Online Empirical Evaluation of Tracking Algorithms

Wu Hao, *Student Member, IEEE,*
Aswin C. Sankaranarayanan, *Student Member, IEEE,*
and Rama Chellappa, *Fellow, IEEE*

All authors are with the Center for Automation Research and the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742. This work is partially funded by a contract from the JHU Applied Physics Laboratory.

Abstract

Evaluation of tracking algorithms in the absence of ground truth is a challenging problem. There exist a variety of approaches for this problem, ranging from formal model validation techniques to heuristics that look for mismatches between track properties and the observed data. However, few of these methods scale up to the task of visual tracking where the models are usually non-linear and complex, and typically lie in a high dimensional space. Further, scenarios that cause track failures and/or poor tracking performance are also quite diverse for the visual tracking problem. In this paper, we propose an online performance evaluation strategy for tracking systems based on particle filters using a time-reversed Markov chain. The keu intuition of our proposed methodology relies on the time-reversible nature of physical motion exhibited by most objects, which in turn should be possessed by a good tracker. In the presence of tracking failures due to occlusion, low SNR or modeling errors, this reversible nature of the tracker is violated. We use this property for detection of track failures. To evaluate the performance of the tracker at time instant t, we use the posterior of the tracking algorithm to initialize a time-reversed Markov chain. We compute the posterior density of track parameters at the starting time $t = 0$ by filtering back in time to the initial time instant. The distance between the posterior density of the time-reversed chain (at $t = 0$) and the prior density used to initialize the tracking algorithm forms the decision statistic for evaluation. It is observed that when the data is generated by the underlying models, the decision statistic takes a low value. We provide a thorough experimental analysis of the evaluation methodology. Specifically, we demonstrate the effectiveness of our approach for tackling common challenges such as occlusion, pose and illumination changes and provide the Receiver Operating Characteristic (ROC) curves. Finally, we also show the applicability of the core ideas of the paper to other tracking algorithms such as the Kanade-Lucas-Tomasi (KLT) feature tracker and the mean-shift tracker.

Index Terms

Performance Evaluation, Tracking, Particle Filters, Model Validation

I. INTRODUCTION

Visual tracking forms one of the most important components in a wide range of application domains. Robust tracking of features form the primary input to classical vision problems such as structure from motion and registration. In addition, tracking finds use in diverse application areas such as surveillance, markerless motion capture and medical imaging. The need for robust tracking algorithms that work over a broad spectrum of application domains cannot be under-

stated. However, practical realities and the diverse nature of data dictates that even the most sophisticated algorithm will have failure modes where the tracking performance is poor and the algorithm loses track. *In this paper, we address the problem of automatic evaluation of tracking algorithms with the goal of detecting track failures and evaluation of tracking performance without the need for ground truth.*

There are multiple reasons why a self-evaluation framework is needed. Its most straightforward use is in online characterization of tracking performance to enable a system to sanitize the tracker output in the event of failure. Further, in the context of distributed sensor network, evaluation of the performance of the tracking algorithm (associated with each modality) can be used to characterize its reliability for the tasks of multi-modal fusion. Self-evaluation can also be used to rank different tracking algorithms based on their performance. In this sense, self-evaluation can be used to choose a tracking algorithm with better performance at run time. It also potentially allows for tracking algorithms to tune their parameters to the specifics of an individual video (as opposed to a training set, which may or may not capture the nuances of a single instance). While ground truth allows the same, it is not self-contained to the tracking algorithm and is not extensible easily.

There exist many evaluation schemes [1] [2] [3] that use ground-truth information to evaluate tracking algorithms, and more importantly rank-order them in terms of performance. The PETS[1] and CLEAR[2] workshops, along with the ETISEO [4] effort focused mainly towards characterizing algorithms in terms of performance *in the presence of ground truth*. The CAVAIR[3] and the VACE[4] efforts were geared towards evaluation of object detection and tracking [5], [6]. In addition to this, there has been some research on distance metrics in matching the ground truth information to the tracker outputs, and in tuning the parameters of the tracking algorithm [7]. However, collection of ground truth is time consuming, and has its own variabilities [8]. Further, performance evaluation using ground truth is not possible for real time field testing or on sequences which are unlabeled. This motivates the need for online performance characterization in the absence of ground truth.

Evaluation of tracking performance and detection of track failure is similar to the problem of

[1]http://petsmetrics.net

[2]http://isl.ira.uka.de/clear07

[3]http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[4]http://www.ic-arda.org/InfoExploit/vace/

model validation, especially when the underlying formulation is in terms of dynamical systems. Tracking performance is bound to deteriorate when the data violates the modeling assumptions signifi cantly. There exist many ways to detect the incompatibility between the models and the observed data. For stochastic nonlinear systems, measurements based on the innovation error forms a common choice as an evaluation metric. The statistics of the innovation error can be cross-checked with those of the model (such as white Gaussian noise), and a hypothesis test can be performed to determine model validity. Similar metrics such as the tracking error (TE) and observation likelihood (OL), and their corresponding cumulative summations in time (CUSUM) have been used for change detection and model validation [9]. TE and OL detect only sharp changes which results in loss of track, and do not register slow changes. A statistic for detection of slow changes called the negative expected log likelihood of state (ELL) and its generalization, gELL are proposed in [9]. ELL is defi ned as a measure of inaccuracy between the posterior at time t and the t-step ahead prediction of the prior state distribution. Interestingly, as we point out later, the evaluation methodology proposed in this paper mirrors the ELL method in spirit.

In [10] [11] [12], under the hypothesis that the model is correct, a random process in the scalar observation space is shown to be a realization of independent identically distributed variables uniformly distributed on interval $[0, 1]$. This result holds for any time series and may be used in statistical tests to determine the adequacy of the model. An extension to vector-valued measurements is presented in [13], where a χ^2-test for multi-dimensional uniform distribution is used to determine if the system behaves consistently. However, when it comes to visual tracking, as the observation could be in a very high-dimensional image space, the computation of the test statistics is infeasible. In [14], an entropy based criterion is used to evaluate the statistical characteristics of the tracked density function. The defi nition of good performance for tracking a single object is that the posterior distribution is unimodal and of low variance. In contrast, a multi-modal and a high variance distribution implies poor or lost tracking. In practice, tracking in the presence of multiple targets and clutter does lead to the presence of multi-modality in the target's posterior density. This, however, does not necessarily imply poor tracking.

While model validation and change detection literature offers formal and rigorous approaches to formulate the problem, in many cases, the underlying models for tracking are unable to handle wide variations that occur in visual tracking. Further, given the complexity of the visual information, it is virtually impossible to accurately model all the information in all its variabilities.

Towards this end, there has been a body of research that exploits the inherent characteristics of tracking output to automatically characterize the performance. In [15], Erdem et al. address an on-line performance evaluation method for contour tracking. Metrics based on color and motion differences along the boundary of the estimated object are used to localize regions where segmentation results are of similar quality, and combined to provide a quantitative evaluation of boundary segmentation and tracking. As an extension, [16] uses a feedback loop to adjust the weights assigned to the features used for tracking and segmentation. This method of evaluation is specific to contour-based tracking systems. Wu and Zheng present a method for self-evaluation in [17]. This empirical method evaluates the trajectory complexity, motion smoothness, scale constancy, shape and appearance similarity, combining each evaluation result to form a total score of the tracking quality. However, this heuristic method can only be applied to a static camera system.

In this paper, we propose an online evaluation methodology that can be applied to many tracking algorithms to detect tracking failures and to evaluate tracking performance. The intuition behind our algorithm lies in the reversibility of the physical motion exhibited by an object. In many cases, this directly corresponds to time-reversibility of the models used in the formulation of the tracking problem. When this tracking problem is defined in terms of dynamical systems exhibiting Markovian properties, we construct a time-reversed Markov chain for the sole purpose of evaluation. The posterior probability density of the time-reversed chain is propagated all the way back to the initial time instant when the tracking algorithm is initialized. The prior used to initialize the tracker is now compared to the posterior of the time-reversed chain to form the evaluation statistic. For a well behaved system, the two probability distributions are expected to show proximity in some statistical sense, with significant discrepancies between them in the presence of tracking error. The proposed approach finds applicability in a host of tracking algorithms that use a dynamical system formulation. In this regard, the use of particle filtering for estimating inferences is very common given the non-linearity of most models and the non-Gaussian noise distribution. The proposed evaluation method involves filtering back to the initial time instant, and gets slower with increasing time. Hence, we also propose an approximation by tracking back and comparing the performance against a point in time where by prior verification we are confident that the performance is good. We analyze the performance of the evaluation methodology by extensive experimentation over a wide variety of videos. It is shown that when

ground truth is available, the track failures detected by our approach correlate significantly with those validated by the ground truth. We also show the applicability of the core ideas for tracking algorithms which are not modeled as dynamical systems. Examples of such algorithms include the KLT feature tracker [18] [19] [20] and the mean-shift tracking algorithm [21] [22]. Finally, we show that the proposed evaluation algorithm can be used for ranking different tracking algorithms based on their performance. This paper is an expanded version of [23].

The paper is organized as follows. We give a brief overview of particle filtering and introduce the terminology used in the paper in Section II. The proposed evaluation methodology and its properties are discussed in Section III. In Section IV, we place the proposed algorithm in the context of previous work on model validation and illustrate connections to other related topics. Finally, we discuss the experimental results that validate the performance of the proposed algorithm in section V.

II. Visual Tracking Using Particle Filters

In this section, we summarize the necessary background of Bayesian filtering methods used in dynamical systems, in particular, the particle filtering method which is used widely in visual tracking systems [24].

In particle filtering [25], we address the problem of Bayesian inference for dynamical systems. Let $x_t \in \mathbb{R}^d$ denote the state at time t, and $y_t \in \mathbb{R}^p$, the noisy observation at time t. We model the state sequence $\{x_t\}$ as a Markovian random process. Further we assume that the observations $\{y_t\}$ to be conditionally independent given the state sequence. Under these assumptions, the models defining the system are given as follows: 1) $p(x_t|x_{t-1})$: The state transition probability density function, describing the evolution of the system from time $t-1$ to t; 2) $p(y_t|x_t)$: the observation likelihood density, describing the conditional likelihood of observation given state; and 3) $p(x_0)$: the prior state probability at $t = 0$.

