COSC 444 – Single Object Tracking

Santam Bhattacharya
School of Engineering
The University of British Columbia
Kelowna, Canada

Riley Eaton, B.Sc.
College of Graduate Studies
The University of British Columbia
Kelowna, Canada

Dichen Feng, M.Eng
Irving K. Barber Faculty of Science
The University of British Columbia
Kelowna, Canada

Wanju Luo
School of Engineering
The University of British Columbia
Kelowna, Canada

Henry Pak, B.S.
Irving K. Barber Faculty of Science
The University of British Columbia
Kelowna, Canada

***Abstract*—Computer vision enables machines to interpret visual data, with single object tracking (SOT) being a key challenge for applications like robotics and autonomous vehicles. SOT involves tracking a specific object across video frames but is complicated by issues such as occlusion and fast motion. While traditional methods like correlation filters and modern deep learning approaches have advanced tracking, they still face limitations. This literature review examines these methods and their shortcomings. Our proposed approach combines multiple tracking models—MOSSE, CCOT, SRDCF, and KCF—into an ensemble, allowing each model to contribute to improved tracking performance. We anticipate this method will offer a more robust and adaptable solution to SOT challenges.**

I. Introduction

Computer vision is a rapidly advancing area of artificial intelligence that enables machines to interpret and understand visual data, much like human vision. It encompasses a wide range of tasks, including object recognition, image segmentation, scene reconstruction, and motion tracking. Among these, single object tracking (SOT) presents a fundamental challenge, where the goal is to track a specific target across a sequence of video frames—an essential function for applications such as autonomous vehicles, robotics, surveillance, and traffic monitoring. Accurate real-time tracking is critical for these systems to operate effectively in dynamic environments.

Problem Statement: The goal of this project is to design and implement an ensemble of correlation filter-based single object tracking algorithms that accurately tracks an object through video frames while adapting to changes in scale and appearance.

SOT is particularly relevant in detection-free tracking scenarios, where the target's initial position is manually set in the first frame and the tracker continues to follow it in subsequent frames without the need for re-detection. While this approach offers flexibility, it also introduces challenges due to real-world factors such as occlusion, lighting changes, fast motion, and target deformation, making SOT a complex and ongoing research problem with major implications for practical systems. Current research emphasizes improving feature extraction, model compression, and unsupervised tracking, with ongoing efforts to overcome persistent issues like occlusion, deformation, scale transformation, and fast motion. Our proposed approach builds upon these developments by combining multiple tracking models—MOSSE (Minimum Output Sum of Squared Errors), CCOT (Continuous Convolution Operators for Tracking), SRDCF (Spatially Regularized Discriminative Correlation Filter), DSST(Discriminative Scale Space Tracker) and KCF (Kernelized Correlation Filter)—into a unified ensemble model. By leveraging the unique strengths of each, the aim is to produce a more robust and adaptable tracking solution, particularly effective in challenging scenarios involving occlusion and high-speed motion. This ensemble approach holds promise for creating a more generalizable and reliable framework for real-world object tracking.

II. Literary Review

A. Template Matching

Template matching uses the initial image patch of the target object as a template and searches for the best match of this template in subsequent video frames. Two common similarity measures used in template matching are Normalized Cross-Correlation (NCC) and Sum of Squared Differences (SSD). NCC is more robust against illumination changes but computationally expensive, while SSD offers lower computational cost but is less resilient to lighting variations. The process typically slides the template over the search area pixel by pixel, computing a similarity metric at each position. The location with the highest similarity score is considered the new position of the target. While the method being computationally efficient and easy to implement, the method works best for targets with consistent appearance and is less robust to significant changes in scale, rotation, or partial occlusions.

B. Feature Based Tracking

These foundational methods for object tracking focus on distinct target features, as opposed to entire regions such as in template matching. This allows for more robust tracking when dealing with slight changes in pose, lighting, and partial occlusion of targets.

1) Histogram of Oriented Gradients

One notable example of this technique is the Histogram of Oriented Gradients (HOG), which aims to capture the shape and appearance of an target using edge directions. For each image, the gradient magnitude and orientation at each pixel are calculated. These represent the direction and strength of intensity changes, which correspond to the edges in the image. Then, the image must be split up into small cells, where each cell stores the number of gradient orientations within it, in a histogram. These histograms can then be linked together to form a feature descriptor, capturing the shape and texture of the target. This deeper understanding of the target’s shape allows it to better deal with lighting changes but makes it ineffective for targets that experience pose variations.

