CYBERNETICS and COMPUTER TECHNOLOGIES

Visual object tracking is an important task in computer vision with diverse applications, including robotics, autonomous navigation, surveillance, and human-computer interaction. This paper provides a survey of methods and benchmark datasets for visual single object tracking (SOT), focusing on both short-term and long-term tracking scenarios. Various tracking approaches are discussed, including correlation-based methods, keypoint tracking, and deep learning techniques such as Siamese networks, transformer-based, and other deep long-term trackers. The overview of popular benchmark datasets, along with corresponding evaluation protocols, is presented. The performance of many discussed algorithms is compared on VOT 2018, LaSOT, and GOT-10k benchmarks. The study reveals that deep learning approaches demonstrate superior performance in complex scenarios, although they often require substantial computational resources. Future research should prioritize enhancing the efficiency and adaptability of tracking algorithms, particularly for long-term scenarios that closely resemble real-world applications.

Keywords: *visual object tracking, single object tracking, correlation filters, keypoint tracking, Siamese networks, transformers.*

UDC 004.93 DOI:10.34229/2707-451X.26.1.7

Y. ROMANIAK

AN ANALYSIS OF VISUAL SINGLE OBJECT TRACKING METHODS

Introduction. Visual object tracking is a crucial computer vision problem with lots of applications, including autonomous navigation, robotics, surveillance, and human-computer interaction. It is an essential component of many Unmanned Aerial Vehicles (UAVs) systems as well. Despite a significant research effort in recent years, the problem remains challenging due to a wide range of factors, including target occlusions, illumination changes, motion blur, background clutter, and object deformations. In addition, the need for real-time performance in many practical applications, especially in resource-constrained environments such as UAVs, imposes strict requirements for both computational efficiency and accuracy.

The goal of visual object tracking is to estimate the position of the object of interest in each frame, given its initial position in the first frame. If the target leaves the field of view (FoV), the algorithm should recognize this situation and provide proper feedback. The most common way to annotate the target's position is to use a bounding box (Fig. 1, a). It is a minimal rectangular area of an image that contains the target. However, sometimes a bounding box cannot accurately capture the shape of the target. Therefore, another approach is tracking with pixel-wise segmentation (Fig. 1, b), which is more robust to complex transformations of the target, but requires more computational resources [1].

There are several categories of object tracking tasks. *Single-object tracking (SOT)* requires following only one target, while the task of *multi-object tracking (MOT)* is to estimate the position of several targets and enumerate them. During *short-term tracking*, it is assumed that the target is always present (at least partially) within the field of view. In contrast, *long-term tracking* considers situations where the target may disappear for an extended period and reappear later. Additionally, long-term trackers are expected to follow the target for an extended period; therefore, the tracker must deal with drifting and significant appearance changes of the target. Long-term tracking has gained increasing research interest as it aligns more closely with real-world scenarios.

This paper is organized as follows. The first part is devoted to short-term and long-term tracking with correlation filters. In the next section, keypoint-based trackers are considered. After this, trackers based on deep neural networks, which include Siamese, transformer-based, and other types of deep long-term trackers are described. In the following sections, popular benchmark datasets are described, and the performance of various tracking algorithms is compared on these datasets. The final section concludes the article and discusses future research.

a – bounding box tracking and positioning

b – bounding box + segmentation tracking and positioning

FIG. 1. Illustration of the target tracking via: *a* – a simple bounding box, *b* – a bounding box with segmentation. Images are from the VOT-LT 2018 dataset [2]

Correlation-based trackers. Discriminative Correlation Filters (DCF) have become a well-known and widely used tool for visual object tracking due to their high speed and good performance. Originally proposed for short-term tracking, DCF-based methods have undergone significant advancements, allowing for their use in more challenging scenarios. In the following section, the core principles and key algorithmic developments of the DCF-based tracking framework are reviewed, with its application to both short-term and long-term tracking.

One of the first papers that introduced the correlation filter framework was by Bolme et al. [3]. The idea can be described as follows. Given an image of a search region X, and an image W, which represents a target object appearance and is called a *filter*, we estimate the location of the target by correlating X with W. The maximum value of the resulting response map indicates the location of the object's center. A filter W acts as a linear classifier that discriminates between the target object and the background. In [3], Bolme formulated W^* as a solution to a least-squares optimization problem, given a training set of images X_i with corresponding ground-truth response maps Y_i (see Fig. 2):

$$W^* = \arg\min_W \sum_i \|X_i * W - Y_i\|^2 + \lambda \|W\|^2,$$

where λ is called a regularization parameter and is introduced to prevent overfitting, and $*$ symbol denotes a correlation between two images. To speed up calculations, the Convolution theorem is applied, which states that correlation between X and W in the spatial domain is equivalent to element-wise multiplication in the Fourier domain. Thus, the optimization problem can be rewritten in the following way:

$$\widehat{W}^* = \arg\min_{\widehat{W}} \sum_i \left\| \hat{X}_i \odot \widehat{W} - \hat{Y}_i \right\|^2 + \lambda \left\| \widehat{W} \right\|^2.$$

The symbol $\hat{\cdot}$ denotes a discrete Fourier transform of the image patch and $\odot$ is an element-wise matrix multiplication. A closed-form solution for the optimization problem is presented in [3] by Bolme et al.:

$$\widehat{W} = \frac{\sum_i \hat{X}_i^* \odot \hat{Y}_i}{\sum_i \hat{X}_i^* \odot \hat{X}_i + \lambda}.$$

Here $\hat{X}^*$ is a complex conjugate of the image $\hat{X}$. This formula involves only element-wise operations, making the filter estimation very efficient in the Fourier domain. If Z is a new search image patch, the target location is given by the maximum value of the response map:

$$Y = \mathcal{F}^{-1}(\widehat{W} \odot \hat{Z}),$$

where $\mathcal{F}^{-1}$ is an inverse discrete Fourier transform. The filter is trained online and is updated each frame to adapt to a changing background.

FIG. 2. A search image *X* with a ground-truth Gaussian-shaped response map *Y*

Further, Henriques et al. [4] introduced the Kernelized Correlation Filter (KCF) algorithm, which extends [3] in several ways. First, it generalizes the original correlation filter formulation to a nonlinear setting by mapping input features into a high-dimensional kernel space using the kernel trick. This enables the tracker to handle more complex appearance variations. Second, it exploits the circulant structure of training samples in the Fourier domain, enabling high-speed training and detection by reducing computational complexity from quadratic to linear. Third, it incorporates a multi-channel HOG feature, improving the robustness of the tracker to illumination changes and background clutter. Generally, any multi-channel feature representation can be incorporated into this framework [4, 5]. The general tracking pipeline, incorporating correlation filters and feature extraction, is presented in Fig. 3.

In real-world scenarios, the target object changes scale during a video. Therefore, additional target scale estimation is required to provide more accurate tracking results. The problem was addressed, in particular, by [6, 7]. Specifically, Li et al. [6] searched for the optimal scaling factor that maximizes the correlation with the filter using a predefined set of scaling candidates. This exhaustive scale search, however, is computationally demanding [7]. Another strategy, implemented in the Discriminative Scale Space Tracker (DSST) [7], learns a separate discriminative correlation filter for translation and a 1-D filter for scale estimation. Training samples for the scale filter are constructed by extracting feature descriptors from image patches, sampled at variable sizes and centered around the target. Relative scale change is selected by maximizing the scale correlation score.

One of the crucial aspects of effective object tracking is image feature extraction. In some early works, e.g. [3], grayscale pixel intensities were used. However, raw pixel values are highly sensitive to illumination

changes and target occlusion, which can significantly degrade a tracker's performance. As mentioned earlier, KCF tracker [4] exploits the Histogram of Oriented Gradients (HOG) feature, used for object detection tasks as well. Another branch of research focused on developing color features that are robust to various illuminance changes. One strategy is to transform an RGB image into another color space, for instance, LAB, HSV (Hue, Saturation, Value), YCbCr (Luminance and two chromatic components). Danelljan et al. [7] show the usage of color attributes or Color Names (CN) features. This is a special color encoding scheme that represents each RGB value as a probabilistic 11-dimensional vector on the set of basic colors: black, blue, brown, grey, green, orange, pink, purple, red, white, and yellow. It is demonstrated that utilizing CN image representation enhances the performance of the correlational tracker [8, 9]. All the described features are often referred to as *hand-crafted features* in the literature.