Given statistical descriptions of the models and noisy observations till time $t, \mathcal{Y}_t = \{y_1, \ldots, y_t\}$, we would like to estimate the posterior density function $\pi_t = p(x_t|\mathcal{Y}_t)$. Under Markovian assumption on the state space dynamics and conditional independence assumption on the observation model, the posterior probability is estimated recursively using the *Bayes Theorem*

$$\pi_t = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_{t-1}} \tag{1}$$

Computation of $p(x_t|y_{1:t-1})$ is called the *prediction* step,

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{2}$$

Equation 2 sets up the recursive step for estimation of the posterior at time t, π_t from that at time $t-1$, π_{t-1}.

$$\pi_t = \frac{p(y_t|x_t)\int p(x_t|x_{t-1})\pi_{t-1}dx_{t-1}}{p(y_t|y_{1:t-1})} \tag{3}$$

Note that, there are no unknowns in (3) since all terms are either specified or computable from the posterior at the previous time step. The problem is that this computation need not have an analytical representation. The particle filter approximates the posterior π_t with a discrete set of particles or samples $\{x_t^{(i)}\}_{i=1}^N$ with associated weights $\{w_t^{(i)}\}_{i=1}^N$ that are suitably normalized. The set $S_t = \{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ is the weighted particle set that represents the posterior density at time t, and is estimated recursively from S_{t-1}. The initial particle set S_0 is obtained from sampling the prior density $\pi_0 = p(x_0)$.

A. Visual Tracking

In this subsection, we briefly discuss some of the common state space approaches to visual tracking focusing in particular on the kind of motion and appearance models commonly used. For rigid objects, most tracking algorithms formulate tracking over a state space that typically comprises of locations on the image plane, the scale and orientation of the target all of which can be re-parametrized as affine deformations of some basic shape. For non-rigid objects, the affine deformation state could be extended to include contour deformation parameters (usually encoded with splines or level sets). Finally, the state space may include components that relate to the appearance of the target, so as to characterize and track the changes in target's appearance with changing pose and illumination.

The state transition model for the dynamical system is usually the motion model describing the kinematics of the target. Depending on the requirements of the application, these could vary from a simple Brownian motion model or a constant velocity model, to activity specific motion models [26] when tracking complicated behaviors that have been learned a priori.

Finally, probably the most important component is the observation model, typically a characterization of the target's appearance encoded either as a gray-scale or color template, or a histogram

of such features. The key property of the observation model is that it provides discriminability of target-specific features over background and other scene constructs. Further, the models are expected to be fairly robust to outliers. Finally, there is the need for robustness to changes in target pose and scene illumination. This can be achieved by explicitly modeling such pose and illumination parameters in the state space of the system, or by having observation models that are invariant (partially or otherwise) to such changes.

In this context, it is important to discuss the role of online appearance models for visual tracking. In many practical systems (especially in surveillance), most targets are opportunistic, with the tracking algorithm having no significant prior characterization of their appearance. In such a scenario, the only identification of target appearance is in the initial frame provided to the target, typically in the form of the prior density of the target. As the target moves in the scene, online appearance models (OAMs) try to adapt to the changing appearance of the target. However, the OAM needs to be updated in order to incorporate new features exhibited by the target without introducing undesirable background artifacts that could potentially cause the tracking algorithm to diverge. This results in two contradicting requirements for the adaptation rules used to update the OAM. The need for updating the appearance models to account for the changing appearance of the target is balanced by the possibility that undesirable background artifacts might be introduced. Invariably, a strategy is chosen that balances these two effects. This leads to scenarios in visual tracking, where the appearance models may no longer represent the same object that was used in initialization. Hence, *this leads to a case where the tracking performance is poor not because of incompatibility between the models and the data (the premise of model validation) or because of lack of smoothness and continuity of tracks (the premise of heuristic works), but because the model characterizing the dynamical system are fundamentally flawed due to undesirable updates*.

In the next section, we outline our approach for performance evaluation, including detection of error such as the one described above. The key point that we like to retain from the discussions given above is the overwhelming role of the prior density in defining the target identity.

III. Time Reversibility for Evaluation

It is insightful to understand the challenges in visual tracking, and where some of the existing tracking algorithms and evaluation schemes fail. We begin with a discussion of failure modalities

of tracking algorithms and the challenges for a self-evaluation scheme.

A. Failure modes of Tracking Algorithms

Visual tracking needs to be robust against a wide variety of operating conditions, dealing with poor video resolution, occlusion, changes in pose and illumination, camera motion and clutter. Under such diverse operating conditions, descriptions of objects, such as appearance, color, shape and texture almost always change unpredictably. At the same time, motion consistency is a feature that most algorithms use to reduce the search space, and it is one feature that is frequently violated when the camera itself is moving.

The range of failures is even more enhanced when the tracking algorithm uses an adaptive and online observation (appearance, shape) model. Adaptive appearance models are crucial for achieving robustness to changing pose and illumination. However, there is almost always the problem of incorporating undesirable features into the model, examples of which could be features that correspond to the background. However, inspite of the large variations in operating conditions, the identity encoded by the appearance and shape information at the initializing frame provides a reference for validation. This forms the basis for the intuition behind the algorithm proposed in this paper.

B. Intuition

Our goal is to provide a general, online evaluation method for visual tracking systems based on dynamical systems. We will refer the Markov chain associated with the tracker algorithm as the "forward" chain. The prior used to initialize the forward chain is the reference distribution which we use to evaluate the performance of the tracking algorithm. In order to evaluate the tracking performance at a time instant (say $t = t_0$) we first need to account for the difference in time instant between the prior ($t = 0$) and the output of the tracker. To achieve this, we construct a time-reversed Markov chain with models that are similar to the forward chain. The key idea is to compute the posterior distribution of this time reversed Markov chain at the initialization time ($t = 0$) and compare it to the prior of the forward chain. For algorithms employing OAMs, the *identity* of the target is defined in the initializing frame and the prior used to initialize the system. This prior information encodes all the knowledge given to the tracking algorithm, and arguably is most critical in determining the performance of the algorithm. In this sense, the

tracking performance can be determined by verifying the output of the tracker at any particular time instant (say $t = t_0$) against the prior with suitable time normalization.

From the point of view of information captured in the tracking algorithms, the underlying intuition is that if, at time t, the tracker contains enough information about the target, then the ability to track well until time t along the forward Markov chain implies that it is very likely to be able to track back to the end along a time-reversed Markov chain equally well.

To get an intuitive understanding of the proposed algorithm, consider a video sequence in which the first frame and the last frame are identical (in camera placement as well as the location of every scene and object point). Good tracking performance would require a tracking algorithm to localize the target in the last frame at the same location as the prior given in the first frame. Such an idea is exploited for detecting drift in feature point tracking in [27]. Our algorithm can be viewed as an extension of that idea for performance characterization.

C. Formalizing the Concept

The forward Markov chain describing the tracking algorithm is defined using the prior density $p(x_0)$, the state model $p(x_t|x_{t-1})$ and the observation models $p(y_t|x_t)$. At time T, given an observation sequence $\mathcal{Y}_T = \{y_1, \ldots, y_T\}$, the posterior is $\pi_T = p(x_T|\mathcal{Y}_T)$. To evaluate the performance of the system, we propose a backward time tracker that uses π_T as its prior and the observation sequence $\mathcal{Y}_T$ in the time reversed order. Using the notation $q(\cdot)$ for probability density functions associated with the time-reversed system, the reverse tracker is formulated as follows. For evaluation at time T, the system is initialized at time $T+1$ and filtered through the observations $\mathcal{Y}_T$.

- **Prior at time $T+1$:**

$$\begin{aligned} q(x_{T+1}) &= p(x_{T+1}|\mathcal{Y}_T) \\ &= \int p(x_{T+1}|x_T)p(x_T|\mathcal{Y}_T)dx_T \end{aligned} \tag{4}$$

- **State Transition Model:** For $t \in (0, T)$,

$$q(x_t|x_{t+1}) = \frac{p(x_{t+1}|x_t)p(x_t)}{p(x_{t+1})} \tag{5}$$

  This can be directly computed from the models for most systems used to define the tracking problem.

- **Observation Model:** We retain the same observation model used in the forward model.

$$\forall t, q(y_t|x_t) = p(y_t|x_t) \tag{6}$$

With this characterization of the system, we can now filter the observation sequence $\mathcal{Y}_T^b = \{y_T, \ldots, y_1\}$ in reverse time. The posterior density function of this filter is of great interest to us. At time t, the posterior density $\pi_t^b = q(x_t|\mathcal{Y}_t^b) = q(x_t|y_T, y_{T-1}, \ldots, y_t)$.

We can now estimate the posterior density at time $t = 0$, π_0^b by recursion. From intuition, we expect this density to be close in some statistical sense to the prior density $p(x_0)$. To this extent, we postulate the following property.

Proposition: Suppose the reverse tracker is initialized with the prior $q(x_{T+1}) = p(x_{T+1})$, then the posterior density of the time-reversed system at time $t = 0$ and the prior density $p(x_0)$ are close to each other on distance metrics comparing the means of the corresponding random variables, provided the underlying model completely fits the data.

Suppose we initialize the reversed time Markov chain using the density $p(x_{T+1})$ as opposed to $p(x_{T+1}|\mathcal{Y}_T)$. It is easy to verify that the final posterior distribution in the time-reversed process is equal to the smoothing result [28] at the beginning of the forward process using all the observations till time T, i.e, $\pi_0^b = p(x_0|y_{1,\ldots,T})$.

Now, π_0^b and the $p(x_0)$ are close in the sense that

$$\int x_0 p(x_0) dx_0 = \int_{\mathcal{Y}_t} \int_{x_0} x_0 \pi_0^b d\mathcal{Y}_t dx_0 \tag{7}$$

Suppose we compare $E(x_0)$ and $E_{\mathcal{Y}_t}(x_0)$, then on an average (over the ensemble set of possible observations) the two means will be the same.

It should be noted that the above result is true only when the reversed time system is initialized with the prior $p(x_{T+1})$. However, for most tracking models, it is the observation model with its characterization of object appearance and/or shape that allows for discrimination of the object from the background. In this sense, the observation model allows for accurate localization (or equivalently, low variance estimation) of the target with the state model used mainly to regularize and smoothen the result. Further, under the assumption that the data $\mathcal{Y}_T$ fits the underlying models, the density $p(x_{T+1}|\mathcal{Y}_T)$ is expected to localize the target better, in the sense of the sharpness of the density around its expected value. Hence, the system defined with prior $p(x_{T+1}|\mathcal{Y}_T)$ is *over-trained* and provides a model that fits the data better.

D. Evaluation Statistic

There exist distance metrics and measures for comparing density functions such as the Kullback-Leibler (KL) divergence and the Bhattacharya distance [29]. However, in our case, the distributions are represented by particles or samples from the density function. In general, given the differences in the individual proposal densities and random number generators, the exact locations at which the densities are sampled will be different. Computing the KL divergence or the Bhattacharya distance for such non-overlapping sample sets would require interpolation (using Parzen windows [29]) or the use of approximations such as the Unscented Transformation [30]. We circumvent this problem with the use of the Mahalanobis distance that depends only on the moments of the distributions.

