Particle dynamics and multi-channel feature dictionaries for robust visual tracking

Srikrishna Karanam
karans3@rpi.edu
Yang Li
yangli625@gmail.com
Richard J. Radke
rjradke@ecse.rpi.edu

Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
110 8th St.
Troy, NY USA

Abstract

We present a novel approach to solve the visual tracking problem in a particle filter framework based on sparse visual representations. Current state-of-the-art trackers use low-resolution image intensity features in target appearance modeling. Such features often fail to capture sufficient visual information about the target. Here, we demonstrate the efficacy of visually richer representation schemes by employing multi-channel feature dictionaries as part of the appearance model. To further mitigate the tracking drift problem, we propose a novel dynamic adaptive state transition model, taking into account the dynamics of the past states. Finally, we demonstrate the computational tractability of using richer appearance modeling schemes by adaptively pruning candidate particles during each sampling step, and using a fast augmented Lagrangian technique to solve the associated optimization problem. Extensive quantitative evaluations and robustness tests on several challenging video sequences demonstrate that our approach substantially outperforms the state of the art, and achieves stable results.

1 Introduction

Recent advances in the application of compressive sensing to traditional computer vision problems such as face recognition [24, 26] inspired several visual tracking approaches based on sparse representations. The core idea of these approaches is to build an appearance model of the object using several pre-defined templates. The problem of tracking the object is then cast as finding a sparse approximation in the subspace spanned by the templates. In [13], Mei and Ling introduced the l_1 tracker, demonstrating impressive tracking results. Given an appearance model $\mathbf{A} = [\mathbf{t}_1 \cdots \mathbf{t}_n] \in \mathbb{R}^{m \times n}$ of an object formed using a set of templates $\mathbf{t}_i \in \mathbb{R}^m, i = 1, \ldots, n$, they express a tracking result $\mathbf{y} \in \mathbb{R}^m$ as $\mathbf{y} = \mathbf{A}\mathbf{x} + \varepsilon$, where $\mathbf{x} \in \mathbb{R}^n$ is the sparse coefficient vector that is to be recovered, and $\varepsilon \in \mathbb{R}^m$ is used to account for partial occlusions. The l_1 tracking algorithm hypothesizes that $\mathbf{x}$ and ε are sparse for a good tracking candidate, and recovers them by solving an l_1 regularized least squares problem. Subsequently, the candidate with the least projection error in the template subspace is chosen as a tracking target, and a Bayesian state inference model in a particle filter framework is used to track the object over time.

In spite of the impressive progress achieved by the l_1 tracker and its recent variants, several issues remain that often lead to tracking failures. In this paper, we introduce a novel particle filter approach to mitigate such problems. The key contributions of our proposed method are:

Dynamic model. Intuitively, it is easy to see that incorporating dynamic information will make a tracking algorithm more robust to the drift problem. However, most related approaches [3, 11, 13, 20, 28, 29, 30] do not take this information into account, employing only a fixed-variance Gaussian distribution to represent the state transition model. To mitigate this problem, we propose to adaptively learn the variance from past states using a dynamic state transition model. Specifically, we employ an autoregressive model in conjunction with block Hankel matrices to continuously learn the dynamics from past data.

Appearance model. Most existing approaches use extremely low resolution image intensity features (e.g. 12 × 15 in [3, 13, 20], 32 × 32 in [12, 28, 31]) as part of the appearance model. Such features do not capture sufficient visual information required to reliably track the object and avoid drift. To mitigate this problem, we propose a three-channel appearance dictionary comprised of image intensity information, normalized image gradient magnitudes, and histograms of oriented gradients to construct an appearance model of the object. We demonstrate that using rich visual representations as part of the appearance model improves tracking accuracy and stability.

Adaptive candidate filtering for speed. Typically, tracking algorithms in a particle filter framework use a fixed number of particles to approximate the posterior distribution (e.g., 600 in [13, 20], 400 in [11, 28, 30]). This number is often a trade-off between tracking accuracy and computational complexity, and limits the use of rich visual representations as part of the appearance model. In this work, we demonstrate that many particles are not necessary to reliably track an object, given the initial location. Specifically, we propose to adapt the number of particles required during the state estimation process using the Kullback-Leibler (KL) distance measure [9]. In addition, we use a fast augmented Lagrangian technique to solve the associated optimization problem, demonstrating superior tracking results at higher speeds than related sparse representation based approaches.

We validate our method on several challenging publicly available video sequences and demonstrate that our method achieves a significant 10% improvement in the area under the curve of the success plot compared to the current state of the art.