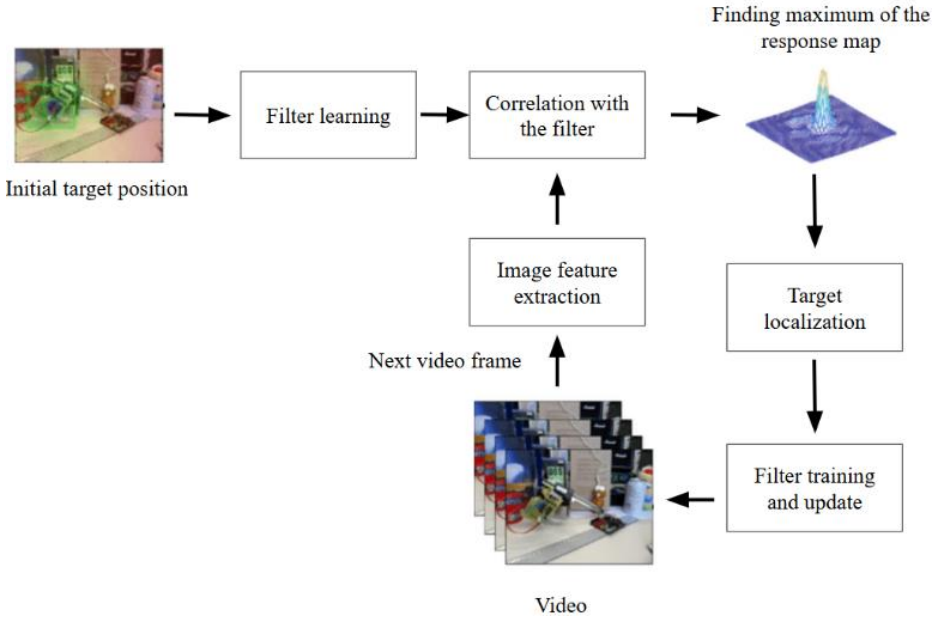

FIG. 3. General pipeline of tracking with correlation filters

An alternative and modern approach to feature extraction is to use deep convolutional neural networks (CNNs) [10]. Architectures such as ResNet [11], trained offline on a large-scale image dataset like ImageNet, have achieved great performance in image classification. These networks have a hierarchical structure: the output from the convolution layer l, followed by pooling and a non-linear activation function, serves as the input to the layer l 1. Additionally, ResNet introduces residual connections to cope with the vanishing gradient problem in very deep neural networks [11]. While earlier layers are trained to capture simple patterns, such as edges and lines, the deeper ones extract high-level semantic information about an image, e.g., object class. Ma et al. [10] utilized intermediate outputs from another network, VGG-Net, then trained separate correlation filters similarly to the KCF tracker and propagated a target location to earlier layers to obtain a more accurate target pose. Further, deep feature extractors have become an essential component of deep learning-based trackers. Those types of trackers are discussed in later sections.

Danelljan et al. [12] proposed a theoretical framework for learning continuous correlation filters in the spatial domain by employing a feature interpolation model of the training samples. The filter is learned online and iteratively using the Conjugate Gradient method. Their tracker, CCOT, improves tracking performance in several ways. First, the framework enables a more natural integration of multi-resolution features, such as HOG or color features, rather than manual feature resizing to fit the filter size. Second, the

continuous response map enables localization of the target at a sub-pixel level, thereby improving the accuracy of the location estimation. The main drawback of the CCOT tracker is its low speed, resulting from the optimization of a large number of parameters, which can also potentially lead to overfitting [12]. The following work by Danelljan et al. [13] presents an Efficient Convolution Operator (ECO) tracker, which aims to address the aforementioned issues of the CCOT tracker. First, instead of learning one large filter for each feature channel, ECO decomposes it into smaller and more efficient operators, which significantly reduces the number of model parameters. Second, ECO employs a more effective generative model that aggregates the most representative information about the target, in contrast to storing previous samples with decayed weights. Third, ECO updates its model at sparser intervals, instead of frame-by-frame updates. The authors demonstrated that using handcrafted features for training enables the tracker to achieve a speed of 60 FPS on a single CPU, making it suitable for real-time applications.

Long-term correlation-based trackers. Basic DCF trackers perform well as long as the target remains in the field of view (or is only partially occluded) and a video sequence is relatively short. This happens due to several reasons. First, in most cases, a DCF tracker updates the filter at each frame, without verifying the estimation quality. This leads to error accumulation, tracker drifting, and eventual target loss in long-term scenarios. Second, in the case of long-term target occlusion, the algorithm requires a separate re-detection logic, e.g., a detection algorithm, trained either in an online or offline manner, to estimate the target's new location and scale. Third, the target's feature representation may not be invariant to complex appearance variations or lighting changes.

Many adaptations of the DCF framework have been proposed for long-term tracking scenarios [14–18]. Ma et al. [14] employed two separate correlation filters for object's translation and scale estimation, but the former one has a more conservative update scheme, depending on the confidence of the latter filter. The tracker's confidence is measured as a maximum value of the response map. When this value drops below a predefined threshold, a random fern-based online detector is activated. The training set for the detector consists of confident positive and negative samples, drawn from the tracker's result. In a similar work [15], an online Support Vector Machine (SVM) algorithm is used as a re-detector. In the Fully-Correlational Long-Term Tracker (FuCoLoT) [16], proposed by Lukežič et al., a detector is implemented as a set of correlation filters, trained over different temporal windows. These filters are updated at various frequencies, including one filter that is never updated (the initial model). This strategy helps achieve resistance to occlusions and disappearance and enables recovery from potential target loss.

Another strategy for improving long-term tracking is to use complementary trackers. For instance, Bertinetto et al. [17] employed two types of target models: *a template-based model* that relies on a Histogram of Oriented Gradients representation and *a color model*, represented by a global color histogram or specifically quantized RGB colors as features. This model captures color distribution and is invariant to spatial permutations and target deformations. Similarly, in [18] a histogram of quantized colors in the Hue-Saturation-Value (HSV) color space is used. In both cases, two separate linear filters are trained, and their scores are fused via convex combination.

Conclusion remarks on tracking with correlation filters. Tracking with correlation filters is a well-studied branch of visual tracking, with a variety of methods that emerged since 2014. Their ability to efficiently locate the object of interest using Discrete Fourier Transform in real-time and to combine different hand-crafted and deep-based image features made them popular for computationally constrained environments [19]. However, in the long-term tracking correlation filters need additional modifications to re-detect objects and reduce error accumulation due to filter online updates. The discussed long-term trackers account for these challenges, but do not solve them completely.

Keypoint-based trackers. Another approach to object tracking involves the use of interesting points, feature points, or *keypoints*. Unlike in the correlation-based setting, where the tracked object is described by

an image patch and a corresponding filter, the object model in keypoint-based methods consists of a set of points (pixel coordinates or small image regions) with some distinctive properties. This representation can be either *sparse* – tracking only a small subset of pixels – or *dense* – estimating the motion of every pixel. In general, keypoint-based tracking consists of detecting keypoints in each new frame, estimating their motion (e.g. through matching), and using this information to establish the object's position.

A crucial aspect of keypoint tracking is the choice of a keypoint detector. The simplest approaches involve finding points or image patches with significant gradient variations in several directions, such as corners [20]. More modern and reliable methods, such as Scale-Invariant Feature Transform (SIFT) [21], Oriented FAST and Rotated BRIEF (ORB) [22], are more robust to noise and significant scale and orientation change of the points between two frames. Additionally, they are equipped with both keypoint detectors and *keypoint descriptors*. A *descriptor* is a numerical vector that compactly encodes distinctive characteristics of a local image region around a detected keypoint. It is used to match keypoints between frames by computing the distance between descriptors (e.g., Euclidean for real-valued vectors or Hamming for binary ones). Significant advancements have been made with deep learning methods that enable training both the keypoint detector and descriptor in a joint, end-to-end manner. For example, the SuperPoint [23] detector shows better performance than classic SIFT or ORB in terms of the number of correct matches and robustness to noise.

One of the earliest and most prominent feature tracking frameworks is presented in several articles by Kanade, Lucas, Shi, and Tomasi [20, 24]. Having a video sequence $I(x, y, t)$, where x, y are the coordinates of the image grid, and t is a frame number, the goal is to find a vector (u^*, v^*) that will minimize the similarity error between patches of neighboring frames:

$$(u^*, v^*) = \arg \min_{(x,y)\in W} \sum_{(x,y)\in W} [I(x, y, t) - I(x + u, y + v, t + 1)]^2 ,$$

where W denotes a small (for instance, 15 by 15 pixels) window around some keypoint. The iterative solution, also known as *Lucas-Kanade optical flow*, involves solving the following system of linear equations:

$$\begin{bmatrix} \sum_{(x,y)\in W} I_x(x,y)^2 & \sum_{(x,y)\in W} I_x(x,y)\cdot I_y(x,y) \\ \sum_{(x,y)\in W} I_x(x,y)\cdot I_y(x,y) & \sum_{(x,y)\in W} I_y(x,y)^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum_{(x,y)\in W} I_x(x,y)\cdot I_t(x,y) \\ \sum_{(x,y)\in W} I_y(x,y)\cdot I_t(x,y) \end{bmatrix},$$

where I_x, I_y and I_t are image derivatives (gradients) in x, y and t directions, and u, v are unknown motion components of the window W. The quality of the solution is determined by the eigenvalues λ_1 and λ_2 of the left-hand matrix. Very small eigenvalues correspond to flat and textureless image regions with no strong intensity variation. In contrast, large eigenvalues mean strong intensity variation in one direction (edge) or several directions (corner). For tracking purposes, Shi and Tomasi [20] proposed to select corner points such that satisfy the following criteria:

$$\min(\lambda_1, \lambda_2) \geq \lambda,$$

where λ is some predefined threshold. Fig. 4 shows an example of such points. Additionally, the initial assumption for the Lucas-Kanade optical flow is brightness constancy within the window and slow motion. To address this, image pyramids are used to estimate the motion at different scales.