Denoting p as the prior distribution $p(x_0)$ and π as the posterior of the time reversed chain $q(x_0|\mathcal{Y}_T)$, the distance $d(p,\pi)$ between the two distributions can be computed as:

$$\begin{aligned} d(p,\pi) &= (\mu_p - \mu_\pi)^T \Sigma_p^{-1} (\mu_p - \mu_\pi) + \\ & \quad (\mu_p - \mu_\pi)^T \Sigma_\pi^{-1} (\mu_p - \mu_\pi) \end{aligned} \tag{8}$$

where μ_p and Σ_p are the mean and the covariance matrix of the distribution p and μ_π and Σ_π are those of the distribution π, all of which can be easily computed or estimated from the particles or in some cases, analytically.

An outline of the proposed evaluation framework is in Table I.

The proposed framework extends gracefully even to other dynamic systems where the inference is not driven by particle filters. For example, if the system is linear Gaussian, then the posterior can be computed using a Kalman filter. The time-reversed system is also linear Gaussian, and its posterior can also be computed using a Kalman filter. In this case, the time-reversed posterior and the prior can be compared using (8). Given the Gaussian nature of both distributions, as an alternative similarity score, one could analytically compute their KL divergences too. It might be possible to provide theoretical guarantees for the algorithm in this simple case. Finally, in Section III-F, we show the applicability of the core idea for other tracking frameworks such as the KLT and the Mean-Shift algorithms.

TABLE I

OUTLINE OF THE PROPOSED EVALUATION ALGORITHM.

To evaluate the performance of the tracker at time T, the density π_T is represented by the samples $\{x_t^{(i)}\}_{i=1}^N$,

1) Propagate the particles using $p(x_{T+1}|x_T)$ to get samples from $p(x_{T+1}|\mathcal{Y}_T)$,

$$\tilde{x}_{T+1}^{(i)} \sim p(x_{T+1}|x_T^{(i)}), i = 1, \ldots, N \tag{9}$$

2) Using the prior represented by the particle set $\{\tilde{x}_{T+1}^{(i)}\}_{i=1}^N$, iterate the steps 3, 4 and 5 for $t \in \{T, T-1, \ldots, 1\}$,
3) Proposition: At time t, propose a new particle set $\{\tilde{x}_t^{(i)}\}_{i=1}^N$ using the state transition model,

$$\tilde{x}_t^{(i)} \sim q(x_t|\tilde{x}_{t+1}^{(i)}), i = 1, \ldots, N \tag{10}$$

4) Weight Computation: Compute the weight $w_t^{(i)}$ associated with the particle $\tilde{x}_t^{(i)}$,

$$w_t^{(i)} = q(y_t|x_t^{(i)}) \tag{11}$$

5) Normalize the weights and resample them to obtain an unweighted particle set.
6) Using the particle set $\tilde{x}_0^{(i)} \sim q(\tilde{x}_0|\mathcal{Y}_T)$, compute mean $\hat{\mu}_\pi$ and covariance matrix $\hat{\Sigma}_\pi$ using sample statistics.
7) The evaluation statistic is computed using (8).

E. Fast Approximation

The proposed evaluation framework poses a requirement to process (or track) across all the frames seen by the tracking algorithm. For such an algorithm, the computational requirements increase linearly with the number of frames (see Figure 1). This makes it increasingly harder for the evaluation algorithm to satisfy real time constraints.

However, a set of sufficient (though not necessary) conditions can be designed to alleviate this problem. We argue that if the performance at time T is good, then not only does the final posterior match well with the prior density, but that the posterior densities of the forward and reverse tracker should match at all intermediate time instants. A fast approximation is now proposed using this observation. Suppose at time t_0, the performance of the system is evaluated to be good, then for an evaluation at a future time instant $t' > t_0$, the time t_0 can be used as a reference point in the place of the $t = 0$ (see Figure 2). Extending this concept, we can recursively shift the reference point to keep a constant upper bound on the computational time

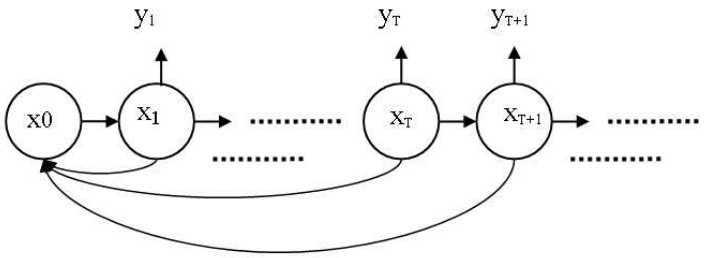


Fig. 1. Schematic of the reference point used in the proposed algorithm. Evaluation of the performance of the tracker requires validation with the prior density using a time reversed chain for suitable time normalization.

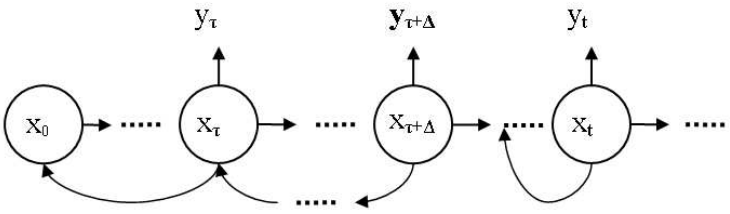


Fig. 2. Schematic of the reference point used in the faster approximation to the proposed algorithm. As opposed to the implementation described in Figure 1, the approximation shifts the reference point from $t = 0$ to create multiple reference points separated by time interval of $\Delta t = \Delta$. This keeps the overall computational requirements for the evaluation scheme bounded.

for the evaluation. Let Δt be the time interval between successive reference points, i.e, the time instants $t_0 = 0, \Delta t, 2\Delta t, 3\Delta, \ldots,$ are used as the reference points. For a time instant t', the reference point chosen is $\Delta t \lfloor t'/\Delta t \rfloor$. However, the suitability of the approximation depends on the length Δt. The trade-off here is between the computation time, which is proportional to Δt and the ability to detect slow changes that are of the order Δt. A clever choice of Δt can go a long way in reducing the computational requirements of the proposed algorithm.

Finally, even with the approximation scheme described above, it might be difficult to achieve real-time processing for the evaluation at every time instant. However, online evaluation in real time is possible if we do not perform evaluation at every frame. For most practical systems, evaluation needs to be performed at regular time intervals. Choosing a fast approximation scheme with Δt as the time difference between reference points as well as the time instants when

evaluation is performed can go a long way in reducing the computing requirements for evaluation.

F. Extensions beyond particle filtering

1) Evaluations of the Kanade-Lucas-Tomasi tracker: The basic idea of the proposed evaluation methodology can be used for tracking algorithms that do not use particle filtering. Here, we show how to apply the evaluation method for feature point tracking using the algorithm [18] [19] [20]. KLT is among the most widely used feature point trackers for many systems and applications and we use it for showcasing our evaluation algorithm.

The original KLT algorithm works under the assumption of brightness constancy and small motion (typically, translation), that is, $I(t, \mathbf{p}) = I(t+1, \mathbf{p} + \mathbf{\Delta p})$ where $I(t, \mathbf{p})$ is the intensity at pixel coordinate $\mathbf{p} = (\mathbf{x}, \mathbf{y})$ in the frame at time t. Under this assumption a linear system in $\mathbf{\Delta p}$ is solved to obtain the translation. In practice, the assumptions of brightness constancy and small motion used in the derivation of the solution are almost always violated eventually, leading to drift in the tracking of the feature point. In vision problems, especially those pertaining to geometry (such as structure from motion and estimation of epipolar geometry, homography), the presence of drift contributes to measurement errors which could subsequently be exaggerated by the following estimation algorithms.

The proposed evaluation methodology provides an elegant way to evaluate the tracking performance. As in the case of the particle filter, we formulate a KLT tracker for tracking back from the current frame to the initial frame. On the one hand, if the assumptions of brightness constancy and small motion are indeed satisfied and that the tracking remains stable and free of drift, the KLT tracker is expected to work well both forward and in reverse. Brightness consistency as a constraint is inherently time reversible and with sufficient smoothness on the function $(I, \mathbf{x})$ and its derivatives, it can be shown that the forward and time reversed systems behave similarly under small motion assumptions.

On the other hand, in the presence of drift due to model failure, when we do the reversed tracking, the tracker does not go back to the initialization point due to the unmodeled errors that affect the tracking. Therefore, the strategy used earlier for evaluation of particle filtering-based trackers along with the fast approximation techniques is also applicable to the KLT tracker. Finally, the interested reader is referred to a feature point algorithm described in [31] that uses this concept of time reversal for achieving robust tracking.

Finally, KLT as a tracking algorithm is a point tracker and does not estimate uncertainty in any specific form (such as density or covariances). We base our evaluation statistic on just the Euclidean distance between the initial point provided to the tracker and the result of the time-reversed KLT tracker. The Euclidean distance between the two points replaces the Mahalanobis distance used for the Particle filtering scenario.

2) Evaluations of the Mean-Shift tracker: The proposed evaluation algorithm can also be extended to the mean-shift tracker [21] [22]. The mean-shift tracker contains two major components: target representation and localization. Histogram based appearance representation is adopted for the target. Target localization is achieved through a mean-shift optimization process. In mean-shift tracking algorithm, the target model is usually considered as centered at the spatial location 0 and represented by its pdf q, which can be approximated by its m-bin histograms as below:

$$\text{target model:} \qquad \hat{\mathbf{q}} = \hat{q_u}, u = 1...m \qquad \sum_{u=1}^{m} \hat{q_u} = 1 \tag{12}$$

In practice, a target is usually represented by an ellipsoidal or rectangle region in the image. With some manipulation, the probability of the feature $u = 1...m$ in the target model can be denoted by some analytical function of its location variables. In the subsequent frame, a target candidate defined at location $\mathbf{z}$ is characterized by the pdf $p(\mathbf{z})$:

$$\text{target candidate:} \qquad \hat{\mathbf{z}}(\mathbf{y}) = \hat{p_u}(\mathbf{z})_{u=1...m} \qquad \sum_{u=1}^{m} \hat{p_u} = 1 \tag{13}$$

Similarly, we can compute the probability of the feature $u = 1...m$ in the target candidate. In [21] [22], Bhattacharya coefficient is adopted to evaluate the similarity likelihood between the target model and candidate:

$$\hat{\rho}(\mathbf{z}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{z}), \hat{\mathbf{q}}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{z})\hat{q}_u} \tag{14}$$

And the distance between target model and candidates can be defined as:

$$d(\mathbf{z}) = \sqrt{1 - \rho[\hat{\mathbf{p}}(\mathbf{z}), \hat{\mathbf{q}}]} \tag{15}$$

To find the location of the target in the current frame is to minimize the distance measure with respect to $\mathbf{z}$. The Mean-shift tracker solves this minimization problem by the mean-shift procedure after linearizing the Bhattacharya coefficient using a Taylor series expansion around the location $\hat{\mathbf{z}}_0$ of the target in the previous frame. Hence, this algorithm works well when the

target candidate $\hat{p}_u(\mathbf{z})_{u=1...m}$ does not change drastically from the initial $\hat{p}_u(\hat{\mathbf{z}}_0)_{u=1...m}$, which is often valid between consecutive frames.