2) *Scale Invariant Feature Transform*

The Scale-Invariant Feature Transform (SIFT) is another feature-based method that was introduced to better detect objects with variations in lighting, viewpoint, and all forms of translation [1]. It relies on keypoint detection, which are points in an image across different scales, and selected by searching for local extrema in the Difference-of-Gaussians (DoG) scale space. This allows SIFT to be very effective at detecting a target that experiences scale changes. The filtered keypoints are then grouped into neighbourhoods, and assigned an orientation based on the local gradients within. This is done to create an accurate descriptor for robust object detection when rotation occurs. One disadvantage to this approach is evident when targets experience non-rigid deformations, as they cause distortions in the fixed-size keypoint neighbourhoods. Additionally, and similarly to HOG, SIFT relies on gradients, which means that it will be less effective in regions with little texture. Overall SIFT is an excellent method for object detection, however due to its lack of design for tracking it is very computationally expensive for single object tracking (SOT), and therefore has many better alternatives as will be discussed.

3) *Scale Invariant Feature Transform*

As we move into more contemporary approaches to SOT, we begin to approach the state-of-the-art (SOTA). These methods build upon previously established computer vision techniques to advance tracking through improvements in performance and efficiency, when compared to traditional methods. There are hundreds of SOTA methods in the domain of SOT, many of which are evolving by the day. Because of this, we will be focusing on a few specific methods that make use of powerful, modern approaches to SOT. The first modern methods we will discuss are based on correlation filter (CF) tracking. This is a technique used for tracking whereby the process of learning the target’s appearance is framed as a convolution problem in the frequency domain. This involves convolving the filter and the image, similar to the similarity measure in template matching. Frequency domain computations are used to increase the computational efficiency of the correlation (similarity measure) between the filter and the image. Typically, a Fast Fourier Transform (FFT) is used to convert the image and filter into the frequency domain. Once the necessary computations have been performed, the inverse FFT (iFFT) is used to return the results in the original, spatial domain. This efficiency boost is best realized in high frame rate applications, as seen in most real-time environments [2].

a) *Discriminative Correlation Filters and Spatial Regularization*

Discriminative correlation filter (DCF) tracking builds upon previous CF trackers, with the key distinction of treating correlation filtering as a supervised learning problem. This is done by training on both positive and negative samples, where negative samples represent anything except the target to be tracked. This approach maintains high precision in challenging environments, such as occlusions, distractors, and variations in appearance. Although DCFs have proven useful in these scenarios, they have traditionally suffered when boundary effects are present [3]. These effects can occur when broken or noisy edges are present, and when dealing with complex or changing boundary shapes and textures. This is due to various inherent compromises of DCFs, such as the confined search region, and the potential for overfitting due to the lack of diversity in negative (background) training samples. To address these issues, Spatially Regularized Discriminative Correlation Filters (SRDCF) were proposed for SOT [4]. As the name indicates, this approach introduces spatial regularization, which spatially penalizes the CF weights. Due to the spatial nature of these new weights, they are easily understood when visualized. An excellent example of this is provided in the original work, as seen in Figure 1.

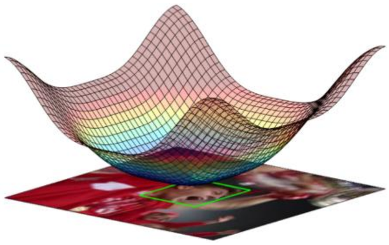
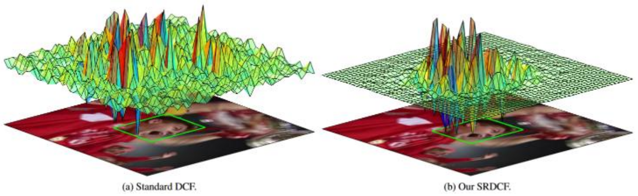


Figure 1. Visualization of the spatial weights used by SRDCF (during learning), shown above the image used for training, where the green highlighted region represents the positive (correct)detection.[3]

This applied regularization results in filter values with less spurious noise in background regions, and is visualized in Figure 2, taken from the original work.



