Visual Tracking using D2-Clustering and Particle Filter

Ramin Raziperchikolaei and Mansour Jamzad
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
Email: razi@ce.sharif.edu, jamzad@sharif.edu

Abstract - *Since tracking algorithms should be robust with respect to appearance changes, online algorithms has been investigated recently instead of offline ones which has shown an acceptable performance in controlled environments. The most challenging issue in online algorithms is updating of the model causing tracking failure because of introducing small errors in each update and disturbing the appearance model (drift). In this paper, we propose an online generative tracking algorithm in order to overcome the challenges such as occlusion, object shape changes, and illumination variations. In each frame, color distribution of target candidates is obtained and the candidate having the lowest distance to the object distribution is considered as the object. In addition, in our work, the particle filter structure is used in which the samples are weighted proportional to their distance to the model. The model which is a color distribution is updated using D2-clustring algorithm. The most distinctive features of our algorithm are: 1) Updating the model using D2-clustering, 2) Avoiding drifting by using the color distribution of the target in the first and last frame, and 3) Detecting of occlusion by considering distance between the model and the best candidate. Experimental results show that our tracker outperforms other algorithms in videos containing challenging scenarios.*

Keywords - *Visual tracking, D2-clustering, Particle filter, Adaptive methods, Mallows distance*

I. INTRODUCTION

Object tracking is an open problem in machine vision which has many practical applications such as surveillance, driver assistance, human computer interaction, etc. Although previous studies have shown major successes achieved for object tracking in the case that prior information exits [1], tracking becomes very challenging if all the information about the target should be provided in the first frame.

From one point of view, all tracking algorithms can be divided into two types: generative and discriminative. In the modeling procedure, if only the object information is used, the algorithm is known to be generative. On the other hand, if the background information is used as well as object information the algorithm is discriminative. From another point of view, all tracking algorithms can be categorized into either online or offline approaches. In offline methods, model is created in the first frame, and there will be no update through all the subsequent frames. In [2] and [3] offline generative methods have been presented in which the object model is created from color or intensity histograms and this model remains constant until the end of algorithm. An offline discriminative algorithm for car tracking is presented in [4] which motion coefficients are computed with maximizing SVM score. Offline methods fail to track objects in videos that contain variations in object or environment such as shape changes or illumination variations. To avoid failure of object tracking under these situations, online methods could be suggested as an alternative, in which the model is updated frame by frame. Online discriminative methods update their classifiers incrementally by using new positive and negative samples obtained from object and background in each frame. In [5], an online method for distinguishing object from background is presented selecting the best feature among 49 features in RGB color space in each frame. In [6], object tracking is performed by using a strong classifier, which is a combination of 5 weak classifiers; it replaces the worst weak classifiers with the new ones in each frame. In [7], an online boosting algorithm is presented for real time tracking which updates selectors with Haar-like features extracted from object and background in each frame.

Online generative algorithms consider a template having minimum distance to the model as the object, then update the model with information included in this template. In [8], an incremental principal component analysis method learns the eigenbases online during object tracking process. In [9], each target candidate is represented as a linear combination of the templates which are updated incrementally. Kwon et al. [10] integrate multiple basic trackers, constructed from basic observations and motion models, into one compound tracker for online tracking.

The most challenging issue in updating of the model is the drift problem [11]. Tracker finds object in the current frame and uses information acquired from new location of the object to update the model. Tracker introduces small errors in its decisions affecting the model during update phase. Finally, accumulation of these errors led to failure of tracking. To overcome this challenge, semi supervised methods are introduced which take the first frame patches as labeled samples and all the other frames patches as unlabeled [12] [13]. In [12], a semi supervised boosting algorithm is proposed. At first, this algorithm assigns labels to patches with respect to offline and online classifiers trained separately, then it updates selectors. Since labeling is too dependent to the labels of first frame patches, the results of the algorithm are very similar to offline methods. In MILBoost Tracker [14], positive and negative samples are placed in positive and negative bags. Then the weak classifiers are trained online to learn a discriminative model. In multiple instance learning problems, positive bags can contain negative samples, so this method is relatively robust to drift. It should be mentioned that MILBoost assumes that all the samples in positive bags are positive during model updating which this assumption can cause drift.