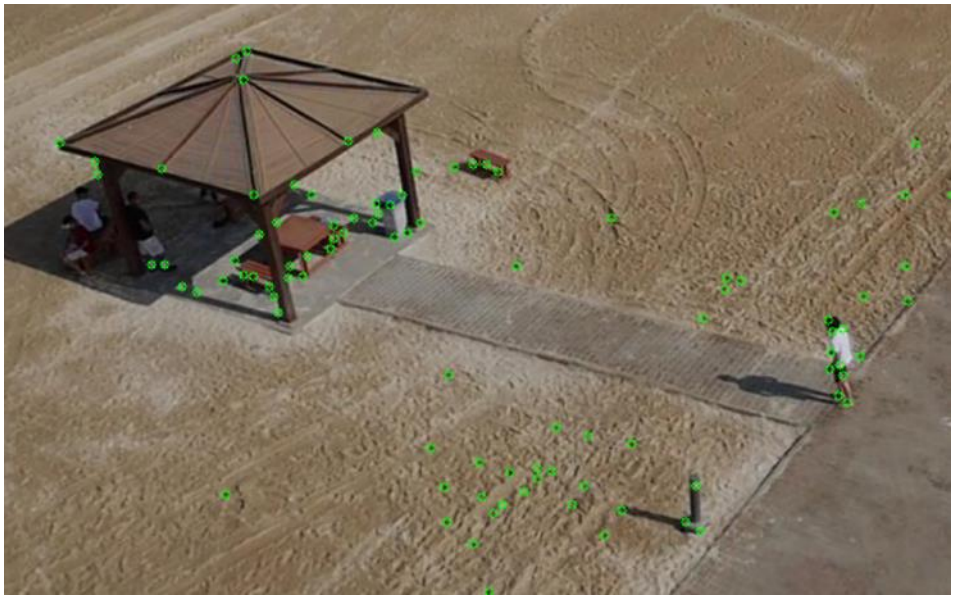

FIG. 4. An image scene with features from Shi-Tomasi point detector [20]. The image is from the VOT-LT 2018 dataset [2]

The described Kanade-Lucas-Tomasi (KLT) tracking framework became a foundation for many further trackers. Kalal et al. [25] propose to evaluate individual point tracks using Euclidean distance between forward ($I_t \rightarrow I_{t+1}$) and backward ($I_{t+1} \rightarrow I_t$) flows. The object's displacement and scale changes are computed as the median direction and scale of 50 % best points, based on the forward-backward error. Nebehay et al. [26, 27] use both optical flow between consecutive frames and matching with the initial frame to track object's feature points. This combination of static and adaptive object models enables more robust long-term tracking. Additionally, each matched keypoint votes for the new object center location. To remove outliers, a clustering algorithm is used to determine subset of points with consistent predictions. Hong et al. [28] demonstrated the complementary use of correlation filter and keypoint-based frameworks. Their multi-store tracker (MUSTer) consists of a short-term component based on a correlation filter and a long-term component, which is defined by a foreground and background keypoint feature database. For feature tracking, both feature matching with the database and sparse optical flow are considered. In contrast to [26, 27], the feature database is updated over time using the memory model, which allows it to retain features that are useful for matching and discard those that are not.

Some works consider the joint use of keypoints and *superpixels* [29]. A superpixel is a group of connected pixels that share similar characteristics such as color, texture, or brightness. Every object can be segmented into superpixels; however, they are not appropriate for matching due to the low discriminative power [29]. On the other hand, it is hard to obtain a rich set of keypoints for textureless or low-contrast images. Some tracking methods, for instance SPiKeS [29], construct a superpixel descriptor that consists of a color histogram, a set of keypoints inside or near the superpixel, and their magnitude and orientation relative to the superpixel center. The matching is performed by comparing color histograms and the relative positions of matched keypoints near the superpixel. At each frame, both sets of keypoints and augmented superpixels are matched with the initial model. A superpixel voting is used to determine a new center location of the object.

Conclusion remarks on keypoint-based trackers. Unlike correlation filters, keypoint-based trackers enable part-based tracking of objects, which means that the object of interest can be tracked by using only

a subset of keypoints. This is useful in scenarios where objects are partially occluded. Using powerful keypoint detectors such as SIFT or ORB can increase robustness to noise and establish orientation and scale invariance. For long-term tracking, various update schemes of object's feature representation must be considered. On the other hand, keypoint representation is not as rich as a representation by correlation filter, for example, for low-textured objects. Additionally, using only keypoints for tracking small objects is also challenging for the same reasons.

Deep learning-based trackers. In recent years, deep machine learning methods have become a major research focus in the visual object tracking domain. By leveraging the powerful feature representation and learning capacity of deep neural networks, deep trackers achieve better tracking accuracy, greater robustness to appearance changes, and generalization across diverse scenarios. As a result, they often surpass traditional methods in benchmark evaluations. In contrast to classic DCF trackers, deep trackers typically are trained only offline using large-scale datasets. Due to the vast number of model parameters, an online update can be computationally infeasible. Therefore, a rich dataset with annotated target positions and different tracking scenarios must be used for model training. In this section, several methods are described, including Siamese trackers, transformer-based trackers, and some other network architectures for long-term tracking.

Siamese trackers. The primary goal of a Siamese network is to compare pairs of inputs, e.g., images or feature vectors, and measure their similarity. Each input is processed via identical subnetworks with shared weights. Then, a similarity metric is used to compare the outputs. In 2016, Bertinetto et al. [30] proposed a Fully Convolutional Siamese Network (SiamFC) for object tracking (Fig. 5). The object's template and a search region are processed via a shared feature extractor φ, and their similarity is computed using cross-correlation with an additional bias parameter. This results in a response map, where a maximum corresponds to the location of the object:

$$R = \varphi(X) * \varphi(Z) + b.$$

The network is trained offline on a large dataset of pairs of neighboring video frames with the known object location. During inference, an object's template is extracted from the first frame and doesn't change during tracking. A search region is cropped and centered around the estimated position of the object. Despite the simplicity, the SiamFC tracker showed excellent performance and real-time speed on Graphical Processing Units (GPUs). A quite popular tracker, GOTURN [31], relies on a similar two-branch architecture, but fuses feature representations and learns to estimate bounding box coordinates directly with a feed-forward regression network.

In the following years, many modifications of Siamese networks were proposed [32–36]. SiamRPN network [32] utilizes the SiamFC as a pre-trained feature extractor and Region Proposal Network (RPN) as a location estimator. During training, the RPN classifies a set of predefined image regions of different scales (anchors) as positive (containing the object) and negative, and provides coordinate adjustments of the anchor with respect to the ground truth. Distractor-aware Siam RPN [33] introduced strategies to improve feature learning by addressing imbalanced data distribution, a separate distractor-aware module, and a local-to-global search strategy to handle target occlusion in long-term scenarios. SiamMask [34] extends the architecture of SiamRPN and improves the offline training procedure by augmenting the loss function with a binary segmentation task. Therefore, the network performs both visual object tracking and video object segmentation in real-time. Yu et al. [35] addressed a key limitation of classic Siamese architectures – the independent feature extraction of both the template and the search region, a characteristic of classic Siamese architectures. Their model, SiamAttn, introduces a new Siamese attention mechanism that incorporates deformable self-attention and cross-attention computations. This mechanism is capable of aggregating rich contextual interdependencies between the target template and the search region and adaptively updates the target template. For more accurate tracking, SiamAttn contains a region refinement module that calculates depth-wise cross-correlations between the attentional features. Overall, the use of an attention mechanism for modelling dependencies between a region and the target is further developed in the transformer model.

