A Research of Data Representation, Event-Based Vision and Hardware Implementation

Hanze Jin *

School of Electrical Engineering and Information Technology, RWTH Aachen University, Aachen, Germany

* Corresponding Author Email: jinhz58@outlook.com

Abstract. The high-speed object tracking is not only a problem of tracking fast-moving targets, but also a system-level challenge involving visual sensing, data representation, motion modeling, target association, and hardware implementation. This paper reviews object-recognition-driven methods for high-speed object tracking, with a focus on data representation, motion information extraction, event-based vision, and hardware system design. First, frame-based image sequences, optical flow, and event streams are compared in terms of their ability to represent target motion under high-speed conditions. Then, traditional motion cues, deep representation methods, and lightweight event-driven features are discussed from the perspectives of robustness, real-time performance, and deployability. The paper further analyzes the role of high-speed visual sensors, event cameras, FPGAs, and hardware–software co-design in practical tracking systems. Existing studies show that event-based vision and dedicated hardware acceleration can reduce redundant computation and improve temporal response, but stable association, resource efficiency, and real-world deployment remain open challenges. This review argues that future high-speed tracking systems should be developed through the co-design of sensors, algorithms, and hardware architectures.

Keywords: High-Speed Object Tracking, Event-Based Vision, Event based Cameras.

1. Introduction

Object tracking is one of the fundamental problems in computer vision. It aims to continuously estimate the position, scale, pose, and motion trajectory of a target in consecutive images or video sequences. As a key component of visual perception systems, object tracking has direct practical value in applications such as intelligent surveillance, autonomous driving, robotic navigation, and human–computer interaction [1, 2]. Once the tracking results become unstable, subsequent tasks such as behavior analysis, risk warning, and path planning may also be affected. Therefore, object tracking has long remained a central research direction in computer vision.

Conventional object tracking methods are mostly built on frame-based images, and continuous localization is usually achieved through object detection, feature extraction, state estimation, and trajectory updating. These methods have developed into relatively mature technical pipelines in standard scenarios. However, they still show clear limitations under high-speed motion and complex illumination conditions. Fixed-frame-rate sampling can cause fast-moving targets to undergo large displacements between adjacent frames, leading to motion blur, loss of edge information, and abrupt scale changes. Meanwhile, full-frame acquisition introduces a large amount of static background information, resulting in substantial data redundancy and computational load, which further affects real-time performance [3]. Therefore, high-speed object tracking is not simply a matter of increasing detection speed. Instead, it is a comprehensive problem shaped by the joint constraints of temporal resolution, feature representation, association stability, and system efficiency.

In recent years, with the development of deep learning, object tracking has gradually shifted from traditional methods based on handcrafted features to learning-based methods centered on detection, representation learning, and trajectory association. In multi-object tracking, the tracking-by-detection paradigm has become dominant: objects are first detected, and trajectories are then maintained by combining appearance features with motion models. StrongSORT shows that improving the detector, embedding model, and association-completion strategies within the DeepSORT framework can effectively enhance multi-object tracking performance [4]. This reflects a broader trend toward the coordinated optimization of detection, embedding, and association, rather than the optimization of a single tracker alone.

In addition to algorithmic advances, visual sensors are also reshaping high-speed object tracking. Unlike conventional cameras that output full frames at a fixed frame rate, event cameras asynchronously generate events when pixel-level brightness changes occur. This sensing mechanism provides high temporal resolution, high dynamic range, low latency, and low power consumption [5]. Compared with frame-based images, event streams are sparser but preserve motion-related information, making them advantageous for high-speed motion, rapidly changing illumination, and edge deployment. Existing studies show that event-stream-based ROI extraction, optical flow estimation, and lightweight trajectory updating can achieve real-time object tracking at relatively low computational cost while reducing the effects of motion blur.

However, high-speed object tracking still faces several unresolved challenges. Frame-based images are vulnerable to distortion in fast-motion scenarios, whereas event data are asynchronous, sparse, and less structured. This makes stable extraction of discriminative features difficult [5]. At the algorithmic level, high-speed motion, occlusion, dense targets, and complex backgrounds often lead to false detections, missed detections, identity switches, and trajectory interruptions. The association stage is especially fragile in multi-object tracking [4]. From a system perspective, good results on public datasets do not necessarily translate into practical deployment, where throughput, latency, power consumption, and hardware resources must be considered together. In response, recent studies have increasingly explored event-based vision representations, the joint optimization of detection and association, and real-time event-based processing architectures for embedded platforms such as FPGAs [6].