978-1-4673-5604-6/12/$31.00 ©2012 IEEE

In this paper, we propose an online generative method for real time object tracking in videos in which appearance changes, illumination variations, and occlusion exist. In our work, in the first frame, object's hue distribution is computed and considered as the appearance model. In subsequent frames, candidate target with the smallest distance to the model is selected as the object which is used for updating the model. To compute the distance between distributions, the Mallows distance measure is used [15]. The update phase of proposed method is based on D2-clustering algorithm which takes some discrete distributions and returns the best prototypes for input distributions [16]. To update the model, object distribution in the first frame is also given to D2-clustering algorithm as well as appearance model (represented as a distribution) and current frame object distribution. Then, D2-clustering returns one prototype which is the updated model and is used in tracking of the object in the next frame. Using the first frame distribution in each update reduces the effect of small errors and makes the algorithm robust to drift. Under the condition that we don't use D2-clustering, tracker fails in videos containing appearance changes since it would be offline and the model is assumed to be constant. In addition, in each update, if we don't use the color distribution of the target in the first frame, a failure will happen due to the drift problem. We should mention that if distance between the model and the best candidate template is higher than a threshold, we realize that occlusion has occurred and the model is not updated. Finally, particle filter framework is used for propagating sample distributions over time.

The rest of the paper is organized as follows. Particle filter and D2-clustering algorithms are reviewed in sections II and III, respectively. Section IV explains D2-tracking algorithm proposed in this paper in details. In section V, we present the experimental results that indicate the robustness of our algorithm in comparison with other methods. Conclusions are drawn in section VI.

II. PARTICLE FILTER

The goal of tracking using Bayesian inference system is to obtain the posterior distribution of object state in time t denoted by X_t, given all the measurements up to time t denoted by $Z_{1:t}$. Thus, it is supposed that $p(X_{t-1}|Z_{1:t-1})$ is available and the posterior distribution $p(X_t|Z_{1:t})$ should be constructed recursively in two steps: prediction and update. The probability $p(X_t|Z_{1:t})$ regarding Markov prior and Bayes rule is computed as

$$p(X_t|Z_{1:t}) \propto p(Z_t|X_t) \int p(X_t|X_{t-1}) p(X_{t-1}|Z_{1:t-1}) dX_{t-1}. \quad (1)$$

In particle filter [17] posterior probability $p(X_t|Z_{1:t})$ is approximated by a set of weighted samples $\{X_t^k\ \pi_t^k\}_{k=1}^K$ where X_t^k and π_t^k show k^{th} particle's state and weight at time t respectively. Weights are computed as

$$\pi_t^k = \pi_{t-1}^k \frac{p(Z_t|X_t^k)\, p(X_t^k|X_{t-1}^k)}{q(X_t^k|X_{t-1}^k, Z_t)} \quad (2)$$

where $q(.)$ is the importance density that samples are drawn from it. We choose importance density to be the prior which yields $\pi_t^k = p(Z_t|X_t^k)$.

In our paper, the state of particles at time t is represented as a three dimensional vector $X_t = (X_t^x, X_t^y, X_t^s)$ where X_t^x, X_t^y, X_t^s indicate x, y position and scale of the target respectively. The motion model $p(X_t|X_{t-1})$ is supposed to be a Gaussian distribution as follows:

$$p(X_t|X_{t-1}) = G(X_{t-1}, \sigma^2) \quad (3)$$

where $X_{t-1}$and σ^2 are the mean and variance of the distribution. σ^2 is also a three dimensional vector which shows the variance of x, y position and scale. The observation likelihood $p(z_t|x_t^k)$ is set according to distance between particle distribution and model. If the distance is low the weight of particle should be high so we use the following equation for observation likelihood

$$p(z_t|x_t^k) = \exp(-Dist(m, p_k)) \quad (4)$$

where m is the model, p_k is the k^{th} particle and $Dist(.)$ function calculates distance between two distributions.