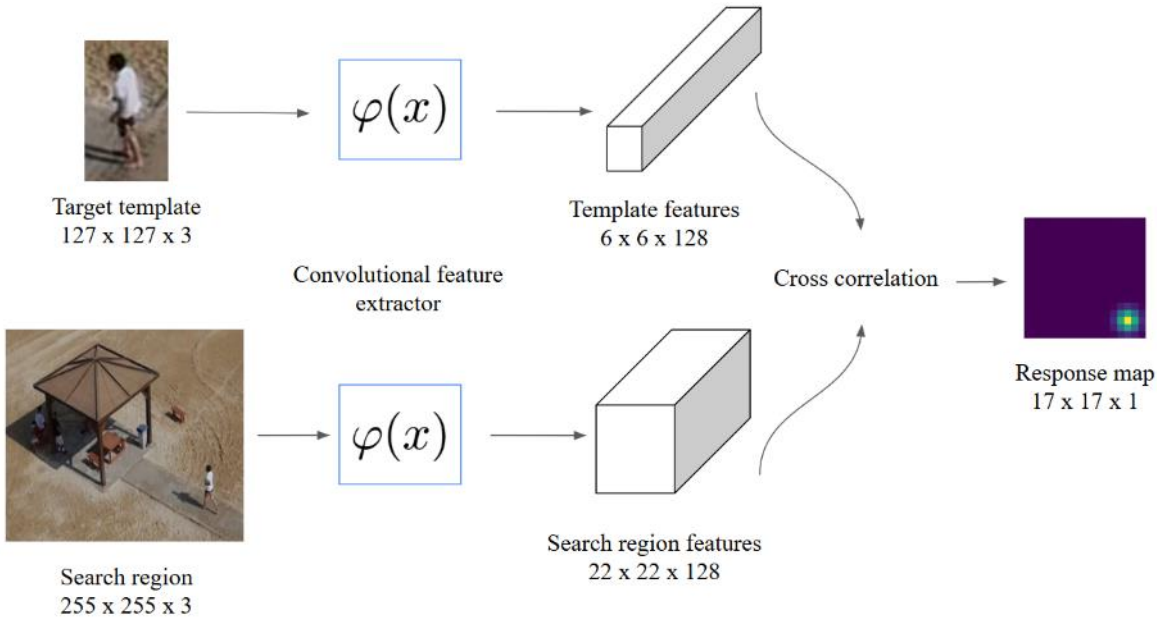

FIG. 5. A structure of the SiamFC neural network [30]

Siam R-CNN, developed by Voigtlaender et al. [36], employs a tracking-by-re-detection approach to improve long-term tracking. They present a Siamese re-detection architecture where a Region Proposal Network, similar to one in SiamRPN or two-stage object detectors like Faster R-CNN, generates potential object locations in the search frame. After that, a Siamese re-detection head then compares these proposals against a learned object template and performs bounding box regression. Given these potential target locations along with their similarity scores, an algorithm based on dynamic programming assesses a comprehensive history of potential object trajectories (tracklets) for both the target and simultaneously tracked distractor objects. This approach, called Tracklet Dynamic Programming, determines the optimal target path by considering cumulative similarity scores, appearance, and motion consistency across frames. Despite its good performance in long-term occlusion scenarios, Siam R-CNN is computationally demanding and slower than many other Siamese trackers, making it less suitable for real-time applications.

Transformer-based trackers. The transformer model, developed by Vaswani et al. [37], was primarily designed for machine translation. Unlike previous architectures for language modelling, it replaced all convolutional and recurrent operations with an attention mechanism to capture contextual dependencies more effectively. The *encoder* uses self-attention to model relationships within the input sequence, while the *decoder* combines self-attention with cross-attention to capture dependencies between the input and output sequences. Not only has the model become the foundation for large language models, but it has also influenced advancements in image recognition and object tracking tasks [38, 39].

Chen et al. [38] presented one of the first adaptations of the Transformer for object tracking. Their method, TransT, consists of three main components. *The backbone network* extracts image features from the target template and the search image independently. *The feature fusion* network enhances feature representation through *ego-context* using a multi-head self-attention module [38] and fuses these representations via a *cross-feature* module using multi-head cross-attention. *The prediction head* is composed of classification and regression branches. The regression network directly predicts the normalized coordinates of the candidate region, while the classification decides if the region corresponds to foreground or background.

Yan et. al [39] developed a spatio-temporal transformer network (STARK) that combines spatial and temporal information for accurate target localization (Fig. 6). The encoder module processes the initial target, current frame, and a dynamically updated template. The decoder predicts the target's spatial position

via *query embeddings*. The prediction head estimates the bounding box and controls template updates. STARK directly predicts bounding box location, without post-processing, which leads to stable and reliable results. The use of a dynamic template updated during tracking enables the tracker to adapt to appearance changes of the target over time, which is crucial for long-term tracking. The tracker shows high performance on multiple benchmarks, including LaSOT and GOT-10k, and real-time speed (30–40 FPS) on GPUs.

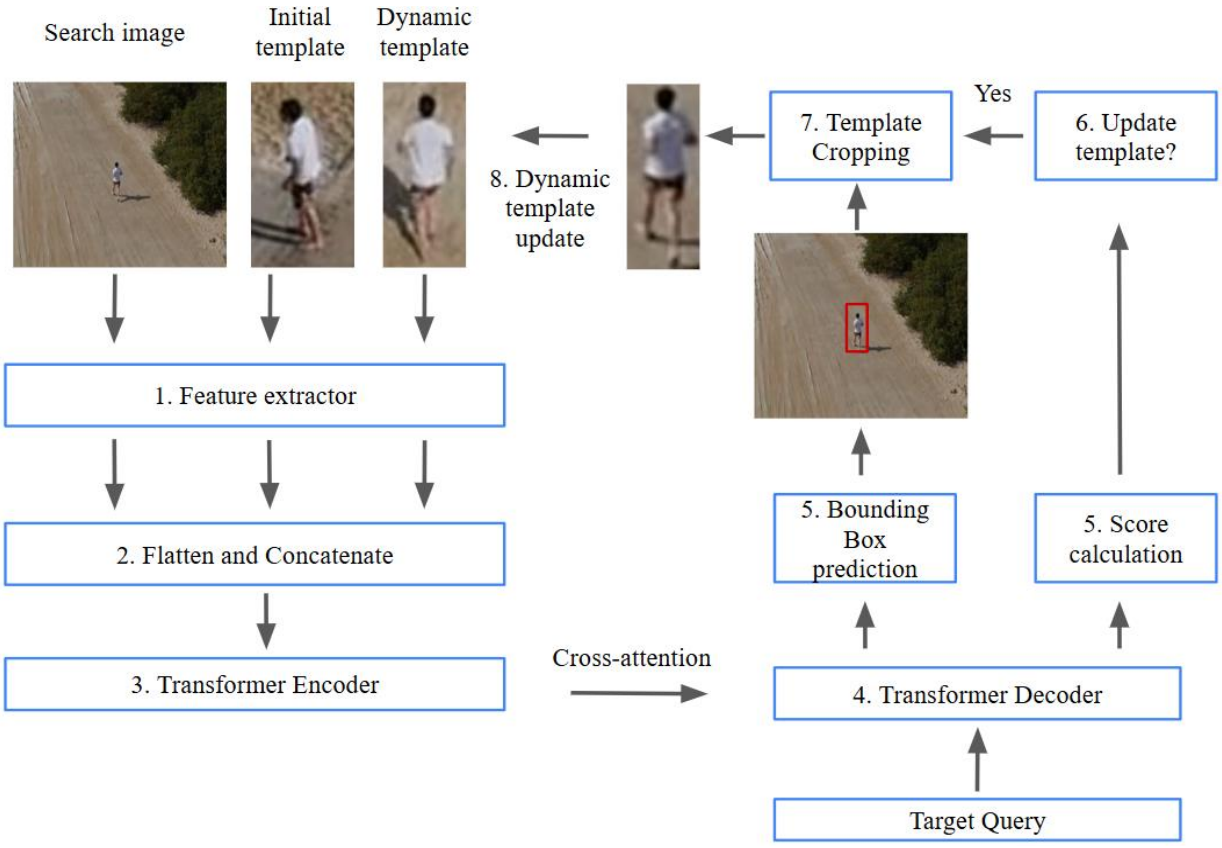

FIG. 6. The STARK tracker architecture [39]

To summarize, tracking with transformers is an ongoing research topic in visual object tracking, with numerous new models emerging.

Deep long-term trackers. Some of the described trackers, such as DaSiamRPN, Siam R-CNN, and STARK, are designed to handle long-term tracking challenges, e.g., target occlusion and appearance variations. This section describes some more deep learning-based approaches to long-term object tracking.

Fan et al. [40] proposed the Parallel Tracking and Verification (PTAV) approach, designed for robust long-term tracking. The framework combines a fast short-term correlation tracker with a verifier and a coordinator, all of which operate in parallel. The tracker continuously estimates the target location and sends its results to a coordinator. The verifier, which is built upon the Siamese network, is responsible for re-detecting the target in case of tracking failure. The use of the Region Proposal Network allows the verifier to process multiple candidate regions at once, thereby reducing its computational complexity. The coordinator monitors the tracker's performance using a tracking confidence score. If the score is below a certain threshold, it sends a request to the verifier to initiate a re-detection process. If the verifier cannot recover the target, the coordinator maintains the last known good tracking result and expands the search region. The parallel execution of all the components helps achieve a balance between the tracker's robustness in the long-term scenarios and its speed.

