Channel-Graph Regularized Correlation Filter for Visual Object Tracking

BY

ARJUN TYAGI

MTech in Computer Science and Engineering

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Supervisor:
Dr A.V. Subramanyam
Assistant Professor IIIT-Delhi

A thesis submitted in partial fulfilment of
the requirements for the degree of
MTech

Department of computer science and Engineering
Indraprastha Institute of Information Technology
Delhi, India

August, 2020

Certificate

This is to certify that the Thesis titled "**Channel-Graph Regularization Correlation Filter for Visual Object Tracking**" being submitted by Arjun Tyagi (MT18023) to the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree. The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree.

June, 2019

Dr A.V. Subramanyam
Department of Electronics and Computer Science
Indraprastha Institute of Information Technology Delhi
New Delhi 110020

Acknowledgements

I want to express my deepest gratitude to my advisor Dr A.V. Subramanyam, for his guidance and support. This thesis would not have been possible without his continued support, patience, valuable suggestions and advice. One could not wish for a better and friendlier supervisor. I would like to specially thank Monika Jain for helping me whenever needed. Last but not least, I would like to thanks my friends and family members for their constant support.

Abstract

Correlation filter (CF) based tracker often disregard or weakly incorporate the importance of feature channels as well as channel similarity. To address this, we propose a channel-graph regularization correlation filter-based visual object tracker (CGRCF). In our work, we study two-channel regularization methods. First is the channel regularization that determines the vital feature channels. Second is the graph-regularization that increases the probability of assigning similar weights based on the properties of feature channels. The proposed tracker can be efficiently solved in the Fourier domain using ADMM (Alternate Direction Method of Multiplier) and achieves a real-time tracking speed of 28FPS. We conduct extensive experimentation on the TC128, VOT2017 and VOT2019 datasets. The proposed tracker demonstrates promising results and performs better than several state of the art CF trackers as well as end-to-end deep learning trackers.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Object tracking is an essential and complex problem in the field of computer vision. It has a wide range of application such as autonomous driving, surveillance and robotics [1–3]. Object tracking aims to determine the position of the object in subsequent frames, given the initial position of the object. The object tracking can broadly be classified into end-to-end deep learning trackers [4–6] and Correlation filters (CF) based trackers [7–14]. The deep learning trackers use features from the deep networks that significantly improve their representation power. Deep learning trackers can be further divided into two groups. The first group contains trackers [15–17] that do an online update by updating the model at every frame. Online training will give a robust representation of the object but on the other hand, requires more computation time. The second group contain the trackers [4, 5, 18] that leverage the use of extensive offline training to learn the representation of the objects. The major problem with the deep learning trackers is that they have a very low tracking speed because of the high computational cost involved in training. The trackers in the second group of deep learning resolve this problem by using offline training for the network, but this approach cannot adapt to the change in target appearance due to lack of online training.

Correlation Filters (CF) based tracker learn the appearance of the object using the filters trained on the object sample images. In recent years, the performance of the CF-based trackers has increased drastically [9, 14, 19]. The most significant advantage of the CF trackers is that they can be learned efficiently in the frequency domain via Fast Fourier transform (FFT) [20]. The filters and the images are converted to the Fourier domain and they are used to solve the objective function, and the output is again mapped back to the spatial domain using the Inverse Fast Fourier Transform (IFFT) [21]. This transition from FFT to IFFT can be

done in $O(n \log n)$ by using the divide and conquer algorithm [22] where n is the size of the image in terms of the total number of features. CF trackers can use deep features due to efficient and faster learning capability that also helps in learning more robust filters which can give robustness to the visual tracking challenges like Illumination Variation, Scale Variation, Occlusion, Deformation, Motion Blur, Fast Motion, In-Plane Rotation, Out of view rotation and Background Clutters. CF-trackers use deep learning features [23, 24] and handcrafted features [25–27] jointly to learn filters, but all of these features may not be useful. Learning the model using all of the features may result in a noisy model and tracker drift. Based on this observation, several spatial and temporal regularization have been investigated [9, 10]. These models help in selecting the features but do not take into consideration the relationship between the different feature channels that results in assigning different weights to the similar feature channels.

In this work, we propose a graph-channel regularization technique to handle the problem of determining important feature maps and the unequal weight assignment by using a combination of regularizations. We propose channel-regularization which helps in selecting the essential feature channels and the graph regularization which helps in assigning the similar weights to the similar feature channels. Some channels are sensitive to foreground or background while some channels are sensitive to noise. Channel regularization selects features to suppress the channel-wise redundancy or noise. Graph regularization represents the n-dimensional feature channels in the form of a graph where each channel is represented as a graph node. The edge weights are assigned by using the distance between the feature channels. This graph is represented in the form of a matrix by using the Laplacian operator [28] and further used to compute the weights for each feature channels. By using the channel-graph regularization, we propose a correlation filter based tracker (CGRCF). We also show how channel-graph regularization can be used to significantly improve the performance of existing trackers. The proposed and the modified trackers show promising results on popular datasets like Temple colour [29] and Visual Object Tracking 2017 [30] and 2019 [31] datasets . These datasets contain a variety of challenges that can help in better understanding the tracker's performance [32].

1.0.1 Research Contribution

The following are the contributions of this thesis:

- We propose a correlation filter based visual object tracker by using the channel and graph regularization.
- To show the effect of channel and graph regularization on the trackers, we reformulate state of art trackers BACF [8] and GFSDCF [9].
- A thorough analysis of the Proposed tracker along with the reformulated tracker is done on the Visual Object Tracking and Temple colour datasets [29–31].

CHAPTER 2

Visual Object Tracking Background

2.1 Literature Review

Discriminative correlation filter (DCF) based approaches have been successfully applied to the field of object tracking [29, 33, 34]. In object tracking, there is always a trade-off between robust tracker performance and speed of tracker. A tracker with frame process rate of more than 30 frames per second is known as a real-time tracker. Some of the trackers offer real-time speed [14, 19] but give less promising results whereas, others provide decent tracking accuracy, but they are not real-time [13, 35]. The object trackers are broadly categorized into two categories: the end to end deep-learning-based trackers and the correlation filter-based trackers. They both have their advantages and disadvantages as the end to end tracker have more robust features, but they are extremely slow whereas, CF trackers are faster but do not include online training of features. In our work, we use correlation filter trackers because of their enormously flexible behaviour and robustness.

In [14], Dai et al. proposed a Minimum Output Sum of Squared Error (MOSSE) tracker that can track up to 700 FPS by using only greyscale samples to train the filters. The tracker detects occlusion based upon the peak-to-sidelobe ratio, the tracker pause and resumes where it left off when the object reappears. In [19], Henriques et al. proposed KCF tracker that shows promising performance with a high tracking rate of more than 150FPS. They trained their model with translated and scaled sample patches. Instead of training tracking model on the raw pixel, KCF uses the multi-channel HOG feature. The authors make a circulant matrix using these features and diagonalize it using the Discrete Fourier Transform to make it space and computationally efficient.

MOSSE [14] and KCF [19] use features from the gray scale images. Similarly [36] introduce the color naming features to achieve robust tracking in color videos. In [37], Tang et.al. introduce multi-kernel into learning KCF. The authors also reformulate the multi-kernel version of CF objective function with its upper bound, alleviating the negative mutual interference of complementary features. In [38], the authors proposed a region of interest-based pooling method. The pooling operation is used on the cropped ROI regions. It helps in compressing the model by preserving the localization. Then using these ROI pooled samples, they learn the correlation filters for tracking. This way, authors were able to use more features but the tracking speed is as low as 5FPS. Similar to using CF technique, several works also explore the concept of Reinforcement Learning in object tracking. [39–41] proposed trackers that use reinforcement and deep reinforcement learning. [39] proposed a neural network-based tracking model that comprises of a CNN(Convolutional Neural Network) for best features, an RNN(Recurrent Neural Network) to construct a video memory state and an RL agent that helps in making decisions about the target localization state.

One of the other approaches in object tracking is based on end-to-end deep learning. In the past decade, deep learning has become very popular. Recently, [4, 42, 43] proposed trackers that are based on end-to-end tracking framework. These trackers fine-tune the features after a specific interval of frames that results in a more robust tracker. In [44], Li et al. proposed a gradient-guided network which uses the information from the gradient to update the template in the current frame. Template generalization method is applied during the offline training, which helps in avoiding over-fitting. In [45], Zhang et al. proposed a deep learning tracker called UpdateNet, which overcomes the shortcomings of the conventional linear update rule and learn the updating step as an optimization problem.

Our work focuses on CF based tracker, where we proposed a novel channel-graph regularization based tracker and reformulate three trackers to show how we can use graph and channel regularization to improve the performance. Below is the literature of the tracker used in the proposed work.

2.1.1 BACF

A background aware CF tracker that was proposed in [8], separates out the background from the foreground by using a representation matrix P. The objective function for BACF is,

$$E(H) = \left\| y - \sum_{k=1}^{c} x_k * (P^T h_k) \right\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^{c} \|h_k\|_2^2 \tag{2.1}$$

Where, $x_k \in R^T$ and $h_k \in R^D$ are the k-th channel of vectorized image and filters. $y \in R^T$ is desired response of correlation filters. $P \in R^{D \times T}$ binary matrix which crops the D elements of vectorized image.

2.1.2 GFSDCF

Group Feature Selection method for Discriminative Correlation Filters (GFSDCF) [9] uses both, spatial and channel regularization to select the features and is formulated as follows:

$$E(H) = \left\| y - \sum_{k=1}^{c} X^k * W^k \right\|_2^2 + \lambda_1 \sum_{k=1}^{c} \left\| \tilde{W}_t^k \right\|_F + \lambda_2 \sum_{i=1}^{N} \sum_{j=1}^{N} \|w_{ijt}\|_2 + \lambda_3 \sum_{k=1}^{c} \left\| W_t^k - W_{t-1}^k \right\|_F^2 \tag{2.2}$$

Where, t is the response at t-th frame. $X_t^k \in R^D$ is k-th channel of vectorize image and $W_t^k \in R^D$ is the k-th filter channel. λ_1, λ_2 and λ_3 are the regularization terms. W and $\tilde{W}_t \in R^{D \times C}$ are C channels feature vector and spatial-regularization . w_{ij_t} is channel-regularization term.

CHAPTER 3

Proposed Method

This chapter contains a detailed formulation of the novel channel and graph regularization based CF tracker along with the modified formulation of BACF [8] and GFSDCF [9] using the channel and graph regularization. The formulation contains the objective function, scale estimation, Lagrangian multiplier update and the model update.

3.1 BACF-Channel Regularized

A fundamental drawback of correlation filter based tracker is that the target background model is not modeled over time. Background aware correlation filter (BACF) [8] uses the negative samples around the target to learn filters that are robust and more generalized. BACF efficiently model the variance in the foreground and background over time.

3.1.1 Objective function

The objective function for Channel regularized BACF tracker is defined as follows:

$$E(h,q) = \frac{1}{2}\left\| y - \sum_{k=1}^{C} q_k (x_k * (P^T h_k)) \right\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{C} \|h_k\|_2^2 + \frac{\beta}{2}\sum_{k=1}^{C} \|q_k\|_2^2 \qquad (3.1)$$

Where, C is the total number of channels and, λ and β are the regularization parameters. $q_k \in R^{1\times 1}$ is a scalar regularization value of k-th feature channel. The eq (3.1) can be efficiently solved using Parseval's theorem [7] that is used to represent the energy equivalent in Fourier domain to time domain.

$$E(\hat{G}, H, q) = \frac{1}{2}\left\|\hat{y} - \sum_{k=1}^{C} \hat{x}_k \circledast \hat{g}_k\right\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{C}\|h_k\|_2^2 + \frac{\beta}{2}\sum_{k=1}^{C}\|q_k\|_2^2 \tag{3.2}$$

$$\text{s.t., } \hat{g}_k = \sqrt{T}FP^T h_k q_k, \text{ and } \hat{x}_k = \sqrt{T}Fx_k,\ k = 1, 2, 3, .., C$$

Where, $F \in R^{T\times T}$ is a Fourier matrix which transforms a T dimensional signal to Fourier domain. The symbol $\hat{}$ denotes that the signal is in the Discrete Fourier Domain. $H = [h_1, h_2, ..., h_C]$ contain learned filters for all C channels and $\hat{G} = [\hat{g}_1, \hat{g}_2, ..., \hat{g}_C]$ is an auxiliary variable matrix that helps to obtain the decomposition of $E(H, q)$ which can be efficiently solved using ADMM [46]. Variables $\hat{G}, H$ and q can also be solved using ADMM. The equivalent Lagrangian form of eq (3.2) can be written as:

$$E(\hat{G}, H, q, \hat{S}) = \frac{1}{2}\left\|\hat{y} - \sum_{k=1}^{C} \hat{x}_k \circledast \hat{g}_k\right\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{C}\|h_k\|_2^2 + \frac{\mu}{2}\sum_{k=1}^{C}\left\|\hat{g}_k - \sqrt{T}FP^T h_k q_k + \frac{\hat{s}_k}{\mu}\right\| + \frac{\beta}{2}\sum_{k=1}^{C}\|q_k\|_2^2 \tag{3.3}$$

Where, λ, β and μ are the regularization parameters. The Fourier transform of the Lagrange multiplier is represented as $\hat{S} = [\hat{s}_1, \hat{s}_2, ..., \hat{s}_C] \in R^{T\times C}$. Variables $\hat{G}, H$ and q can be solved individually using ADMM as follows:

Solving for H: Fixing the variable $q, \hat{G}$ and $\hat{S}$ the optimal solution for H^* can be obtained by,

$$h_k^* = \underset{\mathbf{h}_k}{\operatorname{argmin}} \frac{\lambda}{2}\sum_{k=1}^{C}\|h_k\|_2^2 + \frac{\mu}{2}\sum_{k=1}^{C}\left\|\hat{g}_k - \sqrt{T}FP^T h_k q_k + \frac{\hat{s}_k}{\mu}\right\| \tag{3.4}$$

Differentiating equation (3.4) w.r.t h_k we get,

$$\lambda h_k - \mu\sqrt{T}PF^T q_k(\hat{g}_k - \sqrt{T}FP^T h_k q_k + \frac{\hat{s}_k}{\mu}) = 0 \tag{3.5}$$

$$\lambda h_k - \mu\sqrt{T}Pq_k F^T \hat{g}_k + \mu TPP^T q_k^2 h_k - \sqrt{T}PF^T \hat{s}_k q_k = 0 \tag{3.6}$$

$$\lambda h_k - \mu TPq_k g_k + \mu TPP^T q_k^2 h_k - Tq_k P\hat{s}_k = 0 \tag{3.7}$$

$$(\lambda I + \mu TPP^T q_k^2)h_k = Tq_k P(\mu\hat{g}_k + s_k) \tag{3.8}$$

$$h_k^* = \frac{Tq_kP(\mu\hat{g}_k + \hat{s}_k)}{\lambda I + \mu TPP^T q_k^2} \tag{3.9}$$

where, h_k^* is the optimal value of the k-th filter and I is a $T \times T$ identity matrix. $\hat{}$ denotes that the variable is in Discrete Fourier domain. P denotes the projection matrix that helps in separating foreground from background.