III. D2-CLUSTERING

D2-clustering algorithm is proposed in [16] for Image annotation. Since our method is based on this algorithm, in this section, we review it briefly. Suppose that there are N signatures denoted by $B_i, i = 1, \ldots, N$ each contains d discrete distributions. We show j^{th} distribution of i^{th} signature as

$$B_{ij} = \left\{\left(v_{i,j}^{(1)}, p_{i,j}^{(1)}\right), \ldots, \left(v_{i,j}^{(m_{i,j})}, p_{i,j}^{(m_{i,j})}\right)\right\} \quad (5)$$

where $v_{i,j}^{(k)}, k = 1, \ldots, m_{i,j}$ are vectors on which the distribution B_{ij} takes positive probability $p_{i,j}^{(k)}$. Dimension of vector $v_{i,j}$ and cardinality of support set $m_{i,j}$ are not necessarily similar in different distributions of a signature. Distance between two signatures B_i and B_j which shows their dissimilarity is computed by summation of squared distances between individual distributions

$$\widetilde{D}(B_i, B_j) = \sum_{l=1}^{d} D^2(B_{i,l}, B_{j,l}) \quad (6)$$

where d is the number of distributions a signature contains and $D(B_{i,l}, B_{j,l})$ measures dissimilarity of two distributions. As we will describe in the next section, we use mallows distance similar to [16]. The aim of D2-clustering algorithm is to find a set of prototypes $A = \{\alpha_\eta\}_{\eta=1}^{\widetilde{m}}$ for a signature set $B = \{B_i\}_{i=1}^{n}$ regarding the following optimization function

$$L(B, A^*) = \min_A \sum_{i=1}^{n} \min_{\eta=1,\ldots,\widetilde{m}} \widetilde{D}(B_i, \alpha_\eta). \quad (7)$$

This optimization function selects a prototype set A^* which minimizes the distance between each signature to its nearest neighbor prototype. Full algorithm for finding this prototype set is presented in [16].

(a) Initial frame (b) Partial occlusion (c) Full occlusion

Fig. 1. Detection of partial and full occlusion using Mallows distance

IV. D2-TRACKING

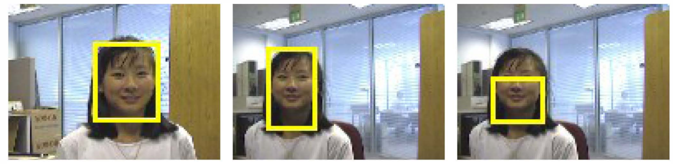
In this section, we completely illustrate our object tracking algorithm. As stated before, all templates are represented by a discrete distribution. In [16] segmentation is used to obtain regions and probabilities are extracted from each region separately. In [3], histogram of intensities is considered as a discrete distribution. In this paper, we first convert the color space of input frame to HSV. Then, H (hue) histogram of template is computed by quantizing the hue value into 32 bins. Each pixel of template lies in a bin and finally histogram is normalized. This histogram is considered as a discrete distribution which assigns a probability to each bin. The average of hues that can lie in a specific bin is taken as the value of that bin. Instead of using histograms to measure the similarity of templates, we take into account their distribution distance by Mallows distance. Using mallows distance has two advantages:

1) Each bin of histogram is compared with all the bins of the other histogram and dissimilarity between each pair of bins is taken into account

2) Each bin takes part in the total distance calculation of histograms with respect to its probability so that bins with higher probabilities play greater roles than those with lower probabilities.

In the first frame, discrete distribution of the object is taken as the model. In the next frame, discrete distributions of all target candidates are computed and the template having lowest distance to current model is selected as the object. Then, before updating of the model with new template, occurrence of occlusion is investigated. If the distance between model and template is greater than a threshold, then we can conclude that occlusion has occurred and the model is not updated. Fig. 1-a shows first frame of Face sequence. Fig. 1-b and Fig. 1-c show partial and full occlusion, respectively. In this example, the Mallows distance between template and model became 7 for partial occlusion and this distance has increased to 80 in full occlusion. In frames that no occlusion has occurred, the distance is between 0 to 5. Under the conditions that the distance is higher than 5, we are confident that partial or full occlusion has occurred and the model is not updated. If there is no occlusion in the current frame, model will be updated.

To have an adaptive model, it should be updated using object in the current frame. In addition, to avoid drifting it should use the target information in the first frame. Distribution of the model as well as object in the first frame and current frame are given as input to D2-clustering algorithm. This algorithm returns a distribution which is used



(a) Initial frame (b) Adaptive model (c) Constant model

Fig. 2. Effect of updating on results

(a) Initial frame (b) Proposed method (c) Drift problem

Fig. 3. Effect of first frame on updating

as the appearance model for finding object in the next frame. Fig. 2 represents the difference between a constant model and an adaptive model. Fig. 2-a shows the first frame form Girl sequence. Fig. 2-b shows the result of our proposed method on frame number 37 and Fig. 2-c shows the result of our method without updating of the model (offline). As depicted, by changing the appearance or size of the object, offline method fails to track while online case continues to tracking.