Yan et al. [41] introduced Skimming-Perusal Tracking, a long-term tracking framework that consists of perusal and skimming modules. The perusal part is an offline-trained SiamRPN tracker and a dedicated verifier that determines whether the target is present or absent in the local search region. If the target is absent, the skimming part generates a set of sliding windows in the global search region, which are verified using a deep convolutional network. Only the best three candidate windows are selected for further processing by the perusal part and tracking correction.

Huang et al. [42] considered the object tracking task as a global instance target search, without relying on any temporal information from the previous frames. Their GlobalTrack algorithm comprises two modules: a Query-Guided Region Proposal Network and a Query-Guided Region-CNN. In this context, the term *query* refers to the discriminative features of the object template extracted from the first frame. The first module generates region proposals modulated by query features, which helps to select regions that are more likely to contain the target. The second module performs bounding box refinement and calculates the confidence score for each proposal from the previous step. The algorithm selects the bounding box with the highest score as a tracking result. The model relies on the query extracted from the first frame and does not apply any additional post-processing steps after the bounding box selection.

Dai et al. [43] considered the issue of online trackers updating in long-term scenarios. Although online trackers are flexible and can handle target appearance variations and background changes, frequent or unreliable updates can lead to error accumulation, tracker drift, and performance degradation. To mitigate this issue, the authors proposed to use a meta-updater that decides whether the tracker's parameters should be updated at the current frame. This meta-updater is an offline-trained binary classifier based on a recurrent neural network, which relies on a history of diverse observations. These observations include geometric cues (changes of the target's bounding box), discriminative cues (response map quality), and appearance cues (similarity measure between current target appearance and the template). The training dataset for the meta-updater is generated using the same short-term tracker used in the long-term tracking algorithm. The presented framework can be generalized and integrated with any short-term tracker.

Dunnhofer et. al [44] presented another approach for long-term tracking by combining the outputs of two complementary deep trackers. Both components report the bounding box of the target and tracking confidence. The first tracker has high target localization capability, but its confidence score is less consistent with the location prediction; for example, it may report high confidence even when the predicted location is inaccurate. In contrast, the second tracker is less precise, but has a more consistent confidence score. Given bounding boxes and confidence scores from both trackers, an online-updated binary classifier based on a deep neural network returns a probability of whether the target is presented in each of the image patches defined by the bounding boxes. The result is then used to correct the tracker's output or to determine that the target is out of view and perform re-detection using the tracker with higher localization accuracy.

Conclusion remarks on deep learning-based trackers. Deep learning-based methods have made significant progress in terms of tracking accuracy and robustness. This became mainly due to the availability of large-scale labeled datasets and the development of effective network architectures. This progress has expanded the applicability of modern trackers in scenarios where reliability is critical. However, as will be discussed in the next section, the increase in accuracy often comes at the cost of higher computational complexity.

Benchmark datasets. The availability of large and high-quality datasets is essential for tracker development, evaluation, and debugging. They can also serve as a benchmark that allows to compare different trackers and present results. Typically, object tracking datasets consist of video sequences of variable length, where the target's position is labeled in each video frame either with a bounding box or a segmentation mask. Additionally, some metainformation might be provided, for example, types of visual distractors presented in the video. This helps to filter the dataset properly for further processing. Typically, benchmark datasets are accompanied by standardized evaluation protocols, allowing for a fair comparison of various algorithms. Below, an overview of widely used datasets for visual tracking is provided, including those focused on short-term and long-term tracking.

Visual Object Tracking (VOT) is an annual challenge that provides standardized datasets, an evaluation protocol, and a leaderboard to assess the performance of both short-term and long-term trackers [45]. Since its launch in 2013, this platform has played a central role in advancing the field of object tracking.

The VOT framework introduces several categories of challenges and corresponding datasets:

1. VOT-ST: a short-term benchmark that contains shorter videos without long-term target occlusions.
2. VOT-LT: a long-term benchmark consisting of longer videos with target disappearance and reappearance.
3. VOT-RGBD and VOT-RGBT: these benchmarks offer four-channel videos, including color RGB and an additional channel, either depth (D) or an infrared channel (T).
4. VOTs – a target position is annotated with a segmentation mask.

New challenges focus on tracking with segmentation instead of bounding box estimation and take into account topological transformations of objects, such as cut or disassembled forms. Starting from 2023, the challenge no longer distinguishes between single-object and multi-object tracking, nor between short-term and long-term tracking.

Generic Object Tracking (GOT) benchmark is a large-scale dataset that offers over 10,000 video segments with more than 1,5 million manually labeled bounding boxes [46]. Object classes from training and test sets do not overlap; therefore, it promotes generalization in tracker development. The overall number of classes is 563. Additionally, the dataset provides annotations of visible ratios, which are a percentage indicating the approximate proportion of an object that is visible. This information can help to develop a model that is more robust to target occlusion.

Large-scale Object Tracking (LaSOT) benchmark also aims to provide a comprehensive platform for training and evaluating deep trackers [47]. The dataset contains 1,550 video sequences totaling over 3,87 million frames. The overall number of object categories is eighty-five, with each category represented by an equal number of sequences to ensure category balance and reduce bias. LaSOT presents several real-world tracking challenges, including occlusion, scale variation, motion blur, illumination change, background clutter, and target disappearance.

The Unmanned Aerial Vehicle Detection and Tracking (UAVDT) [48] benchmark is designed for evaluating object detection and tracking algorithms in aerial videos captured by drones. It addresses challenges unique to UAV-based scenarios, such as small object sizes, dynamic camera motion, and complex backgrounds. It contains 100 video sequences, resulting in 80,000 frames that are fully annotated with bounding boxes and up to 14 attributes (weather condition, camera view, flying altitude, vehicle category, whether the target is occluded or not etc.). The benchmark is suitable for three computer vision tasks: object detection, single-object tracking, and multiple-object tracking.

Tracking methods comparison. VOT Benchmark. VOT Challenge employs different evaluation protocols for assessing short-term and long-term trackers. For short-term tracking, the protocol adopts the so-called reset strategy. When a tracking failure is detected, which is defined as a zero overlap between the predicted and ground-truth bounding boxes, the tracker is reset after five frames. To avoid bias from re-initialization, the ten frames following the reset are excluded from metric computation. Three primary metrics are used for short-term tracker evaluation [2]:

1. Accuracy (A) is the average overlap between predicted and ground-truth bounding boxes.
2. Robustness (R) is the average number of tracking failures per video sequence. In this case, lower robustness means better tracking performance.
3. Expected Average Overlap (EAO) is an integrated metric that estimates the expected overlap a tracker would achieve on a typical short-term sequence without resets. EAO combines both per-frame accuracy and robustness into a single score, making it the primary ranking criterion in VOT short-term evaluations.

In contrast to VOT short-term scenarios, the main requirement for long-term trackers is the ability to re-detect the target or to report that the target is missing. Long-term trackers should also provide a confidence score that reflects whether the target is within the field of view. Therefore, a VOT long-term protocol uses another set of quality metrics:

1. Precision (Pr): mean intersection-over-union (IoU) over frames where the tracker predicted the target's location.
2. Recall (Re): mean IoU over frames where the target is within field of view.
3. F-score: a combination of precision and recall: $F = \frac{2 \cdot Pr \cdot Re}{Pr + Re}$.

Table 1 presents the results on the VOT 2018 short-term benchmark, where trackers' performance is sorted by EAO values. Table 2 demonstrates long-term performance on the VOT 2018 LT benchmark, with results sorted by F-score. All the results can be found in the official report dedicated to the VOT 2018 challenge [2] and corresponding works.

