computers

MDPI

Review

Object Tracking Using Computer Vision: A Review

Pushkar Kadam *,†, Gu Fang *,† and Ju Jia Zou †

School of Engineering, Design and Built Environment, Western Sydney University, Locked Bag 1797, Penrith, NSW 2751, Australia; j.zou@westernsydney.edu.au
* Correspondence: 18745753@student.westernsydney.edu.au (P.K.); g.fang@westernsydney.edu.au (G.F.)
† These authors contributed equally to this work.

Abstract: Object tracking is one of the most important problems in computer vision applications such as robotics, autonomous driving, and pedestrian movement. There has been a significant development in camera hardware where researchers are experimenting with the fusion of different sensors and developing image processing algorithms to track objects. Image processing and deep learning methods have significantly progressed in the last few decades. Different data association methods accompanied by image processing and deep learning are becoming crucial in object tracking tasks. The data requirement for deep learning methods has led to different public datasets that allow researchers to benchmark their methods. While there has been an improvement in object tracking methods, technology, and the availability of annotated object tracking datasets, there is still scope for improvement. This review contributes by systemically identifying different sensor equipment, datasets, methods, and applications, providing a taxonomy about the literature and the strengths and limitations of different approaches, thereby providing guidelines for selecting equipment, methods, and applications. Research questions and future scope to address the unresolved issues in the object tracking field are also presented with research direction guidelines.

Keywords: object tracking; computer vision; image processing; data association; deep learning

check for updates

Citation: Kadam, P.; Fang, G.; Zou, J.J. Object Tracking Using Computer Vision: A Review. *Computers* **2024**, *13*, 136. https://doi.org/10.3390/computers13060136

Academic Editors: Selene Tomassini and M. Ali Akber Dewan

Received: 9 May 2024
Revised: 22 May 2024
Accepted: 26 May 2024
Published: 28 May 2024

1. Introduction

Object tracking using computer vision is one of the most important functions of machines that interact with the dynamics of the real world, such as autonomous ground vehicles [1], autonomous aerial drones [2], robotics [3], and missile tracking systems [4]. For machines to operate and adapt according to real-world dynamics, it is essential to monitor changes. These changes are usually the motions that must be sensed through different sensors, followed by the machines responding according to these changes [4]. Computer vision mimics the human ability to observe these changes. Humans intuitively understand the change in their environment due to different senses, which helps them navigate their world. Vision is one of the primary senses that allow humans to navigate their environment. To design autonomous machines that perform human tasks such as driving [1,3,5–10], fishing [11], agricultural activities [2], and medical diagnoses [12–16], computer vision can help increase productivity. The inclusion of computer vision in human–computer interaction, robotics, and medical diagnoses provides humans with better tools for completing tasks efficiently and making decisions with better insights. Therefore, it is essential to investigate different methods, tools, and potential applications to evaluate their limitations and future scope for object tracking problems in computer vision to improve work efficiency and develop an autonomous system that works well with humans.

Different insights can be gained by looking at a holistic view of object tracking in computer vision that complements various aspects of the problem. Therefore, this review synthesises and categorises information regarding different aspects, such as sensors, datasets, approaches, and applications of object tracking problems in computer vision. The main contributions of this review are as follows:

- A systemic literature review in object tracking based on hardware usage, datasets, image processing and deep learning methods, and application areas.
- Recommendations and guidelines for selecting sensors, datasets, and application methodologies based on their advantages and limitations.
- A taxonomy for sensor equipment and methodologies.
- Research questions and future scope to address unresolved issues in the object tracking field.

This review highlights the development of object tracking methods in computer vision over the last ten years. The review takes major journal articles published since 2013 in object tracking in computer vision and aims to outline the progress made in this field. This review highlights the different approaches, methods, equipment, datasets, and object tracking applications. By highlighting current development, the review consolidates the data on methods, applications, and types of vision sensors, enabling engineers and software developers to make informed choices while developing their systems for different applications. Furthermore, this review identifies different limitations in current methods and proposes future developments to help push the boundaries of object tracking.