2 Related Work

Several improvements have been proposed to the original l_1 tracker. Zhang *et al.* [28] exploited the low-rank nature of the appearance model $\mathbf{A}$ and also explicitly took background information into account to mitigate the drift problem. Bao *et al.* [3] used a fast variant of the proximal gradient optimization algorithm to achieve impressive improvements in the tracking speed. There have also been efforts to incorporate subspace representations into the appearance model. Wang *et al.* [20] used principal component analysis (PCA) basis vectors as part of the appearance model and demonstrated the efficacy of using both subspace learning and sparse representation in achieving robust tracking. Inspired by the success of multi-task learning [5] in image annotation [16] and image classification [27] problems, Zhang *et al.* [30] exploited the interdependencies of particles to learn joint particle representations in a multi-task framework, demonstrating significant computational gains over the l_1 tracker. Zhong *et al.* [31] used sparse generative and discriminative models in a collaborative fashion to effectively deal with drastic appearance changes. More recently, Wang *et al.* [21] demonstrated the efficacy of using online dictionary learning algorithms in updating the target appearance model. A comprehensive discussion and experimental evaluation of these and several other related tracking approaches can be found in [18, 25].

3 Approach overview

We formulate visual tracking as a sparse representation problem in a particle filtering framework. Given the initial location of the target to be tracked, we warp the image into a 64×64 pixel template, thereby representing the position of the target in each frame using a four-dimensional state vector $\mathbf{s_t} \in \mathbb{R}^4$. By perturbing the initial location by a few pixels (typically, 1–3), we form m such templates. In our experiments, we set $m = 10$. We then construct three appearance dictionaries using these templates: an intensity channel dictionary $\mathbf{A}^1 = [\mathbf{t}_1^1 \cdots \mathbf{t}_m^1]$, a normalized gradient magnitude dictionary $\mathbf{A}^2 = [\mathbf{t}_1^2 \cdots \mathbf{t}_m^2]$, and a Histogram of Oriented Gradients (HOG) [6] dictionary $\mathbf{A}^3 = [\mathbf{t}_1^3 \cdots \mathbf{t}_m^3]$, where each dictionary $\mathbf{A}^j \in \mathbb{R}^{d_j \times m}$. Now, given a potential target particle $\mathbf{y}$, we compute its intensity feature vector $\mathbf{y}^1$, normalized gradient magnitude vector $\mathbf{y}^2$, and HOG vector $\mathbf{y}^3$. In each feature channel, we hypothesize that a good target candidate can be represented as a sparse linear combination of the dictionary templates, and recover the sparse vector by solving the following convex optimization problem:

$$(\mathbf{x}^{j*}, \varepsilon^{j*}) = \arg\min_{\mathbf{x}^j, \varepsilon^j} \|\mathbf{x}^j\|_1 + \|\varepsilon^j\|_1 \text{ s.t. } \mathbf{y}^j = \mathbf{A}^j \mathbf{x}^j + \varepsilon^j, j = 1,2,3 \tag{1}$$

where ε^j is used to account for error and partial occlusion, and $\mathbf{x}^j$ is the sparse coefficient vector we wish to recover.

Tracking in a particle filtering framework proceeds by generating several hypotheses, and testing each for its likelihood. In our context, each hypothesis is a candidate particle, represented by its state vector. To generate potential candidate particles, most related approaches employ a fixed-variance Gaussian distributed state transition model. Instead, we search for potential candidate particles using an adaptive state transition model incorporating dynamic information. Our key insight is that learning from the dynamics of past state vectors can lead to improved and efficient search for new candidate particles, leading to both accuracy and speed benefits. Formally, if $\mathbf{s}_t \in \mathbb{R}^4$ is the current state vector, we estimate the elements of the next state vector $\mathbf{s}_{t+1}$ using the following transition model:

$$s_{t+1}(i) = s_t(i) + r(i)\sigma_{t+1}(i) \tag{2}$$

where $s_{t+1}(i)$ is the i^{th} element of $\mathbf{s}_{t+1}$, $r(i) \sim \mathcal{N}(0,1)$ is a normally distributed random number, and $\sigma_{t+1}(i)$ is the i^{th} element of the variance vector $\sigma_{t+1} \in \mathbb{R}^4$ that we estimate on-the-fly using the dynamics of the past states.