TABLE 1. VOT 2018 short-term challenge evaluation results

Tracker	Tracker Type	Accuracy	Robustness	Expected Average Overlap
SiamRPN [32]	Deep, Siamese	0.586	0.276	0.383
ECO [13]	Correlation Filter	0.484	0.276	0.280
C-COT [12]	Correlation Filter	0.494	0.318	0.267
SiamFC [30]	Deep, Siamese	0.503	0.585	0.188
Staple [17]	Correlation Filter	0.530	0.688	0.169
KCF [4]	Correlation Filter	0.447	0.773	0.135
DSST [7]	Correlation Filter	0.395	1.452	0.079

TABLE 2. VOT 2018 long-term challenge evaluation results

Tracker	Tracker Type	Precision	Recall	F-score
CoCoLoT [44]	Deep	0.741	0.729	0.735
STARK [39]	Deep, Transformer	0.691	0.652	0.696
LTMU [43]	Deep	0.710	0.672	0.690
Siam R-CNN [35]	Deep, Siamese	0.670	0.667	0.668
SPLT [41]	Deep	0.633	0.600	0.616
FuCoLoT [16]	Correlation Filter	0.539	0.432	0.480
SiamFC [30]	Deep, Siamese	0.636	0.328	0.433

LaSOT Benchmark. The LaSOT evaluation procedure uses three kinds of evaluation protocols: *no-constraint*, *full-overlap,* and *one-shot* protocols [47]. In the first protocol, all 1,400 videos in LaSOT are utilized for evaluation. In the full-overlap protocol, only 280 sequences from the test set are used to evaluate the performance of trackers. Another 1,220 videos can be used for tracker training. The testing set for the one-shot protocol consists of 150 specially collected videos, enabling the use of 1,400 LaSOT videos for training. Moreover, there is no overlap between the target object classes of the training and testing subsets. In this paper, the results of the one-shot protocol are presented in Table 3.

Across all evaluation protocols, three standard metrics are used to compare tracker performance: precision (PRE), normalized precision (N-PRE), and success (SUC). Precision is calculated as the percentage of

frames in which the center location error between the predicted and ground-truth bounding boxes is within a predefined threshold (typically 20 pixels). Normalized precision accounts for object scale variations and normalizes the precision by the size of the ground-truth bounding box (e.g., the diagonal length of the bounding box). The success rate is defined as the ratio of frames where the intersection-over-union between the ground-truth and predicted bounding boxes exceeds a certain threshold (typically 0.5). It is common to plot these metrics as a function of threshold and to obtain a precision plot, a normalized precision plot, and a success plot. These plots provide a more detailed view of a tracker's behavior across different levels of tolerance, offering insight beyond a single scalar score. More details can be found in [47] and in the official LaSOT leaderboard [49].

TABLE 3. LaSOT evaluation results using one-shot protocol

Tracker	Tracker Type	PRE	N-PRE	SUC
STARK-ST101 [39]	Deep, Transformer, long-term	–	0.77	0.671
TransT [38]	Deep, Transformer, short-term	0.690	0.738	0.649
SiamAttn [35]	Deep, Siamese, short-term	–	0.648	0.560
LTMU [43]	Deep, long-term	0.473	0.499	0.414
DaSiamRPN [33]	Deep, Siamese, long-term	0.420	0.443	0.356
GlobalTrack [42]	Deep, long-term	0.411	0.436	0.356
SiamMask [34]	Deep, Siamese, short-term	0.392	0.420	0.332
SPLT [41]	Deep, long-term	0.297	0.339	0.272
SiamFC [30]	Deep, Siamese, short-term	0.269	0.311	0.230
ECO [13]	Correlation Filter, short-term	0.240	0.252	0.220
PTAV [40]	Deep, long-term	0.222	0.220	0.195
ECO_HC (hand-crafted features) [13]	Correlation Filter, short-term	0.210	0.222	0.182
HCFT [10]	Correlation Filter, short-term	0.183	0.181	0.159
Staple [17]	Correlation Filter, short-term	0.165	0.187	0.158
LCT [14]	Correlation Filter, long-term	0.140	0.155	0.143
fDSST [7]	Correlation Filter, short-term	0.138	0.152	0.135
KCF [4]	Correlation Filter, short-term	0.134	0.148	0.127
CN [8]	Correlation Filter, short-term	0.129	0.140	0.121

GOT-10k benchmark. There are three main metrics used to compare trackers' performance on the GOT-10k benchmark: The Average Overlap (AO) simply refers to the average overlap rate between ground truth and predicted bounding boxes across all video frames [46]. The Success Rate (SR) denotes the proportion of frames where the overlap exceeds a certain threshold; commonly used thresholds are 0.5 and 0.75. Table 4 presents the evaluation results on the GOT-10k dataset using these metrics. Additionally, the speed of each tracker, expressed in frames per second (FPS), is provided along with the device name used for testing. More information regarding other trackers can be found in the official GOT-10k leaderboard [50].

Some studies, report results using the mean average overlap (mAO) and mean success rate (mSR). These metrics calculate the average score per video sequence class and then take the mean across all classes,

providing a fairer evaluation on unbalanced datasets. However, the official GOT-10k leaderboard and many papers use simpler AO and SR metrics; therefore, they are provided in this paper as well.

TABLE 4. GOT-10k evaluation results

Tracker	Tracker Type	Average Overlap (AO)	$SR_{0.5}$	$SR_{0.75}$	Speed, FPS	Device
TransT [38]	Deep, Transformer, short-term	0.723	0.824	0.682	47.27	Titan RTX
STARK-ST50 [39]	Deep, Transformer, long-term	0.680	0.777	0.623	41.8	GPU (Unknown model)
Siam R-CNN [36]	Deep, Siamese, long-term	0.649	0.728	0.597	2.79	GeForce GTX 1080Ti
SiamRPN (with ResNet50 backbone) [32]	Deep, Siamese, short-term	0.516	0.620	0.334	26.68	GeForce GTX 1080Ti
SiamMask [34]	Deep, Siamese, short-term	0.514	0.587	0.366	15.37	NVIDIA Tesla P100
GOTURN [31]	Deep, Siamese	0.347	0.375	0.124	108.55	GeForce GTX Titan X
SiamFC [30]	Deep, Siamese	0.348	0.353	0.098	44.15	GeForce GTX Titan X
CCOT[12]	Correlation Filter, short-term	0.325	0.328	0.107	0.68	16 cores 2.0 GHz CPU
ECO_HC (handcrafted features) [13]	Correlation Filter, short-term	0.286	0.276	0.096	44.55	16 cores 2.0 GHz CPU
fDSST [7]	Correlation Filter, short-term	0.206	0.187	0.075	30.43	16 cores 2.0 GHz CPU
KCF [4]	Correlation Filter, short-term	0.203	0.177	0.065	94.66	16 cores 2.0 GHz CPU

Conclusions. To summarize, visual object tracking remains a challenging problem despite significant progress made over the last decade. The results from the evaluation benchmark clearly show the following:

1. Significant progress in the object tracking domain has been made by deep learning methods, starting with Siamese models. Transformer-based models are currently among the state-of-the-art. Unlike correlation filter approaches, most deep learning methods are trained offline on large-scale datasets for tracking. Due to the large number of parameters in deep trackers, their online updates are often computationally inefficient.

2. Although state-of-the-art deep trackers achieve decent performance on benchmark datasets, they are often not suitable for computationally constrained environments, such as UAVs. To overcome this issue, more lightweight feature extractors and model quantization techniques can be considered [51], though often at the cost of tracking accuracy.

3. In contrast, classic approaches, such as correlation filters, are well-suited for real-time applications and can be effectively adapted for long-term tracking scenarios that are common in practice. Most correlation-based methods are trained online, which allows the model to be flexible and adapt to changes in the environment. However, in the long-term tracking, the problems of model update frequency and efficient

discriminative features must still be addressed. On the other hand, correlation-based trackers can be combined with keypoint detection, which can improve long-term tracking.

The analysis of modern long-term trackers shows that target re-detection and minimization of error drift are the most critical aspects of accurate tracking. Moreover, many practical applications employ computational constraints. Future research in the visual object tracking domain should focus on developing real-time, long-term tracking solutions that account for these constraints.

Authorship contribution. Yevhen Romaniak – research, analysis, writing.

Data availability. The data that supports the findings of this study is openly available in publications devoted to particular trackers, VOT 2018 Challenge report [2] and official public leaderboards [49, 50].

Funding. The author(s) received no financial support for the research, authorship and/or publication of this article.