Solving for $\hat{G}^*$: The optimal solution for $\hat{G}$ can be obtained by fixing H, q and $\hat{S}$ in equation (3.3)

$$\hat{G}^* = \underset{\hat{\mathbf{G}}}{\operatorname{argmin}} \frac{1}{2} \left\| \hat{y} - \sum_{k=1}^{C} \hat{x}_k \circledast \hat{g}_k \right\|_2^2 + \frac{\mu}{2} \sum_{k=1}^{C} \left\| \hat{g}_k - \sqrt{T} F P^T h_k q_k + \frac{\hat{s}_k}{\mu} \right\| \tag{3.10}$$

The equation (3.10) has a high computational complexity. To solve it faster we can change to process pixel-wise and modify the equation as follows:

$$V_j^*(\hat{G}) = \underset{\mathbf{V_j}(\hat{\mathbf{G}})}{\operatorname{argmin}} \frac{1}{2} \left\| \hat{y}_j - V_j(\hat{X}^T) V_j(\hat{G}) \right\|_2^2 + \frac{\mu}{2} \sum_{k=1}^{C} \left\| V_j(\hat{G}) + V_j(\hat{M}) \right\|_2^2 \tag{3.11}$$

where, $V_j(\hat{X}) = [\hat{x}_{1j}, \hat{x}_{2j}, ..., \hat{x}_{Cj}]^T$ and $V_j(\hat{G}) = [\hat{g}_{1j}, \hat{g}_{2j}, ..., \hat{g}_{Cj}]^T$ are vector with C elements. $V_j(\hat{M})$ = $V_j(\frac{\hat{S}}{\mu}) - V_j(\sqrt{TFP^THq})$ and $V_j(\frac{\hat{S}}{\mu}) = [\frac{\hat{s}_{1j}}{\mu}, \frac{\hat{s}_{2j}}{\mu}, ..., \frac{\hat{s}_{Cj}}{\mu}]$. Solving equation (3.11) for G we get,

$$[\mu I + V_j(\hat{X}) V_j(\hat{X})^T] V_j(\hat{G}) = \hat{y}_j \hat{V}_j)(\hat{(}X)) - \mu V_j(\hat{M})$$

$$V_j^*(\hat{G}) = (\mu I + V_j(\hat{X}) V_j(\hat{X})^T)^{-1} (\hat{y}_j V_j(\hat{X}) - \mu V_j \left(\frac{\hat{S}}{\mu} \right) + \mu V_j(\sqrt{T} F P^T H q)) \tag{3.12}$$

The equation can be simplified by using the Sherman-Morrison formula [8] and change the equation (3.12) as follows,

$$V_j^*(\hat{G}) = \frac{1}{\mu} \left(I - \frac{V_j(\hat{X}) V_j(\hat{X})^T}{\mu + V_j(\hat{X}^T) V_j(\hat{X})} \right) (\hat{y}_j V_j(\hat{X}) - \mu V_j \left(\frac{\hat{S}}{\mu} \right) + \mu V_j(\sqrt{T} F P^T H q)) \tag{3.13}$$

Where, $V_j^*(\hat{G})$ denote equivalent value of the optimal value of G.

Solving for q: To solve for q we need to fix $\hat{G}, H$ and $\hat{S}$, Optimal q can be obtained as follows:

$$q_k^* = \underset{\mathbf{q_k}}{\operatorname{argmin}} \frac{\mu}{2} \sum_{k=1}^{C} \left\| \hat{g}_k - \sqrt{T} F P^T h_k q_k + \frac{\hat{s}_k}{\mu} \right\| + \frac{\beta}{2} \sum_{k=1}^{C} \|q_k\|_2^2 \tag{3.14}$$

Differentiating w.r.t to q we get,

$$q_k^* = \frac{\mu\sqrt{T} h_k^T P g_k + T h_k^T P s_k}{\mu\sqrt{T} h_k^T P P^T h_k + \beta} \tag{3.15}$$

Where, q_k^* denotes the optimal singular value of penalty on the k-th feature channel.

3.2 GFSDCF-Channel Regularized

GFSDCF-CR incorporates group feature selection using the channel and graph regularization and a low rank regularization to achieve temporal smoothness of the learned filters during tracking. GFSDCF original formulation use l-2 regularization on the feature channel. So, to show how we can use graph regularization, we reformulate the formulation as follows:

3.2.1 Objective function

The objective function for the GFSDCF with the channel-graph regularization is defined as,

$$\tilde{W} = \underset{\mathbf{W}}{\operatorname{argmin}} \left\| \sum_{k=1}^{C} q_k (W^k * X^k) - Y \right\|_2^2 + \lambda_1 R_s(W) + \lambda_2 R_c(W) + \lambda_3 R_T(W) + \lambda_4 R_q(W) \tag{3.16}$$

Where, $Y \in R^{N \times N}$ Gaussian shaped expected label. $W^k \in R^{N \times N}$ and $X^k \in R^{N \times N}$ are the k-th feature channel and vectorized signal. $R_s(W)$ is the spatial regularization term, $R_c(W)$ is the group regularization term for channel selection, $R_T(W)$ is the temporal regularization term and $R_q(W)$ is the graph regularization term. $W \in R^{N \times N \times C}$ is the multi channel feature tensor.

3.2.1.1 Channel Regularization

The regularization term for spatial group feature selection can be defined as,

$$R_s(W) = \sum_{i=1}^{N} \sum_{j=1}^{N} \|w_{ij}\|_2^2 \tag{3.17}$$

Where, $w_{ij} = [w_{ij1}, w_{ij2}, ..., w_{ijC}]$, is the channel selection term and define as,

$$R_c(W) = \sum_{k=1}^{C} \left\| W^k \right\|_F \tag{3.18}$$

Where, $W^k \in R^{N \times N}$ is the k-th feature channel.

3.2.1.2 Temporal smoothness

The temporal smoothness is obtained to improve the robustness of the correlation filters. To promote the temporal coherence in the filters, a low rank constraint is imposed as,

$$R_T(W) = \sum_{k=1}^{C} \left\| W_t^k - W_{t-1}^k \right\|_F^2 \tag{3.19}$$

Where, W_t and W_{t-1} are the feature tensor at t and $t-1$ frame.

3.2.1.3 Graph regularization

Channel regularization don't take into consideration the relation between the features. So, to ensure that the features sharing the same property will have the same penalty, we introduce graph regularization[47]. Let us consider N-dimensional data points as $W \in R^N$, These points are used to construct a nearest neighbour graph G with N vertices, where each vertex represent a data point. The adjacency weight matrix of the graph G is computed as,

$$Z_{i,j} = \begin{cases} dist(W_i, W_j), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{3.20}$$

Where, $dist(W_i, W_j)$ is use to compute the distance between the two data points. In our formulation we are using heat kernel weighing [48, 49], The equation (3.23) become,

$$Z_{i,j} = \begin{cases} \exp^{-\frac{\left\| W_i - W_j \right\|_2^2}{\sigma_i \sigma_j}}, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{3.21}$$

Where, σ_i and σ_j are the decay factors on the weights. We use the Laplacian operator to represent the graph in the matrix form as,

$$L = D - Z \tag{3.22}$$

Where, Z is the adjacency weight matrix of size $N \times N$ and D is the degree matrix which is computed as $D = diag(d_1, d_2, ..., d_C)$. where degree of i^{th} element is $d_i = \sum_{j=1}^{C} Z_{ij}$.
To obtain a representation for regularization from the graph G, we need to minimize the

following objective function:

$$\frac{1}{2}\sum_{i=1}^{C}\sum_{j=1}^{C}(W_i - W_j)^2 Z_{ij} = Tr(WLW^T) \tag{3.23}$$

The Laplacian regularizer can be factorized as,

$$\begin{aligned} R_q(W) &= Tr(WLW^T) \\ Tr(WLW^T) &= \sum_{k=1}^{C} Lq_k q_k^T \end{aligned} \tag{3.24}$$

To solve the objective function, we use augmented Lagrangian [50] and introduce a slack variable $W^{'} = W$. The optimization function changes to:

$$\begin{aligned} L = \left\| \sum_{k=1}^{C} q_k (W_t^k \circledast X_t^k) - Y \right\| + \lambda_1 \sum_{k=1}^{C} \left\| W_t^{'k} \right\|_F + \lambda_2 \sum_{i=1}^{N}\sum_{j=1}^{N} \left\| w_{ij_t}^{'} \right\| + \lambda_3 \sum_{k=1}^{C} \left\| W_t^k - W_{t-1}^k \right\|_F^2 \\ + \lambda_4 \sum_{k=1}^{C} Lq_k q_k^T + q_k^T 2\lambda_4 \sum_{j\neq k} L_{kj} q_j + + \frac{\mu}{2} \sum_{k=1}^{C} \left\| W_t^k - W_t^{'k} + \frac{\Gamma}{\mu} \right\|_F^2 \end{aligned} \tag{3.25}$$

Where, Γ^k is the Lagrangian multiplier for k-th channel and μ is the penalty. Using the ADMM, the optimal solution can be obtained as,

$$\hat{w}_{ij_t} = \left(I - \frac{\hat{x}_{ij_t}\hat{x}_{ij_t}^H}{\left(\lambda_3 + \frac{\mu}{2}\right) N^2 + \hat{x}_{ij_t}^H \hat{x}_{ij_t}} \right) \left(\frac{\hat{x}_{ij_t}\hat{y}_{ij}}{N^2} + \mu \hat{w}_{ij_t}^{'} - \mu \Gamma_{ij}^{'} + \lambda_3 q_k \hat{x}_{ij_{t-1}} \right) \left(\frac{1}{\lambda_3 + \mu} \right) \tag{3.26}$$

The optimal value can be obtained by using the equation (3.31),

$$\hat{x}_{ij_t}^{'} = max \left(0, 1 - \frac{\lambda_1}{\mu \left\| \hat{P}^k \right\|_F} - \frac{\lambda_2}{\mu \left\| \hat{p}_{ij} \right\|_2^2} \right) \hat{P}_{ij}^k \tag{3.27}$$

Where, λ_1 and λ_2 are the regularization parameters and μ is the penalty term. The term $\hat{P}_{ij}^k$, which is used in equation(3.33), is defined as $\hat{P}_{ij}^k = q_k \hat{w}_{ij}^k + \frac{\Gamma_{ij}^k}{\mu}$.

$$q_d = \frac{\hat{W}_d^T \hat{X}_d^T \hat{Y} - 2\lambda_4 \sum_{d\neq k} L_{kd} q_d}{\hat{W}_d^T \hat{X}_d^T \hat{X}_d \hat{W}_d + \lambda_4 + 2\lambda_4 L_{kk}} \tag{3.28}$$

3.3 Proposed - Channel-Graph Regularized CF tracker

In the previous section, we formulated the channel regularization version of BACF [8]. We also showed how we can apply graph regularization to the existing trackers like GFSDCF [9]. The results in section (4.0.1) show significant improvement in the performance of CF based tracker. The channel regularization helps us reduce the actual number of features used for tracking. But, it does not take into consideration the relation between the features. To solve this problem, we proposed a novel tracker with channel and graph regularization, that not only selects the important feature channels but, also gives the same weight to channels with the same properties.

3.3.1 Objective function

The objective function for the channel-graph regularized correlation filter is defined as,

$$E(h,q) = \frac{1}{2}\left\| y - \sum_{k=1}^{C} q_k (x_k * (P^T h_k)) \right\|_2^2 + \alpha Tr(HLH^T) + \beta R(q) \tag{3.29}$$

where, α and β are the regularization parameters. $Tr(HLH^T)$ can also be represented as $\sum_{i,j=1}^{C} L_{ij} h_j^T h_i$ and similarly the Laplacian regularizer $R(q)$ can be obtained as $\sum_{i,j=1}^{C} L_{ij} q_j q_i^T$. Each filter channel is updated individually by keeping the other channels fixed. So, equation (3.35) can be simplified as,

$$\begin{aligned} E(h,q) = \frac{1}{2}\left\| y - \sum_{k=1}^{C} q_k (x_k * (P^T h_k)) \right\|_2^2 + \alpha L_{kk} h_k^T h_k + 2\alpha h_k^T \sum_{j\neq k}^{C} L_{kj} h_j + \\ \beta L_{kk} q_k^2 + 2\beta q_k \sum_{j\neq k}^{C} L_{kj} q_j \end{aligned} \tag{3.30}$$

To represent (3.36) in the frequency domain, we use Parseval's theorem and modify the equation as,

$$E(h,q) = \frac{1}{2}\left\| y - \sum_{k=1}^{C} \hat{x}_k \circledast \hat{g}_k \right\|_2^2 + \alpha L_{kk} h_k^T h_k + 2\alpha h_k^T \sum_{j\neq k}^{C} L_{kj} h_j + \beta L_{kk} q_k^2 + 2\beta q_k \sum_{j\neq k}^{C} L_{kj} q_j \tag{3.31}$$

$$\text{s.t. } \hat{g}_k = \sqrt{T}FP^T h_k q_k \text{ and } \hat{x}_k = \sqrt{T}F\hat{x}_k,\ k = 1, 2, ..., C$$

The Lagrangian form of the equation (3.37) can be written as

$$E(h,q) = \frac{1}{2}\left\| y - \sum_{k=1}^{C} \hat{x}_k \circledast \hat{g}_k \right\|_2^2 + \frac{\mu}{2}\sum_{k=1}^{C}\left\| \hat{g}_k - \sqrt{T}FP^T h_k q_k + \frac{\hat{s}_k}{\mu}\right\| + \alpha L_{kk} h_k^T h_k + 2\alpha h_k^T \sum_{j\neq k}^{C} L_{kj} h_j + \beta L_{kk} q_k^2 + 2\beta q_k \sum_{j\neq k}^{C} L_{kj} q_j \tag{3.32}$$

where $\hat{S} = [\hat{s}_1, \hat{s}_2, ..., \hat{s}_C,]$ are the Lagrange multipliers. The optimal solution for the equation (3.38) can be obtained by breaking it into sub-problems using ADMM.

Solving for H: The optimal solution for H^* can be obtained by fixing $\hat{G}$,q and $\hat{S}$,

$$h_k^* = \underset{\mathbf{h_k}}{\operatorname{argmin}} \frac{\mu}{2}\sum_{k=1}^{C}\left\| \hat{g}_k - \sqrt{T}FP^T h_k q_k + \frac{\hat{s}_k}{\mu}\right\| + \alpha L_{kk} h_k^T h_k + 2\alpha h_k^T \sum_{j\neq k}^{C} L_{kj} h_j \tag{3.33}$$

By differentiating the equation (3.39) w.r.t h_k we get,

$$h_k^* = \frac{Tq_k(\mu\hat{g}_k + \hat{s}_k) - 2\alpha \sum_{j\neq k}^{C} L_{kj} h_j}{\mu TPP^T q_k^2 + 2\alpha L_{kk}} \tag{3.34}$$

Where, h_k denotes the optimal value of filters for the k-th channel, L_{kj} denotes the value of Laplacian matrix at (k, j)-th index and α is the regularization parameter.