Another key contribution of our approach is that we demonstrate the computational tractability of using visually richer representation schemes as part of appearance modeling. We achieve this by adaptively pruning candidate particles using the Kullback-Leibler (KL) distance measure for probability density functions, and solving Equation 1 using a fast augmented Lagrangian technique.

4 Algorithm description

4.1 Review of particle filtering

Particle filters are useful tools for tracking the state of a dynamic system modeled in a Bayesian framework. Given past observation vectors $\mathbf{y}_{1:t-1}$, the probability of the state $\mathbf{s}_t$ is predicted as $p(\mathbf{s}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{s}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{s}_{t-1}$. Now, given the observation vector $\mathbf{y}_t$ at time t, the probability is updated using $p(\mathbf{s}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{s}_t)p(\mathbf{s}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{t-1})}$. The particle filter approximates $p(\mathbf{s}_t|\mathbf{y}_{1:t})$ by a set of samples $\mathbf{s}_t^i, i = 1,2,\ldots,N$. Each sample has an associated weight w_t^i which is computed as $w_t^i = w_{t-1}^i \frac{p(\mathbf{y}_t|\mathbf{s}_t^i)p(\mathbf{s}_t^i|\mathbf{s}_{t-1}^i)}{q(\mathbf{s}_t|\mathbf{s}_{1:t-1},\mathbf{y}_{1:t})}$, where $q(\mathbf{s}_t|\mathbf{s}_{1:t-1},\mathbf{y}_{1:t})$ is an importance distribution from which the samples $\mathbf{s}_t^i$ are drawn.

Here, we assume $q(\mathbf{s}_t|\mathbf{s}_{1:t-1},\mathbf{y}_{1:t}) = p(\mathbf{s}_t^i|\mathbf{s}_{t-1}^i)$, and hence the previous equation becomes $w_t^i = w_{t-1}^i p(\mathbf{y}_t|\mathbf{s}_t^i)$. In other words, the weights become proportional to the likelihood of the observation vector $\mathbf{y}_t$ given the state vector $\mathbf{s}_t$. We compute this observation likelihood using the following expression: $p(\mathbf{y}_t|\mathbf{s}_t) = \exp\left(-\sum_{j=1}^{3} \alpha_j \|\mathbf{A}^j\mathbf{x}^j - \mathbf{y}_t^j\|_2^2\right)$, where α_j is a positive constant, and $\mathbf{A}^1, \mathbf{A}^2, \mathbf{A}^3$ are the three appearance dictionaries discussed next.

4.2 Modeling target appearance

As noted in the previous section, most of the related tracking algorithms use simple image intensity information of extremely low resolution patches to form the appearance model. In our approach, we use visually richer representations by computing several channels of quantized gradient orientations, normalized image gradients, and intensity features. Our motivation for this approach stems from two observations. First, intuitively, a richer appearance representation can capture more visual data from the target, thereby potentially mitigating the drift problem. A second, more compelling, reason is related to a recent result from information theory. In [23], Wright and Ma showed that if the columns of an appearance dictionary $\mathbf{D}$ are highly correlated, any sparse signal $\mathbf{s}$ can be recovered by solving the following l_1-minimization problem:

$$\min \|\mathbf{s}\|_1 + \|\mathbf{e}\|_1 \text{ s.t. } \mathbf{y} = \mathbf{D}\mathbf{s} + \mathbf{e} \tag{3}$$

Additionally, they also show that as the dimension of the dictionary $\mathbf{D}$ increases, the percentage of errors that the problem represented by Equation 3 can correct approaches 100%. Their basic assumption that the columns of $\mathbf{D}$ should be correlated is valid in our problem. As noted previously, we form the appearance dictionaries by perturbing the initial location by a few pixels, thereby leading to highly overlapped templates and correlated feature channels.

First, for each 64 × 64 template image, we take the intensity vectors t_i^1 and aggregate them into the intensity feature dictionary $\mathbf{A}^1 = [\mathbf{t}_1^1 \cdots \mathbf{t}_m^1]$. Next, we take the image gradient information into account by computing the normalized gradient magnitudes t_i^2 for each template pixel. These values are then vectorized and aggregated to form the normalized gradient magnitude dictionary $\mathbf{A}^2 = [\mathbf{t}_1^2 \cdots \mathbf{t}_m^2]$. If $g(p,q)$ represents the gradient magnitude at pixel (p,q), the normalized gradient magnitude is computed as $\tilde{g}(p,q) = \frac{g(p,q)}{(s(p,q)+f)}$, where s is a smoothed version of the gradient magnitude g, and f is a small positive constant. Specifically, s is computed by convolving g with a $k \times k$ triangular filter. In our experiments, we set $k = 5$. These normalized gradient magnitudes $\mathbf{t}_i^2$ are then used to compute several channels (6, in our experiments) of gradient histograms. These values are then vectorized and aggregated to form the HOG dictionary $\mathbf{A}^3 = [\mathbf{t}_1^3 \cdots \mathbf{t}_m^3]$. Therefore, we model the appearance of the target to be tracked using the three-channel dictionaries $\mathbf{A}^1$, $\mathbf{A}^2$, and $\mathbf{A}^3$.

