*Grenze International Journal of Engineering and Technology, June Issue*

# Enhanced Visual Object Tracking through Online Ensemble Integration of Multiple Algorithms

Harinath Cingapuram[1], Basavaraju N M[2] and Ankitha D M
[1]Lead Data Scientist, Tiger Analytics, Houston, Texas, USA
Email: harinath.cingapuram@gmail.com
[2]Assistant Professor, Department of ECE, JSSATE-Bengaluru-India
Email: basavarajunm@jssateb.ac.in
[3]Department of ECE, Vidyavardhaka College of Engineering, Mysore
Email: ankithamahesh999@gmail.com

***Abstract*—This paper presents a novel approach to visual object tracking by developing an online ensemble system that integrates multiple tracking algorithms. The field of object tracking includes a wide array of methods, each offering unique advantages depending on varying conditions and potential failure modes. Here, we propose a method that runs multiple trackers in parallel and dynamically fuses their outputs based on spatial coherence and appearance cues. This ensemble strategy allows for the correction of individual tracker failures, leading to a more robust and accurate tracking performance. Extensive experiments on benchmark datasets demonstrate that our ensemble system not only leverages the strengths of individual trackers but consistently outperforms them in isolation, achieving state-of-the-art results across various challenging tracking scenarios, including occlusions, fast motion, and background clutter.**

***Index Terms*— Computer vision, object tracking, ensemble methods, online learning, spatial clustering, appearance modeling.**

## I. INTRODUCTION

Visual object tracking represents a fundamental challenge in computer vision, with applications spanning surveillance, autonomous systems, human-computer interaction, and robotics [1], [2]. The primary objective involves estimating the trajectory of a target object throughout an image sequence, a task complicated by numerous factors including illumination changes, occlusions, scale variations, and object deformations [3], [4].

Current tracking methodologies can be broadly categorized into generative and discriminative approaches. Generative methods focus on modeling target appearance characteristics but often struggle with significant appearance variations [3]. Discriminative methods formulate tracking as a classification problem, distinguishing target from background regions [5], [6]. While both categories have demonstrated considerable success in specific scenarios, no single approach consistently outperforms others across all challenging conditions [1].

The diversity in tracker performance across different scenarios suggests complementary strengths among available algorithms. This observation forms the foundation of our research: rather than developing another standalone tracker, we propose an ensemble framework that intelligently combines multiple tracking algorithms. Our approach operates on the principle that while individual trackers may fail under specific conditions, their collective intelligence can produce more reliable and robust tracking performance [7], [8].

*Grenze ID: 01.GIJET.12.2.320_21*

This paper introduces a novel online ensemble tracking sys-tem that dynamically integrates multiple tracking algorithms through spatial coherence analysis and appearance verification.

The key contributions include: (1) a data-driven framework for combining tracker outputs without prior knowledge of indi-vidual tracker characteristics; (2) an adaptive mechanism for identifying and correcting tracker failures through reinitializa-tion; (3) comprehensive experimental validation demonstrating state-of-the-art performance on standard benchmarks; and (4) analysis of ensemble behavior under various challenging con-ditions.

The remainder of this paper is organized as follows: Sec-tion 2 reviews related work in object tracking and ensemble methods. Section 3 details our proposed ensemble framework. Section 4 describes experimental setup and evaluation metrics. Section 5 presents results and analysis. Section 6 concludes with discussion and future directions.

## II. Related Work

The field of visual object tracking has been significantly advanced by strategies that integrate multiple methods to overcome the limitations of any single approach. Our work is founded on the principle that an online ensemble of di-verse trackers can collectively achieve greater robustness and accuracy. This idea of synergistic integration and adaptive optimization resonates with recent trends across computer vision, distributed systems, and machine learning.