Solving for $\hat{G}^*$: The optimal solution for $\hat{G}$ can be obtained by fixing H, q and $\hat{S}$ in equation (3.38)

$$\hat{G}^* = argmin_{\hat{G}} \frac{1}{2}\left\| \hat{y} - \sum_{k=1}^{C} \hat{x}_k \circledast \hat{g}_k \right\|_2^2 + \frac{\mu}{2}\sum_{k=1}^{C}\left\| \hat{g}_k - \sqrt{T}FP^T h_k q_k + \frac{\hat{s}_k}{\mu}\right\| \tag{3.35}$$

The equation (3.41) has a high computational complexity. To solve it faster, we can change to process pixel-wise and modify equation as follows:

$$V_j^*(\hat{G}) = \underset{\mathbf{V_j(\hat{G})}}{\operatorname{argmin}} \frac{1}{2}\left\| \hat{y}_j - V_j(\hat{X}^T) V_j(\hat{G}) \right\|_2^2 + \frac{\mu}{2}\sum_{k=1}^{C}\left\| V_j(\hat{G}) + V_j(\hat{M}) \right\|_2^2 \tag{3.36}$$

where, $V_j(\hat{X}) = [\hat{x}_{1j}, \hat{x}_{2j}, ..., \hat{x}_{Cj}]^T$ and $V_j(\hat{G}) = [\hat{g}_{1j}, \hat{g}_{2j}, ..., \hat{g}_{Cj}]^T$ are one dimensional vector with C elements and $V_j(\hat{M})$ = $V_j(\frac{\hat{S}}{\mu}) - V_j(\sqrt{TFP^THq})$. Solving equation (3.42) for G, we get,

$$V_j^*(\hat{G}) = (\mu I + V_j(\hat{X})V_j(\hat{X})^T)^{-1}(\hat{y}_j V_j(\hat{X}) - \mu V_j(\frac{\hat{S}}{\mu}) + \mu V_j(\sqrt{T}FP^THq)) \tag{3.37}$$

The equation can be simplified by using the Sherman-Morrison formula which changes the equation(3.43) as,

$$V_j^*(\hat{G}) = \frac{1}{\mu}\left(I - \frac{V_j(\hat{X})V_j(\hat{X})^T}{\mu + V_j(\hat{X}^T)V_j(\hat{X})}\right)(\hat{y}_j V_j(\hat{X}) - \mu V_j\left(\frac{\hat{S}}{\mu}\right) + \mu V_j(\sqrt{T}FP^THq)) \tag{3.38}$$

Solving for q: To get the optimal value of q, we differentiate the equation (3.38) w.r.t q as,

$$q^* = \underset{\mathbf{q_k}}{\operatorname{argmin}} \frac{\mu}{2}\sum_{k=1}^{C}\left\|\hat{g}_k - \sqrt{T}FP^Th_kq_k + \frac{\hat{s}_k}{\mu}\right\|_2^2 + \beta L_{kk}q_k^2 + 2\beta q_k\sum_{j\neq k}^{C} L_{kj}q_j \tag{3.39}$$

$$q_k^* = \frac{\mu\sqrt{T}h_k^TPg_k + Th_k^TPs_k - 2\beta\sum_{j\neq k}^{C} L_{kj}q_j}{\mu\sqrt{T}h_k^TPP^Th_k + 2\beta L_{kk}} \tag{3.40}$$

3.3.2 Lagrangian update

The Lagrangian parameter are updated by differentiating equation (3.3) w.r.t to S,

$$\begin{aligned} \hat{s}_k^* = \underset{\mathbf{s_k}}{\operatorname{argmin}} \frac{\mu}{2}\sum_{k=1}^{C}\left\|\hat{g}_k - \sqrt{T}FP^Th_kq_k + \frac{\hat{s}_k}{\mu}\right\| \\ \hat{g}_k - \sqrt{T}FP^Th_kq_k + \frac{\hat{s}_k}{\mu} = 0 \\ \hat{s}_k^* = \mu\left(\sqrt{T}FP^Th_kq_k - \hat{g}_k\right) \end{aligned} \tag{3.41}$$

Where, $\hat{s}_k^*$ denotes the optimal value for the k-th Lagrangian multiplier and F denotes the Fourier matrix which is multiplied to convert a matrix from spatial domain to Fourier domain.

Lagrangian variable at frame t can be updated as,

$$\hat{S}^t = S^{t-1} + \hat{S}^* \tag{3.42}$$

Where, $\hat{S}^*$ is the optimal value of S obtained at t-th frame. Similarly, μ is updated as,

$$\mu^t = min(\mu_{max}, \delta\mu^{t-1}) \tag{3.43}$$

Where, μ_{max} denotes the maximum possible value for the penalty term μ. Both the parameter S and μ are updated in the iterations of ADMM.

3.3.3 Target Localization

The response map for localizing the target is computed as,

$$\hat{r} = \left(\sum_{k=1}^{C} \hat{x}_k \circledast \hat{g}_k \right) \tag{3.44}$$

where, the maximum response of r determines the filter response map. To get the location of the target in the current frame, we convolve the response map $\hat{r}$ with the frame and, the location with the maximum response is considered to be the location of the object.

3.3.4 Model update

For current frame the model is updated as,

$$X^t_{model} = (1 - \eta)X^{t-1}_{model} + \eta\hat{X}^t \tag{3.45}$$

where, η is the learning rate and superscript t denote the current model template.
The reformulated tracker in Section (3.1,3.2) also uses the same way to update the Lagrangian parameter and updating the model.

3.3.5 Feature engineering

- **Proposed tracker** It uses a combination of hand crafted features and the deep learning features. We use multi-dimentional HOG features to estimate the scale. To learn the correlation filter we use a combination of Norm1 layer of VGG-M, which provides a good object representation and, Conv4-3 layer of VGG-16 which provides good distinguishing features.
- **BACF - CR:** Similar to Proposed tracker.
- **GFSDCF - CR:** For the scale estimation, GFSDCF-CR uses a combination of multi-channel HOG and color features. To train the correlation filters, we use res4ex layer of resnet-50.

3.3.6 Parameter settings

Our tracker implementation is based on MATLAB-2018b and runs on a computer with an Intel Xenon 3.60GHz CPU,32 GB of RAM and an NVIDIA 1080-Ti with 11GB of memory.

Parameter	Value
α	0.001
β	0.01
η	0.0186
λ	0.001
λ_1	10
λ_2	1
λ_3	[16 12]
δ	10
θ	0.001
μ (initially)	1
μ_{max}	10000
ADMM iterations	3
HOG channels	31
VGG-m & VGG-16 channels	128
resnet50 channels	128
color channels	10

TABLE 1: Tracker parameter values

CHAPTER 4

Results

In this chapter, we discussed the results of the proposed tracker and show a comparison with the different state of the art trackers BACF [8] and GFSDCF [9]. Object tracking is considered to be a challenging task and, the dataset used to evaluate trackers should cover a wider domain of tracking challenges. This chapter contains the performance evaluation of the trackers on the three datasets: TC128 [29] , VOT2017 [30] and VOT2019 [31]. These datasets contain sequences that are categorized in different challenges. We show the results of the trackers on the complete dataset as well as on individual challenges like:

- **IV:** Illumination Variation
- **SV:** Scale Variation
- **OCC:** Occlusion
- **DEF:** Deformation
- **MB:** Motion Blur
- **FM:** Fast Motion
- **IPR:** In-Plane Rotation
- **OPR:** Out-of-Plane Rotation
- **OV:** Out-of-View
- **BC:** Background Clutters
- **LR:** Low Resolution

The evaluation of the trackers on multiple challenges give a better view of the performance and robustness. These evaluations are done using the popular evaluation metrics like success and precision. A brief introduction of these evaluation metrics are given in the next section.

4.0.1 Evaluation matrices

There are several popular evaluation metrics used in visual tracking and are use widely used in the literature. These metrics assume that a manual annotation is given for a tracking sequence. Let us define the description of the object state in a sequence.

$$\Lambda = \{(R_t, x_t)\}_{t=1}^{N} \tag{4.1}$$

where, N is the length of the sequence. $x_t \in R^2$ and R_t denote the center and the bounding box of the object of the object at time t, respectively. The region overlap based measures address the normalization problem [51–53]. These measures compute the overlap between the ground-truth and the predicted target's region by the tracker.

$$\Phi(\Lambda^G, \Lambda^T) = \{\phi\}_{t=1}^{N}, \phi_t = \frac{R_t^G \cap R_t^T}{R_t^G \cup R_t^T} \tag{4.2}$$

The region overlap measure takes care of both, size and position of the ground-truth and predicted bounding box. The measure becomes zero when the tracker drifts and object gets lost completely. The measure doesn't shows an arbitrary error at tracking failures. For pixel classification, the equation (4.2) can be interpreted as,

$$\frac{R_t^G \cap R_t^T}{R_t^G \cup R_t^T} = \frac{TP}{TP + FN + FP} \tag{4.3}$$

where, TP is the true positive which defines the number of predicted positive sample that are actually positive. FN is the false negative that is the samples falsely classified as negative. FP is the false positive that is the sample that are actually negative but classified as positive. One of the other popular measure we used is precision [52] which is defined as,

$$Precision = \frac{TP}{TP + FP} \tag{4.4}$$

Where, a prediction is consider as positive if the distance between the center of predicted and ground-truth is less that equal to 20. To summarize the overlap measures over complete sequence, we use average overlap [33, 54],

$$\bar{\phi} = \frac{\phi_t}{N} \tag{4.5}$$

Where, t is the overlap at t-th frame and N is the length of sequence.

VOT toolkit: Evaluate a tracker by initializing it for the first frame and letting it run for the end of the sequence and, it resets the tracker if it drift off the target. The performance is evaluated by using the overlap between the ground-truth and the bounding boxes predicted from the tracker. VOT handles the problem of bias and variance by using the reset-based average overlap that does not get hampered by the varying sequence length.The toolkit runs two experiments on the sequences: baseline and unsupervised. The unsupervised experiments test tracker on noise errors, random initialization, etc. Results of VOT toolkit are interpret as follows,

Expected Overlap curve: EO curve is plot by using the expected values of overlap on the different length sequences. Expected overlap (EO) for a sequence of length n is define as,

$$E = \big[E[S_1], E[S_2], ..., E[S_n]\big] \tag{4.6}$$

Where, $E[S_i]$ denote expected value of overlap computed on a sequences of length i. Which is computed as,

$$E[S_i] = \frac{1}{n_i} \sum_{k=n_i}^{n_i} Seq_k w_k \tag{4.7}$$

Where, Seq_k and w_k denote the k-th sequence and weight associated with sequence.

Accuracy: Accuracy is define as the average overlap of all the tracking sequences, which can be computed as,

$$Accuracy = \frac{1}{n} \sum_{i=1}^{n} \Phi(\Lambda_i^G, \Lambda_i^T) \tag{4.8}$$

Where, n is the total number of sequences. Λ_i^G and Λ_i^T are ground truth and predicted bounding boxes of i-th sequence. $\Phi(.)$ denotes the overlap.

Robustness: Average failure on all sequences.

$$Robustness = \frac{1}{n} \sum_{i=1}^{n} F_i \tag{4.9}$$

Where, F_i is the number frames where tracker failed. Failure frames are the frames where overlap is zero.

AR curve: AR curve is a plot between accuracy and robustness of a tracker.

4.0.2 Improvement using Channel Regularization

In this work, we reformulate state of the art CF trackers [8–10] by using channel and graph regularization. In this subsection, we will show improvement in the performance of the already existing CF-based trackers by our modified regularization formulation. Here, we specifically talk about the performance gain in terms of success and precision rates.

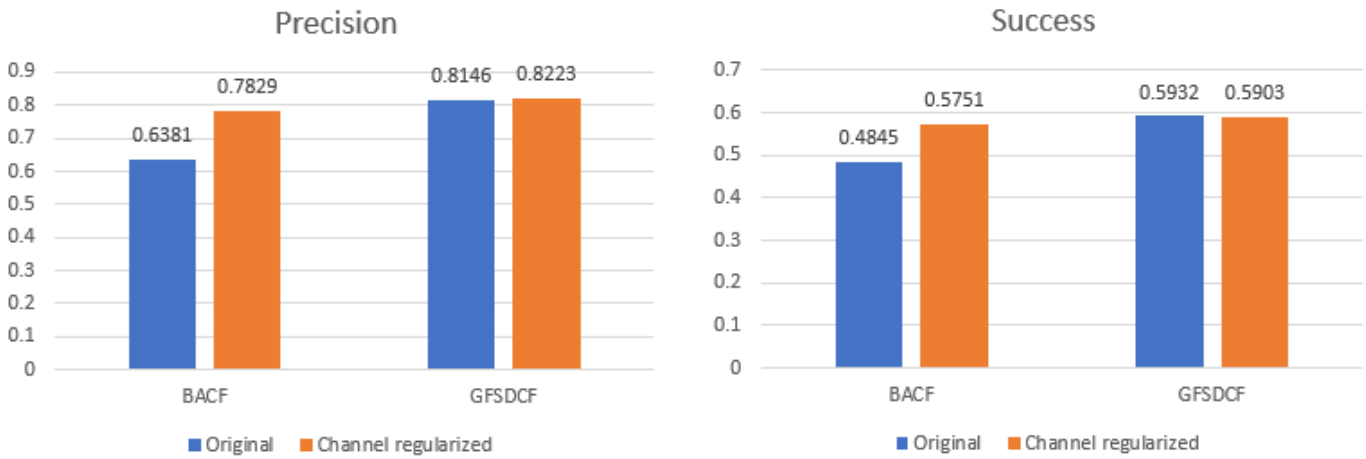


FIGURE 1: Improvement on Success and Precision by using channel regularization

The above plots show improvement in the success rate and the precision which is computed using a threshold of 20 pixel. The table below show the values of the precision and success.

Tracker	BACF	BACF-CR	GFSDCF	GFSDCF-CR
Success	0.4845	0.5751	0.5932	0.5903
Precision	0.6381	0.7829	0.8146	0.8223

TABLE 2: Success and Precision values for comparison

BACF tracker achieve an outstanding improvement of 15.7% on success rate and a gain of 18.5% on precision rate. Although GFSDCF does not show a decent improvement on success rate, but it showed a significant improvement on precision. From the above table, we can conclude that the regularization helps in improving the performance of already existing CF-based trackers.

4.0.3 Results on TC128 dataset

This section contains the evaluation results of the twenty state of the art correlation filter based tracker and the deep learning trackers. TC128 [29] consist of 129 sequences which contain more than 50k frames. We use the success and precision as evaluation metrics and the area under the curve (AUC) is use to get a better understanding of the overall performance.

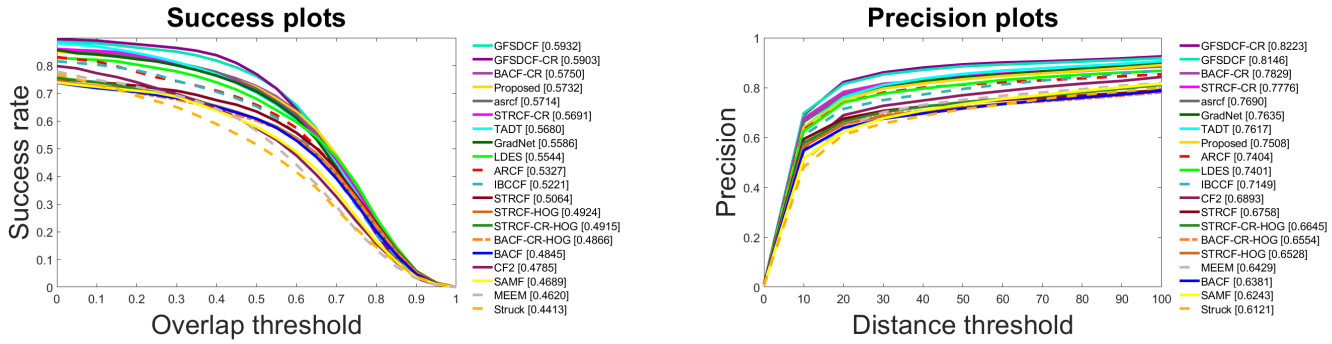


FIGURE 2: Overall Success and Precision plot for TC128 Dataset

To get a better understanding of the performance of the proposed tracker, we rank them on the basis of their success rate. The below table contains a comparison between the top 9 tracker with the proposed tracker. Top three tracker are shown in Red, Blue and Green colours.

Tracker	GFSDCF	Proposed	ASRCF	TADT	GradNet	LDES	ARCF	IBCCF	STRCF
Success	0.5932	0.5732	0.5714	0.5680	0.5586	0.5544	0.5327	0.5221	0.5064
Rank	1	2	3	4	5	6	7	8	9

TABLE 3: Ranking on TC128 using success plot AUC

The proposed tracker secured a good rank in the comparison with the state of the art trackers. The proposed tracker outperformed the deep learning tracker [44, 55]. It also secured a good position in the comparison with the CF trackers [7, 8, 10, 12, 55, 56]. The deep learning trackers are considered to have robust features. A comparison with them shows that the Proposed tracker not only has robust features but, also has a good correlation filter learning approach. Similarly, a comparison on the different challenges on the TC128 dataset is shown in the appendix (1.1).

4.0.4 Results on VOT2017 dataset

This section contains results of the VOT2017 toolkit [30]. There are 23 trackers used to show a comparison between the Proposed and channel-regularized trackers. VOT2017 is a popular dataset and comes with a toolkit to evaluate the trackers.