Against this background, this paper reviews object-recognition-driven methods for high-speed tracking from two perspectives: visual signal modeling and tracking system design. It summarizes the main ideas, applicable scenarios, and limitations of representative studies, and outlines the technical issues that remain open. The rest of the paper is organized as follows. Section 2 discusses data representation and motion information extraction in high-speed object tracking. Section 3 reviews object-recognition-driven tracking algorithms and their system-level implementation mechanisms.

2. Motion Modeling in High-Speed Tracking

High-speed object tracking depends first on how visual information is observed, represented, and converted into motion cues. In this chapter, data representation and modeling refer to two closely related processes: the acquisition of raw visual observations and their transformation into images, event streams, or spatio-temporal features; and the construction of motion models that describe changes in target states over time. For high-speed object tracking, the input format is therefore not just a sensor choice. It shapes how targets are recognized, how trajectories are associated and predicted, and whether the system can meet real-time deployment requirements.

2.1. Visual Data Acquisition and Representation

High-speed object tracking depends first on the way visual data is acquired. Conventional vision systems usually capture full-frame images at a fixed frame rate and estimate target states from positional changes between consecutive frames. When a target moves very fast, however, the displacement between adjacent frames increases markedly, which may lead to blurred edges, missing details, and discontinuous motion representation. Full-frame sampling also records a large amount of static background information that is irrelevant to motion, increasing the cost of data transmission and computation in subsequent detection, feature extraction, and trajectory updating. In high-speed scenarios, visual data therefore needs to be not only captured, but also represented in a suitable form. A useful representation should describe target changes at a finer temporal granularity while preserving as much motion-relevant information as possible [5].

In traditional object tracking, frame-based image sequences are still the most popular type of data. The video, in this context, is considered to be a temporally homogeneous series of two-dimensional images. Every frame is stored with target appearance, texture of the background, and light source data, and the following algorithms can be constructed based on the object detection, appearance model, motion prediction, and data association. StrongSORT can be regarded as the tracking-by-detection paradigm, which is based on this frame-sequence representation. Du et al. based on DeepSORT presented a more powerful detector, a ReID embedding model, and association-completion and trajectory-smoothing methods to allow detection, representation, and association in the context of multi-object tracking to be optimized jointly. This example implies frame-based sequences are still efficient in the case when the quality of the images is constant and the desired speed is moderate. Simultaneously, they rely on inter-frame continuity and sharp imaging, which also reveal certain drawbacks in the situations of high speeds [4].

Optical flow is concerned with time-varying pixel or local-area motions as compared with simply updating the target positions in response to the detection boxes. It can then be considered as a motion based visual representation. Instead of classifying the target, optical flow estimates its motion, direction, and local structural variations on the image plane based on the change in the intensity of the image or the activity of the event on the image plane. This can be handy when analyzing fast-moving objects, as it can further describe a target, but instead of describing its location, it describes its motion. Conventional optical flow techniques tend to rely on the assumption of brightness constancy and compute the motion field by using gradient constraints or local plane fitting. They have the ability to indicate the velocity of moving objects in an ideal scenario without prior information of the scene. Practically, though, they are computationally expensive, need quite powerful hardware support and are sensitive to noise, restricting their applications in real-time high-speed tracking.

These limitations become more evident in high-speed scenarios. Frame-based images are prone to motion blur and loss of information with increasing target speed, and the inter-frame intensity change-based optical flow estimation is less robust. The other type of input to optical flow estimation is event cameras. They do not wait until the entire image frames are finished, but, as pixel-level brightness changes happen, output events with position, timestamp and polarity information. By so doing, they capture motion variations at a more detailed time scale. Zhang et al. further explained that fine-grained motion information can be encoded by the exact timestamps of event cameras and hence, these cameras are applicable in visual motion estimation in the conditions of high speed and low-light.

Zhang et al. also significantly decreased the computational cost of event-based optical flow estimation by introducing a spiking neural network to predict event-camera optical flow and train it using spatio-temporal backpropagation. They used their approach to report average endpoint error (AEE) values of 0.76, 1.13, 0.95, and 0.45 on the Indoor Flying 1, Indoor Flying 2, Indoor Flying 3, and Outdoor Day 1 sequences, respectively, on the MVSEC dataset. These were not as high as Spike-FlowNet which reported 0.84, 1.28, 1.11, and 0.49 on the same sequences, but there was still a performance difference between it and the CNN-based methods like STRN-FlowNet. The findings indicate that the spiking neural networks can maintain beneficial motion information in the event-based optical flow estimation, and their primary benefit is the energy efficiency. The energy analysis presented by Zhang et al. demonstrates that the rate of spiking activity of their approach is as low as 0.38%-0.75. As accumulation operations are dominant in SNNs compared to multiplyaccumulate operations, which are energy-intensive, they can theoretically significantly decrease the theoretical computational energy consumption of CNNs with the same structure.