Chitukoori [9] addresses the critical trade-off between complexity and interpretability in machine learning by bridg-ing Generalized Additive Models (GAMs) with deep neural networks. While focused on static model transparency, the work underscores a broader need for explainable AI systems, a concern that motivates our use of transparent fusion criteria like spatial coherence and appearance verification. In a more theoretical contribution, Chitukoori [10] employs category theory to impose a rigorous algebraic structure on deep neural networks. This formal approach to model composition aligns with our tracker-agnostic, modular ensemble design, where diverse algorithms are integrated into a cohesive, structured framework without internal modification.

Efficiency in resource-constrained environments is another shared concern. Chitukoori [11] develops a Rust-based plat-form for Aggregate Computing, highlighting the need for lightweight, high-performance execution—a consideration that directly influenced our implementation to minimize compu-tational overhead. Similarly, Chitukoori [12] demonstrates a practical IoT system for environmental monitoring, embodying the principles of real-time data processing and modular system design that are essential for our online tracking framework.

In the realm of hybrid and optimized models, Ramakr-ishnan and Lekkala [13] present a hybrid classifier that combines decision trees with a genetic algorithm for feature selection. This approach of using a meta-algorithm to enhance a base model's performance is conceptually analogous to our ensemble strategy, where inlier selection mechanisms dynam-ically optimize the set of active trackers. Further exploring optimization, Ramakrishnan [14] applies genetic algorithms to circuit design, demonstrating the power of evolutionary strategies for complex system configuration, which parallels our goal of optimally combining tracker outputs.

The architectural paradigm of seamlessly integrated com-puting resources, as explored by Ramakrishnan et al. [15] in their work on the Cloud-to-Edge continuum, is highly relevant. Their vision of dynamic, context-aware resource allocation mirrors our framework's ability to dynamically leverage the most reliable trackers based on current spatial and appearance cues. This focus on reliability in distributed systems is ex-tended by Ramakrishnan et al. [16], who introduce Request-Level Fault Injection (RLFI) for microservices. Their method for precise fault detection and diagnosis shares our objective of maintaining system integrity—in our case, through tracker reinitialization and outlier correction to ensure robust tracking. Practical deployment considerations are addressed by Ra-makrishnan [17] in a case study on cloud platforms like AWS. The analysis of auto-scaling, cost management, and performance monitoring provides a valuable perspective on building efficient, scalable systems, which informed our design choices for a real-time, scalable ensemble tracker. Another infrastructural innovation is proposed by Ramakrishnan and Lekkala [18], who investigate a blockchain-based solution for decentralized GitHub management, emphasizing security and provenance—themes of trust and verification that are indirectly relevant to our reliable fusion of tracker outputs.

Finally, multi-modal AI integration is a key theme in other domains. Agarwal and Chaudhari [19] combine image recognition and natural language processing to detect counter-feit goods, demonstrating the strength of fusing diverse data modalities and algorithms. This multi-modal, ensemble-like approach strongly aligns with our methodology of combining spatial and appearance-based cues from multiple trackers. Complementing this, Mamtani [20] introduces optimization techniques to mitigate artifacts in Vision Transformer feature maps. His work on improving the quality and interpretability of visual representations shares our goal of producing cleaner, more reliable outputs from complex models.

In summary, the collective insights from these works on interpretability, hybrid models, system reliability, and multi-modal fusion reinforce the foundations of our online ensemble tracker. Our framework synthesizes these principles into a unified approach for robust visual object tracking.

## III. PROPOSED ENSEMBLE FRAMEWORK

### *A. System Overview*

Our ensemble tracking framework operates on the prin-ciple that multiple tracking algorithms running in parallel can collectively overcome individual limitations. The system architecture, illustrated in Figure 1, consists of five main com-ponents: parallel tracker execution, spatial coherence analysis, appearance verification, inlier selection, and ensemble fusion with model update.