4.3 Dynamic adaptive state transition model

As noted earlier, most of the related tracking approaches [3, 11, 13, 20, 28, 29, 30] use a simple Gaussian distributed state transition model. Specifically, if $\mathbf{s}_t$ is the current state vector, the elements of the next state vector $\mathbf{s}_{t+1}$ are estimated as

$$s_{t+1}(i) = s_t(i) + r(i)\sigma_0(i) \tag{4}$$

where $r(i) \sim \mathcal{N}(0,1)$ is a normally distributed random number and $\sigma_0(i)$ is the i^{th} element of $\sigma_0 \in \mathbb{R}^4$, a fixed variance vector set manually. However, such a fixed-variance state transition model can cause significant drift errors in the approximation of the particle filter, resulting in severe tracking failures. In our work, we propose a simple modification to Equation 4 as follows:

$$s_{t+1}(i) = s_t(i) + r(i)\sigma_{t+1}(i) \tag{5}$$

where $\sigma_{t+1} \in \mathbb{R}^4$ is a dynamic adaptive variance vector computed as

$$\sigma_{t+1} = \max(\min(\sigma_0\sqrt{e_t}, \sigma_{\max}), \sigma_{\min}) \tag{6}$$

where $\sigma_{\max} \in \mathbb{R}^4$ and $\sigma_{\min} \in \mathbb{R}^4$ are upper and lower bounds on σ_{t+1} we impose manually, and max and min are element-wise operators. This adaptive variance model increases or decreases the search area for new particle sampling depending on whether the prediction error is large or small. To compute the scalar e_t, we take into account past dynamics to model the temporal evolution of the state vector $\mathbf{s}_t$. If $\tilde{\mathbf{s}}_t$ corresponds to the particle with the highest observation probability at time t, and $\hat{\mathbf{s}}_t$ is the estimated state vector at time t using past state vectors, then we have $e_t^j = \|\mathbf{y}_{\tilde{\mathbf{s}}_t}^j - \mathbf{y}_{\hat{\mathbf{s}}_t}^j\|_2$, where $j \in \{1,2,3\}$ and $\mathbf{y}_{\mathbf{s}}^1$, $\mathbf{y}_{\mathbf{s}}^2$, and $\mathbf{y}_{\mathbf{s}}^3$ are respectively the intensity, normalized gradient, and HOG feature vectors of the candidate image corresponding to the state $\mathbf{s}$. We then add up these feature channel errors to give the combined prediction error $e_t = e_t^1 + e_t^2 + e_t^3$ at time t.

To estimate $\hat{\mathbf{s}}_t$, we exploit the temporal evolution of the state vector $\mathbf{s}_t$. Specifically, we consider an n^{th} order autoregressive model as follows:

$$\hat{\mathbf{s}}_t = c_1\mathbf{s}_{t-1} + c_2\mathbf{s}_{t-2} + \cdots + c_n\mathbf{s}_{t-n} \tag{7}$$

To compute the coefficient vector $\mathbf{c} = [c_n, c_{n-1}, \ldots, c_1]$, we form the block Hankel matrix [1] associated with the state vector $\mathbf{s}$:

$$\mathbf{H_s}^{p,q} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_q \\ \mathbf{s}_2 & \mathbf{s}_3 & \cdots & \mathbf{s}_{q+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_p & \mathbf{s}_{p+1} & \cdots & \mathbf{s}_{q+p-1} \end{bmatrix} \tag{8}$$

In our work, we set $p = t-n-1$ and $q = n$. With the Hankel matrix defined as in Equation 8 and the autoregressive model defined as in Equation 7, we make the following straightforward observation:

$$\mathbf{H_s}^{t-n-1,n} \begin{bmatrix} c_n \\ c_{n-1} \\ \vdots \\ c_1 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{n+1} \\ \mathbf{s}_{n+2} \\ \vdots \\ \mathbf{s}_{n+(t-n-1)} \end{bmatrix} \tag{9}$$