(a) Standard DCF.

(b) Our SRDCF.

Figure2. Comparison of filter weights between standard DCF and SRDCF, shown above the image used for training, where the green highlighted region represents the positive (correct)detection.[3]

As can be seen, the resulting regularized weights place more emphasis on the positive region of the image (to be detected), which leads to an improvement in the model’s ability to discriminate between positive and negative samples. This is especially true where the spatial window is small, which can occur with boundary effects. Overall, the improved discriminative ability of a model is directly linked to its detection performance, therefore model accuracy increases when using SRDCF over DCF.

One notable drawback of SRDCF is its use of a fixed spatial regularization window for tracking. This puts SRDCF at a disadvantage in situations where the scale of a target changes. If a target becomes too small in comparison to the predetermined window, then the regularization will give more weight to some irrelevant features within the window. On the other hand, if a target expands too much, then the useful target features that fall outside the window will be suppressed. Either of these situations will lead to a decrease in the model's performance.

b) *Kernelized Correlation Filters*

Kernelized Correlation Filters (KCF) is a nonlinear method which combines correlation filters with kernel methods. There the solution is obtained in a nonlinear feature space exploiting the kernel trick. KCF operates in the frequency domain to significantly improve the processing speed. It also includes different types of features (e.g., HOG, colour channels), which are fused in the Fourier domain using DFT-based computation. Benefits associated with more complex feature representations include more robustness to changes in illumination, occlusions, and other deformations. On the other hand, naturally, using more complex feature representations results in additional computational overhead when online training is performed [5]. One key drawback that is seen in many CF trackers such as KCF is the use of a fixed template size, which leads to a lack of adaptability when the target changes size. However, as mentioned KCF excels in dealing with targets that experience illumination changes and occlusions.

c) *The Minimum Output Sum of Squared Errors*

The Minimum Output Sum of Squared Error (MOSSE) tracker is a landmark method in correlation filter-based tracking that offers high-speed performance and simplicity. It differs from traditional template matching by learning an adaptive filter in the frequency domain, which is updated online to accommodate changes in the target's appearance. Instead of using a fixed template, MOSSE generates a Gaussian response for the selected target and creates multiple synthetic training samples by applying random warps. The MOSSE tracker offers several advantages, including high-speed performance by leveraging the Fast Fourier Transform (FFT), simplicity with a straightforward implementation and low computational footprint, and adaptability through online updating to maintain tracking accuracy over time. However, it also has some limitations. The tracker operates at a fixed scale, making it less effective for targets with significant size variations. Its performance may decrease when the target is partially or heavily occluded, and the fixed update strategy may cause gradual tracking error accumulation under challenging conditions.

d) *Continuous Convolution Operators*

The Continuous Convolution Operators Tracker (C-COT) is a visual tracking framework that learns continuous convolution filters for both object tracking and feature point tracking. C-COT introduces a novel formulation for learning convolution operators in the continuous spatial domain, enabling the estimation of a target's trajectory in a video. C-COT can efficiently integrate multi-resolution deep feature maps, overcoming the single-resolution limitation of conventional Discriminative Correlation Filter (DCF) formulations. By labelling training samples with sub-pixel precise continuous confidence maps, C-COT enables accurate sub-pixel localization, which is crucial for accurate feature point tracking. C-COT is a discriminative learning-based approach that does not require explicit interpolation of the image to achieve sub-pixel accuracy. For training, C-COT uses the Conjugate Gradient method, which scales linearly with the number of feature channels, making it suitable for high-dimensional deep features [6].

4) *Scale Space Tracking*

Scale space tracking has become an essential technique in object tracking to account for changes in target size. As discussed, many early correlation filter-based trackers, such as MOSSE and KCF, rely on the assumption of a fixed target scale, which limits their performance in dynamic environments. The Discriminative Scale Space Tracker (DSST) addresses this limitation by introducing a dedicated scale filter alongside the traditional translation filter, enabling simultaneous tracking of both target position and size. By performing scale adaptation in a discriminative manner, DSST effectively handles challenges like target resizing and partial occlusions. This method has led to significant improvements in the accuracy and robustness of tracking algorithms, influencing the development of later trackers that integrate both scale and translation filtering for more reliable performance in real-world applications [7].