The mean-shift tracker is popular due to its computational efficiency and ease of implementation. However, there are two major limitations that usually cause the traditional mean-shift tracker to fail [32]. The first limitation is that the basic mean-shift procedure assumes that the scale of the object remains unchanged during tracking, which may not be true in many real cases. To handle scale change, it will bring uncertainty to the convergence of the tracker. The second limitation is that the traditional mean-shift tracker uses radially symmetric kernels which cannot adequately represent various object shapes. Therefore, like other tracking algorithms, the traditional mean-shift tracker may also often encounter difficulties during tracking, which makes it necessary to evaluate its performance in real time.

Based on the same idea we used for particle filter based trackers and the KLT tracker, we evaluate the mean-shift tracker using the distance between the forward and backward tracker. Here, the status of the tracking object can be characterized by the location (assuming the scale remains constant). Hence, simple Euclidean distance between the forward and backward kernel modes which are found by the mean-shift method is used for evaluation.

IV. Discussion

In this section, we discuss some of the properties of the evaluation algorithm. In particular we highlight potential similarities between our algorithms and tools in existing literatures. For example, ideas similar to time-reversal have been applied to the image registration problem where it is desirable for the forward and the backward maps to be inverses of each other [33] [34].

A. Similarity to the ELL

The proposed evaluation methodology is similar to the ELL statistic [9] in spirit, both involving posterior of the tracking algorithm and the prior at time $t = 0$. ELL propagates the prior density to time t and computes the inaccuracy between the t-step predicted prior and the posterior π_t. In contrast, the proposed methodology time reverses the posterior π_t back to the initial time using a time-reversed system and compares it against the prior at time $t = 0$. The main difference in

our formulation is the t-step reverse prediction is *conditioned* on the observed data, while the t-step prediction in ELL is unconditional.

B. Time-reversed Markov chain

The main idea behind the evaluation methodology involves time-reversed models. The concept, at a first glance, seems similar to *time-reversible* Markov chains [35]. Time-reversal is a concept that is common to both the evaluation methodology as well as time-reversible chains. Time-reversal produces a Markov chain whose state transition density is given in (5). However, time-reversibility of a Markov chain is a stronger statement on the nature of the state transition density. Specifically, the Markov chain is said to be time reversible when the so called *detailed balance* property is satisfied, that is, there exists a probability density p_s such that

$$p(\mathbf{x}_t = x_1|\mathbf{x}_{t-1} = x_2)p_s(\mathbf{x}_2) = p(\mathbf{x}_{t-1} = x_2|\mathbf{x}_t = x_1)p_s(\mathbf{x}_1) \tag{16}$$

The proposed evaluation methodology does require a well conditioned model for the time reversed Markov chain. However, it does not need the property of detailed balance to be satisfied for the particular model. In this sense, the concept of time-reversible Markov chains and the evaluation methodology proposed in this paper are completely different ideas.

C. Smoothing filter

In Bayesian smoothing algorithms, the quantity of interest is $p(x_t|y_{1:T})$, the posterior of the state conditioned on all observations $y_{1:T}$, including those in the future. Computation of these smoothing posteriors involves running a forward PF and a backward PF and fusing their respective posteriors systematically [36]. However, in the smoothing algorithm, there are no new constraints that are used, in the sense, that the dynamical system model (prior + state transition + observation models) is still the same. However, the proposed evaluation method depends on this concept of time-reversibility of the physical models, which is a property that is extraneous to the basic definition of the dynamical systems. In this regard, the concept of smoothing filters and the evaluation methodology are two disparate concepts; it is possible to apply the evaluation methodology to the smoothing filter.

D. Failure Modes of the Proposed Algorithm

While the proposed evaluation algorithm works well across a wide range of tracking algorithms (see Section V), there are some cases when it fails. Such failures vary with the selected tracking model and the specifi cs of data. In particular, we discuss two cases where the evaluation algorithm can potentially fail.

The fi rst scenario deals with tracking algorithms that lock onto the initial position, thereby losing track of a moving object. However, the time reversed tracker used for evaluation will also remain locked at the initial position (of the forward tracker), and give low evaluation scores, indicating a good tracking performance. This is clearly a failure mode of the evaluation methodology, although for an unreasonable tracking algorithm. However, it highlights a potential

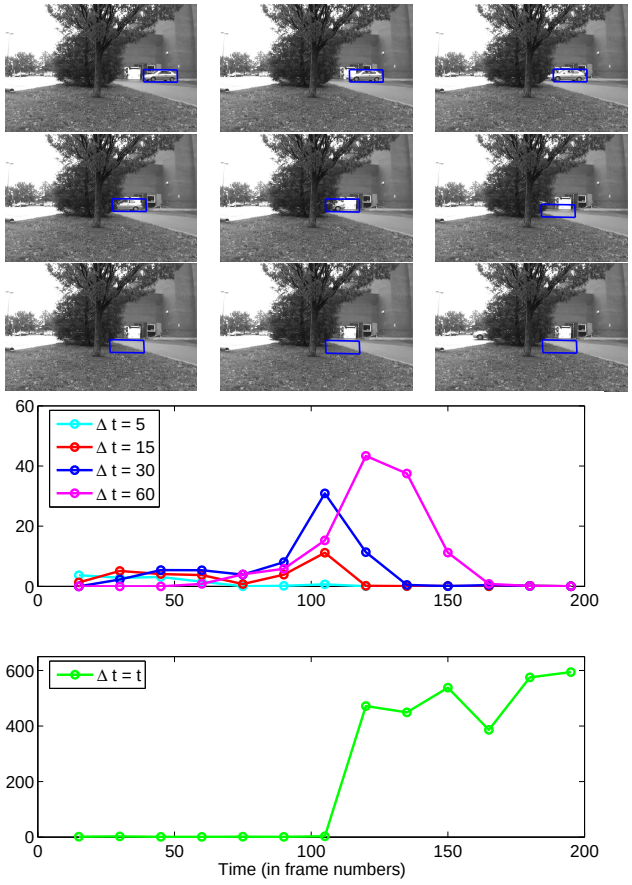


Fig. 3. Performance evaluation over occlusion. Target is completely occluded by frame number 100. (Top left to bottom right) Tracking results at frame numbers 1, 20, 40, 60, 80, 100, 120, 135 and 150 (Bottom row) Evaluation results using the proposed algorithm ($\Delta t = t$) and its fast approximations ($\Delta t = 5, 15, 30, 60$).

scenario where the evaluation methodology might fail.

A second instance of failure involve trackers that are completely guided by their motion model. This could possibly happen due to observations being rejected as outliers by the observation model, or in cases where a data association step associates a missing data state with the tracker. In such a case, the time reversibility of the motion model (most commonly used motion models are time reversible in the sense that the same model with different parameters can explain the time reversed motion) would naturally guide the tracker back to its initial value.

A more realistic situation involves a combination of the two above-mentioned scenarios. Consider an example, where a tracking algorithm loses an object in the initial few frames of a video. For the remaining frames of the video, the output of the tracking algorithm is unpredictable. However, without sufficient observations to guide the estimate, the state transition model becomes the pre-dominant model in governing the evolution of the posterior density. For tracking algorithms that use a Brownian motion model on the state transition, the mean of the posterior does not change (and hence, remains close to the prior $p(x_0)$). The evaluation score in this case can possibly be of low value.

In short, the proposed method is very useful for many types of tracking problems; with certain potential failure modes that can be detected using simple heuristics. It is also noteworthy that the proposed evaluation might fail for a particular instance of data-algorithm pair, it does not have a consistent failure mode (say such as occlusion or illumination).

V. Experiments

In this section, we present experimental results of the proposed performance evaluation method with particle filtering-based visual trackers, the Kanade-Lucas-Tomasi (KLT) and the mean-shift tracker. We first show that the proposed evaluation algorithm can detect various common failure modes in visual tracking systems using particle filters. We use the algorithm proposed in [37] as the representative tracking algorithm for this set of experiments. This algorithm uses a six-dimensional state space for capturing affine deformations, with a Brownian motion model for the state dynamics. The observation model is a template based OAM, which is a specific mixture of Gaussian model proposed in [38].

A. *Evaluation under common Tracking scenarios*

Figure 3 shows results for a video where the target is completely occluded. We used our evaluation algorithm once every 15 frames. The target undergoes occlusion around 100th frame. The proposed statistic and its fast approximations register peaks or sharp rises in value around this frame. It is noteworthy that evaluation using $\Delta t = 5$ does not seem large enough to capture the tracking failure. However, a higher value of Δt registers the loss of track. Finally, as expected, inference using fast approximations is not useful after a track failure is registered. This is because that reference point against which the algorithm is being compared is corrupted.

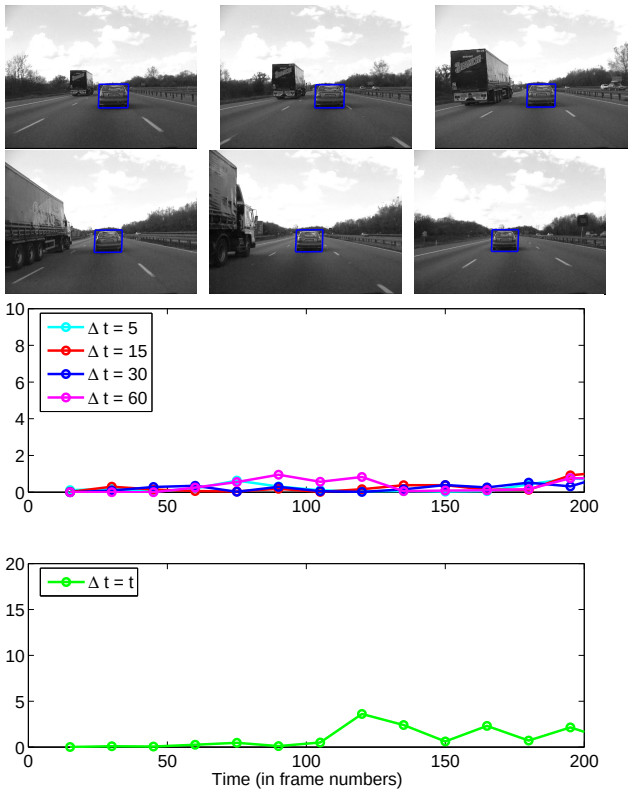


Fig. 4. Performance evaluation over slow pose change. (Top three rows) Tracking results at frame numbers 1, 40, 80, 120, 160 and 200 (Bottom rows) Evaluation results using the proposed algorithm ($\Delta t = t$) and its fast approximations ($\Delta t = 5, 15, 30, 60$).