The coefficient vector $\mathbf{c}$ can be recovered from Equation 9 using a simple least squares error minimization technique. Specifically, the coefficient vector $\mathbf{c}$ can be written as $\mathbf{c} = \mathbf{H}_p \begin{bmatrix} \mathbf{s}_{n+1} & \mathbf{s}_{n+2} & \cdots & \mathbf{s}_{n+(t-n-1)} \end{bmatrix}^\top$, where $\mathbf{H}_p$ is the psuedo-inverse of ${\mathbf{H}_\mathbf{s}}^{t-n-1,n}$, given by $\mathbf{H}_p = ({\mathbf{H}_\mathbf{s}}^{t-n-1,n\top}{\mathbf{H}_\mathbf{s}}^{t-n-1,n})^{-1}{\mathbf{H}_\mathbf{s}}^{t-n-1,n\top}$. A key advantage of our approach is that by continuously learning the variance σ_{t+1} from past data, we can adapt to time-varying dynamics more effectively when compared to the fixed variance model of Equation 4, thereby better positioning our algorithm to deal with the tracking drift problem.

4.4 Adaptive candidate filtering

In our algorithm, the number of candidate particles to sample plays a crucial role. Choosing a higher number of particles in a given search area helps to exhaustively search for potential target particles, but has a downside in that the computational complexity also increases, making the overall tracking algorithm impractical to use. On the other hand, randomly choosing a lower number of particles at each sampling step is also not prudent, possibly leading to severe tracking drift. Several related tracking approaches choose a fixed number (400-600) [3, 11, 12, 14, 21, 29, 31] of particles as a compromise between computational complexity and tracking accuracy. However, even this number is quite high, resulting in impractical average tracking run-times.

Here, we incorporate the dynamic model described in the previous section into the adaptive particle filtering framework of Fox [9]. We divide the state space into discrete bins and at each new frame, determine the number of particles required so that the KL-distance between the maximum likelihood estimate of the particle-based posterior and the true posterior probability does not exceed a threshold error ν. Specifically, we choose the desired number of particles from a chi-square distribution as $N = \frac{1}{2\nu}\chi^2_{k-1,1-\delta}$, where k is the number of bins with support, ν is the desired approximation error, and $(1-\delta)$ is the probability with which the KL-distance approximation can guarantee an error less than ν. A proof of this expression can be found in the supplementary material. In our experiments, we found that performing spatial binning using only the translational parameters of the state vector resulted in acceptable approximations. Therefore, to determine if a new particle $\mathbf{s}_t$ falls into a bin, we determine the vector $\mathbf{r}_b = [\frac{\mathbf{s}_{tt_x}}{\sigma_{tt_x}} \frac{\mathbf{s}_{tt_y}}{\sigma_{tt_y}}]$, where $\mathbf{s}_{tt_x}, \sigma_{tt_x}$, and $\mathbf{s}_{tt_y}, \sigma_{tt_y}$ represent the translational parameters of $\mathbf{s}_t$ and σ_t respectively, and check if $\mathbf{r}_b$ exists in the set of currently binned particles.

4.5 Efficiently solving the optimization problem

Solving the optimization problem of Equation 1 efficiently is key to achieving fast tracking. In our work, we use a fast augmented Lagrangian technique to solve the problem of Equation 1. Specifically, we eliminate the equality constraints by introducing a Lagrange multiplier, and optimize the resulting cost function in an iterative fashion.

Formally, for each candidate particle, we have the intensity feature vector $\mathbf{y}^1$, the normalized gradient magnitude feature vector $\mathbf{y}^2$, and the HOG feature vector $\mathbf{y}^3$. In each channel $\mathbf{y}^j$, we solve the optimization problem of Equation 1. For this problem, the augmented Lagrangian can be written as

$$\mathcal{L}_\zeta(\mathbf{x}^j, \varepsilon^j, \rho) = \|\mathbf{x}^j\|_1 + \|\varepsilon^j\|_1 + \frac{\zeta}{2}\|\mathbf{y}^j - \mathbf{A}^j\mathbf{x}^j - \varepsilon^j\|_2^2 + \rho^\top(\mathbf{y}^j - \mathbf{A}^j\mathbf{x}^j - \varepsilon^j) \qquad (10)$$