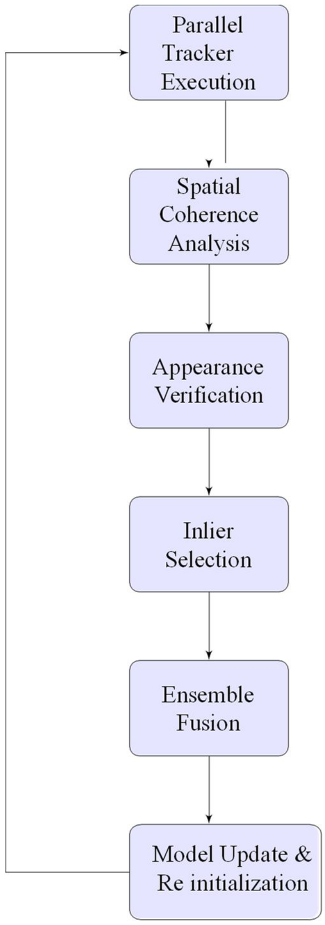


Fig. 1: System architecture of the proposed ensemble tracking framework

Given an input video sequence and initial target specification, the system maintains a pool of $n$ trackers $T = \{t_1, t_2, ..., t_n\}$. At each frame, all trackers independently pro-cess the current image and produce state estimates $X = \{x_1, x_2, ..., x_n\}$, typically represented as bounding boxes. These estimates form the raw input for subsequent ensemble processing stages.

The framework is designed to be tracker-agnostic, requiring only that individual trackers provide bounding box estimates and support reinitialization. This flexibility allows integration of diverse tracking methodologies, from correlation filter-based approaches to deep learning-based methods, without modifying their internal architectures.

### B. Spatial Coherence Analysis

Spatial coherence serves as the primary cue for identifying reliable trackers within the ensemble. We observe that in most tracking scenarios, correct tracker estimates tend to cluster around the true target location, while erroneous estimates scatter across the image plane. This spatial distribution pattern enables effective clustering-based analysis.

We employ complete-link hierarchical agglomerative clus-tering (CL) [21] to group tracker estimates based on spatial proximity. The dissimilarity between two bounding boxes $x_i$ and $x_j$ is defined using the intersection-over-union (IoU) metric:

This measure captures both positional and scale differences between bounding box estimates. A symmetric $n \times n$ dissimi-larity matrix $D$ is constructed from all pairwise comparisons. The clustering algorithm groups bounding boxes with dissim-ilarity below threshold $v = 0.8$, typically resulting in a small number of coherent clusters $C = \{c_1, c_2, \dots, c_m\}$.

The clustering stage effectively partitions trackers into spa-tially coherent groups, with the largest cluster often corre-sponding to correct target estimates. However, spatial coher-ence alone may be insufficient in scenarios where multiple trackers fail consistently or when background clutter creates false clusters.

### C. Appearance Modeling and Verification

To complement spatial analysis, we incorporate appearance verification using a discriminative model trained to distinguish target from background regions. This component addresses cases where spatial clustering might be misleading, such as when multiple trackers fail in a coordinated manner or when background regions resemble the target.

We employ Histogram of Oriented Gradients (HOG) fea-tures combined with a Support Vector Machine (SVM) clas-sifier for appearance modeling [22]. The classifier is initially trained using positive samples from the first frame target re-gion and negative samples from surrounding background areas. For each tracker estimate $x_i$, we extract HOG features from the corresponding image region and compute an appearance score $s_i$ representing the classifier's confidence that the region contains the target.

The choice of HOG features and SVM classification is motivated by their demonstrated effectiveness in tracking-by-detection frameworks [23], [24]. HOG features provide robustness to illumination changes and partial occlusion, while SVM classifiers offer strong discriminative capability with efficient online updates.

Appearance scores $S = \{s_1, s_2, \dots, s_n\}$ provide an inde-pendent measure of tracker reliability, orthogonal to spatial coherence. These scores are particularly valuable for identify-ing correct trackers when they constitute a minority within the ensemble.