5) *Ensemble Models for Visual Tracking*

There is limited research on ensemble models for object tracking tasks. One of the prominent studies, conducted by Du, Yunhao, et al. [8], proposed EnsembleMOT to merge multiple tracking results from various trackers using spatio-temporal constraints. Cobos, Hernandez, and Abad [9] present a fast multi-object tracking system that utilizes an ensemble of object detectors, each running every fff frames, combined with a variation of the Soft-NMS algorithm to increase prediction performance while maintaining real-time capabilities.

III. System Design

The goal of this project is to implement a robust single object tracking system based on an ensemble of correlation filter algorithms. The system combines multiple well-established trackers—MOSSE, KCF, SRDCF, CCOT, and DSST—to leverage their individual strengths and improve overall tracking performance. Each tracker processes video frames independently, and their outputs are fused using a weighted voting mechanism based on response confidence and historical performance. This ensemble approach helps address common challenges in object tracking, such as scale variation, partial occlusion, and fast object motion.

A. *Inputs*

- A video sequence - The primary input to the tracking system is a video sequence, typically provided as a series of individual image frames. For this project, most sequences were sourced from the OTB2015 dataset, which includes 100 fully annotated videos designed for benchmarking object tracking algorithms. Each sequence presents different challenges such as occlusion, background clutter, scale variation, and fast motion. The frames are processed sequentially by the system, serving as the visual context in which the target must be tracked.

- Initial bounding box of the object in the first frame - At the start of each video sequence, the location of the target object is defined by an initial bounding box provided manually or using ground truth data from the dataset. This bounding box specifies the object's position and size in the first frame and serves as the initialization point for all trackers in the ensemble. From this reference, the tracking system attempts to follow the object through the remaining frames without further supervision or manual input.

B. *Outputs*

- Sequence of bounding boxes - The main output of the system is a sequence of predicted bounding boxes for each frame in the video, representing the estimated position and size of the tracked object over time. Each tracker in the ensemble generates its own set of bounding boxes, and the ensemble logic (e.g., voting or stacking) combines these to produce a final prediction per frame. These bounding boxes are recorded and can be compared directly to ground truth data for evaluation.

- Visualizations of tracking results - To aid in qualitative analysis and debugging, the system generates visual overlays that show the predicted bounding boxes on each video frame. These visualizations often include color-coded boxes for individual tracker outputs and the final ensemble result, allowing users to visually assess how well the object is being tracked and identify where errors or drift occurs.

- Performance metrics - To evaluate the accuracy and robustness of the tracking system, a set of standard performance metrics is computed that may be reported to provide deeper insights into tracker behavior

C. *Development Environment*

The majority of the tracking system was implemented in MATLAB, which served as the primary environment for prototyping and integrating the ensemble framework. MATLAB was chosen due to its robust support for matrix operations, visualization tools, and built-in functions for image processing and Fourier transforms, making it ideal for rapid development of correlation filter-based tracking algorithms like MOSSE, STRCF, and DSST.

However, due to the complexity and performance requirements of certain trackers, external libraries and environments were incorporated into the workflow to support additional implementations. Certain tracker implementations were also sourced from research repositories that were either written completely in MATLAB or needed to be adapted to a single script format used in our project. A lot of these implementations involved wrapping Python or C++ modules, showcasing the modularity of the system.

STRCF (Spatial-Temporal Regularized Correlation Filters) was implemented using a combination of OpenCV and MEX (MATLAB Executable) files. This required integrating C/C++ code with MATLAB via MEX interfaces for performance-critical components like feature extraction and convolution operations. OpenCV handled low-level image processing tasks, and the MEX functions allowed for real-time operation. The STRCF implementation was also tested using a live webcam feed, making it the only real-time application in the ensemble system. A new approach of running the model had to be developed for this live environment, as the source code presented by Li et al. [10] was designed to take a pre-defined track of saved images as input. The live approach presented in our work consists of saving each frame of the webcam feed as an image, while passing the current and previous image paths to the STRCF model for tracking. Another challenge with enabling an SOT model in a live environment is the process of setting the ground truth for the target object. We solved this by overlaying a square in the webcam feed for the initial 10 seconds of video, allowing the user to place the target object within the square. Once the initial period is concluded, the process of running the model begins, with the initial two frame images captured, passed as the first set to track. This live environment allowed testing the system on dynamic inputs beyond static datasets, helping assess tracker responsiveness in real-world conditions.