The event-based optical flow cannot be regarded as an easy substitute to the traditional optical flow, but it is a continuation of the motion information presentation in high-speed conditions. The classic optical flow they use will estimate motion between successive frames and is more appropriate when image quality does not change and the computational resources are available. In comparison, event-based optical flow directly exploits temporal information provided by asynchronous event streams, which is why it is more appropriate when tracking low-latency, high-speed motion, and edges are to be deployed. In high-speed object tracking, this implies that the system can not only measure the spatial location of the target, but also its dynamic behavior at relatively low power usage, which can be used to give timely motion information to predict trajectories, associate with the target, and control the system in real time.

Event cameras have further changed how visual data is acquired and represented in high-speed object tracking. Unlike conventional cameras, which output full-frame images at a fixed frame rate, event cameras trigger events asynchronously only when pixel-level brightness changes exceed a predefined threshold. Each event records its spatial position, polarity, and timestamp. Gallego et al. note that event cameras provide microsecond-level temporal resolution, extremely high dynamic range, low power consumption, and reduced motion blur, giving them clear advantages in low-latency, high-speed, and complex-illumination scenarios [7]. Because event streams are triggered by changes in the scene, they filter out much of the static and redundant background information at the data level, making the visual input more focused on target edges and the motion process itself. In this sense, visual sensing shifts from periodically recording the entire scene to recording local changes on demand, which better matches the temporal sensitivity and real-time requirements of high-speed object tracking.

Event-driven data representation has been demonstrated to be beneficial in real-world tracking. Duarte et al. suggested a hand-tracking approach that is founded on ROI extraction and feature update with the help of one event camera. This method outlines the hand area of interest based on the activity of boundary events and subsequently derives spatial trajectory characteristics in the area of interest. They demonstrate that real-time tracking is possible with a significant decrease in the volume of data without significant increases in the trajectory error [5]. The REMOT architecture by Gao et al. is an even further extension of event cameras that links the output of event cameras to real-time multi-object tracking on an FPGA. REMOT processes events in a distributed fashion, enhancing throughput and saving energy, through hardware-software co-design [8]. The examples above imply that not only event-driven data representation is not inappropriate to describe high-speed motion, but that it is inherently compatible with parallel hardware architectures.

Visual data acquisition and representation determines to a great extent the performance threshold of high-speed tracking of objects. Image sequences based on frames offer full appearance information, and can be used with established detection-association systems, but the fixed sampling process used is prone to motion blur and latency at high-speed situations. Optical flow representation emphasizes motion tendency but its accuracy remains to be dependent on the quality of the original images. Compared to event cameras, event cameras enhance temporality resolution on the sensing side, as well as reduce redundant data by asynchronous triggering, which are more appropriate in low-latency tracking and real-time edge analysis. In this light, the event-driven models can better generalize high-speed cases compared to the traditional uniformly sampled models, which justifies why the recent research on high-speed tracking has been shifted to event-based vision more and more than frame-based vision [8].

2.2. Motion Information Extraction and Feature Representation

Object tracking is not simply a matter of tracking the target in every frame or the segment of an event-stream after visual data has been received. What is more important is to extract motion and appearance information of a series of observations in a form that does not change over time. According to classical surveys on tracking objects, tracking can usually be influenced by high motion, changes of appearance, non-rigid deformation, occlusion, change of illumination, and clutter in the background. This is why target representation, feature selection and motion models are of prime importance in tracking result stability [9]. These challenges are more severe in fast-paced situations: the target can experience large displacements over a brief period, its local contours and scale can vary quickly, and background interference and occlusion can readily cause discontinuous measurements. Unless motion and appearance features are extracted reliably, even available detection results can prove hard to match correctly and result in mismatches and identity switches or discontinuities in trajectories.