Figure 4 shows the results of evaluation for a sequence in which a target exhibits a small change in pose, easily tracked by the tracker. As expected, the proposed evaluation methodology generates a test statistic which takes low values indicating a good tracking performance. Figure 5 shows evaluation results on an aerial sequence in which the tracker loses track of the target due to the jerky motion of the camera. The test statistics registers sharp peaks around the point

where the loss of track happens.

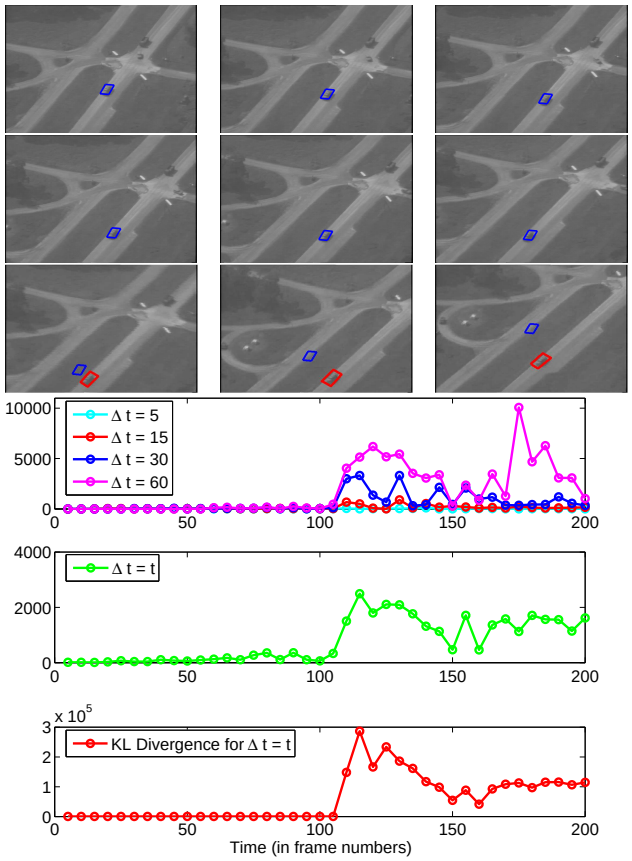


Fig. 5. Performance evaluation in an aerial sequence. The tracker loses track of the object around frame 110 due to jerky camera motion. (Top three rows) Tracking results at frame numbers 1, 20, 40, 60, 80, 100, 120, 140 and 160. The true target location is marked in red after the algorithm loses track. (Bottom row) Evaluation results using the proposed algorithm $(\Delta t = t)$, its fast approximations and the KL divergence between prior density and posterior of time reversed chain.

The proposed algorithm was tested on sequences in the PETS-2001 data set and the evaluation is compared with the ground truth. The comparison with the ground truth is done by computing the distance between the center of the target as hypothesized by the tracker to the ground truth. Figures 6 and 7 show the results on two sequences from the dataset. In Figure 6, the tracker tracks the object fairly well. Both the proposed statistic and the comparison against the ground truth take a low value. Figure 7 shows the evaluation results for a scenario involving tracking failure. While all statistics register the failure of track, the proposed statistic registers the track failure before the ground truth. This is because of the specifi c evaluation criterion used with the ground truth, which involves comparing only the centers of the target, while the bounding box

is inaccurate before the loss of track (frame 60).

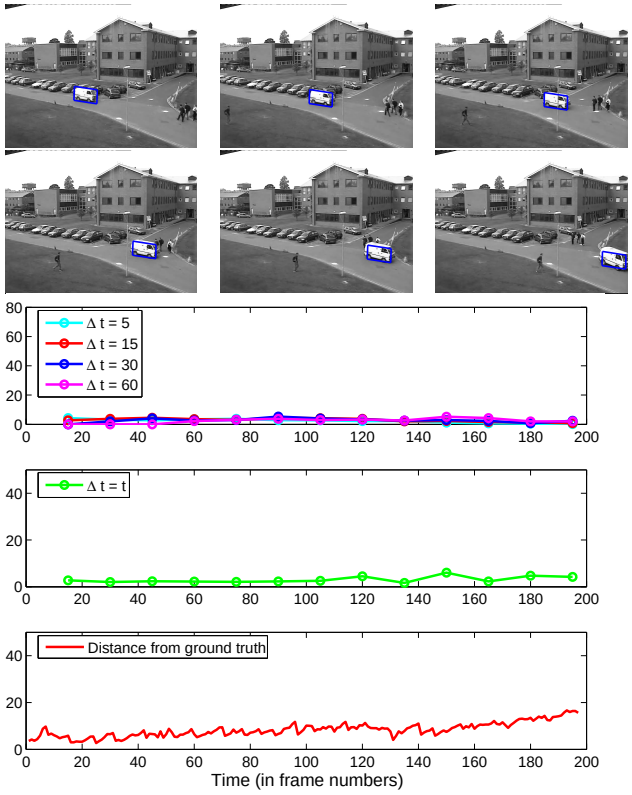


Fig. 6. Performance evaluation on a PETS sequence including ground truth. (Top three rows) Tracking results at frame numbers 1, 30, 60, 90, 120 and 160. (bottom three rows) Evaluation results using proposed statistics and its fast approximations and the ground truth. Tracking performance remains fairly constant as shown by both the ground truth and the proposed evaluation strategy.

B. Receiver Operating Characteristic

To further give a statistical evaluation of the proposed evaluation method, we organized a data set containing 40 sequences obtained from various scenarios, like outdoor/indoor, vehicle/human, optical/infrared, static/moving camera, ground/airborne, etc. These video sequences were each obtained from standard video datasets such as the PETS 2001, 2002 dataset, the aerial sequences from the VIVID dataset, the TSA dataset and other videos collected at the University of Maryland. Each sequence composes of 200 frames. The first frames of each sequence are shown in Figure 8.

Ground truth for each video was obtained manually, and comprises of a tight bounding box (parallelogram) around the target at frames $1, 20, 40, \ldots, 200$. A detection event corresponds to

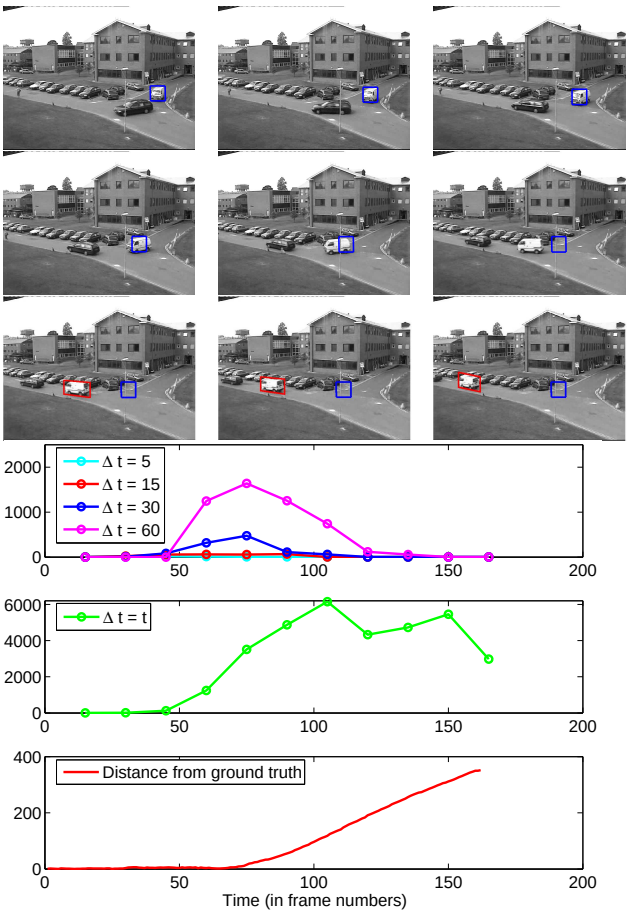


Fig. 7. Performance evaluation for a PETS sequence including ground truth. (Top three rows) Tracking results at frame numbers 1, 20, 40, 60, 80, 100, 120, 140 and 160. The true target location is marked in red after the algorithm loses track. (bottom three rows) Evaluation results using proposed statistics and its fast approximations and the ground truth.

detecting the failure of the tracking algorithm. The true state of nature is obtained by using the spatial overlap between the ground truth and the region assigned as the target by the MAP estimate of the tracking algorithm. A low overlap between the two confirms that the tracking performance is poor, and is denoted as a detection of failure.

After obtaining the evaluation statistic values, we vary a threshold to get different detection and false alarm rates and plot the ROC curve. We plot operating curves under various operating scenarios.

1) Length of the Video: We performed experiments characterizing the performance of the evaluation algorithm as the length of the video increases. This is to quantify the possible small degradation of performance as the length of the video increases. Figure 9 shows the ROC curves

Fig. 8. The collected data set for obtaining ROC curve of the proposed evaluation method.

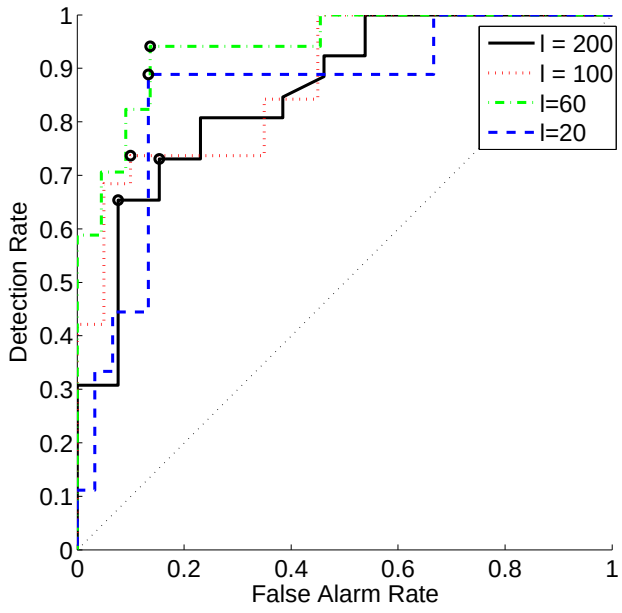


Fig. 9. We characterize the performance of evaluation as the length of the video (number of frames) changes. The encircled points are the (Bayesian operating points) for equi-prior, and $0-1$ cost structure.

for videos of length $l = (20, 60, 100, 200)$ frames. Also marked are the Bayes' operating point for equi-prior and $0-1$ cost structure. This allows us to get a quantitative assessment of the

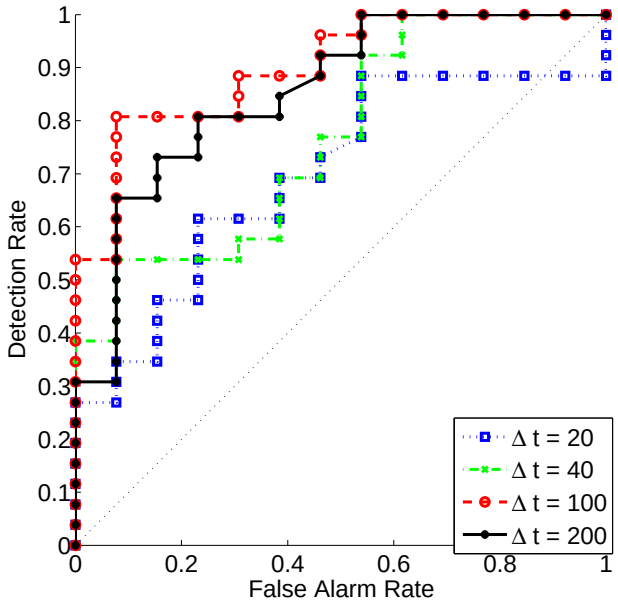


Fig. 10. The ROC curves of the proposed evaluation method with OAM-based particle filtering. The evaluation was performed at the final frame of a 200 frame long video. Each line corresponds to a fast approximation scheme with different approximation length. Note that performance does not degrade much between the basic evaluation strategy $\Delta t = 200$ and an approximation $\Delta t = 100$.