References

1. Suslenko O. Comparison of Neural Network Architectures for Spatial Orientation of Unmanned Aerial Vehicles and Ground Drones. *Cybernetics and Computer Technologies*. 2025. 4. P. 99–105. (in Ukrainian) https://doi.org/10.34229/2707-451X.25.4.9
2. Kristan M., Leonardis A., Matas J. et al. The Sixth Visual Object Tracking VOT2018 Challenge Results. *Computer Vision – ECCV 2018 Workshops*. Cham : Springer International Publishing, 2019. ISBN 978-3-030-11009-3. P. 3–53. https://doi.org/10.1007/978-3-030-11009-3_1
3. Bolme D.S., Beveridge J.R., Draper B.A., Lui, Y. M. Visual object tracking using adaptive correlation filters. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. P. 2544–2550. https://doi.org/10.1109/CVPR.2010.5539960
4. Henriques J.F., Caseiro R., Martins P., Batista J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **37** (3). P. 583–596. https://doi.org/10.1109/TPAMI.2014.2345390
5. Galoogahi H.K., Sim T., Lucey S. Multi-channel Correlation Filters. *2013 IEEE International Conference on Computer Vision2013 IEEE International Conference on Computer Vision (ICCV)*. (Sydney, Australia, 12.2013). Sydney, Australia : IEEE, 2013. P. 3072–3079. https://doi.org/10.1109/ICCV.2013.381
6. Li Y., Zhu J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. *Computer Vision – ECCV 2014 Workshops*. Cham : Springer International Publishing, 2015. P. 254–265. https://doi.org/10.1007/978-3-319-16181-5_18
7. Danelljan M., Hager G., Khan F.S., Felsberg M. Discriminative Scale Space Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **39** (8). P. 1561–1575. https://doi.org/10.1109/TPAMI.2016.2609928
8. Danelljan M., Khan F.S., Felsberg M., Van de Weijer J. Adaptive Color Attributes for Real-Time Visual Tracking. *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Columbus, OH, 06.2014). Columbus, OH : IEEE, 2014. P. 1090–1097. https://doi.org/10.1109/CVPR.2014.143
9. Danelljan M., Häger G., Khan F.S., Felsberg M. Coloring Channel Representations for Visual Tracking. *Image Analysis*. P. 117–129. https://doi.org/10.1007/978-3-319-19665-7_10
10. Ma C., Huang J.-B., Yang X., Yang M.H. Hierarchical Convolutional Features for Visual Tracking. *2015 IEEE International Conference on Computer Vision (ICCV)2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile : IEEE, 2015. P. 3074–3082. https://doi.org/10.1109/ICCV.2015.352
11. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Las Vegas, NV, USA, 06.2016). Las Vegas, NV, USA : IEEE, 2016. P. 770–778. https://doi.org/10.1109/CVPR.2016.90
12. Danelljan M., Robinson A., Khan F.S., Felsberg M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. *Computer Vision – ECCV 2016*. P. 472–488. https://doi.org/10.1007/978-3-319-46454-1_29
13. Danelljan M., Bhat G., Khan F.S., Felsberg M. ECO: Efficient Convolution Operators for Tracking. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Honolulu, HI, 07.2017). Honolulu, HI : IEEE, 2017. P. 6931–6939. https://doi.org/10.1109/CVPR.2017.733
14. Ma C., Yang X., Zhang C., Yang, M. H. Long-term correlation tracking. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Boston, MA, USA, 06.2015). Boston, MA, USA : IEEE, 2015. P. 5388–5396. https://doi.org/10.1109/CVPR.2015.7299177

15. Ma C., Huang J.-B., Yang X., Yang M. H. Adaptive Correlation Filters with Long-Term and Short-Term Memory for Object Tracking. *International Journal of Computer Vision*. **126** (8). P. 771–796. https://doi.org/10.1007/s11263-018-1076-4
16. Lukežič A., Zajc L.Č., Vojíř T., Matas J., Kristan M. FuCoLoT – A Fully-Correlational Long-Term Tracker. *Computer Vision – ACCV 2018*. P. 595–611. https://doi.org/10.1007/978-3-030-20890-5_38
17. Bertinetto L., Valmadre J., Golodetz S., Miksik O., Torr P.H. Staple: Complementary Learners for Real-Time Tracking. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Las Vegas, NV, USA, 06.2016). Las Vegas, NV, USA : IEEE, 2016. P. 1401–1409. https://doi.org/10.1109/CVPR.2016.156
18. Kyyko V., Matsello V. Real-Time Tracking of Objects in Video Based on Adaptive Histogram Features. *Kibernetika i vyčislitel'naâ tehnika*. 2023. **3** (213). P. 4–19. (In Ukrainian) https://doi.org/10.15407/kvt213.03.004
19. Yang J., Tang W., Ding Z. Long-Term Target Tracking of UAVs Based on Kernelized Correlation Filter. *Mathematics*. **23** (6). P. 3006. https://doi.org/10.3390/math9233006
20. Shi J., Tomasi C. Good features to track. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. (Seattle, WA, USA, 1994). Seattle, WA, USA : IEEE Comput. Soc. Press, 1994. P. 593–600. https://doi.org/10.1109/CVPR.1994.323794
21. Lowe D.G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. **60** (2). P. 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94
22. Rublee E., Rabaud V., Konolige K., Bradski G.R. ORB: An efficient alternative to SIFT or SURF. *2011 IEEE International Conference on Computer Vision (ICCV)*. (Barcelona, Spain, 11.2011). Barcelona, Spain : IEEE, 2011. P. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544
23. DeTone D., Malisiewicz T., Rabinovich A. SuperPoint: Self-Supervised Interest Point Detection and Description. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. (Salt Lake City, UT, USA, 06.2018). Salt Lake City, UT, USA : IEEE, 2018. P. 337–33712. https://doi.org/10.1109/CVPRW.2018.00060
24. Lucas B.D., Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision. *IJCAI'81: 7th international joint* conference *on Artificial intelligence.* Vancouver, Canada, 1981. P. 674–679. https://hal.science/hal-03697340
25. Kalal Z., Mikolajczyk K., Matas J. Forward-Backward Error: Automatic Detection of Tracking Failures. *2010 20th International Conference on Pattern Recognition (ICPR)*. (Istanbul, Turkey, 08.2010). Istanbul, Turkey : IEEE, 2010. P. 2756–2759. https://doi.org/10.1109/ICPR.2010.675
26. Nebehay G., Pflugfelder R. Consensus-based matching and tracking of keypoints for object tracking. *2014 IEEE Winter Conference on Applications of Computer Vision (WACV)*. (Steamboat Springs, CO, USA, 03.2014). Steamboat Springs, CO, USA : IEEE, 2014. P. 862–869. https://doi.org/10.1109/WACV.2014.6836013
27. Nebehay G., Pflugfelder R. Clustering of static-adaptive correspondences for deformable object tracking. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Boston, MA, USA, 06.2015). Boston, MA, USA : IEEE, 2015. P. 2784–2791. https://doi.org/10.1109/CVPR.2015.7298895
28. Hong Z., Chen Z., Wang C., Mei X., Prokhorov D., Tao D. MUlti-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Boston, MA, USA, 06.2015). Boston, MA, USA : IEEE, 2015. P. 749–758. https://doi.org/10.1109/CVPR.2015.7298675
29. Derue F.-X., Bilodeau G.-A., Bergevin R. SPiKeS: Superpixel-Keypoints structure for robust visual tracking. *Machine Vision and Applications*. **29** (1). P. 175–186. https://doi.org/10.1007/s00138-017-0884-9
30. Bertinetto L., Valmadre J., Henriques J.F., Vedaldi A., Torr P.H. Fully-Convolutional Siamese Networks for Object Tracking. *Computer Vision – ECCV 2016 Workshops*. P. 850–865. https://doi.org/10.1007/978-3-319-48881-3_56
31. Held D., Thrun S., Savarese S. Learning to Track at 100 FPS with Deep Regression Networks. *Computer Vision – ECCV 2016*. Cham : Springer International Publishing, 2016. P. 749–765. https://doi.org/10.1007/978-3-319-46448-0_45
32. Li B., Yan J., Wu W., Zhu Z., Hu X. High Performance Visual Tracking with Siamese Region Proposal Network. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Salt Lake City, UT, 06.2018). Salt Lake City, UT : IEEE, 2018. https://doi.org/10.1109/CVPR.2018.00935
33. Zhu Z., Wang Q., Li B., Wu W., Yan J., Hu W. Distractor-Aware Siamese Networks for Visual Object Tracking. *Computer Vision – ECCV 2018*. P. 103–119. https://doi.org/10.1007/978-3-030-01240-3_7
34. Hu W., Wang Q., Zhang L., Bertinetto L., Torr, P. H. Siammask: A framework for fast online object tracking and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2023. **45** (3), P. 3072–3089. https://ieeexplore.ieee.org/document/10036241