In the case of multi-object tracking, detecting the targets is not the only issue of concern, but also the need to identify whether the observations over time are associated with the same object. Multi-object tracking is a combination of related tasks, according to Luo et al., such as localizing an object, maintaining its identity and generating its trajectory. It implies that the feature representation should be a description of the current target state and be able to support cross-frame association and long-term trajectory maintenance [10]. How a system perceives the identical target is then key to designing tracking methods. It can be based on changes in position and velocity patterns, appearance embeddings and deep features or joint spatio-temporal representations. As deep learning was developed, Ciaparrone et al. demonstrate that the deep models have found extensive application in multi-object tracking in detection, feature extraction, and data associations, replacing traditional hand-designed features by more discriminative learning-based features [11]. In tracking high-speed objects, feature representation is not, then, a technical selection, but a trade-off between real-time performance, robustness and deployability.

Motion extraction, such as frame differencing, background modeling, and optical flow, is based on traditional visual cues, which remain popular to this day. Optical flow provides local motion fields of vectors based on brightness-change constraints and is able to capture the direction and velocity of motion of a target between successive time frames which is helpful in high-speed motion estimation. Frame differencing isolates motion areas by calculating the pixel-to-pixel difference between successive frames. It is computationally cheap and can be easily applied in real time, however it is also susceptible to changes in illumination and noise. Background modeling isolates foreground targets, by constructing a reference model of the stationary scene. It is effective in the fixed scene surveillance, but it loses stability with change of view points or in complicated scenarios. These are structurally simple, interpretable and cheap methods to achieve, thus they are still applicable in resource limited situations. Their weakness is evident as target speed rises, motion blur is more extreme or as the percentage of occlusion rises where robustness tends to fall significantly.

Motion information can be obtained in other ways in event-based vision. Since event streams are caused by changes in brightness, they already have spatio-temporal data concerning moving edges. The work of Zhang et al. demonstrates that motion information does not necessarily need to be received following a full-frame reconstruction. The feature encoding and state estimation can be done directly at the event level, which is more suited to the dynamic variations of fast-moving targets [3]. This indicates a change of high-speed tracking towards the reconstruction of images prior to motion analysis to a modelling of motion directly on event streams.Object tracking also increasingly relies on high-level representations learned by deep models. StrongSORT is a representative example. It improves the appearance and motion branches of DeepSORT by using a stronger detector and a ReID embedding network for target appearance representation, together with global association and interpolation strategies to handle occlusion, missed detections, and trajectory fragmentation [4]. This shows that feature representation in modern object tracking is no longer limited to handcrafted low-level motion cues. It has become a composite representation built from CNN features, ReID embeddings, and spatio-temporal modeling. These deep features improve recognition capability in complex backgrounds and dense-target scenarios, but they also increase computation, training dependence, and system resource overhead.

In high-speed object tracking, more complex feature representations are not always preferable. Deep features improve target distinguishability, especially in complex backgrounds and dense-target scenarios, and help recognition and association remain more stable. Yet CNNs, ReID networks, and spatio-temporal models also increase computational cost and hardware dependence. For high-speed tracking systems designed for low latency, low power consumption, or edge deployment, relying on complex networks at every stage may cause real-time performance to become the main bottleneck before accuracy gains can be fully realized. This has led to another direction in event-based vision:

exploiting the sparsity and motion sensitivity of event streams to perform target localization and trajectory updating in a more lightweight way.

Duarte et al.'s hand-tracking study offers a useful example. Their method performs preprocessing, ROI definition, and trajectory smoothing based on event activity, without relying on complex deep networks, yet still achieves low-cost real-time tracking [5]. In scenarios where motion boundaries are relatively clear and event-active regions remain stable; the event stream itself can provide enough motion cues for state estimation around changing regions. REMOTE takes this idea closer to hardware implementation. It uses attention units to aggregate states and maintain targets within local event regions, allowing event feature extraction and trajectory updating to be combined with parallel processing on FPGAs [8]. In this sense, Duarte's work demonstrates the feasibility of lightweight event features for real-time tracking, while REMOT shows their potential for hardware deployment.

Motion information extraction and feature representation have not followed a simple path in which traditional methods are replaced by deep learning methods. Different application requirements have led to several parallel routes. Traditional methods rely mainly on explicit motion cues such as frame differencing, background modeling, and gradient-based optical flow. They are structurally clear, computationally inexpensive, and easy to implement, which makes them useful in resource-constrained systems. Their robustness, however, is limited under high-speed motion, frequent occlusion, or complex backgrounds, where motion blur, noise, and discontinuous observations become more prominent.

Deep representation methods learn more discriminative motion and appearance features through CNNs, ReID, and spatio-temporal modeling. They are stronger in target distinguishability, occlusion recovery, and long-term association, but they also increase computational complexity and deployment cost. Chen et al.'s survey on visual object tracking also notes that existing tracking methods include both traditional non-deep frameworks and deep learning frameworks, with clear differences in accuracy, robustness, and efficiency [12]. Method selection should therefore depend on the specific tracking scenario rather than on a single criterion.