In our experiments, we set $\zeta = 2m/\|\mathbf{y}^j\|_1$ [26]. Subsequently, $\mathbf{x}^j$ and ε^j can be recovered in an iterative fashion using the following scheme:

$$\varepsilon^j_{i+1} = \arg\min_{\varepsilon^j} \mathcal{L}_\zeta(\mathbf{x}^j_i, \varepsilon^j, \rho_i) = \text{shrink}\Big(\mathbf{y}^j - \mathbf{A}^j\mathbf{x}^j_i + \frac{1}{\zeta}\rho_i), \frac{1}{\zeta}\Big) \tag{11}$$

$$\mathbf{x}^j_{i+1} = \arg\min_{\mathbf{x}^j} \mathcal{L}_\zeta(\mathbf{x}^j, \varepsilon^j_{i+1}, \rho_i) \tag{12}$$

$$\rho_{i+1} = \rho_i + \zeta(\mathbf{y}^j - \mathbf{A}^j\mathbf{x}^j_{i+1} - \varepsilon^j_{i+1}) \tag{13}$$

where $(\text{shrink}(\mathbf{t}, \alpha))_i = \text{sgn}(t_i)\max\{|t_i| - \alpha, 0\}$, $i = 1, 2, \ldots, n$. As noted above, the update step for ε^j_{i+1} has an analytic solution, and we solve the update step for $\mathbf{x}^j_{i+1}$ using the Fast Iterative Shrinkage Thresholding Algorithm (FISTA) [4]. A detailed derivation and analysis of the entire process can be found in the supplementary material.

4.6 Robust dictionary update

As with any template based approach, it is essential that we update the appearance dictionary as we progress through the video sequence, since the dictionary defined in the first frame will not accurately represent the target appearance over time. In our work, we use an intuitive approach to update the appearance dictionary. In each frame, we use the ε^{j*} determined by solving Equation 1 to compute an error/occlusion ratio l^j by finding the ratio of the number of non-zero entries in ε^{j*} to its length. In each feature channel, we find the angle distance θ_j between the vectors representing the most likely candidate particle $\mathbf{y}^j$ and the appearance template $\mathbf{t}^j_i$ with the highest coefficient vector x^j_i. We then update the template $\mathbf{t}^j_i$ with $\mathbf{y}^j$ only if θ_j exceeds a threshold t_{θ_j}, and l^j is below a threshold t_{l_j}.

5 Experiments and Results

5.1 Experimental setup

We implemented our tracking approach in MATLAB. All experiments are performed on an Intel Core 2 Duo 2.66 GHz CPU with an installed RAM of 8 GB.

Datasets. We consider 25 publicly available video sequences[1] that represent several challenging aspects in visual tracking: illumination variation, scale variation, occlusion, non-rigid object deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter, and low resolution.

Evaluation methodology: We quantitatively evaluate the results of our tracking approach using several evaluation metrics and robustness tests. First, we report the average center location error (CLE), defined as the average Euclidean distance between the tracked center location of the target (B_t) and the ground truth (B_{gt}), and the average success rate (SR), defined as the average fraction of frames that have an overlap ratio $o_r = \frac{\text{area}(B_t \cap B_{gt})}{\text{area}(B_t \cup B_{gt})} > 0.5$ [8]. We then report the overall mean performance in terms of the area under the curve (AUC) of the success plot, in which we plot the overlap precision, defined as the fraction of frames that have overlap ratio above a threshold, versus overlap ratio for thresholds ranging from 0 to 1. We also perform the temporal robustness (TRE) and spatial robustness (SRE) tests [25]

[1] Videos demonstrating our tracking performance can be found in the supplementary material.

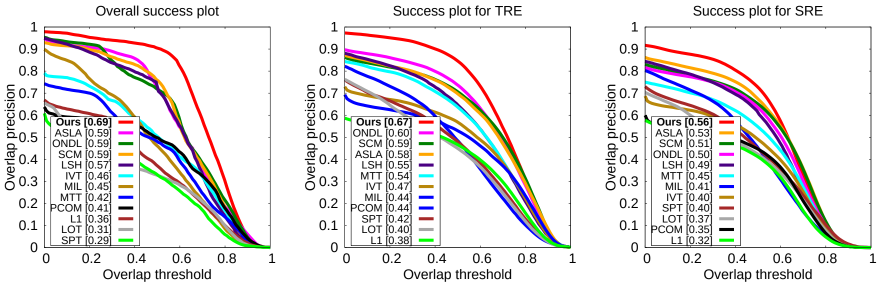
to evaluate the sensitivity of our approach to scale and position initialization errors. In the TRE tests, we divide each sequence temporally into 10 subsegments, and run our tracker for each segment. In the SRE tests, by shifting and scaling the initial bounding box, we sample 12 different locations of the initial bounding box, and run our tracker in each case. We compare the results of our approach for each of these evaluation metrics against several recently proposed trackers: L1 [3], MTT [30], ONDL [21], SCM [31], LSH [10], ASLA [12], PCOM [19], LOT [15], SPT [22], MIL [2], and IVT [17]. We chose these algorithms due to their state-of-the-art performance and public availability of source codes for re-implementation.