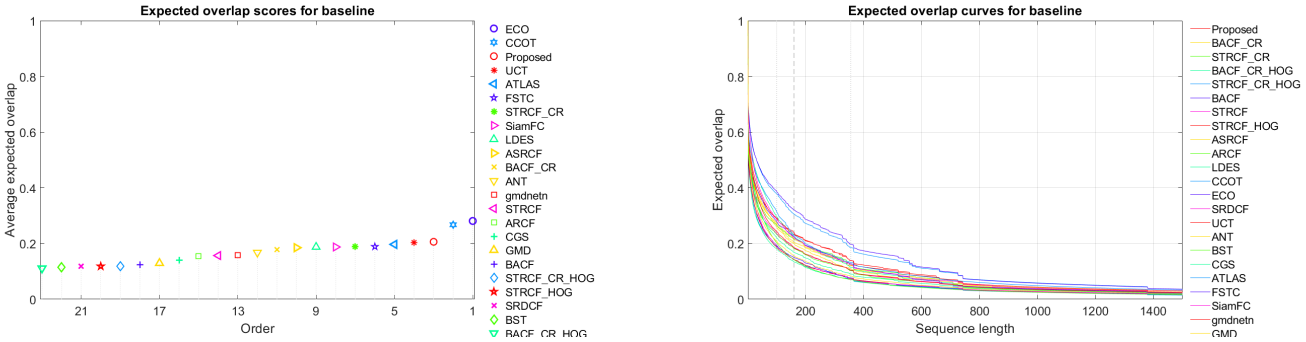


FIGURE 3: VOT2017 expected overlap curves

The figure shows the performance of the tracker using the expected overlap score. The expected overlap score for baseline shows the comparison based on the AUC obtained from the overlap curve. To show the performance of the Proposed tracker on the VOT2017, we rank the top 9 trackers on the basis of their overlap scores.

Tracker	ECO	CCOT	Proposed	UCT	ATLAS	FSTC	LDES	ASRCF	ANT
Success	0.2809	0.2674	0.2061	0.2042	0.1969	0.1889	0.1875	0.1851	0.1676
Rank	1	2	3	4	5	6	7	8	9

TABLE 4: EAO ranking on VOT2017

The Proposed tracker is giving good performance and outperformed several state of the art trackers [7, 12, 57, 58]. ECO [35] and CCOT [59] are giving better results than Proposed tracker. CCOT is a CF tracker it gives decent results but it has a tracking speed of less than 1 FPS. On the other hand, ECO is an end-to-end deep learning tracker and gives a tracking speed of 8FPS whereas, our Proposed tracker has an excellent balance between the tracking speed and the robustness. The toolkit gives the report on the basis of different evaluation metrics like overlap. Toolkit also gives a comparison on the different challenges. This section only contains the comparison on the expected overlap. More results on the different challenges can be found in the appendix (1.2).

4.0.5 Results on VOT2019 dataset

This section contains results on the VOT2019 dataset obtained from the toolkit [31]. There is comparison of the Proposed and reformulated tracker with the other trackers on the basis of expected overlap.

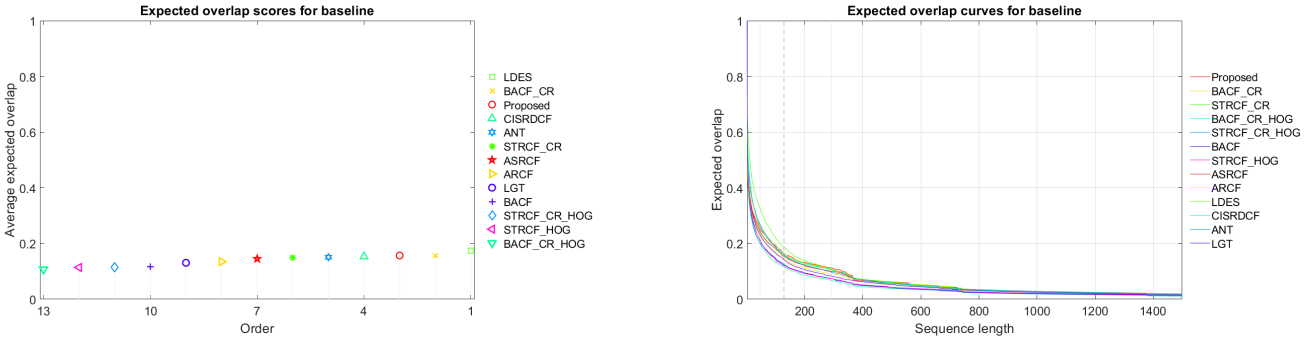


FIGURE 4: VOT2019 expected overlap curves

The figure shows a comparison on the basis of expected overlap. The is the plot between the expected overlap and the length of the sequence which shows the overlap of the tracker results and the ground-truth on the different length sequences. The below table shows the comparison of the Proposed tracker with the 7 other trackers.

Trackers	LDES	Proposed	CISRDCF	ANT	ASRCF	ARCF	LGT	BACF
Success	0.1747	0.1569	0.1533	0.1509	0.1451	0.1351	0.1308	0.1162
Rank	1	2	3	4	5	6	7	8

TABLE 5: EAO ranking on VOT2019

The Proposed tracker gives a decent performance on the VOT2019 dataset. The LDES [57] tracker secure the first rank but on the other hand, it does not give real-time speed. A detailed analysis of the speed and robustness is shown in the next section. More results on the VOT2019 dataset can be found in the appendix (1.3).

4.1 Tracking speed Comparison

In this section, we will do a comparison between the tracking speed of different state of the art trackers with our proposed tracker. A tracker is called real-time is it's tracking speed is greater than or equal to 30FPS.

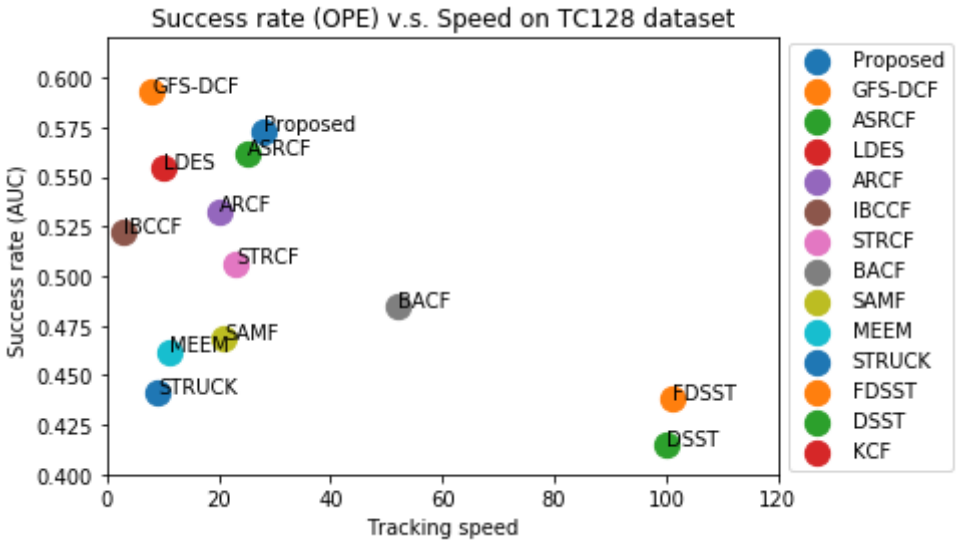


FIGURE 5: Tracking speed comparison

The above figure shows the plot between the success rate and the tracking speed in frame per second. In the object tracking, there is a trade off between the tracking speed and the robustness. The below table show the ranking of the trackers on the basis of speed.

Trackers	Proposed	ASRCF	ARCF	STRCF	BACF	SAMF	MEEM	FDSST	DSST	KCF
FPS	28	26	20	23	52	21	11	111	100	150
Rank	5	6	9	7	4	8	10	2	3	1

TABLE 6: Ranking on tracking speed with CF trackers

KCF [19] , DSST [60] , FDSST [61] track at a speed of more than 100 FPS but do not give a good tracking accuracy whereas, the trackers like GFSDCF [9], LDES [57] give a robust tracking but the tracking speed is very less than real time. Our Proposed tracker (CGRCF) gives a perfect balance between the robustness and the speed.

4.2 Conclusion

We proposed two efficient regularizers namely, channel regularizer and graph regularizer. We also proposed a novel tracker using these regularizers. Additionally, we demonstrated how these regularizers can be plugged into existing trackers. Based on the robust results and analysis, we can conclude that the channel and graph regularization helps us to build a tracker that can achieve the state of art performance. Most of the regularization techniques used in object tracking do not take into consideration the relation between the features. Our channel-graph regularization formulation helps in selecting the essential features as well as assign similar weights to the similar feature channels. By using ADMM, the proposed tracker can be efficiently solved and able to track at a speed 30FPS.

4.3 Future Work

Our proposed tracker currently uses a combination of handcrafted and deep learning features. These features are used for the scale estimation and for learning correlation filters. While using a variety of features, the probability of features to become less correlated is high. In such scenarios, the graph regularization approach may not work as expected. The GR (graph regularization) based approach works well when the features are correlated. In future, we can introduce a more robust way to compute the relation between the features. That will help in making our proposed approach more robust. Currently, we are using the VGG net [23] for tracking. In future, we may explore features from more sophisticated convolution neural networks like resnet [24]. We could use graph regularization technique with the end-to-end deep learning trackers to get more robust features. In our tracker, we are extracting patches from the localization of the current location of object. But in some scenario where the object movement is very high, we may loose the object. To resolve this problem, we can increase the search area by introducing more advance techniques like YOLO [62]. This may result in a slower tracker but the performance may also increase by a certain factor.

Bibliography

[1] A. Mangawati, Mohana, M. Leesan, H. V. R. Aradhya in 2018 International Conference on Communication and Signal Processing (ICCSP), **2018**, pp. 0667–0671.

[2] Z. Jia, A. Balasuriya, S. Challa, 'Vision based target tracking for autonomous land vehicle navigation: a brief survey', *Recent Patents on Computer Science* **2009**, *2*, 32–42.

[3] S. Handrich, A. Al-Hamadi in 2012 19th IEEE International Conference on Image Processing, IEEE, **2012**, pp. 1981–1984.

[4] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, P. H. Torr, 'Fully-convolutional siamese networks for object tracking', **2016**, 850–865.

[5] Y. Zhang, L. Wang, J. Qi, D. Wang, M. Feng, H. Lu in Proceedings of the European conference on computer vision (ECCV), **2018**, pp. 351–366.

[6] C. Huang, S. Lucey, D. Ramanan in Proceedings of the IEEE International Conference on Computer Vision, **2017**, pp. 105–114.

[7] K. Dai, D. Wang, H. Lu, C. Sun, J. Li in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2019**, pp. 4670–4679.

[8] H. Kiani Galoogahi, A. Fagg, S. Lucey in Proceedings of the IEEE international conference on computer vision, **2017**, pp. 1135–1143.

[9] T. Xu, Z.-H. Feng, X.-J. Wu, J. Kittler in Proceedings of the IEEE International Conference on Computer Vision, **2019**, pp. 7950–7960.

[10] F. Li, C. Tian, W. Zuo, L. Zhang, M.-H. Yang in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2018**, pp. 4904–4913.

[11] J. F. Henriques, R. Caseiro, P. Martins, J. Batista in European conference on computer vision, Springer, **2012**, pp. 702–715.

[12] Z. Huang, C. Fu, Y. Li, F. Lin, P. Lu, 'Learning aberrance repressed correlation filters for real-time uav tracking', **2019**, 2891–2900.

[13] Y. Zou, T. Chen, 'Laser vision seam tracking system based on image processing and continuous convolution operator tracker', *Optics and Lasers in Engineering* **2018**, *105*, 141–149.

[14] D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui in 2010 IEEE computer society conference on computer vision and pattern recognition, IEEE, **2010**, pp. 2544–2550.

[15] L. Wang, W. Ouyang, X. Wang, H. Lu in Proceedings of the IEEE international conference on computer vision, **2015**, pp. 3119–3127.

[16] H. Nam, B. Han in Proceedings of the IEEE conference on computer vision and pattern recognition, **2016**, pp. 4293–4302.

[17] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. Lau, M.-H. Yang in Proceedings of the IEEE International Conference on Computer Vision, **2017**, pp. 2555–2564.

[18] B. Li, J. Yan, W. Wu, Z. Zhu, X. Hu in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2018**, pp. 8971–8980.

[19] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, 'High-speed tracking with kernelized correlation filters', *IEEE transactions on pattern analysis and machine intelligence* **2014**, *37*, 583–596.

[20] H. M. Ozaktas, M. A. Kutay in 2001 European Control Conference (ECC), IEEE, **2001**, pp. 1477–1483.

[21] J. Zheng, T. Su, W. Zhu, X. He, Q. H. Liu, 'Radar high-speed target detection based on the scaled inverse Fourier transform', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2014**, *8*, 1108–1119.

[22] S. Gorlatch, 'Programming with divide-and-conquer skeletons: A case study of FFT', *The Journal of Supercomputing* **1998**, *12*, 85–97.

[23] K. Simonyan, A. Zisserman, 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint arXiv:1409.1556* **2014**.

[24] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi in Thirty-first AAAI conference on artificial intelligence, **2017**.

[25] J. Chang-Yeon, 'Face Detection using LBP features', *Final Project Report* **2008**, *77*, 1–4.

[26] H. Zhou, Y. Yuan, C. Shi, 'Object tracking using SIFT features and mean shift', *Computer vision and image understanding* **2009**, *113*, 345–352.

[27] Z.-R. Wang, Y.-L. Jia, H. Huang, S.-M. Tang in 2008 11th International IEEE Conference on Intelligent Transportation Systems, IEEE, **2008**, pp. 1155–1160.

[28] X. Dong, D. Thanou, P. Frossard, P. Vandergheynst, 'Learning Laplacian matrix in smooth graph signal representations', *IEEE Transactions on Signal Processing* **2016**, *64*, 6160–6173.

[29] P. Liang, E. Blasch, H. Ling, 'Encoding color information for visual tracking: Algorithms and benchmark', *IEEE Transactions on Image Processing* **2015**, *24*, 5630–5644.

[30] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin Zajc, T. Vojir, G. Häger, A. Lukežič, A. Eldesokey, G. Fernandez, The Visual Object Tracking VOT2017 Challenge Results, **2017**.

[31] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, L. Čehovin Zajc, O. Drbohlav, A. Lukezic, A. Berg, A. Eldesokey, J. Kapyla, G. Fernandez, The Seventh Visual Object Tracking VOT2019 Challenge Results, **2019**.

[32] L. Čehovin, A. Leonardis, M. Kristan, 'Visual object tracking performance measures revisited', *IEEE Transactions on Image Processing* **2016**, *25*, 1261–1274.

[33] Y. Wu, J. Lim, M.-H. Yang in Proceedings of the IEEE conference on computer vision and pattern recognition, **2013**, pp. 2411–2418.

[34] M. Danelljan, G. Hager, F. Shahbaz Khan, M. Felsberg in Proceedings of the IEEE international conference on computer vision, **2015**, pp. 4310–4318.

[35] M. Danelljan, G. Bhat, F. Shahbaz Khan, M. Felsberg in Proceedings of the IEEE conference on computer vision and pattern recognition, **2017**, pp. 6638–6646.

[36] M. Danelljan, F. Shahbaz Khan, M. Felsberg, J. Van de Weijer in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2014**, pp. 1090–1097.

[37] M. Tang, B. Yu, F. Zhang, J. Wang in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2018**, pp. 4874–4883.

[38] Y. Sun, C. Sun, D. Wang, Y. He, H. Lu in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2019**, pp. 5783–5791.

[39] D. Zhang, H. Maei, X. Wang, Y.-F. Wang, 'Deep reinforcement learning for visual object tracking in videos', *arXiv preprint arXiv:1701.08936* **2017**.

[40] S. Yun, J. Choi, Y. Yoo, K. Yun, J. Young Choi in Proceedings of the IEEE conference on computer vision and pattern recognition, **2017**, pp. 2711–2720.

[41] L. Ren, X. Yuan, J. Lu, M. Yang, J. Zhou in Proceedings of the European Conference on Computer Vision (ECCV), **2018**, pp. 684–700.

[42] A. He, C. Luo, X. Tian, W. Zeng in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2018**, pp. 4834–4843.

[43] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, S. Maybank in Proceedings of the IEEE conference on computer vision and pattern recognition, **2018**, pp. 4854–4863.