For high-speed object tracking, the key is not to choose more complex features, but to balance target speed, scene complexity, and hardware resources. Edge devices, mobile robots, and low-power real-time systems benefit more from lightweight, sparse, and event-driven representations. When computing power is sufficient, target categories are complex, or occlusion is severe, deep spatio-temporal features can better support recognition capability and trajectory stability. The choice of feature representation should therefore be matched with the sensor modality, computing platform, and practical application constraints.

3. Sensing Devices and Hardware Systems for HIigh-Spreed Object Tracking

This chapter turns to sensing devices and system-level implementation. Whether visual data can represent target motion effectively first depends on the sensor's ability to capture rapid changes. Whether these data can be transformed into stable trajectories then depends on the back-end hardware's ability to process them within limited power and latency budgets. High-speed object tracking is therefore not only a matter of algorithmic accuracy, but also an engineering problem shaped by sensors, data streams, computing platforms, and system architectures.

For fast-moving targets, the first question is whether the system can "see fast enough." If the target is already blurred, missed, or compressed during acquisition, even complex recognition and association algorithms can only compensate for incomplete information. High-speed scenarios also bring increasing data volume, accumulated computational latency, and higher device power consumption. A practical high-speed tracking system must therefore not only sense quickly, but also process data quickly, stably, and efficiently. Based on this logic, this chapter discusses high-speed visual sensors and hardware architectures.

3.1. High-Speed Visual Sensors and Data Acquisition Systems

The first requirement of high-speed tracking of objects is the possibility to record fast changes. The traditional vision systems typically use frame based camera, which captures full images at a constant frame rate and interpolates the motion states using the difference between targets in neighbouring frames. This method is advanced and stable in typical video set-ups, and can be easily combined with object discovery, categorization and tracking of multiple objects algorithms. The drawbacks of frame-based cameras become apparent as the target speed goes up, though. The motion of a target can be very far between two frames, and therefore, its motion is only sampled sparsely. The edges, textures, and contours may also become unstable, and the motion blur may be obtained with continuous motion during exposure. These issues have a direct impact on the following recognition and tracking association in cases where the tracking method is based on detection boxes and appearance characteristics.

The simplest solution is to increase the frame rate. The high-speed cameras have a higher number of images per section of time and reduce the time between the next frames, thus enabling continuous motion to be captured faster. This allows them to be used in industrial inspection, sports analysis and experimental measurement. Nonetheless, high-speed cameras still adhere to the principles of full-frame acquisition. Although it may be just a small part of the scene that will change, the system will have to capture, transmit and process entire images. With higher frame rate, bandwidth, storage, computation and power consumption requirements are also increased. HPC cameras thus, address the issue of not seeing fast enough, yet add another one, too much data and too much processing.

Another method of providing a solution to this trade-off is through event cameras. In contrast to the traditional cameras that produce full-frame images at a fixed rate, event cameras produce events in an asynchronous way only when brightness changes on a pixel-level are above a certain threshold. The pixel location, time and polarity of the brightness change are typically recorded in each event. According to Gallego et al., event cameras are capable of offering temporal resolution at the microsecond level and a dynamic range of approximately 140 dB (compared to approximately 60 dB of traditional standard cameras) and capable of low power consumption and reduced motion blur [7]. This is not only a parameter improvement, but a different way of documenting the visual data: traditional cameras document the whole scene based on time, whereas event cameras document the local information based on changes.

This is a property that corresponds to the high-speed tracking requirements. During high-speed tracking of objects, the central focus is typically not the motionless background, but the dynamic variations of target edges, contours and motion states. Since event cameras only respond to the changing areas, they minimize redundancy at the sensing level of the image and focus the input information more on motion. The survey by Kryjak indicates that event cameras are capable of a dynamic range of approximately 120 dB, as compared to a 50 to 60 dB in traditional frame-based cameras; event timestamps have a resolution of almost 1 microsecond, and practically none of the events are produced when the scene is not changing [6]. A faster camera is not an event camera, but an asynchronous visual sensor with more properties that are desirable in representing high-speed changes.