Fig. 3 depicts the importance of first frame in updating of the model in our proposed method. Fig. 3-a shows the first frame of Basketball sequence and Fig. 3-b shows the result of our method which uses the first frame in each update. In Fig. 3-c only the current frame is used for updating of model and the drift problem causes failure on frame number 133. In experimental results, we show that our method can handle both situations using the first frame and current frame distributions in each update. In the other word, the model is updated frame by frame by considering the fact that it doesn't forget the first frame distribution. Algorithm 1 illustrates the whole process of our tracker.

V. EXPERIMENTAL RESULTS

In this paper, we modeled the object using a signature with one distribution. As described in [16], a signature can contain more than one distribution and all its distributions can be updated using D2-clustering algorithm separately. As a result, we could use hue, intensity, edge, and other distributions simultaneously. In our implementation each hue histogram contains 32 bins.

In Algorithm 1, the $Occlusion\_Threshold$ represents the occurrence of occlusion. In our implementation, this threshold is set to 5 for all experiments. The number of particles and the variance of state variable are also the parameters that should be determined. Our experiments represents that 100 particles are enough for robust tracking. Increasing the number of particles can lead to better results with the expense of decreasing in speed of tracker. Variances of x and y showing position of the target, are set to 3 for all experiments Face

Algorithm 1 Online D2-Tracking

Inputs: Number of particles, Target location in the first frame
Initializations:
distribution[0] = hue distribution of the target in the first frame.
Set all particles state in the first frame $\{x_1^k\}_{k=1}^K$ to state of the target in the first frame
$\{\pi_1^k\}_{k=1}^K = 1/K$
Transmit() updates particle states as described in section 2.
Dist() returns Mallows distance between 2 distributions
D2-clustering() returns a distribution with respect to input distributions as described in sextion 3.

for $t = 1$ to T do
  for $k = 1$ to K do
    $x_t^k = Transmit(x_{t-1}^k)$
    $distribution[k]$ = hue distribution of particle x_t^k
    $distance[k] = Dist(model, distribution[k])$
    $\pi_t^k = \exp(-distance[k])$
  end for
  $index = \operatorname*{Argmax}_k \pi_t^k$
  If $distance[index] < occlusion\_Threshold$
    $model = D2clustering(model, distribution[0], distribution[index])$
  End if
  Output x_t^{index} as object state in i^{th} frame
end for

sequence in which it is set to 1. Although variance of scale is set to 0.1 in all experiments, maximum and minimum scale of particles height and width are restricted. In videos that target scales too much, the size of particles is allowed to change more than the videos in which the size of target remains constant. D2-clustering algorithm needs an initial distribution to start its optimization. We consider the model that we want to update as the initial distribution of D2-clustering in each update. In order to compute Mallows distance, we used the code developed by Rubner [18].

We compared our algorithm with different types of trackers. We selected the offline fragment-based tracker (FragT) [3] and the online incremental visual tracker (IVT) [8] as generative algorithms. Among online discriminative algorithms, the online boosting tracker (BoostT) [7], semi-supervised tracker (SemiT) [12] and multiple instance learning tracker (MIL) [14] were selected. The codes of authors were used to compare these methods with our method. Videos which contain challenging scenarios such as appearance variations, illumination changes, occlusion, etc. were also selected to determine the robustness of algorithms.

Table 1 illustrates the speed of our tracker in comparison with other methods. In all videos, the speed of our tracker is in the second rank after MIL tracker. The average speed of D2-tracker is 23.6 frames per second so we can consider it as a real time tracker.

Table 2 shows the quantitative evaluation of our method which depicts the average center error of different trackers. Our algorithm outperformed the other trackers in 4 videos. In 2 videos, our tracker acquired the second rank with 7 and 9 pixel errors to the best algorithms. The most important point in this table is the average error in the last row in which our algorithm has the best performance. Some algorithms perform well on one or two videos but the average error determines the most robust algorithm having an acceptable performance on all challenging videos. For example, IVT and SemiT show best results on Singer1 and Redteam respectively but the average error distances of these methods are 2 times greater than ours. In qualitative evaluation, we selected three algorithms with best performances from Table 1 and compared our tracker with them. Below, we mention detailed discussion of the video sequences.