[44] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, H. Lu in Proceedings of the IEEE International Conference on Computer Vision, **2019**, pp. 6162–6171.

[45] L. Zhang, A. Gonzalez-Garcia, J. v. d. Weijer, M. Danelljan, F. S. Khan in Proceedings of the IEEE International Conference on Computer Vision, **2019**, pp. 4010–4019.

[46] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, 'Distributed optimization and statistical learning via the alternating direction method of multipliers', *Foundations and Trends® in Machine learning* **2011**, *3*, 1–122.

[47] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, D. Cai, 'Graph regularized sparse coding for image representation', *IEEE transactions on image processing* **2010**, *20*, 1327–1336.

[48] M. Belkin, P. Niyogi in Advances in neural information processing systems, **2002**, pp. 585–591.

[49] T. Zhou, H. Bhaskar, F. Liu, J. Yang, 'Graph regularized and locality-constrained coding for robust visual tracking', *IEEE Transactions on Circuits and Systems for Video Technology* **2016**, *27*, 2153–2164.

[50] R. Glowinski, P. Le Tallec, *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*, SIAM, **1989**.

[51] M. Godec, P. M. Roth, H. Bischof, 'Hough-based tracking of non-rigid objects', *Computer Vision and Image Understanding* **2013**, *117*, 1245–1256.

[52] K. Zhang, L. Zhang, M.-H. Yang, 'Fast compressive tracking', *IEEE transactions on pattern analysis and machine intelligence* **2014**, *36*, 2002–2015.

[53] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, 'Visual tracking: An experimental survey', *IEEE transactions on pattern analysis and machine intelligence* **2013**, *36*, 1442–1468.

[54] D. Wang, H. Lu, M.-H. Yang, 'Online object tracking with sparse prototypes', *IEEE transactions on image processing* **2012**, *22*, 314–325.

[55] X. Li, C. Ma, B. Wu, Z. He, M.-H. Yang in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2019**, pp. 1369–1378.

[56] F. Li, Y. Yao, P. Li, D. Zhang, W. Zuo, M.-H. Yang in Proceedings of the IEEE International Conference on Computer Vision Workshops, **2017**, pp. 2001–2009.

[57] Y. Li, J. Zhu, S. C. Hoi, W. Song, Z. Wang, H. Liu in Proceedings of the AAAI Conference on Artificial Intelligence, *Vol. 33*, **2019**, pp. 8666–8673.

[58] Z. Zhu, G. Huang, W. Zou, D. Du, C. Huang in Proceedings of the IEEE International Conference on Computer Vision Workshops, **2017**, pp. 1973–1982.

[59] M. Danelljan, A. Robinson, F. Shahbaz Khan, M. Felsberg in ECCV, **2016**.

[60] M. Danelljan, G. Häger, F. Khan, M. Felsberg in British Machine Vision Conference, Nottingham, September 1-5, 2014, BMVA Press, **2014**.

[61] M. Danelljan, G. Häger, F. S. Khan, M. Felsberg, 'Discriminative scale space tracking', *IEEE transactions on pattern analysis and machine intelligence* **2016**, *39*, 1561–1575.

[62] M. J. Shafiee, B. Chywl, F. Li, A. Wong, 'Fast YOLO: A fast you only look once system for real-time embedded object detection in video', *arXiv preprint arXiv:1709.05943* **2017**.

[63] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, L. Cehovin Zajc, O. Drbohlav, A. Lukezic, A. Berg et al. in Proceedings of the IEEE International Conference on Computer Vision Workshops, **2019**, pp. 0–0.

1 Appendix

1.1 Results on different challenges in TC128 dataset

This section contains the results of the trackers on the different challenges that are discussed in the Chapter-4. This comparison is done by using the success and precision evaluation metrics. The plots show the success rate and the precision rate on a visual tracking challenge in TC128 dataset, along with a table which shows the top 8 trackers and rank according to their success overlap rate. The values along with the tracker name denotes the area under the curve. AUC helps to compute the overall performance of the tracker on the particular challenge. Top three tracker are denoted in red, blue green colour.

1.1.1 Background Clutter

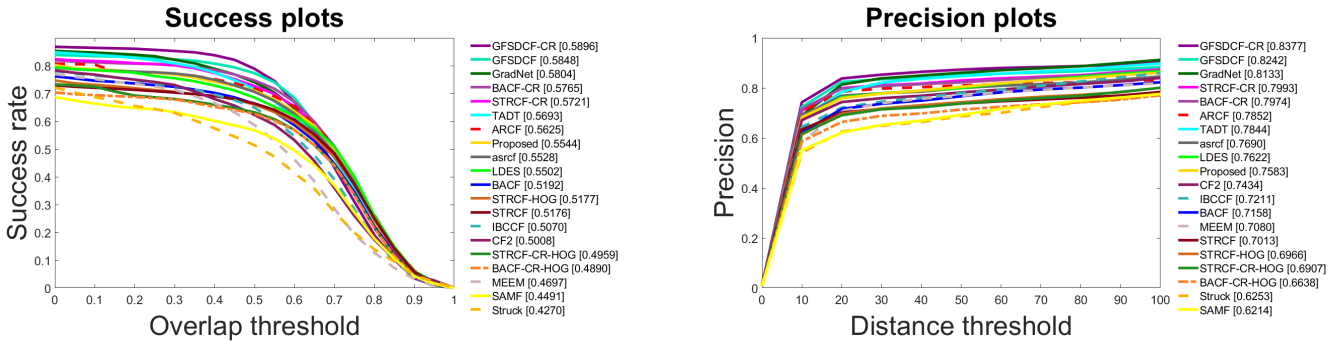


FIGURE 6: Background Clutter

Rank	1	2	3	4	5	6	7	8
Tracker	GFSDCF	ARCF	Proposed	ASRCF	LDES	BACF	STRCF	IBCCF
Success	0.5848	0.5625	0.5544	0.5528	0.5502	0.5192	0.5176	0.5070

TABLE 7: Ranking of trackers - Background Clutters

1.1.2 Deformation

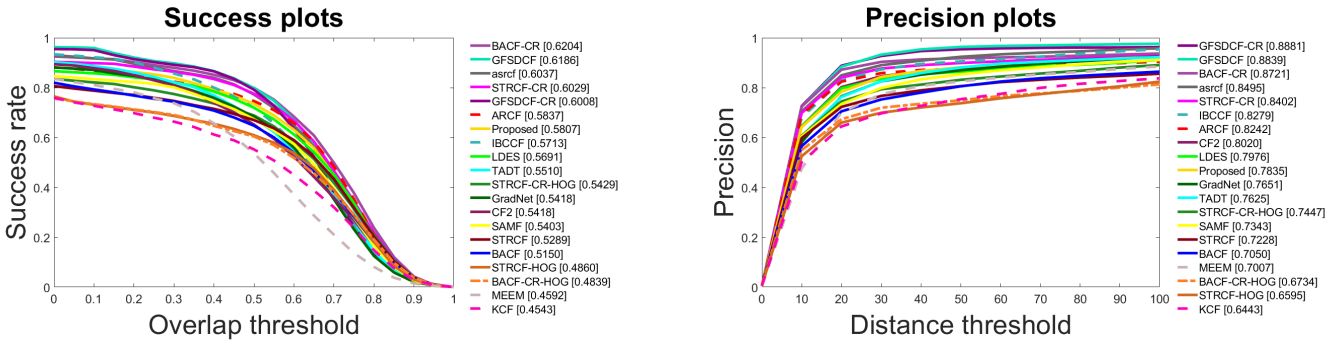


FIGURE 7: Deformation

Rank	1	2	3	4	5	6	7	8
Tracker	GFSDCF	ASRCF	ARCF	Proposed	IBCCF	LDES	CF2	SAMF
Success	0.6186	0.6037	0.5837	0.5807	0.5713	0.5691	0.5418	0.5403

TABLE 8: Ranking of trackers - Deformation

1.1.3 Fast Motion

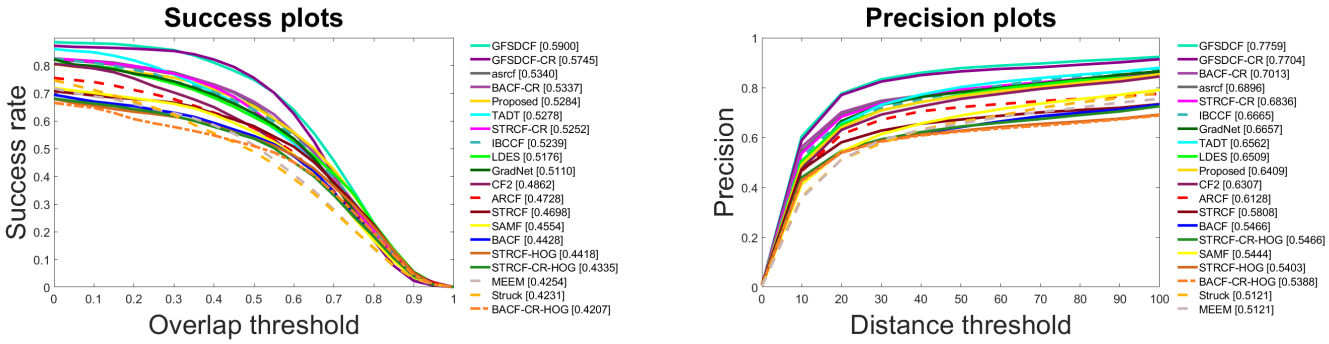


FIGURE 8: Fast Motion

Rank	1	2	3	4	5	6	7	8
Tracker	GFSDCF	ASRCF	Proposed	IBCCF	LDES	CF2	ARCF	STRCF
Success	0.5900	0.5340	0.5284	0.5239	0.5176	0.4862	0.4728	0.4698

TABLE 9: Ranking of trackers - Fast Motion

1.1.4 In-Plane Rotation

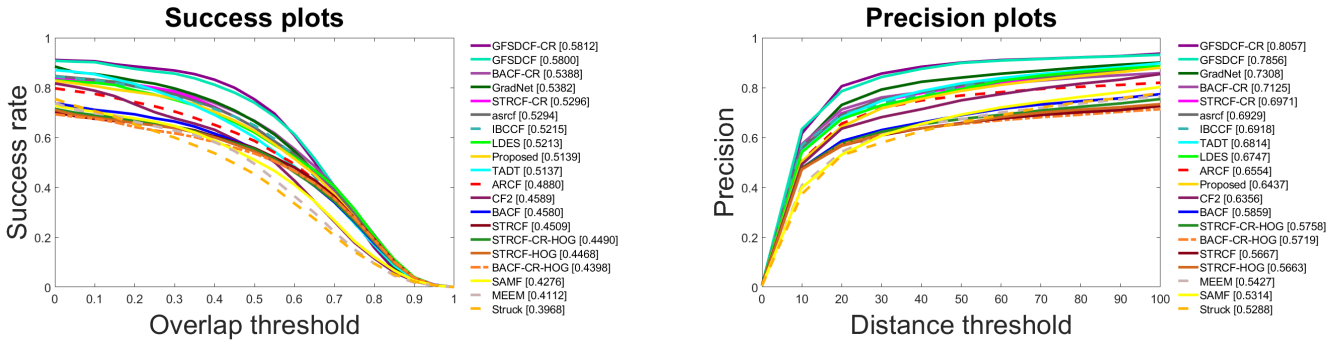


FIGURE 9: In-Plane Rotation

Rank	1	2	3	4	5	6	7	8
Tracker	GFSDCF	ASRCF	IBCCF	LDES	Proposed	ARCF	CF2	BACF
Success	0.5800	0.5294	0.5215	0.5213	0.5139	0.4880	0.4589	0.4580

TABLE 10: Ranking of trackers - In-Plane Rotation

1.1.5 Illumination variation

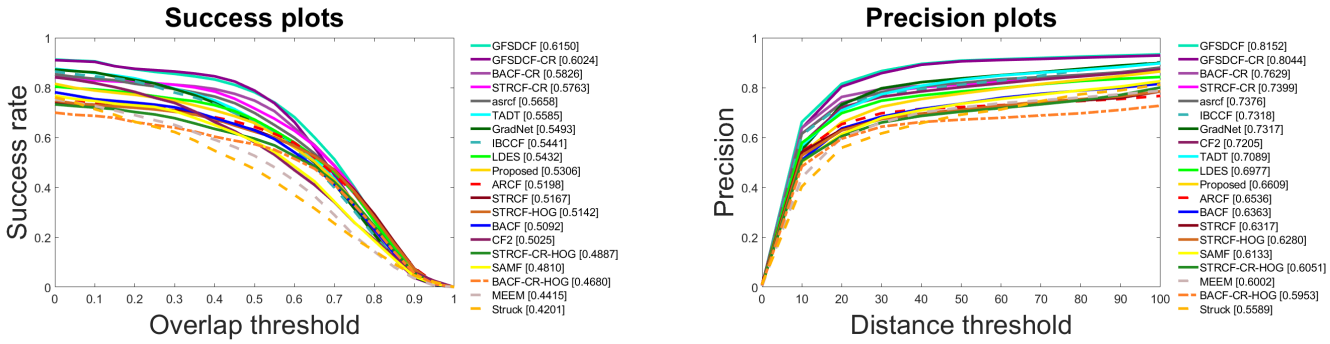


FIGURE 10: Illumination variation

Rank	1	2	3	4	5	6	7	8
Tracker	GFSDCF	ASRCF	IBCCF	LDES	Proposed	ARCF	STRCF	BACF
Success	0.6150	0.5658	0.5441	0.5432	0.5306	0.5198	0.5167	0.5092

TABLE 11: Ranking of trackers - Illumination variance

1.1.6 Low Resolution

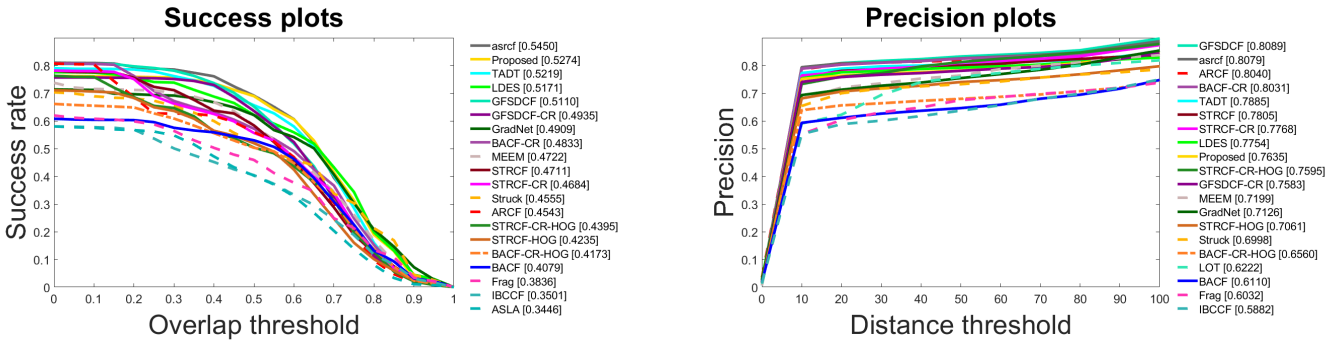


FIGURE 11: Low Resolution

Rank	1	2	3	4	5	6	7	8
Tracker	ASRCF	Proposed	LDES	GFSDCF	MEEM	STRCF	Struck	ARCF
Success	0.5450	0.5274	0.5171	0.5110	0.4722	0.4711	0.4555	0.4543