In this paper, Section 2 outlines different reviews performed in object tracking and distinguishes this review from these previous reviews. Section 3 discusses the types of equipment for different vision sensors and how they impact development. Section 4 provides the overview of available datasets for benchmarking object tracking results. Section 5 lays out the different approaches and methods used in object tracking. Section 6 lists the different areas where object tracking in computer vision is deployed. Section 7 provides a discussion of object tracking methods and datasets. Section 8 provides limitations and future work along with the research questions and recommendations to address them. Section 9 outlines the conclusion of this study. Figure 1 shows the structure of the review.

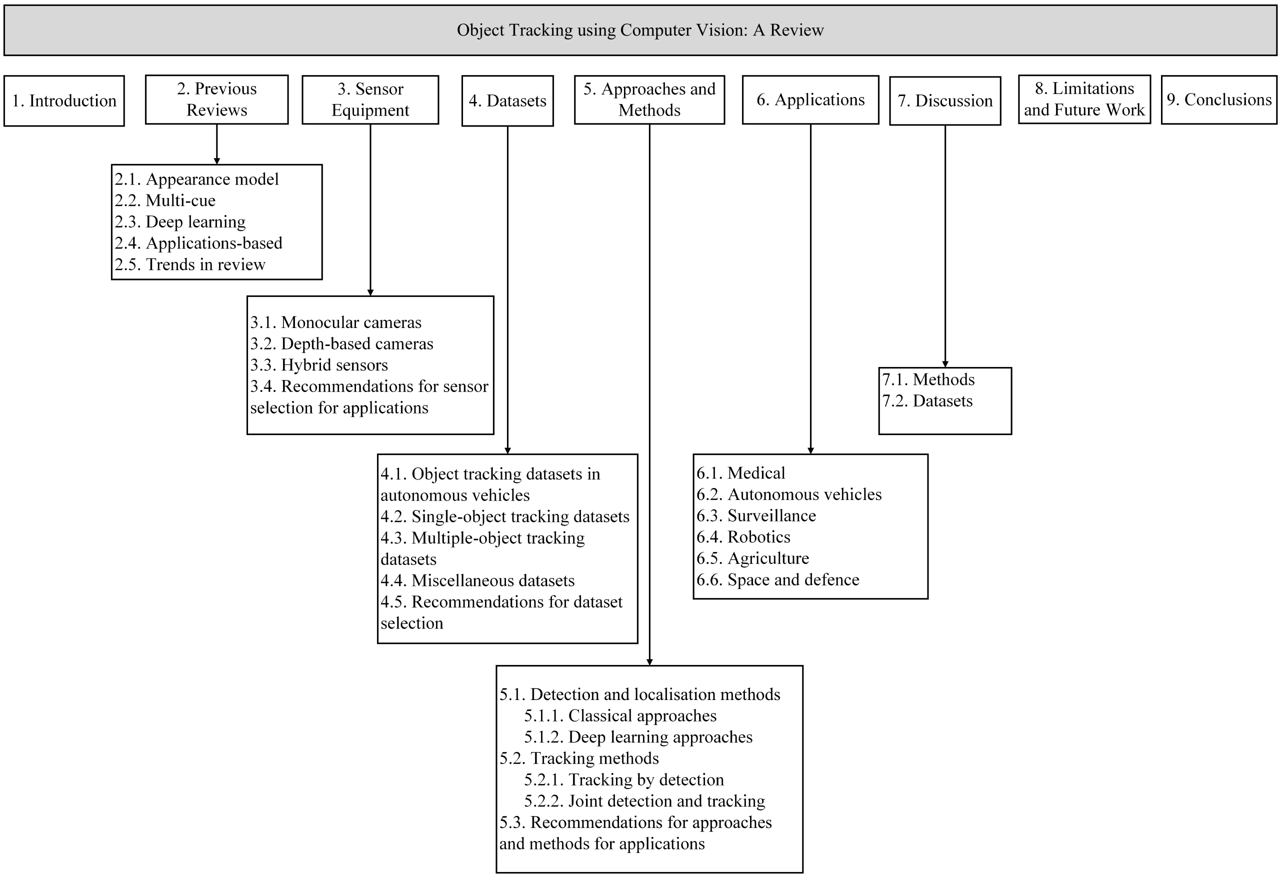
Object Tracking using Computer Vision: A Review
1. Introduction
2. Previous Reviews
3. Sensor Equipment
4. Datasets
5. Approaches and Methods
6. Applications
7. Discussion
8. Limitations and Future Work
9. Conclusions
2.1. Appearance model
2.2. Multi-cue
2.3. Deep learning
2.4. Applications-based
2.5. Trends in review
3.1. Monocular cameras
3.2. Depth-based cameras
3.3. Hybrid sensors
3.4. Recommendations for sensor selection for applications
7.1. Methods
7.2. Datasets
4.1. Object tracking datasets in autonomous vehicles
4.2. Single-object tracking datasets
4.3. Multiple-object tracking datasets
4.4. Miscellaneous datasets
4.5. Recommendations for dataset selection
6.1. Medical
6.2. Autonomous vehicles
6.3. Surveillance
6.4. Robotics
6.5. Agriculture
6.6. Space and defence
5.1. Detection and localisation methods
5.1.1. Classical approaches
5.1.2. Deep learning approaches
5.2. Tracking methods
5.2.1. Tracking by detection
5.2.2. Joint detection and tracking
5.3. Recommendations for approaches and methods for applications