Event cameras have also shown clear advantages in practical tracking tasks. Duarte et al. used a single event camera for hand tracking and achieved real-time position estimation through event preprocessing, noise reduction, ROI extraction, and trajectory smoothing. Their experiments show that ROI localization can reduce the data volume by up to about 96% while preserving useful features. The average trajectory error is about 10 mm in the X–Y plane and approximately 25–40 mm in three-dimensional space [5]. These results suggest that event-driven methods can achieve good accuracy and efficiency in two-dimensional real-time tracking, while depth estimation remains more challenging.

Event cameras are often described as bio-inspired visual sensors. While conventional cameras record images periodically, event cameras resemble the retinal response to dynamic changes more closely. In their study on event-based optical flow, Zhang et al. note that event cameras respond only at locations where brightness changes occur, and that each output event contains position, microsecond-level timestamp, and polarity information. When spiking neural networks directly process such discrete spatio-temporal event streams, they can reduce power consumption by about 99% compared with CNNs of the same structure [3]. Bio-inspired vision therefore changes not only the way visual data are acquired, but also the implementation basis for low-power motion estimation.

The creation of high-speed visual sensors has gone beyond the issue of raising the frame rate. It is a larger shift towards frame-based sensing to event-based and bio-inspired vision. Traditional frame-based cameras give full appearance information and can be used with full-grown detection and recognition algorithms, but cannot be used in high-speed applications due to frame rate and exposure time constraints. High-speed cameras not only enhance the sampling density, but also introduce additional load on the data transmission, computation, and energy consumption. Event cameras minimize redundancy in the acquisition phase and enhance time resolution, and are more suitable to low-latency, high-dynamic-range and edge-deployment applications. In the case of high-speed object tracking sensor selection is not simply a point of device configuration, but a point of origin that defines the data format, the algorithm path and the system architecture.

3.2. Hardware Architectures and Implementation of High-Speed Tracking Systems

The sensor defines the ability to capture target changes at a particular speed and the hardware architecture defines the speed at which changes can be processed. The high-speed tracking typically involves data reception, preprocessing, object detection, feature extraction, motion estimation, data association and trajectory updating to be done in a very short time. Any of these stages can be delayed, leading the output trajectory to miss the actual target state, and impacting the warnings, control, or decision making. That is why, high-speed tracking systems should take into account not only the accuracy of the algorithms, but the throughput, latency, power consumption and hardware resources.

In conventional vision systems, CPUs and GPUs are the most widely used computing platforms. CPUs are general-purpose and well suited to task scheduling and process control, but their parallel processing capability is limited for large-scale image convolution and high-throughput visual computation. GPUs offer much stronger parallelism and are widely used to run deep learning detectors, CNN-based feature extraction networks, and multi-object recognition models in tracking-by-detection systems. Their relatively high-power consumption and heat dissipation requirements, however, make them less suitable for long-term low-power real-time deployment in drones, mobile robots, vehicle-mounted edge devices, and embedded surveillance scenarios.

FPGAs have therefore become an attractive platform for high-speed vision systems. They are reconfigurable, support parallel and pipelined processing, and usually consume less power than general-purpose platforms. For specific tracking tasks, dedicated data paths can be designed so that image preprocessing, convolution computation, event filtering, ROI updating, and trajectory maintenance are executed in parallel at the hardware level. This reduces the scheduling overhead that would otherwise occur on general-purpose processors.

Hardware systems for high-speed tracking are closely tied to the sensor modality. For frame-based images, the main approach is to accelerate deep models in hardware. Modern tracking increasingly depends on detection, classification, and deep feature extraction. In frameworks such as StrongSORT and ByteTrack, the speed and stability of the detector directly affect tracking performance. Sali et al. note that FPGAs, with their low power consumption, deterministic latency, and reconfigurability, have been used for real-time detection, classification, and tracking in autonomous driving, robotics, and video surveillance [13]. These systems often rely on model quantization, pruning, parallel convolution, pipelined processing, and dataflow optimization to reduce computational load and run deep models on resource-constrained platforms.

This route benefits from mature frame-based detection and recognition frameworks. Detectors provide candidate boxes, ReID networks extract appearance features, and motion models together with matching algorithms perform trajectory association. In StrongSORT, the lightweight AFLink and GSI modules add only about 1.7 ms and 7.1 ms per image on MOT17, respectively, showing that association-completion modules can improve trajectory continuity with limited extra cost [4]. The limitation, however, is that the system still has to process complete frame-based images. Even with hardware acceleration, high-frame-rate input continues to place considerable pressure on bandwidth and computation.