TABLE 12: Ranking of trackers - Low Resolution

1.1.7 Motion Blur

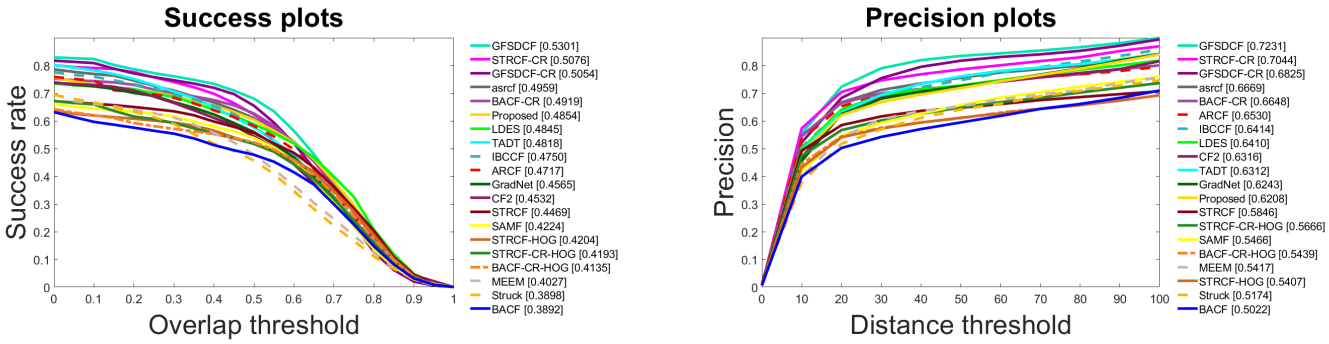


FIGURE 12: Motion Blur

Rank	1	2	3	4	5	6	7	8
Tracker	GFSDCF	ASRCF	Proposed	LDES	IBCCF	ARCF	CF2	STRCF
Success	0.5301	0.4959	0.4854	0.4845	0.4750	0.4717	0.4532	0.4469

TABLE 13: Ranking of trackers - Motion Blur

1.1.8 Occlusion

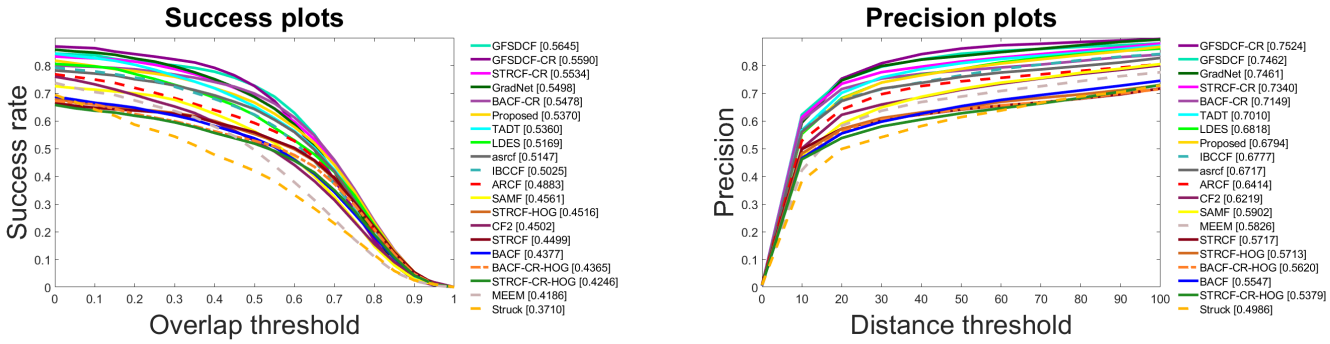


FIGURE 13: Occlusion

Rank	1	2	3	4	5	6	7	8
Tracker	GFSDCF	Proposed	LDES	ASRCF	IBCCF	ARCF	SAMF	CF2
Success	0.5645	0.5370	0.5169	0.5147	0.5025	0.4883	0.4561	0.4502

TABLE 14: Ranking of trackers - Occlusion

1.1.9 Out-of-Plane

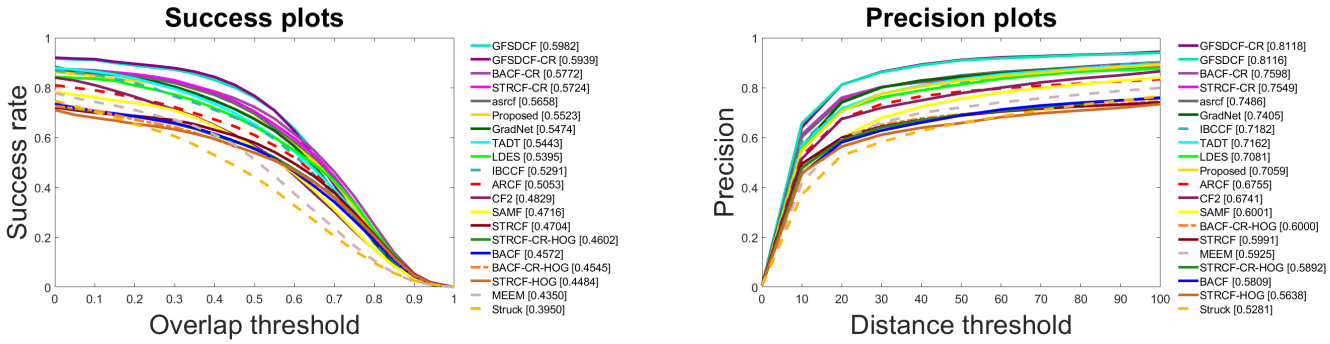


FIGURE 14: Out-of-Plane

Rank	1	2	3	4	5	6	7	8
Tracker	GFSDCF	ASRCF	Proposed	LDES	IBCCF	ARCF	CF2	SAMF
Success	0.5982	0.5658	0.5523	0.5395	0.5291	0.5053	0.4829	0.4716

TABLE 15: Ranking of trackers - Out-of-Plane Rotation

1.1.10 Out-of-View

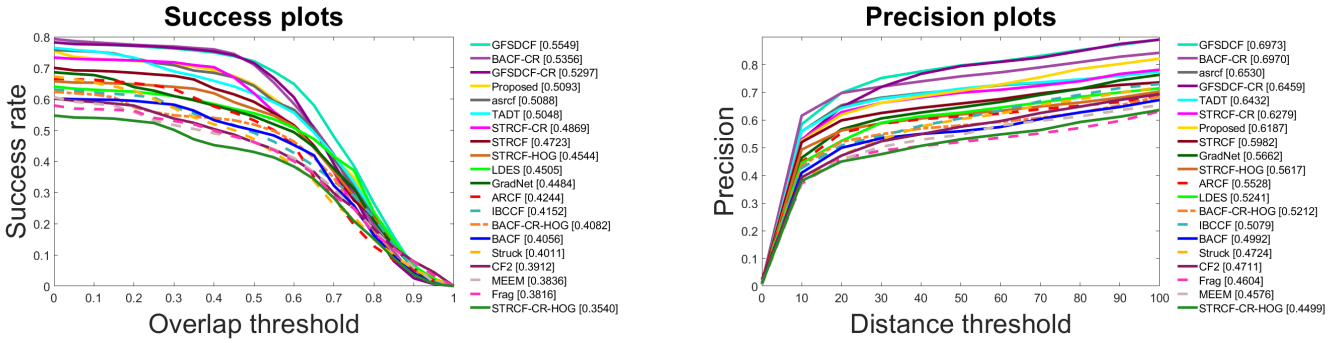


FIGURE 15: Out-of-View

Rank	1	2	3	4	5	6	7	8
Tracker	GFSDCF	Proposed	ASRCF	STRCF	LDES	ARCF	IBCCF	BACF
Success	0.5549	0.5093	0.5088	0.4723	0.4505	0.4244	0.4152	0.4056

TABLE 16: Ranking of trackers - Out-of-View

1.1.11 Scale Variance

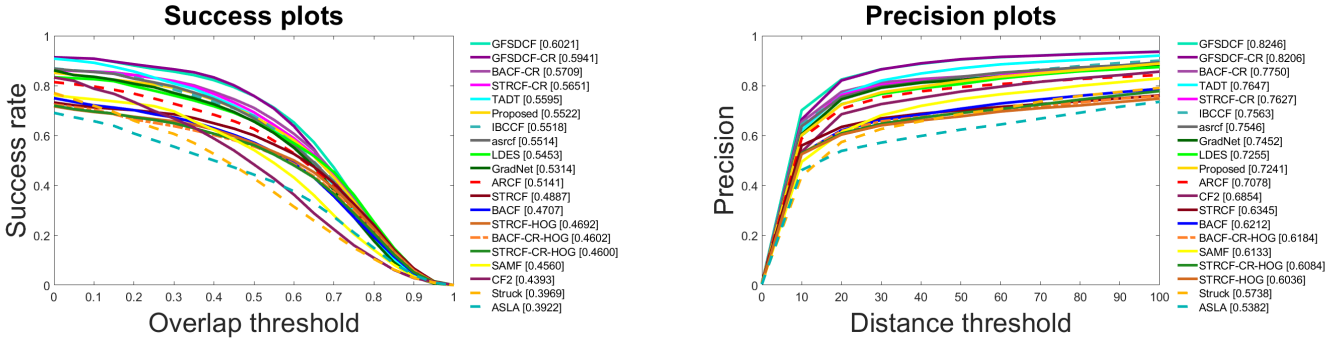


FIGURE 16: Scale Variance

Rank	1	2	3	4	5	6	7	8
Tracker	GFSDCF	Proposed	IBCCF	ASRCF	LDES	ARCF	STRCF	BACF
Success	0.6021	0.5522	0.5518	0.5514	0.5453	0.5141	0.4887	0.4707

TABLE 17: Ranking of trackers - Scale Variation

VOT toolkit evaluates tracker on the challenges CM = Camera Motion, EMP = Empty Tag, IV= Illumination Variation, MC = Motion Change, OCC = Occlusion and SC = Size Change on both baseline and unsupervised experiments.

Evaluation on VOT2017 dataset is as follows:

(1) TABLE 18 shows the accuracy comparison of the proposed approaches with recent trackers for the baseline experiment.
(2) TABLE 19 shows the robustness comparison of the proposed approaches with recent trackers for the baseline experiment.
(3) Figure 17 show the Accuracy-Robustness (AR) plots of individual challenges for baseline experiments.
(4) FIGURE 18 shows the overlap curves of individual challenges for unsupervised experiments.

Evaluation on VOT2019 dataset is as follows:

(1) Figure 19 shows the Accuracy-Robustness (AR) plots of individual challenges for baseline experiments.
(2) TABLE 20 shows the accuracy comparison of the proposed approaches with recent trackers for the baseline experiment.
(3) TABLE 21 shows the robustness comparison of the proposed approaches with recent trackers for the baseline experiment.
(4) FIGURE 20 shows the overlap curves of individual challenges for unsupervised experiments.

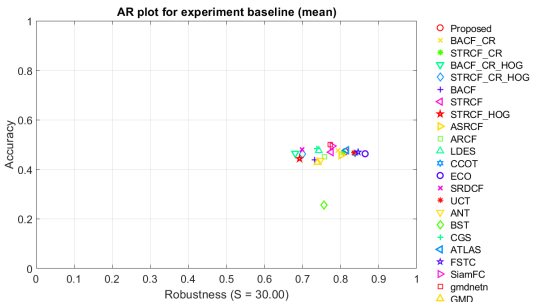
1.2 Results on VOT2017 dataset

	CM	EMP	IV	MC	OCC	SC	Mean	Weighted Mean	Average Pooled
Proposed (CGRCF)	0.5077	0.4503	0.4779	0.4828	**0.4524**	0.4585	0.4716	0.4737	0.4754
BACF-CR	**0.5262**	0.4804	0.4668	0.4748	**0.4420**	**0.4655**	0.4760	**0.4875**	0.4904
STRCF-CR	0.5085	0.4817	0.4831	0.4655	0.4148	0.4541	0.4680	0.4779	0.4825
BACF-CR-HOG	0.5153	0.4870	0.4753	0.4484	0.3999	**0.4615**	0.4646	0.4786	0.4846
STRCF-CR-HOG	0.5236	0.4638	0.4763	0.4656	0.3798	0.4603	0.4616	0.4741	0.4787
Correlation Filter based and Hybrid Trackers									
BACF [8]	0.4845	0.4515	0.4532	0.4638	**0.4339**	0.3440	0.4385	0.4476	0.4526
STRCF-HOG [10]	0.4912	0.4476	0.4758	0.4407	0.3698	0.4337	0.4431	0.4514	0.4552
ASRCF [7]	0.4906	0.4725	0.4809	0.4479	0.4010	0.4506	0.4572	0.4654	0.4701
ARCF [12]	0.4883	0.4633	0.4525	0.4551	0.4100	0.4393	0.4514	0.4615	0.4647
LDES [57]	**0.5370**	0.4956	0.4783	**0.4850**	0.4053	0.4559	0.4762	**0.4929**	**0.5023**
CCOT [**danelljan2016beyond**]	0.5158	0.4994	0.4460	**0.4835**	0.3917	0.4499	0.4644	0.4851	**0.4949**
ECO [35]	0.5131	0.4846	**0.5026**	0.4810	0.3520	0.4451	0.4631	0.4762	0.4848
SRDCF [34]	0.4816	**0.5080**	**0.5796**	0.4492	0.4171	0.4416	**0.4795**	0.4767	0.4867
UCT [**kristan2017visual**]	0.5026	0.4969	0.4726	0.4800	0.4312	0.4257	0.4681	0.4807	0.4887
ANT [**kristan2017visual**]	0.4890	0.4541	0.3986	0.4504	0.4099	0.4166	0.4364	0.4540	0.4622
BST [**kristan2017visual**]	0.2376	0.3091	0.2606	0.2391	0.2412	0.2443	0.2553	0.2627	0.2697
CGS [**kristan2017visual**]	**0.5380**	**0.5018**	0.4831	0.4820	0.4289	**0.4632**	**0.4828**	**0.4979**	**0.5063**
ATLAS [**kristan2017visual**]	0.4926	**0.5079**	**0.5248**	0.4830	0.4098	0.4449	**0.4772**	0.4835	0.4916
FSTC [**kristan2017visual**]	0.4926	0.4826	0.4770	**0.4986**	0.4076	0.4604	0.4698	0.4783	0.4836
GMD [**kristan2017visual**]	0.4607	0.4573	0.4258	0.4411	0.3801	0.4072	0.4287	0.4422	0.4490

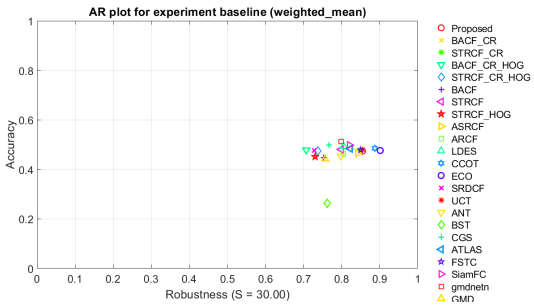
TABLE 18: Accuracy comparison of the proposed approaches with recent trackers for the baseline experiment on VOT-2017 dataset.