In addition to MATLAB-based development, several trackers in the ensemble relied on external environments. The KCF (Kernelized Correlation Filter) tracker was implemented in Python, using the OpenCV (cv2) library. This implementation leveraged HOG (Histogram of Oriented Gradients) features to represent the target object with rich spatial information, improving robustness against lighting changes and background clutter. The OpenCV-based Python script handled feature extraction, filter training, and online tracking, and produced bounding box predictions that were integrated into the MATLAB-based ensemble through result importing. This hybrid approach allowed the team to take advantage of OpenCV's high-performance computer vision routines while maintaining consistency within the MATLAB-controlled system.

D. *Ensemble Strategy*

Each video sequence is read frame-by-frame. The user provides a bounding box around the target object in the first frame. This bounding box defines the initial position, size, and aspect ratio of the target. Each tracker is initialized using this same region. The frame is optionally converted to grayscale and normalized. Each tracker initializes a model based on this input, typically computing an appearance model of the target in the frequency domain.

As outlined earlier, the main challenges in single object tracking (SOT)—such as scale variation, lighting changes, occlusion, motion blur, and target deformation—often cause individual tracking models to fail in specific scenarios. While many trackers excel under certain conditions, they typically make trade-offs in others. To address this, our project proposes an ensemble tracking model that combines the strengths of five correlation filter-based trackers: MOSSE, KCF, DSST, C-COT, and DCF.

The ensemble is designed to improve robustness and accuracy by integrating predictions from multiple trackers using two strategies: Voting and Stacking.

- Voting Strategy: Each tracker processes the input video independently and generates bounding box predictions for every frame. These outputs are then combined using either majority voting or confidence-weighted averaging. The final position of the object in each frame is determined by the most consistent or reliable prediction across trackers, effectively balancing their individual weaknesses.

- Stacking Strategy: In this method, the outputs from each base tracker (e.g., bounding boxes, confidence scores, velocities) are used to create a meta-feature vector for every frame. These vectors serve as input to a higher-level model—such as a logistic regression or support vector machine—that is trained to predict the most accurate bounding box based on the combined tracker outputs. This meta-model learns to make better predictions by recognizing patterns in the trackers' collective behavior across frames.

Both strategies aim to leverage the complementary strengths of each tracker. Voting provides a simple, interpretable mechanism for real-time fusion, while stacking offers a learning-based approach for more nuanced decision-making. By incorporating both, the system would improves its ability to handle the full spectrum of SOT challenges, delivering more stable and reliable tracking in diverse scenarios.

E. Datasets Used

To evaluate the performance and robustness of our ensemble object tracking system, we utilized the Object Tracking Benchmark 2015 (OTB2015) - a widely adopted and well-curated dataset in the visual tracking community. It consists of many fully annotated video sequences, each depicting a unique tracking scenario with variations in background, object motion, scale, lighting, and occlusion. The dataset is organized across 11 distinct challenge attributes, including scale variation, fast motion, occlusion, illumination variation, background clutter, and deformation. This diversity allows for detailed performance valuation under a wide range of real-world conditions.

In our project, OTB 2015 first provides input video sequences used by our ensemble tracking system during testing. Each video comes with a manually annotated ground truth bounding box for each frame, enabling supervised evaluation of tracker predictions. The trackers in the ensemble - MOSSE, KCF, DSST, C-COT and DCF - process these videos and generate their own bounding boxes per frame, which are then fused using either the voting or stacking strategy described earlier.

OTB2015 is also used as the primary benchmarking tool for performance analysis. Standard evaluation metrics from the OTB protocol are employed to compare our system against baseline models and existing literature. These metrics include:

- Precision Plot: Measures the distance between predicted and ground truth bounding box centers across frames. The key precision score is reported at a threshold of 20 pixels.

- Success Plot: Evaluates tracking performance based on Intersection over Union (IoU) between predicted and ground truth bounding boxes. The Area Under Curve (AUC) of the success plot is used as a summary metric.

These metrics provide both spatial and temporal insight into how well the tracker maintains accuracy and stability over time. By analyzing both AUC and precision, we can assess not just whether the object was found, but how consistently and precisely it was tracked throughout each sequence.