TABLE 1. SPEED OF TRACKERS (FRAMES PER SECOND)

Video	FragT	IVT	BoostT	SemiT	MIL	D2-Tracker
Average Time	3.6	6.2	10.8	6.9	66.9	23.6

TABLE 2. AVERAGE PIXEL ERRORS OF TRACKERS

Video	FragT	IVT	BoostT	SemiT	MIL	D2-Tracker
Basketball	78	24	156	149	139	13
Trellis	41	59	96	73	65	14
Face	17	21	17	28	16	15
Girl	20	49	29	19	16	16
Redteam	43	38	24	11	38	20
Singer1	25	9	120	40	194	16
Average	37.3	33.3	73	53.3	78	15.6

Fig. 4 represents the results on Basketball sequence presented in [10] which contains challenges like partial occlusion, cluttered background, and fast motion. Our method tracks the object better than other algorithms with the average of 13 pixel errors to the ground truth. As illustrated in the second row of Fig. 4, FragT and MIL track other players because their shirt color is very similar to the target (background clutter). IVT Performs better than FragT and MIL but it fails to track the object in some frames of the sequence.

Trellis or David-outdoor video [8] is one of the most challenging videos in which object undergoes appearance variations and pose changes. Although, in general, online algorithms perform better than offline ones in videos with appearance changes, FragT which is an offline generative algorithm outperformed all online algorithms except D2-Tracker. The main reason is the drift problem in update procedure of online algorithms which makes failure. As depicted in Fig. 5, our tracker performs well in this video because of avoiding drift problem when updates the model.

Both of Face and Girl sequences contain partial and full occlusion challenges. In Face sequence [3], target remains constant and all the tracking algorithms have acceptable results as shown in Fig. 6. We conclude from the first row of the Fig. 6 that our tracker (D2-tracker) can detect the un-occluded parts of the target in partial occlusion situations. The other trackers specify some parts of the book as the object in each frame. In Girl sequence [19], in addition to challenges such as partial and full occlusion the object moves fast. In Fig. 7, frames 115 and 251 show the target after two full occlusions. In both frames, our tracker found the object immediately. Consequently it is more accurate than other

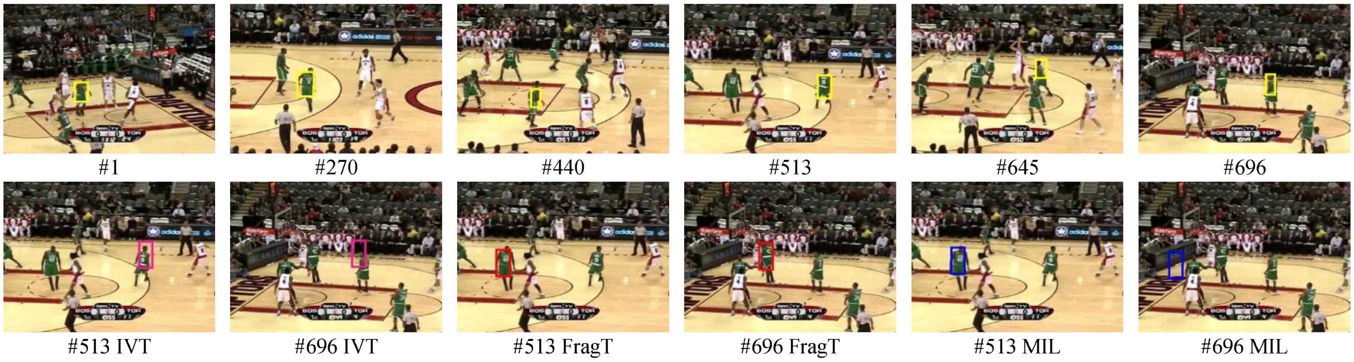


Fig. 4. The tracking results of the Basketball sequence: Our proposed method (row1) in comparison with IVT, FragT, and MIL (row2).