### D. Inlier Selection Strategies

The inlier selection stage identifies reliable trackers whose estimates will contribute to the final ensemble output. We investigate two complementary selection criteria based on spatial clustering and appearance verification.

The cluster size criterion operates on the assumption that the largest spatially coherent cluster contains correct tracker estimates:

$$c^* = \arg\max_{c_i \in C} |c_i| \qquad (2)$$

This approach effectively leverages the wisdom of the crowd, performing well when the majority of trackers follow the target correctly.

The appearance score criterion selects the cluster with the highest maximum appearance score:

$$c^* = \arg\max_{c_i \in C} \max_{x_j \in c_i} s_j \qquad (3)$$

This method prioritizes tracker confidence over consensus, proving valuable when most trackers fail but a few maintain correct tracking.

Both criteria offer distinct advantages depending on tracking conditions. The cluster size method provides smooth trajecto-ries and handles fast motion effectively, while the appearance score method excels in scenarios with background clutter and coordinated tracker failures.

### E. Ensemble Fusion and Model Update

The final ensemble estimate is derived from the selected inlier trackers $c*$. We compute the medoid bounding box as the ensemble output:

$$x^* = \arg\min_{x_i \in c^*} \sum_{x_j \in c^*} d(x_i, x_j) \qquad (4)$$

The medoid selection provides robustness to outliers within the inlier set and ensures spatial consistency in the ensemble output.

Model update occurs periodically to maintain tracking ac-curacy under appearance changes. The appearance model is updated using new positive samples extracted around the cur-rent ensemble estimate when confidence is high, as measured by the average appearance score of inlier trackers or the percentage of trackers in the inlier cluster.

Tracker reinitialization addresses the problem of tracker drift by resetting outlier trackers to the current ensemble estimate. This process occurs every 15 frames, balancing the need for correction with allowing trackers to recover from temporary failures. The asynchronous reinitialization strategy enables failed trackers to resume tracking from the correct target location, effectively steering them back to the target.

## IV. Experimental Setup

### *A. Evaluation Metrics*

We employ standardized evaluation metrics from the track-ing literature to ensure fair comparison with existing methods [1]. Distance precision (DP) measures the percentage of frames

TABLE I: TRACKER POOL COMPOSITION

| Method | Type | Implementation |
|---|---|---|
| Template Matching (TM) [25] | Template-based | C++ |
| Mean Shift (MS) [25] | Density-based | C++ |
| Variance Ratio (VR) [25] | Density-based | C++ |
| Peak Difference (PD) [25] | Point-based | C++ |
| Ratio Shift (RS) [25] | Point-based | C++ |
| ASLA [26] | Sparse representation | MATLAB/C++ |
| Compressive Tracker (CT) [24] | Sparse representation | MATLAB/C++ |
| MEEM [23] | Tracking-by-detection | MATLAB/C++ |
| SPLTT [27] | Tracking-by-detection | MATLAB/C++ |
| KCF [28] | Correlation filter | MATLAB |
| DCF [28] | Correlation filter | MATLAB |
| SCT [29] | Correlation filter | MATLAB |
| CSK [28] | Correlation filter | MATLAB |
| RCSK [30] | Correlation filter | MATLAB |
| MOSSE [31] | Correlation filter | MATLAB |

where the estimated center location is within a threshold distance d from the ground truth:

$$\mathrm{DP}(d) = \frac{1}{N}\left|\{t : \|\mathbf{p}_t - \mathbf{g}_t\| \le d\}\right|, \; d \ge 0 \qquad (5)$$

Overlap precision (OP) measures the percentage of frames where the overlap between estimated and ground truth bound-ing boxes exceeds threshold $b$:

$$\mathrm{OP}(b) = \frac{1}{N}\left\{ f : \frac{|\hat{B}_f \cap B_f|}{|\hat{B}_f \cup B_f|} \ge b \right\}, \; 0 \le b \le 1 \qquad (6)$$

We follow the evaluation protocol of the Online Object Tracking Benchmark [1], using precision plots (DP at 20 pixels threshold) and success plots (area under the OP curve) for comprehensive performance assessment. One-pass evaluation (OPE) with initialization from ground truth in the first frame ensures consistent comparison across methods.