5.2 Quantitative results

In this section, we discuss the quantitative results of our tracking approach. Table 1 shows the CLE and success rate averaged over all the 25 test sequences[2]. Our approach provides very promising results on these metrics, resulting in a mean CLE reduction of 65% and a mean success rate increase of 18% over ONDL [21], the next best performing approach in our experiments. The overall mean success plot is shown in Figure 1 (a). As can be noted from the figure, our approach achieves a significant improvement of 10% for the AUC of the success plot compared to each of SCM [31], ASLA [12] and ONDL [21], the next best performing approaches.

Table 1: Average center location error (in pixels) and success rate (in percentage) for all 25 test sequences. **Red** - Best, *Blue* - Second best. We also show a speed (in fps) comparison with other sparse representation based approaches. "$*$" indicates the algorithm is not based on sparse visual representation.

Criterion	Ours	L1	MTT	ONDL	SCM	ASLA	PCOM*	LSH*	LOT*	SPT*	MIL*	IVT*
Success rate	**93.87**	48.07	59.85	*75.77*	72.11	72.81	55.31	67.53	45.96	42.35	44.51	56.83
CLE	**7.03**	68.66	46.65	*20.07*	26.55	22.82	51.95	22.19	53.78	60.61	36.44	50.72
Speed	2.5	*8.2*	0.4	0.5	0.05	0.7	6.5	7	0.2	0.1	**8.5**	6.5



(a) (b) (c)

Figure 1: Success plots averaged over all the 25 test sequences. In all three plots, the area under the curve (AUC) is reported in the legend. In each of the three tests, our approach outperforms the state of the art.

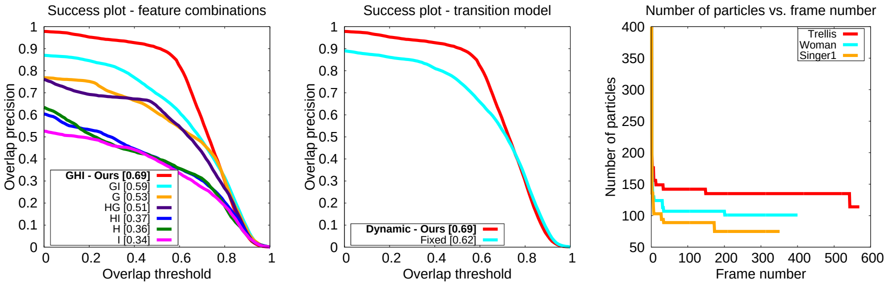
The success plots for both TRE and SRE tests are also shown in Figure 1 (b)-(c). As can be noted from the plots, our approach outperforms the state of the art for both of the

[2] Per-sequence results can be found in the supplementary material.

robustness tests, resulting in mean AUC improvements of 7% and 3% for TRE and SRE, respectively, over the next best performing approach.

Speed. We also compare the average tracking run-time of our approach with the other competing approaches. As can be seen from Table 1, the average run-time of our approach is favorable when compared to other related approaches based on sparse representation, while giving much better tracking accuracy. This can be attributed to the fact that we adaptively prune the number of candidate particles at each frame, and solve the problem of Equation 1 using a fast Lagrangian method.

Validating key contributions. Finally, we provide experimental evidence to corroborate each of the two primary contributions of this paper: the use of multi-channel feature dictionaries, and the use of a dynamic state transition model. To validate the efficacy of multi-channel feature dictionaries, we tested all seven possible combinations of the three feature types, intensity (I), normalized gradient (G), and HOG (H). To validate the efficacy of our transition model, we compared it with the fixed-variance transition model with variance σ_0. Each of these two experiments were performed on all 25 test sequences, and the success plots obtained for each case are shown in Figure 2 (a)-(b). As can be noted from the figure, our algorithmic decisions indeed resulted in significant improvements, with AUC improvements of 10% and 7% in the feature combinations experiment and the transition model experiment respectively. We also provide evidence to corroborate the efficacy of adaptive candidate filtering. In Figure 2 (c), we can see how the number of particles needed for particle filter approximation drops as we proceed through a video sequence. Finally, we note that in our experiments, on an average for all the sequences, we end up requiring less than 25% of initial number of particles (which is set to 400) for approximating the posterior probability density.