	CM	EMP	IV	MC	OCC	SC	Mean	Weighted Mean	Average Pooled
Proposed (CGRCF)	**40.00**	30.00	7.00	21.00	**18.00**	23.00	**23.16**	**29.23**	**97.00**
BACF-CR	53.00	27.00	5.00	28.00	29.00	25.00	27.83	34.26	118.00
STRCF-CR	44.00	31.00	7.00	25.00	**19.00**	20.00	24.33	31.01	105.00
BACF-CR-HOG	99.00	78.00	9.00	51.00	32.00	38.00	51.16	69.72	231.00
STRCF-CR-HOG	79.00	74.00	10.00	44.00	29.00	34.00	45.00	60.77	203.00
Correlation Filter based and Hybrid Trackers									
BACF [8]	77.00	61.00	6.00	43.00	31.00	33.00	41.83	55.77	189.00
STRCF-HOG [10]	85.00	68.00	9.00	46.00	39.00	31.00	46.33	61.33	198.00
ASRCF [7]	45.00	**26.00**	6.00	29.00	23.00	23.00	25.33	30.97	105.00
ARCF [12]	53.00	49.00	8.00	29.00	34.00	20.00	32.16	41.41	141.00
LDES [57]	47.00	46.00	10.00	31.00	31.00	27.00	32.00	39.64	133.00
CCOT [**danelljan2016beyond**]	**26.00**	**19.00**	6.00	**19.00**	**22.00**	**14.00**	**17.66**	**20.41**	**68.00**
ECO [35]	**25.00**	**14.00**	4.00	**18.00**	**22.00**	**9.00**	**15.33**	**17.66**	**59.00**
SRDCF [34]	76.00	86.00	9.00	49.00	32.00	29.00	46.83	64.11	208.00
UCT [**kristan2017visual**]	44.00	29.00	**3.00**	24.00	27.00	**12.00**	**23.16**	29.79	103.00
ANT [**kristan2017visual**]	64.00	**26.00**	8.00	45.00	27.00	30.00	33.33	40.15	135.00
BST [**kristan2017visual**]	74.73	66.93	4.86	41.93	24.66	26.73	39.97	55.50	188.60
CGS [**kristan2017visual**]	73.66	62.46	8.40	39.26	24.13	26.73	39.11	53.37	172.06
ATLAS [**kristan2017visual**]	60.00	30.00	**2.00**	35.00	24.00	22.00	28.83	37.42	127.00
FSTC [**kristan2017visual**]	57.00	**26.00**	**1.00**	**15.00**	30.00	15.00	24.00	31.95	114.00
GMD [**kristan2017visual**]	86.06	50.33	5.40	47.66	30.73	26.20	41.06	54.73	187.46

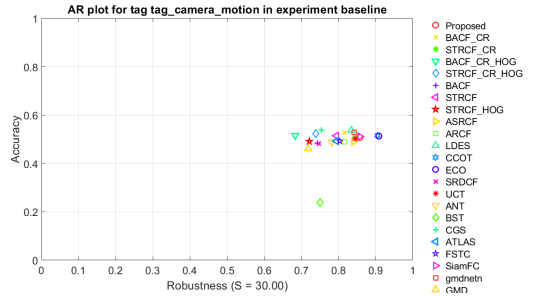
TABLE 19: Robustness comparison of the proposed approaches with recent trackers for the baseline experiment on VOT-2017 dataset.



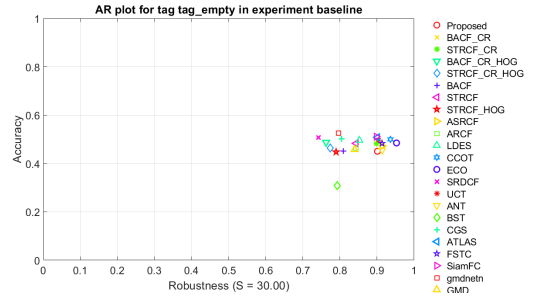
(a) AR plot - Mean



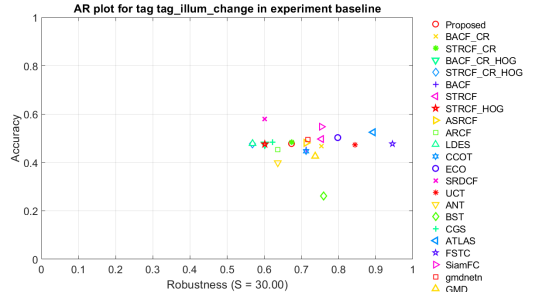
(b) AR plot - Weighted Mean



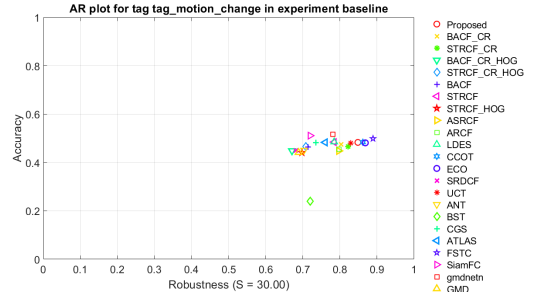
(c) AR plot - Camera Motion (CM)



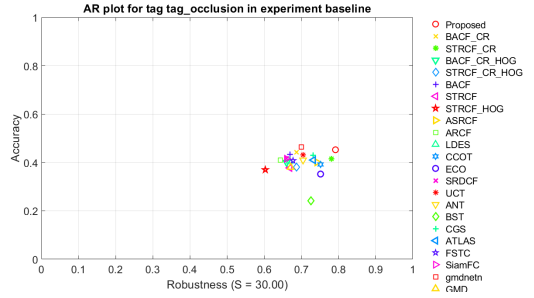
(d) AR plot - Empty Tag (EMP)



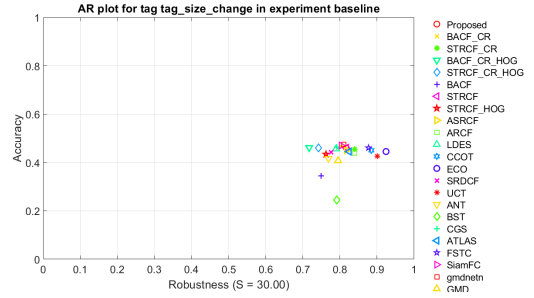
(e) AR plot - Illumination Variance (IV)



(f) AR plot - Motion Change (MC)

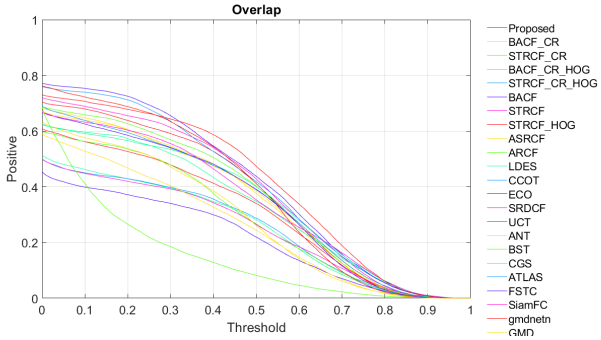


(g) AR plot - Occlusion (OCC)

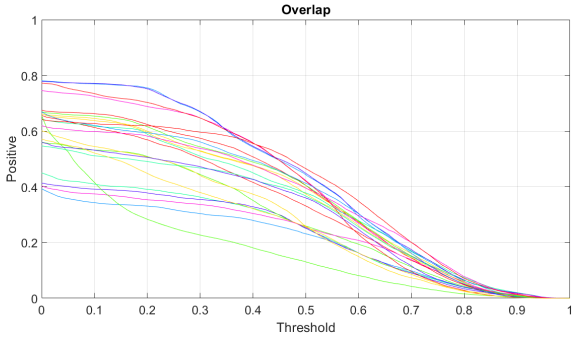


(h) AR plot - Size Change (SC)

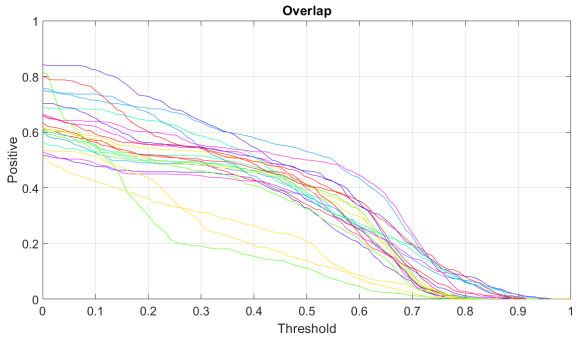
FIGURE 17: AR Plots of individual challenges for baseline experiments



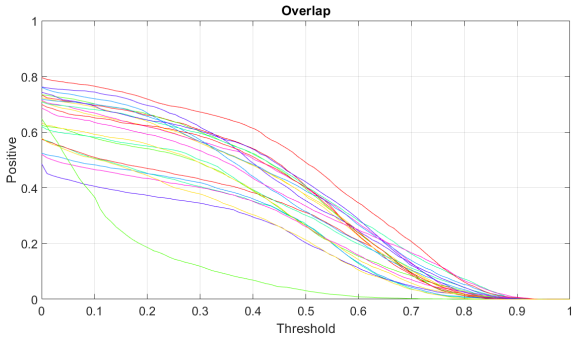
(a) Unsupervised - Average



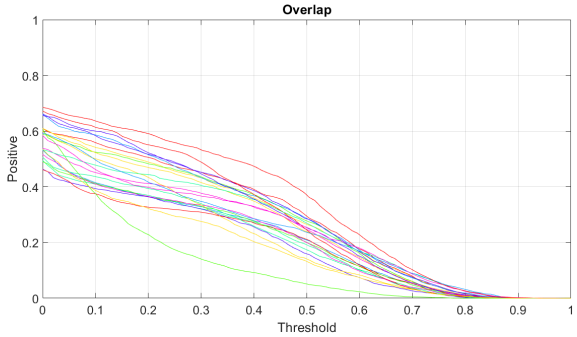
(b) Unsupervised - Empty Tag (EMP)



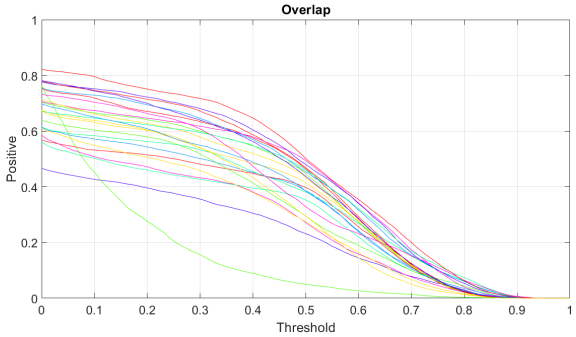
(c) Unsupervised - Illumination Variance (IV)



(d) Unsupervised - Motion Change (MC)



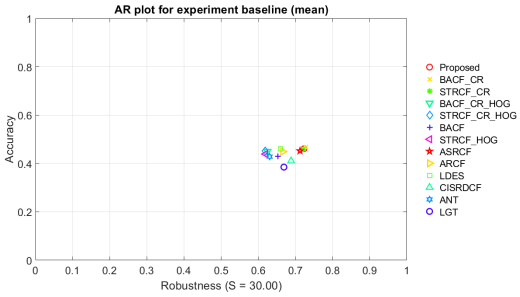
(e) Unsupervised - Occlusion (OCC)



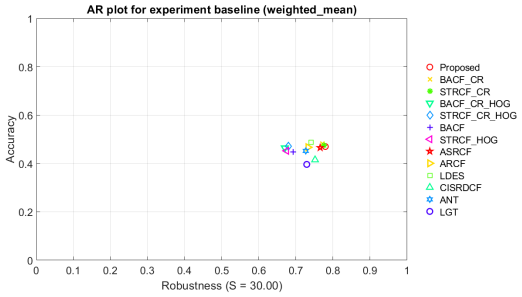
(f) Unsupervised - Camera Motion (CM)

FIGURE 18: Overlap Curves of individual challenges for unsupervised experiments on VOT17

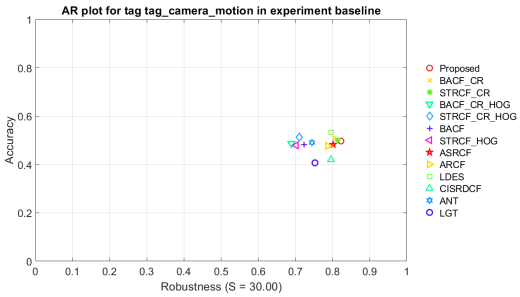
1.3 Results on VOT2019 dataset



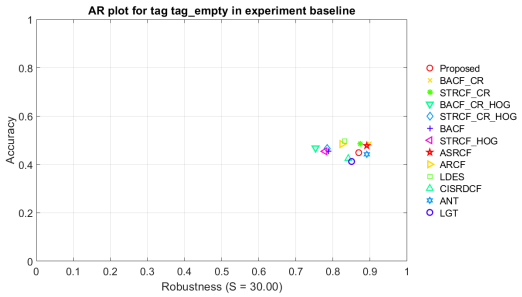
(a) AR plot - Mean



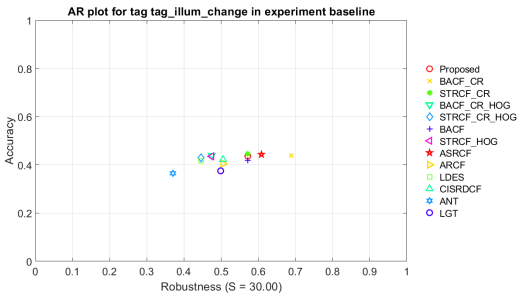
(b) AR plot - Weighted Mean



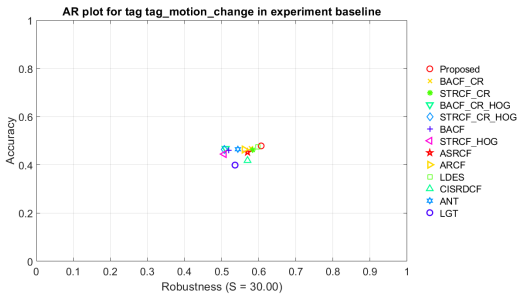
(c) AR plot - Camera Motion (CM)



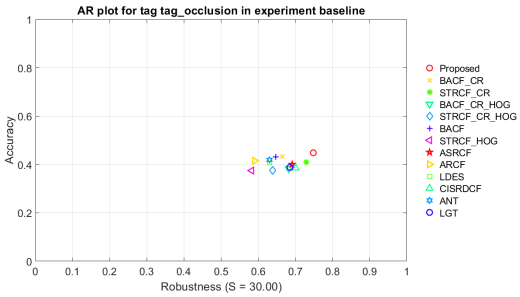
(d) AR plot - Empty tag (EMP)



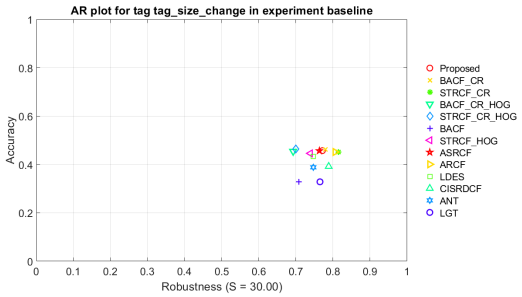
(e) AR plot - Illumination Variance (IV)



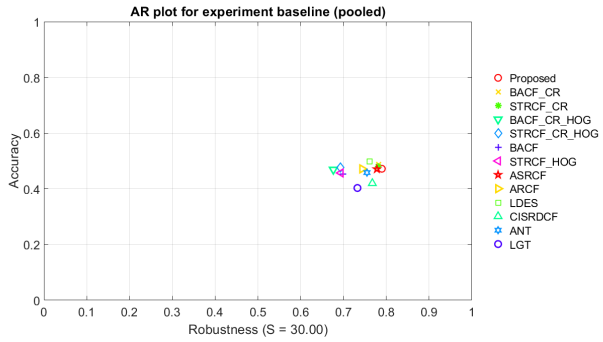
(f) AR plot - Motion Change (MC)



(g) AR plot - Occlusion (OCC)



(h) AR plot - Size Change (SC)



(i) AR plot - Pooled

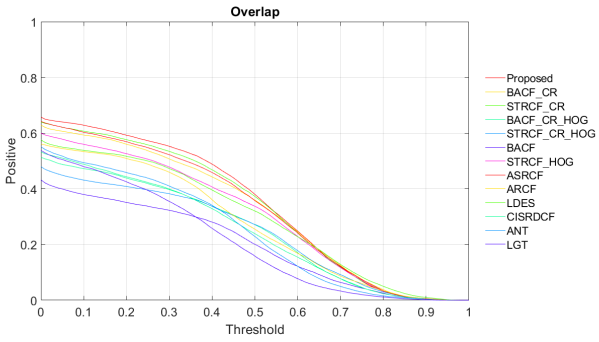
FIGURE 19: AR Plots of individual challenges for baseline experiments

	CM	EMP	IV	MC	OCC	SC	Mean	Weighted Mean	Average Pooled
Proposed (CGRCF)	0.4967	0.4487	0.4362	**0.4788**	**0.4483**	**0.4571**	**0.4610**	0.4698	0.4719
BACF-CR	**0.5106**	**0.4851**	**0.4391**	**0.4689**	**0.4333**	**0.4624**	**0.4666**	**0.4828**	**0.4868**
STRCF-CR	0.4997	**0.4853**	**0.4459**	0.4615	0.4089	0.4505	0.4586	**0.4746**	**0.4800**
BACF-CR-HOG	0.4866	0.4680	0.4388	0.4675	0.3828	0.4542	0.4496	0.4641	0.4690
STRCF-CR-HOG	**0.5121**	0.4650	0.4302	0.4634	0.3748	**0.4639**	0.4516	0.4711	0.4763
Correlation Filter based and Hybrid Trackers									
BACF [8]	0.4821	0.4546	0.4183	0.4607	**0.4306**	0.3290	0.4292	0.4476	0.4533
STRCF-HOG [10]	0.4789	0.4543	0.4353	0.4447	0.3746	0.4458	0.4389	0.4523	0.4570
ASRCF [7]	0.4825	0.4777	**0.4432**	0.4524	0.4003	0.4565	0.4521	0.4652	0.4707
ARCF [12]	0.4762	0.4842	0.4050	0.4637	0.4151	0.4515	0.4493	0.4669	0.4716
LDES [57]	**0.5321**	**0.4972**	0.4177	**0.4745**	0.4071	0.4330	**0.4603**	**0.4882**	**0.4986**
CISRDCF [63]	0.4201	0.4243	0.4224	0.4181	0.3853	0.3919	0.4104	0.4147	0.4198
ANT [63]	0.4906	0.4419	0.3650	0.4648	0.4187	0.3881	0.4282	0.4518	0.4581
LGT [63]	0.4067	0.4121	0.3752	0.3991	0.3879	0.3282	0.3849	0.3960	0.4030

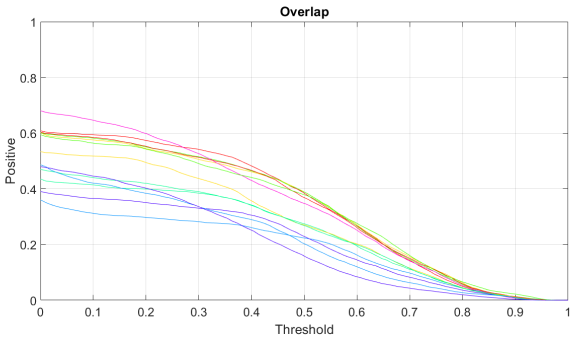
TABLE 20: Accuracy comparison of the proposed approaches with recent trackers for the baseline experiment on VOT-2019 dataset.