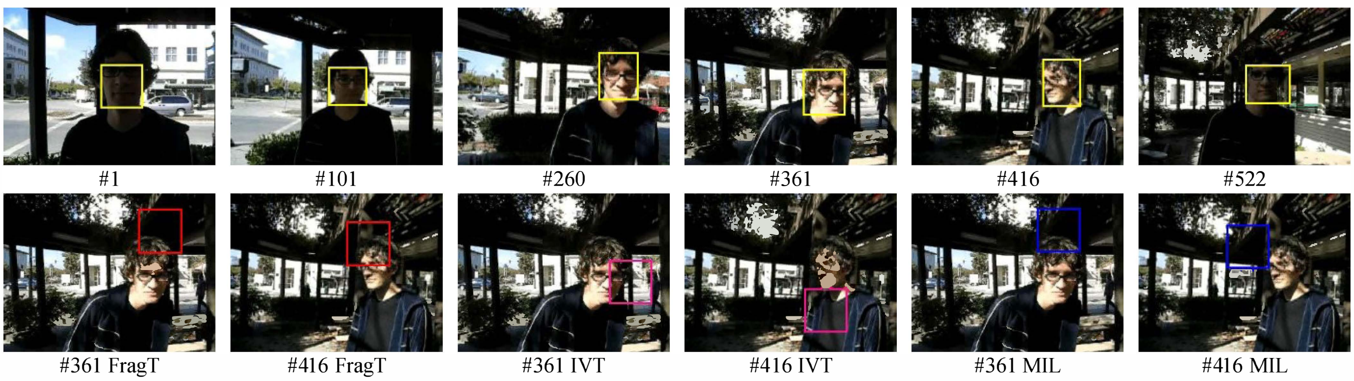


Fig. 5. The tracking results of the Trellis sequence: Our proposed method (row1) in comparison with FragT, IVT, and MIL (row2).

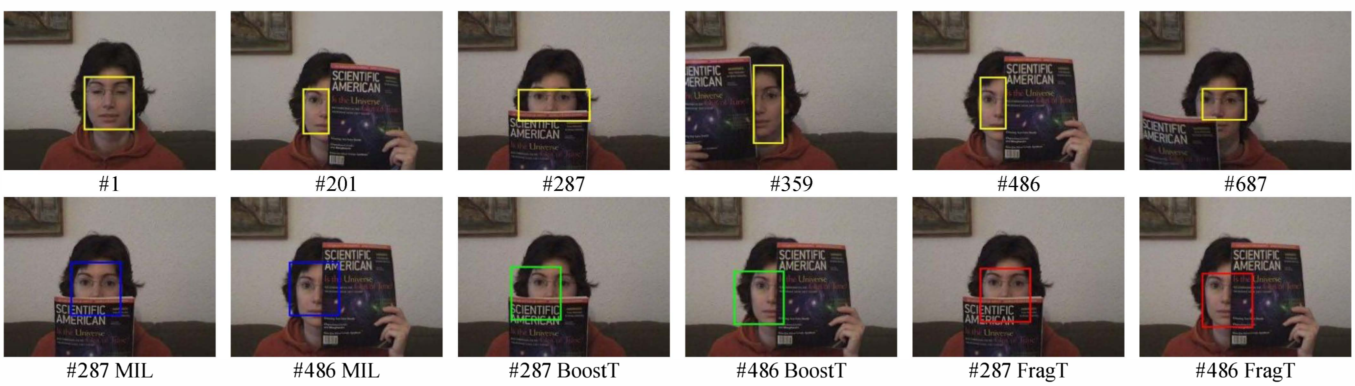


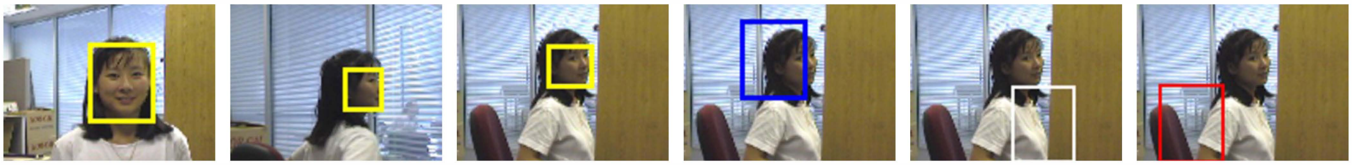
Fig. 6. The tracking results of the Face sequence: Our proposed method (row1) in comparison with MIL, BoostT, and FragT (row2).

methods.

The main challenge in Redteam and Singer1 sequences is the variation of object size. In Redteam [20] the goal is to track a car which its size changes through the sequence. We ignored the first 399 frames and considered frame 400 as the first frame of the sequence. Fig. 8 shows our tracker results in comparison with other methods. In our method particles use scale parameter in their state. Therefore, tracker can find the best size of the object in each frame. Singer1 sequence [10] is another example showing that our algorithm is robust with respect to object size changes. Fig. 9 represents the tracker results on this sequence which contains illumination variations and object size changes. Results show the robustness of our tracker in handling these challenges.