### *B. Datasets and Trackers*

We evaluate our approach on the Online Object Tracking Benchmark (OOTB) [1], comprising 50 challenging sequences with full annotation of 11 attributes including illumination variation, occlusion, scale change, and motion blur. This dataset represents current standards for tracking evaluation and enables direct comparison with state-of-the-art methods.

Our tracker pool, detailed in Table I, includes 15 diverse algorithms representing major tracking paradigms. Selection criteria required publicly available source code, bounding box output, state storage capability, and reinitialization support. This diverse composition ensures the ensemble can leverage complementary strengths across different tracking methodolo-gies.

### C. Implementation Details

We implemented our ensemble framework in MATLAB, with critical components optimized in C++. Spatial clustering uses complete-link hierarchical agglomerative clustering with the dissimilarity measure defined in Equation 1. The appear- and 9 orientation bins, combined with an RBF SVM classifier from LIBSVM [32].

TABLE II: PERFORMANCE ANALYSIS FOR DIFFERENT ENSEMBLE CONFIGURATIONS

| Selection Method | # Trackers | Success (AUC) | Precision (20px) |
|---|---|---|---|
| 3*Appearance Score | 5 | 0.585 | 0.855 |
| | 10 | 0.569 | 0.811 |
| | 15 | 0.560 | 0.804 |
| 3*Cluster Size | 5 | 0.528 | 0.765 |
| | 10 | 0.493 | 0.715 |
| | 15 | 0.480 | 0.678 |

All experiments were conducted on a system with Intel Xeon CPU and 16GB RAM. Processing times were measured excluding individual tracker computations to focus on ensem-ble overhead. The framework maintains real-time capability with average processing times of 0.062s/frame for 5 trackers, 0.122s/frame for 10 trackers, and 0.198s/frame for 15 trackers.

## V. RESULTS AND ANALYSIS

### A. Overall Performance

We evaluate our Online Ensemble Tracker (OET) against individual trackers in our pool and state-of-the-art methods including Struck [33], SCM [34], and TLD [4]. The Figure presents precision and success plots across all 50 OOTB se- individual trackers in our pool and state-of-the-art methods including Struck [33], SCM [34], and TLD [4]. The Figure presents precision and success plots across all 50 OOTB se- quences, demonstrating consistent superiority of our ensemble approach.

Our ensemble achieves ranking scores of 0.585 AUC for success and 0.855 precision at 20 pixels, outperforming all individual trackers and competing methods. The improvement is particularly notable in challenging sequences involving camera rotation (singer1, singer2, fish), complete occlusion (jogging, david3), and fast motion (bolt, football).

Qualitative results illustrate the ensemble's ability to main-tain tracking under conditions that cause individual tracker failures. The framework effectively identifies and leverages reliable trackers while correcting failures through reinitialization.

### B. Ensemble Configuration Analysis

We investigate the impact of ensemble size and inlier se-lection criteria on tracking performance. Table II summarizes results for different pool sizes using both selection criteria.

The appearance score criterion consistently outperforms cluster size selection across all pool configurations. However, adding weaker trackers to the ensemble gradually decreases performance, with a 5

The cluster size method provides smoother trajectories and handles fast motion effectively but struggles with occlusions where multiple trackers may fail consistently. The appearance score method excels in scenarios with background clutter and coordinated failures but may be misled by spurious regions with target-like appearance.

### C. Attribute-based Performance

As it presents attribute-based analysis comparing OET with state-of-the-art trackers across different challenge categories. Our ensemble outperforms competitors in 8 of 11 attributes, demonstrating broad robustness across diverse tracking conditions.