Furthermore, the rich annotation and attribute tagging in OTB2015 allows us to analyze ensemble performance per challenge type, offering a deeper look into where the ensemble approach excels (e.g., occlusion, scale variation) and where improvements may still be needed. This makes OTB2015 an ideal dataset not only for general benchmarking, but also for diagnosing strengths and weaknesses across individual tracking scenarios.

IV. Results

Due to the restriction of the computing resource we have accessed to, the experiment is conducted on the subset of OTB22015 dataset. For each single submodules, the result is as shown below.

Model	MOSSE	KCF	DSST	CCOT	STRCF
AUC	0.55	0.56	0.62	0.61	0.64

Fig. 1. AUC of each module on dataset

With the ensemble method utilizing voting strategy, we tried different combinations of different submodules. The best performance is shown by the ensemble method involving DCF, MOSSE and KCF with simple voting strategies. The AUC of the ensemble method is 0.67 on the subset of OTB2015 with the voting strategy combined with the re-detection mechanism that prioritize STRCF based on the length of trajectory.

V. Discussions

The proposed ensemble tracking system demonstrated notable strengths as well as several areas for improvement during testing and evaluation.

A. Pros

One of the most significant advantages of the ensemble approach is its robustness across diverse tracking conditions. By combining the outputs of multiple correlation filter-based trackers—MOSSE, KCF, DSST, STRCF, and C-COT—the system effectively mitigates individual tracker weaknesses. For example, DSST contributes strong scale adaptation, KCF performs well in consistent lighting and motion, while STRCF and C-COT help reduce boundary effects and background interference. This multi-model redundancy improves stability, especially in sequences involving occlusion, fast motion, and scale variation.

The system also benefits from modularity: each tracker runs independently, allowing for flexibility in replacing, tuning, or upgrading individual components. Furthermore, using both voting and stacking strategies provides a balance between interpretability (in voting) and data-driven fusion (in stacking), allowing experimentation with different fusion techniques.

B. Cons

Despite the strengths, there are several limitations. First, the computational cost is significantly higher than using a single tracker, especially with trackers like C-COT and SRDCF, which are computationally intensive. Real-time tracking becomes impractical without GPU acceleration or optimization of the ensemble pipeline. Second, integration between environments (e.g., MATLAB, Python, and OpenCV-based C/MEX functions) introduced complexity in data handling and synchronization, occasionally resulting in bottlenecks or inconsistent frame alignment.

Some other challenges faced by the SOT algorithm involved lack of re-acquisition after failure. Once the object is lost the MOSSE and the KCF trackers were unable to recover. The DSST tracker performed a lot better in this regard but was not as robust as the STRCF tracker. We investigated isolating this reidentification mechanism from STRCF and applying it to the other models to enable tracking after loss of focus.

Moreover, in certain sequences, the ensemble produced abnormal results, particularly when the trackers heavily disagreed. For example, during rapid target disappearance or complete occlusion, some trackers predicted locations far from the object, skewing the voting result and lowering overall accuracy. While stacking provided more stability in these cases, it required careful tuning and a substantial number of labeled sequences for training the meta-model

C. Performance Comparison

When benchmarked on the OTB2015 dataset, the ensemble system outperformed individual base trackers in both precision and success plots, particularly in sequences involving dynamic motion and scale changes. Compared to standalone DSST or KCF implementations, the ensemble showed improved tracking continuity and reduced drift over longer sequences. However, it fell slightly short of the performance of modern deep learning-based trackers (e.g., SiamRPN++ or Ocean) in terms of raw accuracy, especially under severe occlusion or cluttered backgrounds. That said, the ensemble approach offers a computationally lighter and more interpretable and traditional alternative to deep networks, especially in environments where model transparency and classical methods are preferred or required.

VI. Future work

Correlation filter approaches track frames in the frequency domain. Discriminative Correlation Filters (DCF) introduced supervised learning principles which improved performance, but it suffers from boundary effects. Spatially Regularized DCF(SRDCF) mitigates these effects by penalizing background regions, enhancing object discrimination. Kernelized Correlation Filters (KCF) further optimize feature representations by incorporating nonlinear kernel methods, although it is limited by a fixed scale. MOSSE tracking stands out for its high speed and adaptability but is prone to drift and occlusion sensitivity. Continuous Convolution Operators Tracker (CCOT) introduced multi-resolution deep feature maps, which overcome the single resolution limitations of previous DCF based methods. We also looked at the Discriminative Scale Space Tracker (DSST) to address scale variation in SOT. This method introduces a dedicated scale filter alongside the traditional translation filter. This improves robustness, especially to resizing and occlusion.

