**13**
	MHT	**92.3%**	**78.8%**	**18**	0	**15**	**17**
	Cox's MHT [12]	84.1%	77.5%	17	0	65	45
	Milan [31]	90.3%	74.3%	**18**	0	**15**	22
S2L2	MHT-DAM	**59.2%**	**61.4%**	**10**	**2**	162	**120**
	MHT	57.2%	58.7%	7	**1**	**150**	**134**
	Cox's MHT [12]	38.0%	58.8%	3	8	273	154
	Milan [31]	**58.1%**	**59.8%**	**11**	**1**	**153**	167
S2L3	MHT-DAM	38.5%	**70.8%**	**9**	22	**9**	**8**
	MHT	**40.8%**	**67.3%**	**10**	**21**	**19**	**18**
	Cox's MHT [12]	34.8%	66.1%	6	22	65	35
	Milan [31]	**39.8%**	65.0%	8	**19**	22	27
S1L1-2	MHT-A+M	**62.1%**	**70.3%**	**21**	**9**	**14**	**11**
	MHT-M	**61.6%**	**68.0%**	**22**	12	23	31
	Cox's MHT [12]	52.0%	66.5%	17	14	52	41
	Milan [31]	60.0%	61.9%	**21**	**11**	**19**	**22**
S1L2-1	MHT-DAM	**25.4%**	**62.2%**	**3**	24	**30**	**25**
	MHT	24.0%	58.4%	**5**	**23**	**29**	**33**
	Cox's MHT [12]	22.6%	57.4%	2	**23**	57	34
	Milan [31]	**29.6%**	**58.8%**	2	**21**	34	42

References

[1] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012. 2

[2] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR*, 2014. 2, 7

[2] http://cpl.cc.gatech.edu/projects/MHT/

[3] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Multi-commodity network flow for tracking multiple people. *PAMI*, 2014. 2

[4] J. Berclaz, E. Turetken, F. Fleuret, and P. Fua. Multiple object tracking using K-shortest paths optimization. *PAMI*, 2011. 2

[5] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *Image and Video Processing*, 2008. 6

[6] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999. 1, 3, 4

[7] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *PAMI*, 2011. 2

[8] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, 2011. 2

[9] S. Busygin. A new trust region technique for the maximum weight clique problem. *Discrete Appl. Math.*, 2006. 5

[10] A. Butt and R. Collins. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *CVPR*, 2013. 2, 5

[11] R. T. Collins. Multitarget data association with higher-order motion models. In *CVPR*, 2012. 2

[12] I. J. Cox and S. L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *PAMI*, 1996. 1, 2, 4, 5, 8

[13] C. Dicle, O. Camps, and M. Sznaier. The way they move: Tracking targets with similar appearance. In *ICCV*, 2013. 7

[14] J. Ferryman and A. Ellis. PETS2010: Dataset and challenge. In *AVSS*, 2010. 6

[15] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3D traffic scene understanding from movable platforms. *PAMI*, 2014. 7

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 6

[17] M. Han, W. Xu, H. Tao, and Y. Gong. An algorithm for multiple object trajectory tracking. In *CVPR*, 2004. 2

[18] C. Huang, Y. Li, and R. Nevatia. Multiple target tracking by learning-based hierarchical association of detection responses. *PAMI*, 2013. 2

[19] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *PAMI*, 2005. 2

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[21] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, 2010. 2

[22] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, 2011. 2

[23] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *CVPR*, 2014. 7

[24] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, 2015. 6

[25] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 1, 5

[26] J. Liu, P. Carr, R. T. Collins, and Y. Liu. Tracking sports players with context-conditioned motion models. In *CVPR*, 2013. 2

[27] N. McLaughlin, J. Martinez Del Rincon, and P. Miller. Enhancing linear programming with motion modeling for multi-target tracking. In *WACV*, 2015. 7

[28] A. Milan, L. Leal-Taixé, I. Reid, and K. Schindler. Joint tracking and segmentation of multiple targets. In *CVPR*, 2015. 7

[29] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *PAMI*, 2014. 7

[30] A. Milan, K. Schindler, and S. Roth. Challenges of ground truth evaluation of multi-target tracking. In *CVPR Workshop*, 2013. 6

[31] A. Milan, K. Schindler, and S. Roth. Detection-and trajectory-level exclusion in multiple object tracking. In *CVPR*, 2013. 6, 8

[32] S. Oh, S. Russell, and S. Sastry. Markov Chain Monte Carlo data association for multi-target tracking. *IEEE Transactions on Automatic Control*, 2009. 2, 6, 7

[33] P. R. Ostergard. A new algorithm for the maximum-weight clique problem. *Nordic Journal of Computing*, 2001. 5

[34] D. J. Papageorgiou and M. R. Salpukas. The maximum weight independent set problem for data association in multiple hypothesis tracking. *Optimization and Cooperative Control Strategies*, 2009. 2, 4

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

[36] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 1979. 1, 2

[37] A. Segal and I. Reid. Latent data association: Bayesian model selection for multi-target tracking. In *ICCV*, 2013. 2, 5

[38] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV*, 2011. 2

[39] A. Smeulder, D. Chu, R. Cucchiara, S. Calderara, A. Deghan, and M. Shah. Visual tracking: An experimental survey. *PAMI*, 2014. 2

[40] X. Song, J. Cui, H. Zha, and H. Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *ECCV*, 2008. 2

[41] S. Wang and F. C. Learning optimal parameters for multi-target tracking. In *BMVC*, 2015. 7

[42] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *CVPR*, 2012. 2

[43] J. Yoon, H. Yang, J. Lim, and K. Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *WACV*, 2015. 7

[44] A. R. Zamir, A. Dehghan, and M. Shah. GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*, 2012. 5

[45] L. Zhang and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 2