Notable strengths include handling of illumination variation (IV), out-of-plane rotation (OPR), occlusion (OCC), deformation (DEF), motion blur (MB), in-plane rotation (IPR), out-of-view (OV), and background clutter (BC). The SCM tracker shows advantage in scale variation (SV) due to its affine motion model, highlighting a limitation of our current scale estimation approach.

The ensemble's strong performance across diverse attributes confirms its ability to leverage complementary strengths of constituent trackers, adapting to different challenges through the data-driven fusion mechanism.

### D. Tracker Contribution Analysis

We analyze individual tracker contributions to the ensemble by measuring how often each tracker is selected as inlier, outlier, or medoid. High-performing trackers like KCF [28] and MEEM [23] show low outlier rates (15-20%) and high medoid selection frequency (25-30%), confirming their relia-bility within the ensemble.

Lower-performing trackers exhibit higher outlier rates (40-60%) but still contribute to spatial consensus when correctly tracking the target. The diversity in tracker behavior enables the ensemble to maintain robustness across different scenarios, with various trackers taking leadership under specific condi-tions.

The analysis confirms that while individual tracker quality matters, the ensemble mechanism effectively identifies and leverages reliable trackers regardless of their standalone per-formance, demonstrating the value of diversity in ensemble composition.

## VI. Conclusion and Future Work

We have presented a novel online ensemble framework for visual object tracking that dynamically combines multiple tracking algorithms through spatial coherence analysis and appearance verification. Our approach demonstrates that intelligent fusion of complementary trackers can achieve per-formance surpassing individual algorithms and current state-of-the-art methods.

The key advantages of our framework include its tracker-agnostic design, data-driven fusion mechanism, and ability to correct tracker failures through reinitialization. Extensive evaluation on standard benchmarks confirms consistent performance improvement across diverse challenging conditions, with particular strength in handling occlusions, fast motion, and background clutter.

Several directions warrant future investigation. Scale esti-mation represents a current limitation that could be addressed through multi-scale analysis or dedicated scale estimators. Expanding the tracker pool with recent deep learning-based approaches could further enhance ensemble performance. Ad-ditionally, adaptive weighting of tracker contributions based on historical performance could improve fusion quality.

The ensemble paradigm offers promising avenues for ad-vancing visual tracking robustness. By focusing on intelligent combination rather than individual algorithm development, our work demonstrates that collective intelligence can overcome limitations of individual approaches, moving toward more reliable tracking systems for real-world applications.

## References


[1] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," IEEE Conference on Computer Vision and Pattern Recognition, pp. 2411–2418, 2013.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," ACM Computing Surveys, vol. 38, no. 13, 2006.

[3] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," International Journal of Computer Vision, vol. 77, pp. 125–141, 2007.

[4] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 34, pp. 1409–1422, 2011.

[5] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," IEEE transactions on Pattern Analysis and Machine Intelligence, pp. 983–990, 2010.

[6] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," British Machine Vision Conference, vol. 1, pp. 1–10, 2006.

[7] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "Prost: Parallel robust online simple tracking," IEEE Conference on Computer Vision and Pattern Recognition, pp. 723–730, 2010.

[8] C. Bailer, A. Pagani, and D. Stricker, "A superior tracking approach: Building a strong tracker through fusion," European Conference on Computer Vision, pp. 170–185, 2014.

[9] V. Chitukoori, "Interpretable neural networks: Bridging gams and deep learning," in 2025 IEEE/ACIS 23rd International Conference on Soft-ware Engineering Research, Management and Applications (SERA), 2025, pp. 362–364.

[10] ——, "A category theory-based framework for deep neural networks: Enhancing structure and interpretation," 2025.

[11] ——, "A rust-based distributed execution platform for aggregate com-puting in resource-constrained environments," in 2025 21st International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), 2025, pp. 681–688.