	CM	EM	IV	MC	OCC	SC	Mean	Weighted Mean	Average Pooled
Proposed (CGRCF)	**53.00**	34.00	**9.00**	**64.00**	**22.00**	24.00	**34.33**	**42.16**	**157.00**
BACF-CR	**59.00**	**26.00**	**6.00**	70.00	31.00	23.00	**35.83**	**43.36**	**163.00**
STRCF-CR	**56.00**	**33.00**	**9.00**	**69.00**	**24.00**	**19.00**	**35.00**	**43.25**	**164.00**
BACF-CR-HOG	101.00	69.00	12.00	86.00	29.00	34.00	55.16	73.43	260.00
STRCF-CR-HOG	93.00	59.00	13.00	87.00	34.00	33.00	53.1667	68.38	244.00
Correlation Filter based and Hybrid Trackers									
BACF [8]	88.00	58.00	**9.00**	84.00	33.00	32.00	50.66	65.70	238.00
STRCF-HOG [10]	96.00	61.00	12.00	87.00	41.00	28.00	54.16	70.02	243.00
ASRCF [7]	60.00	**28.00**	**8.00**	72.00	28.00	25.00	36.83	44.58	167.00
ARCF [12]	65.00	47.00	11.00	74.00	40.00	**20.00**	42.83	52.71	197.00
LDES [57]	62.00	45.00	13.00	**66.00**	35.00	27.00	41.33	50.27	182.00
CISRDCF [63]	62.00	42.00	11.00	72.00	**27.00**	**22.00**	39.3333	48.98	176.00
ANT [63]	80.00	**28.00**	16.00	78.00	35.00	27.00	44.00	53.09	187.00
LGT [63]	77.11	39.44	11.20	79.80	28.65	24.80	43.50	54.86	206.85

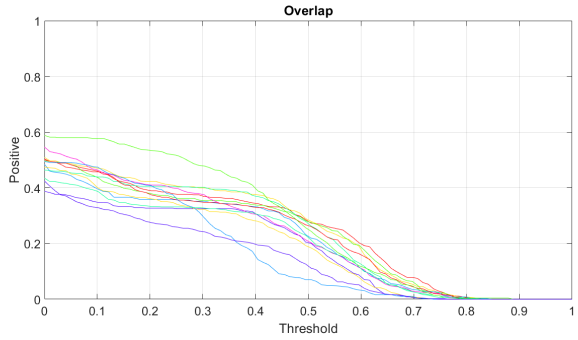
TABLE 21: Robustness comparison of the proposed approaches with recent trackers for the baseline experiment on VOT-2019 dataset.



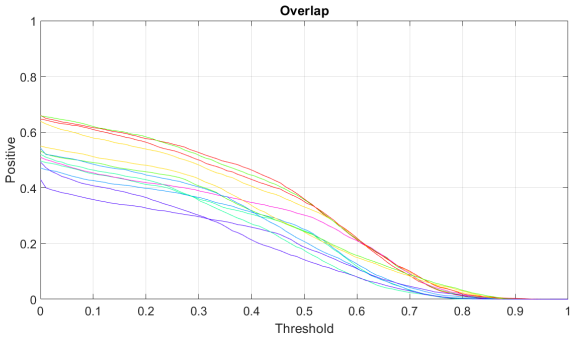
(a) Unsupervised - Average



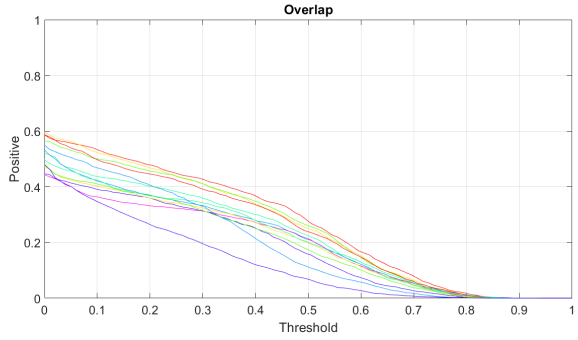
(b) Unsupervised - Empty Tag (EMP)



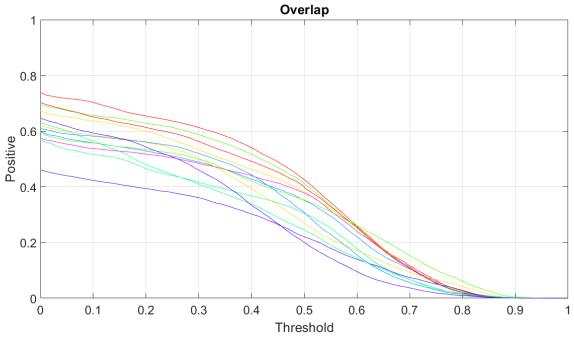
(c) Unsupervised - Illumination Variance (IV)



(d) Unsupervised - Motion Change (MC)



(e) Unsupervised - Occlusion (OCC)



(f) Unsupervised - Camera Motion (CM)

FIGURE 20: Overlap Curves of individual challenges for unsupervised experiments on VOT19

1.4 Results summary

Previous sections show the evaluation of trackers on three different datasets. [29–31]. There are 11 challenges in the TC128 dataset and we used 15 trackers for the comparison where in four of the challenges, Proposed tracker is able to achieve second rank. On the VOT-2017 dataset, the tracker is able to secure top three rank in the five different comparisons on the basis of robustness. On the VOT-2019 dataset, the tracker achieves top three rank on the six different challenges on the robustness. Overall, the tracker gives a decent performance in terms of both tracking speed and robustness. It performs better than most of the correlation filter based trackers. We also reformulate three CF tracker by using the channel regularization and from the results we can conclude that regularization has a positive impact on the performance.

1.5 Quantitative Analysis

This section contains the output of the tracker on the actual video frames. We use [7, 8, 10, 19, 35] trackers to show the comparison with the Proposed tracker. From the results it can be clearly seen that our tracker is giving a robust tracking and performing better than state of the art trackers. In some challenges like occlusion, the tracker is not drifting and giving robust results. An example can be seen in the video *Airport* in Figure 22 where from frame 55 to 75 the object is not visible but, the Proposed tracker still manages to track in the 80-th frame. To show a quantitative comparison on the actual video frames, we use three videos which contain five different challenges i.e. SV, OCC, FM, OPR, IPR. The different trackers predicted bounding boxes are shown by the different colors as shown in the Figure 21.

FIGURE 21: Legend for the bounding boxes

(a) Video Frame - 55

(b) Video Frame - 60

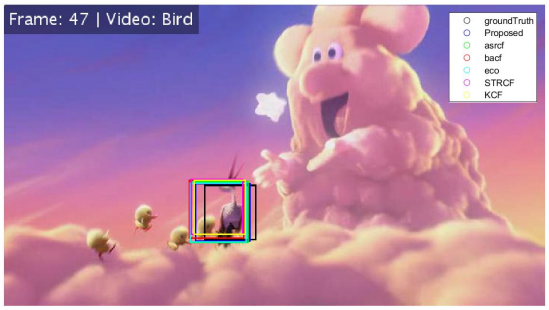
(c) Video Frame - 65

(d) Video Frame - 70

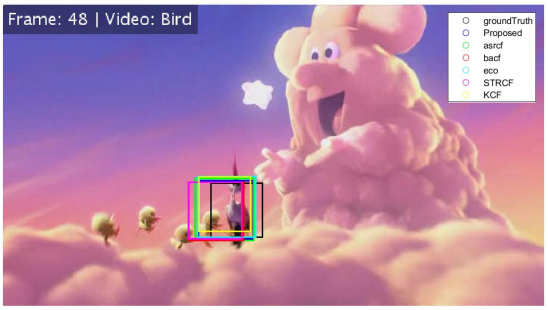
(e) Video Frame - 75

(f) Video Frame - 80

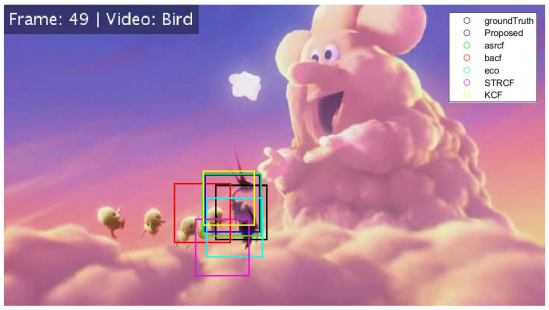
FIGURE 22: Video name: *Airport* ; Challenges: SV, OCC



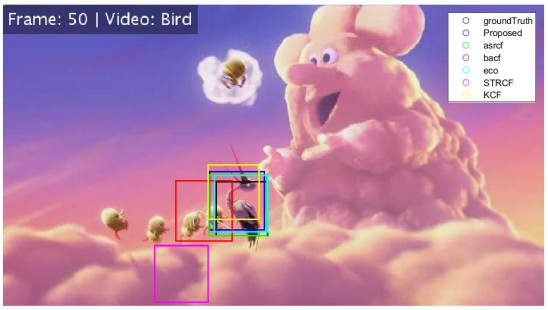
(a) Video Frame - 47



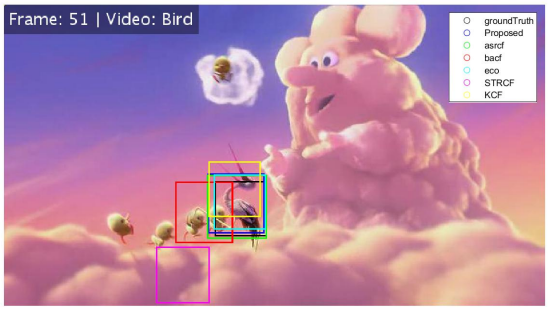
(b) Video Frame - 48



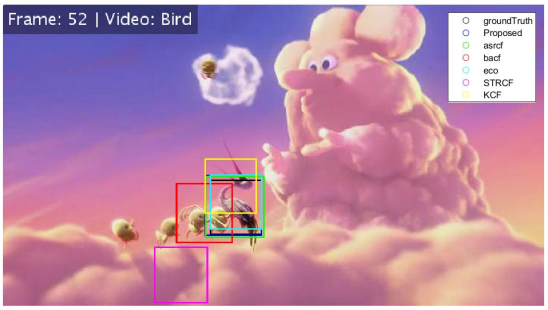
(c) Video Frame - 49



(d) Video Frame - 50

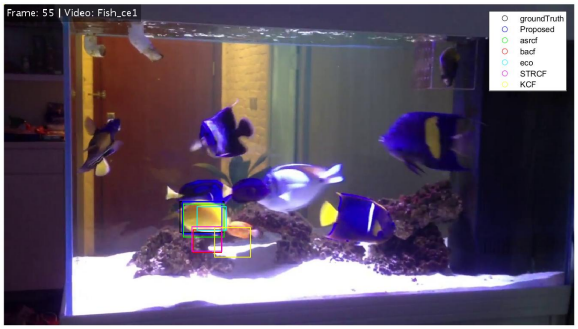


(e) Video Frame - 51

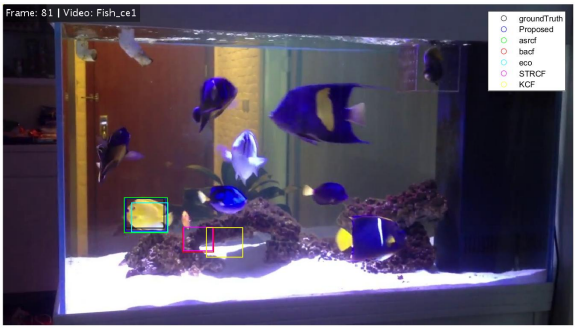


(f) Video Frame - 52

FIGURE 23: Video name: *Bird* ; Challenges: OCC,FM,OPR

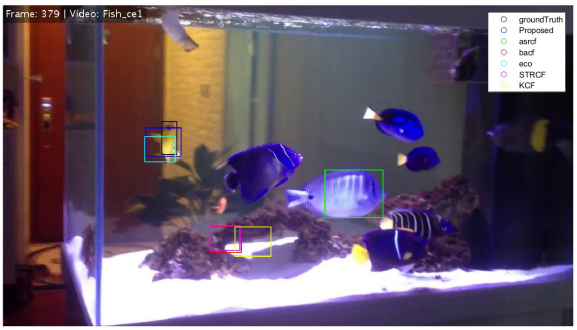


(a) Video Frame - 55

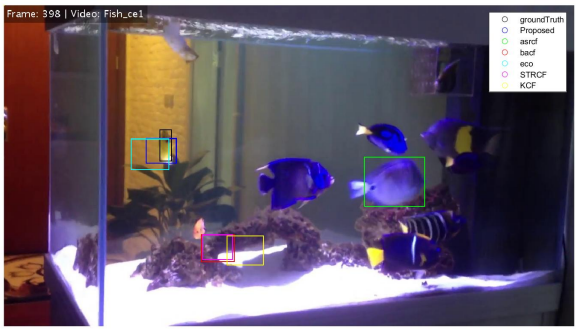


(b) Video Frame - 81

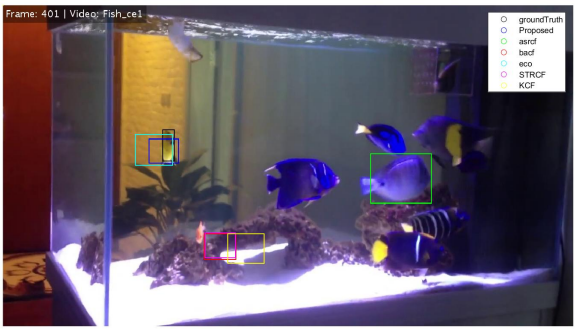
(c) Video Frame - 332



(d) Video Frame - 379



(e) Video Frame - 398



(f) Video Frame - 401

FIGURE 24: Video name: *Fish* ; Challenges: OCC,IPR,OPR,SV