The other route is to build event-driven hardware architectures around event streams. Instead of reconstructing events into images, these systems can process asynchronous events directly. Because event data are sparse, the hardware activates relevant computing units only when events arrive, avoiding repeated computation on static backgrounds. Gao et al.'s REMOT architecture implements real-time event-based multi-object tracking on an FPGA. It uses attention units to track event regions autonomously and relies on hardware–software co-design for event processing, region updating, and target management. REMOT processes 0.43–2.91 million events per second with a power consumption of 1.75–5.45 W, and achieves HOTA scores of 43.1%–73.2% [8].

Per-event processing makes better use of the temporal advantage of event cameras. While traditional methods usually perform detection and association frame by frame, per-event processing updates local states as soon as each event arrives. Xiong et al. proposed BMMOT, a boundary-movement-based multi-object tracking system. Implemented on an AMD ZYNQ FPGA, BMMOT achieves an event processing latency of 19.6–137.2 ns per event, processes up to 51 million events per second, and consumes less than 2.71 W of power. It also maintains tracking accuracy of 52.1%–79.4% on multiple datasets, while saving 16.4%–41.0% of LUTs and 64.5%–78.7% of flip-flop resources compared with existing work [14]. These results show that per-event processing can improve both throughput and resource efficiency.

Practical systems rarely rely entirely on a single type of hardware or a single technical route. Pure software systems are flexible, but they are easily constrained by throughput and power consumption under high-event-rate or high-frame-rate inputs. Pure hardware systems are fast and low-latency, but they often lack flexibility in complex target management, occlusion reasoning, and trajectory merging. Hardware–software co-design therefore provides a more practical compromise. High-frequency, rule-based tasks such as event filtering, convolution computation, ROI updating, and local feature extraction are well suited to hardware implementation, while target initialization, trajectory management, occlusion reasoning, and complex association strategies can be handled by software or embedded processors.

The hardware architecture of a high-speed tracking system is closely coupled with its data representation. Frame-based image input usually requires GPUs or FPGAs to accelerate CNN-based detectors and appearance feature extraction. Event-stream input is better suited to asynchronous processing on low-power parallel platforms such as FPGAs. Tasks involving optical flow, pose estimation, or motion state estimation also require a low-latency data path between feature extraction and state updating. System design in high-speed object tracking should therefore not be judged only by the performance metrics of a single algorithm. It also needs to account for sensor output format, data sparsity, computing platform capability, and deployment environment.

The development of sensing devices and hardware systems for high-speed object tracking is moving beyond the pattern of "high-frame-rate acquisition with general-purpose computing." A clearer direction is "event-driven acquisition with dedicated hardware acceleration." Traditional frame-based systems rely on complete images and mature deep models, so they remain useful when visual information is rich and computing power is sufficient. The event-driven systems are based on the thin event streams and parallel hardware architectures, which is why they are more appropriate in low-latency, low-power, and highly dynamic applications. Future work might not thus be about merely increasing the frame rate or compressing a single model, but about co-design of sensors, data representation, object recognition algorithms, and hardware architectures. This co-design is expected to offer a more reasonable trade-off between accuracy, real time performance and energy efficiency.

4. Conclusion

The paper has surveyed the object-recognition-based approaches to high-speed object tracking, including data representation, motion information extraction, object recognition, and trajectory association, and hardware system implementation. The speed of the target is not the only problem in the high-speed object tracking. The instability is also increased by fast motion in perception, modeling, and implementation of systems. The conventional frame-based imaging is limited by the frame rate and exposure time, which in effect tend to cause motion blur, loss of the edge information, and high inter-frame shift. Local motion tendencies can be described by optical flow methods, and their stability remains to be dependent on the quality of the image and continuous observations. Tracking-by-detection methods based on deep learning enhance the ability of multi-object tracking with detectors, ReID features, and trajectory association, as well as raises the computational load and the deployment pressure.

The visual input brought about by event cameras is different in principle compared to traditional frame based cameras. Their ability to record changes in brightness as asynchronous events gives them an emphasis on moving edges and spatio-temporal variations, and so works well with high-speed motion, complicated illumination and low-latency. Event-based optical flow, ROI-extraction, lightweight-trajectory-updating, and FPGA-based, real-time processing-architectures researches have driven high-speed tracking beyond full-frame image processing to change-driven perception.

The high-speed object tracking is not about the optimization of the performance of one algorithm only. It is a system level issue that is determined by sensors, data representation, feature extraction, target association and computing platforms. There are appropriate conditions of the different technical ways. Frame-based image approaches maintain a relatively complete information on appearance and enable object detection and category-level classification. Event-based vision is more adaptable to the rapidly changing dynamic situations, with its focus on low redundancy and temporal resolution. Deep representation algorithms enhance the distinguishability of targets, but need more powerful computing capabilities. Event-based approaches with a lightweight are more amenable to real-time edge deployment. It is rather not to seek more complicated models or elevated single metrics but to achieve a superior balance between precision, real-time functionality, stability, and power consumption.