[12] ——, Development of an IoT-Based Environmental Monitoring System Using Raspberry Pi for Real-Time Data Analysis, 2025.

[13] R. K. Ramakrishnan and J. J. Lekkala, “Hybrid decision tree and genetic algorithm for enhanced classification accuracy,” TechRxiv, Tech. Rep., 2025.
[14] R. K. Ramakrishnan, “Genetic algorithm optimization in circuit design,” in 2025 1st International Conference on AIML-Applications for Engi-neering & Technology (ICAET), 2025, pp. 1–4.
[15] R. K. Ramakrishnan, A. Nayak, and J. J. Lekkala, “Integrating cloud, edge, and iot,” TechRxiv, Tech. Rep., 2025.
[16] R. K. Ramakrishnan, M. Sadineni, and J. J. Lekkala, “Enhancing distributed system reliability through request-level fault injection and fine-grained tracing,” TechRxiv, Tech. Rep., 2025.
[17] R. K. Ramakrishnan, “Financial and technological considerations for deploying applications on cloud computing platforms: A case study of aws,” TechRxiv, Tech. Rep., 2025.
[18] R. K. Ramakrishnan and J. J. Lekkala, “Decentralized github manage-ment: Blockchain solution,” TechRxiv, Tech. Rep., 2025.
[19] T. Agarwal and A. Chaudhari, “A data analytics approach to detecting counterfeit luxury goods: Analyzing material descriptions and visual patterns,” Journal of Intelligent Systems and Engineering Management, 2025.
[20] S. Mamtani, “Enhancing transformer-based vision models: Addressing feature map anomalies through novel optimization strategies,” arXiv, 2025.
[21] A. K. Jain and R. C. Dubes, “Algorithms for clustering data,” 1988.
[22] P. Dollar, “Piotr’s computer vision matlab toolbox (pmt),” 2009.
[23] J. Zhang, S. Ma, and S. Sclaroff, “Meem: robust tracking via multiple experts using entropy minimization,” European Conference on Computer Vision, 2014.
[24] K. Zhang, L. Zhang, and M. H. Yang, “Real-time compressive tracking,” Lecture Notes in Computer Science, vol. 7574, pp. 864–877, 2012.
[25] R. Collins, X. Zhou, and S. K. Teh, “An open source tracking testbed and evaluation web site,” IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2005.
[26] X. Jia, H. Lu, and M. H. Yang, “Visual tracking via adaptive structural local sparse appearance model,” IEEE Conference on Computer Vision and Pattern Recognition, pp. 1822–1829, 2012.
[27] J. S. Supancic and D. Ramanan, “Self-paced learning for long-term tracking,” IEEE Conference on Computer Vision and Pattern Recog-nition, pp. 2379–2386, 2013.
[28] J. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed track-ing with kernelized correlation filters,” IEEE transactions on Pattern Analysis and Machine Intelligence, pp. 1–14, 2015.
[29] K. Zhang, L. Zhang, M.-H. Yang, and D. Zhang, “Fast tracking via spatio-temporal context learning,” European Conference on Computer Vision, pp. 1–16, 2013.
[30] M. Danelljan, F. S. Khan, M. Felsberg, and J. V. D. Weijer, “Adaptive color attributes for real-time visual tracking,” IEEE Conference on Computer Vision and Pattern Recognition, 2014.
[31] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” IEEE Conference on Computer Vision and Pattern Recognition, pp. 2544–2550, 2010.
[32] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 27:1–27:27, 2011.
[33] S. Hare, A. Saffari, and P. H. S. Torr, “Struck: Structured output tracking with kernels,” IEEE International Conference on Computer Vision, pp. 263–270, 2011.
[34] W. Zhong, H. Lu, and M.-H. Yang, “Robust object tracking via sparsity-based collaborative model,” IEEE Conference on Computer Vision and Pattern Recognition, pp. 1838–1845, 2012.