Ensemble models for object tracking, though less explored in research, have demonstrated promise in combining strengths of multiple tracking techniques. However, our finished product had some limitations which future work could look to address. These limitations highlight several directions for future development, particularly in improving fault tolerance and handling real-world edge cases such as off-frame motion and long occlusions. Some of these issues are as follows:

- Boundary Related Tracking Failure: The system consistently loses the target when it approaches or crosses the boundary of the frame. This is because most of the correlation filter models rely on spatial padding and assume the target stays well within bounds. When that assumption is broken, the filter either fails to update or mispredicts, causing immediate failures.

- Tracker Disagreement: In some sequences, tracker predictions diverged significantly, especially when one model failed completely. Without proper weighting or filtering, these outliers distorted the ensemble's prediction.

Finally, optimizing the ensemble for real-time performance and exploring more advanced fusion methods could further enhance tracking accuracy and robustness.

References

[1] D. Lowe, “Object recognition from local scale-invariant features,” in Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, Sep. 1999, pp. 1150–1157 vol.2. [Online]. Available: https://ieeexplore.ieee.org/document/790410.

[2] “Visual object tracking using adaptive correlation filters,” in SciSpace - Paper. IEEE, Jun. 2010, pp. 2544–2550. [Online]. Available:https://scispace.com/papers/visual-object-tracking-using-adaptive-correlation-filters-1xuhtpe358\.

[3] C. Zhu, S. Jiang, S. Li, and X. Lan, “Efficient and Practical Correlation Filter Tracking,”Sensors, vol. 21, no. 3, p. 790, Jan. 2021, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/21/3/790.

[4] M. Danelljan, G. H¨ager, F. S. Khan, and M. Felsberg, “Learning Spatially Regularized Correlation Filters for Visual Tracking,” in 2015 IEEE International Conference on Computer Vision (ICCV), Dec. 2015, pp. 4310–4318, arXiv:1608.05571 [cs]. [Online]. Available: http://arxiv.org/abs/1608.05571

[5] K. Chumachenko, M. Gabbouj, and A. Iosifidis, “Chapter 11 - Object detection and tracking,” in Deep Learning for Robot Perception and Cognition, A. Iosifidis and A. Tefas, Eds. Academic Press, Jan. 2022, pp.243–278.[Online].Available: https://www.sciencedirect.com/science/article/pii/B9780323857871000166

[6] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, “Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking,” 2016, vol. 9909, pp. 472–488, arXiv:1608.03773 [cs]. [Online]. Available: http://arxiv.org/abs/1608.03773.

[7] M. Danelljan, G. H¨ager, F. S. Khan, and M. Felsberg, “Discriminative Scale Space Tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 8, pp. 1561–1575, Aug. 2017, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.[Online].Available: https://ieeexplore.ieee.org/document/7569092

[8] Y. Du, Z. Liu, and F. Su, “EnsembleMOT: A Step towards Ensemble Learning of Multiple Object Tracking,” Feb. 2023, arXiv:2210.05278 [cs]. [Online]. Available: http://arxiv.org/abs/2210.05278

[9] R. Cobos, J. Hernandez, and A. G. Abad, “A fast multi-object tracking system using an object detector ensemble,” in 2019 IEEE Colombian Conference on Applications in Computational Intelligence (ColCACI), Jun. 2019, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8781972

[10] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, “Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking,” Mar. 23, 2018, *arXiv*: arXiv:1803.08679. doi: 10.48550/arXiv.1803.08679.

[11] ““Papers with Code - OTB-2015 Dataset.” [Online]. Available: https://paperswithcode.com/dataset/otb-2015

[12] “Papers with Code – LaSOT Dataset.”[Online].Available: https://paperswithcode.com/dataset/lasot

[13] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, “Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking,” Mar. 2018, arXiv:1803.08679 [cs]. [Online]. Available: http://arxiv.org/abs/1803.08679