Future research on high-speed object tracking can be organized around three issues: more reliable perception, more stable association, and more deployable systems. At the data level, frame-based images and event streams should be further integrated, which is worth investigating. Frame-based images can give detail on both texture and semantics and event streams can give finer detail to movement changes. The two modalities might be made more temporally synchronized, have their features aligned and noise suppressed which can enhance object recognition and high speed motion perception.

On the algorithmic level, trajectory consistency in complicated situations is one of the issues. Occasionally, high-speed motion is characterized by occlusion, lost detections, sudden change in scale, and thick targets. Using the positional matching of detection boxes alone makes it hard to assure a constant object identity in the long term. These appearance features, motion prediction, event-based optical flow and spatio-temporal context must thus be more tightly coupled together to construct a more robust data association mechanism.

Real-time deployability on a system level will be a determining factor of the practicality of high-speed tracking techniques. The design of an algorithm must thus consider more than accuracy on standardized systems and look at event throughput, processing latency, power consumption, storage bandwidth and use of hardware resources. FPGAs, embedded platforms, spiking neural networks, and hardware–software co-design architectures offer feasible paths toward low-power real-time tracking. The remaining challenge is to maintain algorithmic flexibility and tracking stability under limited resources. Evaluation should also move beyond single accuracy metrics. For high-speed object tracking, system performance is better reflected by a combination of tracking accuracy, identity preservation, response latency, energy consumption, and environmental adaptability.

The future of this field will depend not only on new tracking algorithms, but also on visual tracking systems that can operate reliably in real-world high-speed scenarios. As event cameras, deep learning, and edge computing platforms continue to develop, high-speed object tracking is likely to move toward a more integrated direction, where sensing devices, intelligent algorithms, and hardware architectures are optimized together.

References

[1] Lu H., Li P., Wang D. Review of object tracking algorithms. Pattern Recognition and Artificial Intelligence, 2018, 31(1): 61-76.

[2] Huang K., Chen X., Kang Y., et al. Review of intelligent video surveillance technology. Chinese Journal of Computers, 2015, 38(6): 1093-1118.

[3] Zhang Y., Lv H., Zhao Y., et al. Event-based optical flow estimation with spatio-temporal backpropagation trained spiking neural network. Micromachines, 2023, 14(1): 203.

[4] Du Y., Zhao Z., Song Y., et al. StrongSORT: Make DeepSORT great again. IEEE Transactions on Multimedia, 2023, 25: 8725-8743.

[5] Duarte L., Safeea M., Neto P. Event-based tracking of human hands. Sensor Review, 2021, 41(4): 382-389.

[6] Kryjak T. Event-based vision on FPGAs: A survey. arXiv preprint arXiv:2407.08356, 2024.

[7] Gallego G., Delbruck T., Orchard G., et al. Event-based vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(1): 154-180.

[8] Gao Y., Wang S., So H.K.H. A reconfigurable architecture for real-time event-based multi-object tracking. ACM Transactions on Reconfigurable Technology and Systems, 2023, 16(4): Article 58.

[9] Yilmaz A., Javed O., Shah M. Object tracking: A survey. ACM Computing Surveys, 2006, 38(4): Article 13.

[10] Luo W., Xing J., Milan A., Zhang X., Liu W., Kim T.K. Multiple object tracking: A literature review. Artificial Intelligence, 2021, 293: 103448.

[11] Ciaparrone G., Sánchez F.L., Tabik S., Troiano L., Tagliaferri R., Herrera F. Deep learning in video multi-object tracking: A survey. Neurocomputing, 2020, 381: 61-88.

[12] Chen F., Wang X., Zhao Y., Lv X., Tang X., Niu X., Huang Q., Wu B. Visual object tracking: A survey. Computer Vision and Image Understanding, 2022, 222: 103508.

[13] Sali S.M., Meribout M., Abdul Majeed A. Real-time FPGA-based CNNs for detection, classification, and tracking in autonomous systems: State-of-the-art designs and optimizations.

[14] Xiong W., Zhang J., Zhu X., et al. A high-performance and resource-efficient FPGA-based multi-object tracking system using event cameras. In 2025 35th International Conference on Field-Programmable Logic and Applications, 2025.