35. Yu Y., Xiong Y., Huang W., Scott M.R. Deformable Siamese Attention Networks for Visual Object Tracking. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Seattle, WA, USA, 06.2020). Seattle, WA, USA : IEEE, 2020. P. 6727–6736. https://doi.org/10.1109/CVPR42600.2020.00676
36. Voigtlaender P., Luiten J., Torr P.H.S., Leibe B. Siam R-CNN: Visual Tracking by Re-Detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Seattle, WA, USA, 06.2020). Seattle, WA, USA : IEEE, 2020. P. 6577–6587. https://doi.org/10.1109/CVPR42600.2020.00661
37. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I. Attention is All you Need. *Advances in Neural Information Processing Systems* (2017). Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
38. Chen X., Yan B., Zhu J., Wang D., Yang X., Lu H. Transformer Tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Nashville, TN, USA, 06.2021). Nashville, TN, USA : IEEE, 2021. P. 8122–8131. https://doi.org/10.1109/CVPR46437.2021.00803
39. Yan B., Peng H., Fu J., Wang D., Lu H. Learning Spatio-Temporal Transformer for Visual Tracking. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. (Montreal, QC, Canada, 10.2021). Montreal, QC, Canada : IEEE, 2021. P. 10428–10437. https://doi.org/10.1109/ICCV48922.2021.01028
40. Fan H., Ling H. Parallel Tracking and Verifying: A Framework for Real-Time and High Accuracy Visual Tracking. *2017 IEEE International Conference on Computer Vision (ICCV)*. (Venice, 10.2017). Venice : IEEE, 2017. P. 5487–5495. https://doi.org/10.1109/ICCV.2017.585
41. Yan B., Zhao H., Wang D., Lu H., Yang X. 'Skimming-Perusal' Tracking: A Framework for Real-Time and Robust Long-Term Tracking. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. (Seoul, Korea (South), 10.2019). Seoul, Korea (South) : IEEE, 2019. P. 2385–2393. https://doi.org/10.1109/ICCV.2019.00247
42. Huang L., Zhao X., Huang K. GlobalTrack: A Simple and Strong Baseline for Long-Term Tracking. *Proceedings of the AAAI Conference on Artificial Intelligence*. **34** (7). P. 11037–11044. https://doi.org/10.1609/aaai.v34i07.6758
43. Dai K., Zhang Y., Wang D., Li J., Lu H., Yang X. High-Performance Long-Term Tracking With Meta-Updater. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Seattle, WA, USA, 06.2020). Seattle, WA, USA : IEEE, 2020. P. 6297–6306. https://doi.org/10.1109/CVPR42600.2020.00633
44. Dunnhofer M., Micheloni C. CoCoLoT: Combining Complementary Trackers in Long-Term Visual Tracking. *2022 26th International Conference on Pattern Recognition (ICPR)*. (Montreal, QC, Canada, 21.08.2022). Montreal, QC, Canada : IEEE, 2022. P. 5132–5139. https://doi.org/10.1109/ICPR56361.2022.9956082
45. Kristan M., Matas J., Leonardis A., Vojír T., Pflugfelder R.P., Fernandez G.J., Nebehay G., Porikli F.M., Cehovin L. A Novel Performance Evaluation Methodology for Single-Target Trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **38** (11). P. 2137–2155. https://doi.org/10.1109/TPAMI.2016.2516982
46. Huang L., Zhao X., Huang K. GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **43** (5). P. 1562–1577. https://doi.org/10.1109/TPAMI.2019.2957464
47. Fan H., Lin L., Yang F., Chu P., Deng G., Yu S., Bai H., Xu Y., Liao C., Ling, H. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Long Beach, CA, USA, 06.2019). Long Beach, CA, USA : IEEE, 2019. P. 5369–5378. https://doi.org/10.1109/CVPR.2019.00552
48. Yu H., Li G., Zhang W., Huang Q., Du D., Tian Q., Sebe N. The Unmanned Aerial Vehicle Benchmark: Object Detection, Tracking and Baseline. *International Journal of Computer Vision*. **128** (5). P. 1141–1159. https://doi.org/10.1007/s11263-019-01266-1
49. LaSOT – Large-scale Single Object Tracking. http://vision.cs.stonybrook.edu/~lasot/results.html (accessed: 17.09.2025)
50. GOT-10k: Generic Object Tracking Benchmark. http://got-10k.aitestunion.com/leaderboard (accessed: 17.09.2025)
51. Swati Kumar V. N., Dinesh Kawa S., Engineer P.J. An Efficient Object Tracking on Edge Devices with Quantized Siamese Networks. *2025 Devices for Integrated Circuit (DevIC)*. (Kalyani, India, 05.04.2025). Kalyani, India : IEEE, 2025. P. 604–609. https://doi.org/10.1109/DevIC63749.2025.11012629

Received/Одержано 02.09.2025 Accepted/Прийнято 03.03.2026 Published/Надруковано 27.03.2026

Yevhen Romaniak,
PhD student, Institute of Information Technologies and Systems of the NAS of Ukraine, Kyiv.
https://orcid.org/0009-0007-9543-9356
romanyak.yevhen@gmail.com

УДК 004.93

Є.С. Романяк

Аналіз методів відстеження об’єктів у відео

Інститут інформаційних технологій та систем НАН України, Київ
Листування: romanyak.yevhen@gmail.com

Вступ. Візуальне простеження (трекінг) об’єктів це важлива задача у комп’ютерному зорі, що має широкий спектр застосувань, включаючи автономну навігацію, робототехніку та задачі моніторингу. Завдання полягає в оцінці положення об'єкта у послідовності відеокадрів, виходячи з його положення на початковому кадрі. Незважаючи на значні зусилля дослідників, завдання залишається складним через такі чинники як перекриття цілі, зміни умов освітлення, розмиття в русі та деформації об’єкта. Методи простеження поділяються на короткострокові, де припускається, що ціль залишається у полі зору, і довгострокові, які обробляють зникнення об’єкта і його повторну появу. Дана стаття надає глибокий аналіз різних методів відстеження одного об'єкта, охоплюючи як традиційні підходи, такі як трекери з використанням кореляційних фільтрів та ключових точок, так і сучасні методи глибокого навчання.

Мета роботи – полягає в аналізі різних алгоритмів відстеження об’єкта, як короткострокових, так і довгострокових, а також наборів даних, що використовуються для оцінки якості цих алгоритмів. Стаття має на меті розглянути основні принципи різних підходів до відстеження, включаючи кореляційні фільтри, методи, засновані на відстеженні ключових точок, і різні моделі глибокого навчання, такі як сіамські нейронні мережі, трансформери та інші. Крім того, дослідження представляє огляд популярних еталонних наборів даних, таких як VOT 2018, LaSOT і GOT-10k, та порівнює точність багатьох з розглянутих у статті алгоритмів на цих бенчмарках. Це порівняння висвітлює сильні та слабкі сторони різних підходів до трекінгу і створює основу для майбутніх напрямків досліджень, зокрема, щодо підвищення ефективності, адаптивності та швидкодії алгоритмів відстеження для застосувань у реальних задачах.

Результати. *Кореляційні трекери* відомі своєю високою швидкістю та хорошою точністю. Ці методи використовують перетворення Фур’є для ефективних обчислень, точність яких може бути покращена за допомогою ознакового опису зображення, від HOG ознак, до представлення, отриманого глибокою згортковою нейронною мережею. Однак, вони потребують модифікацій для довгострокового відстеження, щоб справлятися зі зникненням об’єкта і зменшувати накопичення помилок. Хоча деякі з розглянутих методів враховують ці складнощі, вони не вирішують їх повністю. *Трекери, що використовують ключові точки*, відстежують об’єкти, ідентифікуючи та зіставляючи цікаві точки або ознаки в кадрах. Такі методи, як метод Канаде – Лукаса – Томасі, це основа, тоді як детектори SIFT або ORB підвищують стійкість до шуму та змін масштабу. Ці трекери є особливо корисними для сценаріїв, де об’єкт частково перекривається, оскільки вони можуть відстежувати підмножину точок об’єкта. Однак вони можуть мати проблеми з мало текстурованими та маленькими об’єктами. Трекери, засновані на глибокому навчанні, є значним досягненням, що перевершує традиційні методи за точністю та стійкістю завдяки їх можливостям представлення ознак зображення. Деякі глибокі трекери, такі як SiamFC чи SiamRPN, демонструють високу точність та швидкодію у реальному часі на графічному процесорі (GPU). Порівняння алгоритмів на бенчмарках, таких як VOT 2018, LaSOT та GOT-10k, показує, що підходи, засновані на глибокому навчанні демонструють вищу точність у складних сценаріях відстеження, але потребують більш значних обчислювальних ресурсів.

Висновки. Алгоритми простеження об’єктів значно еволюціонували з появою методів глибокого навчання, що дозволило трекерам досягти вищої точності та стійкості порівняно з традиційними методами, наприклад, на основі кореляційних фільтрів чи ключових точок. Поява великого об’єму розмічених даних, наприклад, датасетів VOT, GOT-10k та LaSOT, стало важливим кроком у розроблені глибоких трекерів та стандартизованої основи для оцінки нових алгоритмів. Хоча методи на основі кореляційних фільтрів або ключових точок все ще використовуються, наприклад, у середовищах з обмеженими обчислювальними ресурсами, трекери, засновані на глибокому навчанні, зокрема, сіамські мережі та трансформери, показують найкращу точність. Подальші дослідження повинні зосередитися на оптимізації ефективності та адаптивності цих алгоритмів, щоб зробити їх більш придатними для застосувань у реальному часі та різноманітних реальних сценаріїв.

Ключові слова: візуальне відстеження об’єктів, відстеження одного об’єкту, кореляційні фільтри, відстеження ключових точок, сіамські нейронні мережі, трансформери.