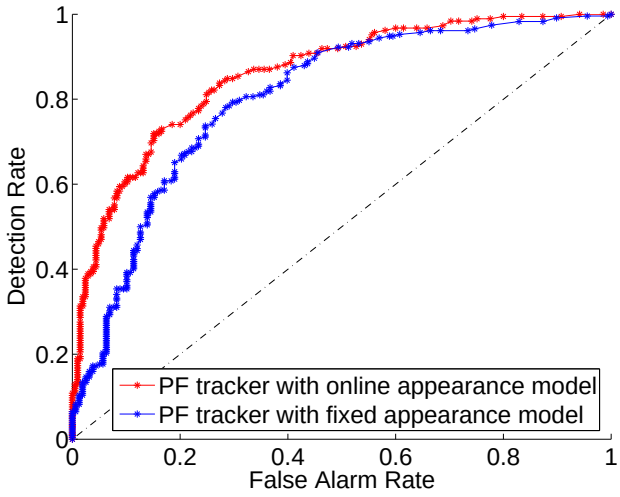


Fig. 11. The performance comparisons of the proposed evaluation method between OAM-PF tracker and FAM-PF tracker. The evaluation performance remains fairly same over two different tracking algorithms hinting at the robustness of the evaluation strategy over different tracking algorithms.

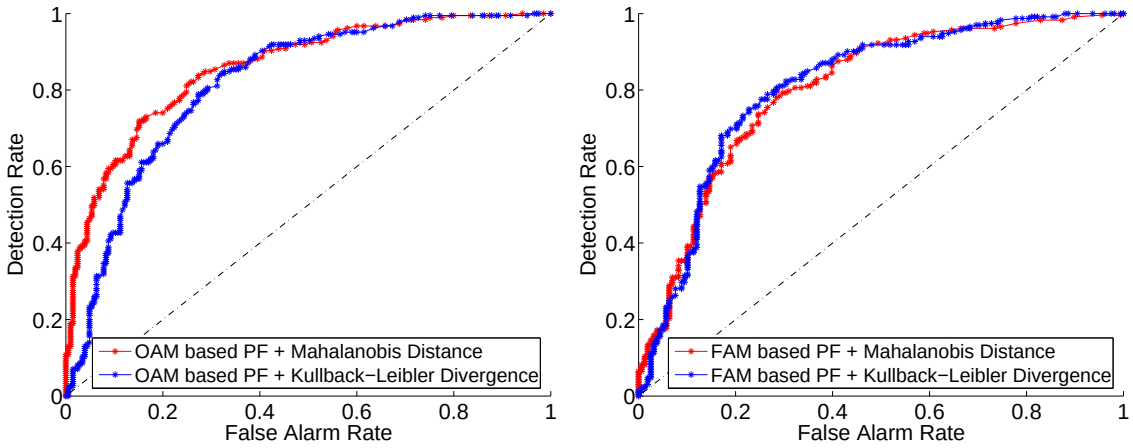


Fig. 12. The performance comparisons of the proposed evaluation method by using Mahalanobis distance and KL divergence based evaluation statistics respectively. Evaluation performance seems fairly similar under either metric. This could possibly hint at the unimodality of the densities around the prior time instant $t = 0$. The similarity in evaluation justifies the use of the faster Mahalanobis distance in the place of the KL divergence which is expensive to compute for point clouds.

Bayes risk and its degradation as the length of the video increases.

Length of the Video	20	60	100	200
Bayes Risk	0.12	0.1	0.18	0.21

TABLE II

THE BAYES RISK OF THE EVALUATION ALGORITHM.

This allows us to interpret the ROC curves better. For example, at $l = 60$ the detection probability is $P_D = 0.94$ at a false alarm probability $P_F = 0.13$, which falls to $P_D = 0.73$ when $l = 200$ (same the same false alarm rate P_F).

2) Fast Approximation: We next show the differences in performance between the basic evaluation method and its fast approximations at various Δt. The curves in Figure 10 show ROC for evaluation at the last frame of the video (at $t = 200$) using the basic algorithm ($\Delta t = 200$) and fast approximations at $\Delta t = 20, 40, 100$. Note that in the fast approximation method, if an intermediate point is declared as a track failure, then all subsequent points are also declared as track failures. This contributes to the poor performance at $\Delta t = 20$. It is seen from the figure that with appropriate intervals, like 100 frames, the performance of the fast approximation strategy

is comparable to the basic framework, while keeping the computation time constant.

3) Tracking Algorithm: We ran the evaluation method for particle filter-based tracking algorithms based on the OAM and a fixed appearance model (FAM). We show both the ROC curves in Figure 11. The test data set is the same for both trackers, while the evaluation performance is different by a small margin. Further, a comparison with the ROC curves in the evaluations of the KLT and mean-shift trackers (shown in Figures 16 and 14) suggests that the performance of the evaluation method may reveal some characteristics of the underlying tracking algorithm. We plan to explore this as a part of future work.

4) Choice of Evaluation Metric: In computing the evaluation statistic, we proposed to use Mahalanobis distance in place of distances such as the KL divergence which compares two densities directly. To test the effectiveness of this Mahalanbois metric, we also computed the KL divergence-based distance when using the basic framework where the computation is feasible given the Gaussian prior distribution. The comparisons in Figure 12 show that there is no significant difference between using the Mahalanobis distance and the KL divergence in our experiments. This could possibly be due to the unimodality of the densities around the prior time instant $t = 0$. The similarity in evaluation justifies the use of the faster Mahalanobis distance in the place of the KL divergence which is expensive to compute for point clouds.

5) Evaluation of Mean Shift Tracker: Using the same data set as used for the particle filtering-based tracker evaluation, we tested the evaluation algorithm on the traditional mean-shift tracker. By excluding some sequences where the traditional mean-shift tracker completely fails from the very beginning, which makes the evaluation completely unreliable as we discussed in the above section, the final test set contains 26 sequences and 260 evaluation points in total. Figure 13 shows the evaluation results for a sequence with slow tracking drift. The corresponding evaluation score for this sequence increases indicating the increasing drift in tracking. We use the same ground truth as in the particle filtering case. The true state of the track (failure or not) was determined by comparing the hypothesized region to the ground truth. Lack of sufficient overlap between the two was labeled as a failed tracker. The evaluation metric was designed based on the distance between the output of the time-reversed mean-shift and the initial guess. The Euclidean distance between the two was used (as the scale of the tracker remains fixed, which makes the Euclidean distance almost equivalent to spatial overlap). As before, we computed the ROC curve using the dataset of 26 sequences (see Figure 14) show-casing the performance of the

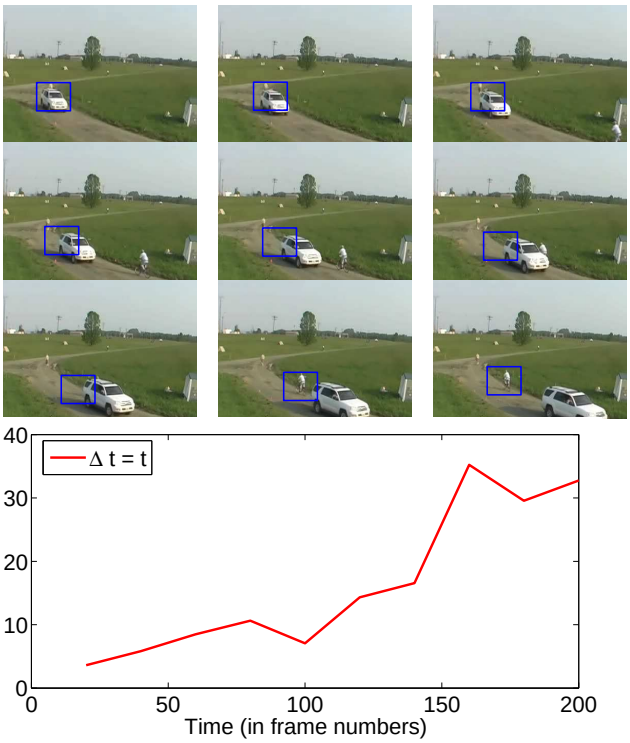


Fig. 13. Performance evaluation for a sequence tracked by the mean shift tracker with slow tracking drift. (Top three rows) Tracking results at frame numbers 20, 40, 60, 100, 120, 140, 160, 180 and 200. (bottom) Evaluation results using the proposed statistics under the basic mode.

evaluation method for the Mean Shift tracker. We designed this work based on the code from http://www.cs.bilkent.edu.tr/ ismaila/MUSCLE/MSTracker.htm.

6) Evaluation of KLT Feature Tracker: We also tried to use the proposed method to detect the tracking failures of the KLT feature tracker. The KLT is a feature point tracking algorithm and hence, we can generate multiple test cases from a single image. For our experiments, we used four images, and selected 200 features per image using the KL feature selection criterion (see Figure 15). Selecting 200 features per image gives us a mix of good and bad feature points (in terms of their tractability). We create a synthetic sequence by translating the images. This gives us the ground truth for the sequence. A feature point is considered to have drifted if it diverges by more than 2 pixels from the ground truth. As before, we use ROC curves to characterize the detection of drift using our evaluation methodology. The ROC curve in Figure 16 indicates that the evaluation method works very well for the KLT tracker. We also show some detection results in Figure 16.