(a) (b) (c)

Figure 2: (a)-(b) Success plots for the feature combinations experiment and the transition model experiment. (c) A plot of the number of particles needed for three example sequences is also shown.

6 Conclusions and Future Work

We proposed and validated the efficacy of incorporating particle dynamical information and rich visual representations about the target in achieving accurate and stable object tracking. Extensive experiments on challenging video sequences demonstrated the superiority of our method to the current state of the art. While we achieve accurate tracking at reasonable frame rates, there is much work to be done to achieve fast real-time tracking. To this end, exploiting the high degree of coherence of the appearance dictionaries in designing fast sparse recovery algorithms could be an exciting direction of study in the future. Additionally, incorporating target color information [7] into our appearance modeling framework could lead to further improvements in tracking accuracy.

Acknowledgement

This material is based upon work supported by the U.S. Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Award 2013-ST-061-ED0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

References

[1] Mustafa Ayazoglu, Binlong Li, Caglayan Dicle, Mario Sznaier, and Octavia I Camps. Dynamic subspace-based coordinated multicamera tracking. In *ICCV*, 2011.

[2] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE T-PAMI*, 33(8):1619–1632, 2011.

[3] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji. Real time robust l_1 tracker using accelerated proximal gradient approach. In *CVPR*, 2012.

[4] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIIMS*, 2(1):183–202, 2009.

[5] Xi Chen, Weike Pan, James T Kwok, and Jaime G Carbonell. Accelerated gradient method for multi-task sparse learning problem. In *ICDM*, 2009.

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014.

[8] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes VOC challenge. *IJCV*, 88(2):303–338, June 2010.

[9] Dieter Fox. Adapting the sample size in particle filters through KLD-sampling. *IJRR*, 22(12):985–1003, 2003.

[10] Shengfeng He, Qingxiong Yang, Rynson W. H. Lau, Jiang Wang, and Ming-Hsuan Yang. Visual tracking via locality sensitive histograms. In *CVPR*, 2013.

[11] Zhibin Hong, Xue Mei, Danil Prokhorov, and Dacheng Tao. Tracking via robust multi-task multi-view joint sparse representation. In *ICCV*, 2013.

[12] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, 2012.

[13] Xue Mei and Haibin Ling. Robust visual tracking using l_1 minimization. In *ICCV*, 2009.

[14] Xue Mei, Haibin Ling, Yi Wu, Erik Blasch, and Li Bai. Minimum error bounded efficient l_1 tracker with occlusion detection. In *CVPR*, 2011.

[15] Shaul Oron, Aharon Bar-Hillel, Dan Levi, and Shai Avidan. Locally orderless tracking. In *CVPR*, 2012.

[16] Ariadna Quattoni, Xavier Carreras, Michael Collins, and Trevor Darrell. An efficient projection for $l_{1,\infty}$ regularization. In *ICML*, 2009.

[17] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.

[18] Arnold W. M. Smeulders, Dung M. Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *IEEE T-PAMI*, 36:1442–1468, 2014.

[19] Dong Wang and Huchuan Lu. Visual tracking via probability continuous outlier model. In *CVPR*, 2014.

[20] Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Online object tracking with sparse prototypes. *IEEE T-IP*, 22(1):314–325, 2013.

[21] Naiyan Wang, Jingdong Wang, and Dit-Yan Yeung. Online robust non-negative dictionary learning for visual tracking. In *ICCV*, 2013.

[22] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. Superpixel tracking. In *ICCV*, 2011.

[23] John Wright and Yi Ma. Dense error correction via l^1-minimization. *IEEE T-IT*, 56(7), 2010.

[24] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE T-PAMI*, 31(2):210–227, 2009.

[25] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[26] Allen Y Yang, Zihan Zhou, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Fast l_1-minimization algorithms for robust face recognition. *IEEE T-IP*, 22(8):3234–3246, 2013.

[27] Xiao-Tong Yuan and Shuicheng Yan. Visual classification with multi-task joint sparse representation. In *CVPR*, 2010.

[28] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Low-rank sparse learning for robust visual tracking. In *ECCV*. 2012.

[29] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via multi-task sparse learning. In *CVPR*, 2012.

[30] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via structured multi-task sparse learning. *IJCV*, 101(2):367–383, 2013.

[31] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, 2012.