VI. Conclusion

In this paper, we presented an online generative tracking algorithm called D2-tracker. The appearance model is template color distribution which is updated in each frame. Our update procedure is based on D2-clustering algorithm. In updating, we avoid drifting by using the target distribution in the first frame and keep the algorithm adaptive by using the target distribution in the last processed frame. These two distributions as well as the current model are the inputs to D2-clustering algorithm. As a result, the output will be the updated model. Using real time D2 tracker can significantly improve tracking an object under the conditions involving challenging scenarios such as illumination variations,



#1 #115 #251 #251 MIL #251 SemiT #251 FragT

(a) Our proposed method (b) Comparison with other methods

Fig. 7. The tracking results of the Girl sequence: Our proposed method in comparison with MIL, Semi, and FragT.

#400 #1088 #1684 #1088 SemiT #1088 BoostT #1088 MIL

(a) Our proposed method (b) Comparison with other methods

Fig. 8. The tracking results of the Redteam sequence: Our proposed method in comparison with SemiT, BoostT, and MIL.

#1 #145 #250 #145 IVT #145 FragT #145 SemiT

(a) Our proposed method (b) Comparison with other methods

Fig. 9. The tracking results of the Singer1 sequence: Our proposed method in comparison with IVT,FragT, and SemiT .

appearance changes, partial and full occlusion etc. in comparison with the existing algorithms.

REFERENCES

[1] J Yang and A Waibel, "A Real-Time Face Tracker," in *Proc. IEEE Workshop Applications of Computer Vision*, 1996, pp. 142-147.

[2] D Comaniciu, V Ramesh, and P Meer, "Kernel-Based Object Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-577, 2003.

[3] A Adam, E Rivlin, and I Shimshoni, "Robust Fragments-based Tracking using the Integral Histogram," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2006, pp. 798-805.

[4] S Avidan, "Support Vector Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064 - 1072, 2004.

[5] R Collins, "Online Selection of Discriminative Tracking Features," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631 - 1643, 2005.

[6] S Avidan, "Ensemble Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261 - 271, 2007.

[7] H Grabner, M Grabner, and H Bischof, "Real-Time Tracking via Online Boosting," in *Conf. British Machine Vision*, 2006, pp. 47-56.

[8] D Ross, J Lim, R.-S Lin, and M.-H Yang, "Incremental Learning for Robust Visual Tracking," *Int'l J. Computer Vision*, vol. 77, no. 1, pp. 125-141, 2008.

[9] M Xue and L Haibin, "Robust Visual Tracking using L1 Minimization," in *12th Int'll IEEE Conf. Computer Vision*, 2009, pp. 1436-1443.

[10] J Kwon and K. M Lee, "Visual Tracking Decomposition," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 1269-1276.

[11] L Matthews, T Ishikawa, and S Baker, "The Template Update Problem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810-815, 2004.

[12] H Grabner, C Leistner, and H Bischof, "Semi-supervised Online Boosting for Robust Tracking," in *Proc. European Conf. Computer Vision(ECCV)*, 2008.

[13] S Stalder, H Grabner, and L van Gool, "Beyond Semi-Supervised Tracking: Tracking Should Be as Simple as Detection, but not Simpler Than Recognition," in *Proc. Int'l IEEE Conference. Computer Vision Workshop*, 2009.

[14] B Babenko, M. H Yang, and S Belongie, "Robust Object Tracking with Online Multiple Instance Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619 - 1632, 2011.

[15] E Levina and P Bickel, "The Earth Mover's Distance Is the Mallows Distance: Some Insights from Statistics," in *Proc. Eighth Int'l IEEE Conf. Computer Vision*, 2001, pp. 251-256.

[16] Wang. J.Z and J.Z Wang, "Real-Time Computerized Annotation of Pictures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 985 - 1002, 2008.

[17] A Doucet, N de Freitas, and N Gordon, *Sequential Monte Carlo Methods in Practice.* New York: Springer-Verlag, 2001.

[18] Y Rubner, "Code available at: http://ai.stanford.edu/~rubner/emd/,".

[19] S Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998, pp. 232-237.

[20] R Collins, X Zhou, and S.K Teh, "An Open Source Tracking Testbed and Evaluation Web Site," in *IEEE workshop. Performance evaluation and surveillance (PETS)*, 2005.