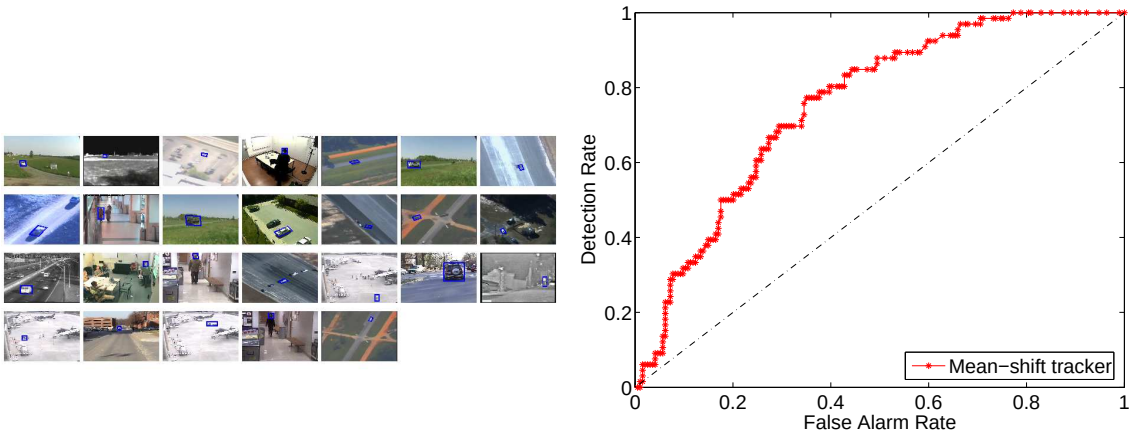


Fig. 14. Evaluation results for the mean shift tracking algorithm over a dataset of 26 videos snapshots from whom are shown in the left-image. The ROC curve of the evaluation algorithm for the tracker using the basic mode. From 26 videos of 200 frames each, we obtained 260 evaluation points which were used to generate the ROC curves.

Fig. 15. Test images used for the KLT tracking algorithm overlaid with the selected feature points. Each image was translated to create a synthetic video providing ground truth for evaluation.

C. Ranking the Performance of Trackers

We have showed above that the proposed online evaluation algorithm can detect tracking failures in the absence of ground truth data. In addition to this, the proposed algorithm can also be used to compare the performance of different trackers. We compared the performances of the three trackers we used in this paper: the particle filter based tracker with OAM, the particle filter based tracker with FAM and the mean-shift tracker. Since the KLT tracker is a feature tracking algorithm and requires a different test set, we did not include it in this ranking experiment.

The experiment was performed as follows. For each tracker, we count the number of tracking failures reported at different false alarm rates over the data set shown in Figure 14. For each tracker, we have 260 evaluation points (26 sequences, with 10 evaluation points each). We can

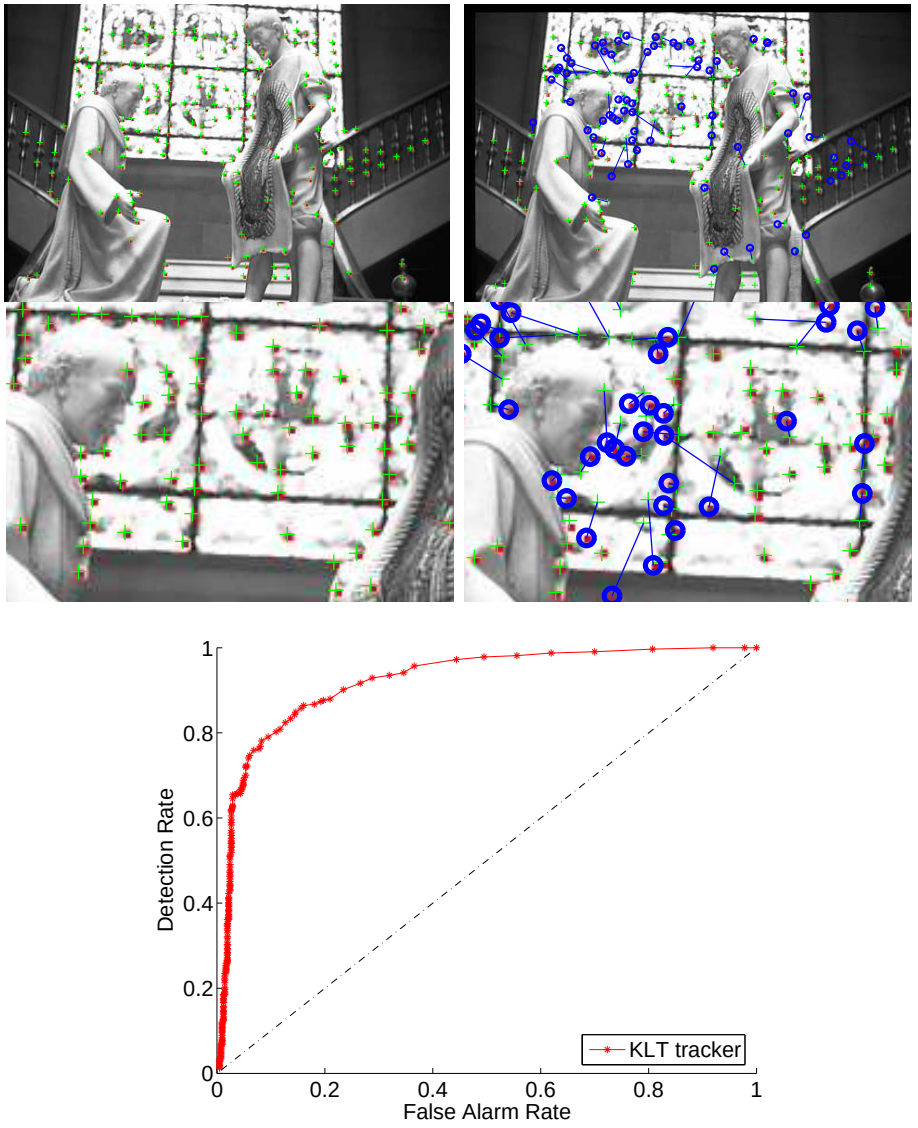


Fig. 16. Performance of the evaluation method for the KLT feature point tracking algorithm. (Top left column) The initial frame and the enlarged details for KLT tracking. The red dot shows the initialization of the KLT tracker and the green plus sign shows the ground truth. (Top right column) The final frame and its enlarged details for KLT tracking. As we can see, many tracking feature points (red dot) have drifted away from their ground truth locations (green plus sign) and been detected by the evaluation algorithm (the blue circles indicate the drifted points which are connected with their ground truth locations by blue line). (Bottom) The ROC Curve of the evaluation method for KLT tracker using 4 images and 800 feature points.

see from the figure that at a false alarm rate of 0.6, the detection rates for all three trackers are very close, therefore we can compare the performance of each tracker in terms of the number of detected tracking failures at this point. Intuitively, a tracking algorithm with more detected track failure should correspond to a poorer tracking performance. From the figure, the ranking

order for the three trackers we used here results in (PF tracker with FAM) $<$ (PF tracker with OAM) $<$ (Mean-shift tracker), from left to right, worst to best performance. Notice that:

$$\begin{aligned} D_{\text{bad}} &= G_{\text{bad}} * \text{Detection Rate} + G_{\text{good}} * \text{False Alarm Rate} \\ G_{\text{bad}} + G_{\text{good}} &= \text{Total number of evaluation points} \end{aligned} \tag{17}$$

where D_{bad} is the detected number of failures, G_{bad} is the real number of failures (ground truth).

With the help of the ROC curves of the proposed evaluation algorithm together with the number of detected failures, we can recover the ground truth number of tracking failures. The results are: 152 (PF tracker with FAM), 90 (PF tracker with OAM) and 66 (Mean-shift tracker). As we can see, the ground truth ranking result of these three trackers gives the same ordering as the proposed evaluation algorithm.

It is noteworthy that the above comparison is valid only because that the detection rates for the tracking algorithms are similar at the false alarm rate of 0.6. At a different operating point where the detection rates are not similar (for the same false alarm) such a comparison becomes invalid as the tracking with a higher detection rate tends to report larger number of detected failures.

D. Summary

To summarize the results, the following properties of the proposed evaluation scheme are highlighted. The proposed evaluation algorithm is shown to detect common failure modes in visual tracking and also compares favorably with ground truth based evaluation. The value of Δt is shown to be critical in the effi ciency of the fast approximations. A value of $\Delta t = 40, 60$ seems reasonably large enough to register failures. It should be noted that fast approximations are meaningless after detection of failure, as the reference point against which they are compared does not correspond to good tracking. The choice of threshold to declare poor performance can be decided for a specifi c tracking system by inspection from the ROC curve. The choice is also influenced by the value of Δt. It can be seen that for all the experiments in this paper, the inference from the proposed evaluation agrees well with subjective evaluation of track failures. The supplemental material includes videos showcasing the working of the evaluation algorithm.

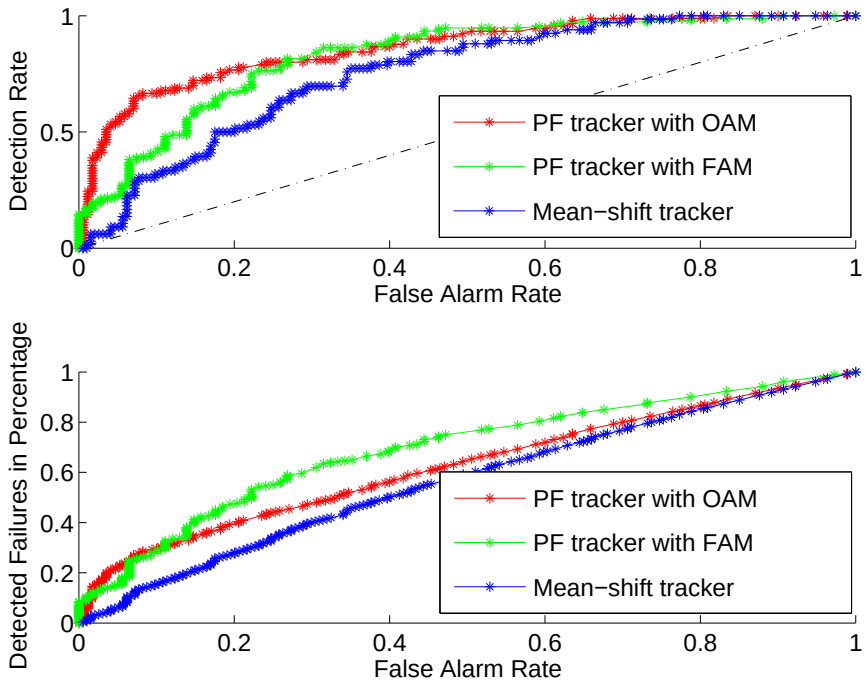


Fig. 17. The ranking result of the three trackers: the particle filter based tracker with OAM and FAM, the Mean-shift tracker. The above is the corresponding ROC curve of each tracker on the data set described in Figure 14. The bottom plot is the number of detected failures (the number is in percentage) using the proposed evaluation algorithm for each tracker at different false alarm rates.

VI. Conclusion

In this paper, we present a method to provide automatic and online evaluation of the tracking performance in visual systems without the knowledge of ground truth. The proposed evaluation algorithm works by verifying the prior at time $t = 0$ against the posterior of a time-reversed chain. The time-reversed chain is initialized using the posterior of the tracking algorithm. We characterize the performance of the algorithm using ROCs under various operating conditions. While the focus in the paper has been on systems using particle filtering, the evaluation method is fairly independent of the tracking algorithms used. In this regard, we show that the algorithm works well for other tracking approaches such as the KLT and the mean shift tracker. We also show that the evaluation methodology can also be used to rank different tracking algorithms according to their performance. We envision the use of the evaluation methodologies proposed in this paper for online verification and ranking of tracking performance. Future directions of

research include tracking algorithms that optimize the evaluation metric so as to minimize the chances of track failure.

REFERENCES

[1] J. Black, T. Ellis, and P. Rosin, "A novel method for video tracking performance evaluation," in *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003, pp. 125–132.

[2] A.T. Nghiem, F. Bremond, M. Thonnat, and R. Ma, "A new evaluation approach for video processing algorithms," in *IEEE Workshop on Motion and Video Computing*, Feb. 2007, pp. 15–15.

[3] T. Schlogl, C. Beleznai, M. Winter, and H. Bischof, "Performance evaluation metrics for motion detection and tracking," in *International Conference on Pattern Recognition*, 2004, vol. 4.

[4] A. T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, "ETISEO, performance evaluation for video surveillance systems," in *IEEE Conference on Advanced Video and Signal Based Surveillance*, Sept. 2007, pp. 476–481.

[5] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, M. Boonstra, and V. Korzhova, "Performance Evaluation Protocal for Face, Person and Vehicle Detection & Tracking in Video Analysis and Centent Extraction (VACE-II) CLEAR-Classification of Events, Activities and Relationships," *Tampa, January*, 2006.

[6] J.C. Nascimento and J.S. Marques, "Performance Evaluation of Object Detection Algorithms for Video Surveillance," *IEEE TRANSACTIONS ON MULTIMEDIA*, vol. 8, no. 4, pp. 761, 2006.

[7] B. Georis, F. Bremond, M. Thonnat, and B. Macq, "Use of an evaluation and diagnosis method to improve tracking performances," in *International Conference on Visualization, Imaging and Image Proceeding*, 2003.

[8] T. List, J. Bins, J. Vazquez, and R.B. Fisher, "Performance evaluating the evaluator," in *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Oct. 2005, pp. 129–136.

[9] N. Vaswani, "Additive change detection in nonlinear systems with unknown change parameters," *IEEE Trans. Signal Processing*, 2006.

[10] C. Andrieu, A. Doucet, S.S. Singh, and V.B. Tadic, "Particle methods for change detection, system identification, and control," *Proceedings of the IEEE*, pp. 423–438, March 2004.

[11] R. Gerlach, C. Carter, and R. Kohn, "Diagnostics for time series analysis," *Journal on Time Series Analysis*, , no. 3, pp. 309 – 330, 1999.

[12] J. Vermaak, C. Andrieu, A. Doucet, and S. Godsill, "Particle methods for bayesian modelling and enhancement of speech signals," *IEEE Trans. Speech and Audio Processing*, , no. 3, March 2002.

[13] F. van der Heijen, "Consistency checks for particle filters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, , no. 1, pp. 140–145, January 2006.

[14] Le Lu, Xiangtian Dai, and Gregory Hager, "A particle filter without dynamics for robust 3d face tracking," *IEEE Proc. Computer Vision and Pattern Recognition Workshops*, 2004.

[15] C.E. Erdem, A. Murat Tekalp, and B. Sankur, "Metrics for performance evaluation of video object segmentation and tracking without ground-truth," *Proc. IEEE Int'l Conf. on Image Processing*, pp. 69–72, October 2001.

[16] C. E. Erdem, B. Sankur, and A. M. Tekalp, "Non-rigid object tracking using performance evaluation measures as feedback," *Proc. IEEE Int'l Conf. on Comput. Vis. and Patt. Recog.*, pp. II–323– II–330, 2001.

[17] H. Wu and Q. Zheng, "Performance self-evaluation of visual tracking systems," in *Army Science Conference*, 2004.

[18] Bruce D. Lucas and Takeo Kanade, "An iterative image registration technique with an application to stereo vision," *International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

[19] Carlo Tomasi and Takeo Kanade, "Detection and tracking of point features," *Carnegie Mellon University Technical Report CMU-CS-91-132*, April 1991.

[20] Jianbo Shi and Carlo Tomasi, "Good features to track," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[21] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 142–149, 2000.

[22] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transaction on Pattern Analysis Machine Intellegence*, pp. 564–575, 2003.

[23] H. Wu, A. C. Sankaranarayanan, and R. Chellappa, "In situ evaluation of tracking algorithms using time reversed chain," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

[24] Michael Isard and Andrew Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[25] Arnaud Doucet, Nando De Freitas, and Neil Gordon, *Sequential Monto Carlo Methods in Practice*, New York, NY: Springer-Verlag, 2001.

[26] A. Veeraraghavan, R. Chellappa, and M. Srinivasan, "Shape-and-behavior encoded tracking of bee dances," *IEEE Transaction on Pattern Analysis Machine Intellegence*, pp. 463–476, Mar. 2008.

[27] Qinfen Zheng and Rama Chellappa, "Automatic feature point extraction and tracking in image sequences for arbitrary camera motion," *International Journal of Computer Vision*, pp. 31–76, June 1995.

[28] Mike Klass, Mark Briers, Nando De Freitas, Arnaud Doucet, Simon Maskell, and Dustin Lang, "Fast particle smoothing: If i had a million particles," *Proc. IEEE Int. Conf. on Machine Learning*, 2006.

[29] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification (2nd Edition)*, Wiley-Interscience, 2000.

[30] J. Goldberger, S. Gordon, and H. Greenspan, "An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures," *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 487–493, October 2003.

[31] H. Wu, R. Chellappa, A.C. Sankaranarayanan, and S.K. Zhou, "Robust Visual Tracking Using the Time-Reversibility Constraint," *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, 2007.

[32] Alper Yilmaz, "Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, Jun. 2007.

[33] S.K. Yeung and P. Shi, "Stochastic inverse consistency in medical image registration," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 188–196. Springer, 2005.

[34] GE Christensen and HJ Johnson, "Consistent image registration," *IEEE Transactions on Medical Imaging*, pp. 568–582, July 2001.

[35] S. M. Ross, "Stochastic processes," 1996.

[36] M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: if I had a million particles," *Proceedings of the 23rd international conference on Machine learning*, pp. 481–488, 2006.

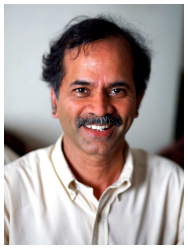
[37] Shaohua Kevin Zhou, Rama Chellappa, and Baback Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. Image Processing*, , no. 11, pp. 1491–1506, November 2004.

[38] AD Jepson, DJ Fleet, and TF El-Maraghi, "Robust online appearance models for visual tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 10, pp. 1296–1311, 2003.

Hao Wu (S' 04) received her B.S. and M.S. degrees in electrical engineering from University of Science and Technology of China in 2000, 2003 respectively. She is pursuing the Ph.D. degree at the department of electrical and computer engineering, University of Maryland, College Park. Her research interests are in computer vision, multimedia data analysis, pattern recognition and statistical machine learning. Since March 2009, she has been a research scientist at Kodak Research Lab, Rochester, NY.

Aswin C. Sankaranarayanan (S' 04) received the B.Tech. degree from the Indian Institute of Technology, Madras in 2003. He is currently a Ph.D candidate in the Department of Electrical and Computer Engineering at the University of Maryland, College Park. He is the recipient of the Distinguished dissertation fellowship from the ECE Dept for 2008-09, and the Future Faculty Fellowship in A. J. Clark School of Engineering from Spring 2007. He was also a participant in IBM's Emerging Leaders In Multimedia Workshop held at the T. J. Watson Research Center in October 2007.

His research interests are in computer vison, geometry, statistics and signal processing.



Rama Chellappa (S'78-M'79-SM'83-F'92) received the B.E. (Hons.) degree from the University of Madras, India, in 1975 and the M.E. (Distinction) degree from the Indian Institute of Science, Bangalore, in 1977. He received the M.S.E.E. and Ph.D. Degrees in Electrical Engineering from Purdue University, West Lafayette, IN, in 1978 and 1981 respectively.

Since 1991, he has been a Professor of Electrical Engineering and an affiliate Professor of Computer Science at University of Maryland, College Park. He is also affiliated with the Center for Automation Research (Director) and the Institute for Advanced Computer Studies (Permanent Member). In 2005, he was named a Minta Martin Professor of Engineering. Prior to joining the University of Maryland, he was an Assistant (1981-1986) and Associate Professor (1986-1991) and Director of the Signal and Image Processing Institute (1988-1990) at University of Southern California, Los Angeles. Over the last 28 years, he has published numerous book chapters, peer-reviewed journal and conference papers. He has co-authored and edited many books on MRFs, face and gait recognition and collected works on image processing and analysis. His current research interests are face and gait analysis, markerless motion capture, 3D modeling from video, image and video-based recognition and exploitation and hyper spectral processing, Prof. Chellappa served as the associate editor of four IEEE Transactions, as a Co-Editor-in-Chief of Graphical Models and Image Processing and as the Editor-in-Chief of IEEE Transactions on Pattern Analysis and Machine Intelligence. He served as a member of the IEEE Signal Processing Society Board of Governors and as its Vice President of Awards and Membership. He is serving a two-year term as the President of IEEE Biometrics Council. He has received several awards, including an NSF Presidential Young Investigator Award, four IBM Faculty Development Awards, an Excellence in Teaching Award from the School of Engineering at USC, two paper awards from the International Association of Pattern Recognition, the Technical Achievement and Meritorious Service Awards from the IEEE Signal Processing Society and the IEEE Computer Society. At University of Maryland, he was elected as a Distinguished Faculty Research Fellow, as a Distinguished Scholar-Teacher, received the Outstanding Faculty Research Award from the College of Engineering, an Outstanding Innovator Award from the Office of Technology Commercialization and an Outstanding GEMSTONE Mentor Award. He is a Fellow of IEEE, the International Association for Pattern Recognition and the Optical Society of America. He has served as a General the Technical Program Chair for several IEEE international and national conferences and workshops. He is a Golden Core Member of IEEE Computer Society and serving a two-year term as a Distinguished Lecturer of the IEEE Signal Processing Society.