Figure 1. Structure of the review.

2. Previous Reviews

There has been a considerable development in object tracking using computer vision. Previous review articles and surveys focus on a niche area of the object tracking problem. A review focusing exclusively on a subarea of the research field is often beneficial in investigating specific gaps in the literature. However, widening the scope of the literature review helps to identify whether a particular approach has an advantage over the others. Furthermore, a review of the field of research provides a roadmap for researchers and engineers to investigate the problem further according to the needs of the application. This section identifies different reviews covering different aspects of the object tracking problem and distinguishes this review from these previous reviews. This section also outlines the main contribution of each review, which acts as a roadmap for different research niches in the object tracking literature.

2.1. Appearance Model

Any object, such as circles, squares, cylinders, and triangles, can be deconstructed to its basic geometry. Identifying these geometric features can assist in detecting the objects in an image frame. These types of visual appearance form object descriptors, which use different features of the object, such as edges and corners, to construct a mathematical model for object identification.

In their survey of appearance models, Li et al. [17] reviewed the literature on visual representation as per their feature-construction mechanism. Since object tracking methods have problems handling complex object appearance changes due to illumination, occlusion, shape deformation, and camera motion, Li et al. [17] concluded that it was essential to effectively model the 2D appearance of tracked objects for successful visual tracking. Their survey focused on the detection methods as a precursor to the tracking-by-detection approach. While appearance models are advantageous in object detection, they are still handcrafted to particular object detection. Handcrafted feature models for face detection will differ from human body detection. While that survey proposed learning techniques such as support vector machines and particle filtering, their learning is dependent upon the training sample selection.

2.2. Multi-Cue

Since the publication of the review by Li et al. [17] in 2013, there have been significant improvements in deep learning methods, which have proven effective in object detection [18,19]. In their survey, Kumar et al. [19] identified the research in multi-cue object tracking that used appearance models in traditional and deep learning approaches. Multi-cue methods rely on multiple cues in the image, such as colour, texture, contour, and object features, to develop descriptors to identify the object. They surveyed methods that used handcrafted features integrated with deep learning-based models to provide robust tracking algorithms.

2.3. Deep Learning

There was a surge in the review of deep learning methods for object tracking, with two reviews in 2021 and three reviews in 2022. Park et al. [20] reviewed the evolution of multiple-object tracking in deep learning by categorising the previous multiple object tracking algorithm in 12 approaches. They also reviewed the benchmark datasets and standard evaluation methods. Kalake et al. [21] reviewed deep learning-based online multiple-object tracking and ranked the networks on different public benchmark datasets. Mandal et al. [22] provided an empirical review of the state-of-the-art deep learning methods for change detection by categorising the existing approaches into different deep learning methods. Furthermore, they provided an empirical analysis of the evaluation settings adopted by existing deep learning methods. Guo et al. [23] reviewed deep learning methods for multiple-object tracking in autonomous driving. Their review categorised the algorithms based on tracking by detection, joint detection and tracking, and transformer-based tracking.

They identified multiple-object tracking datasets and provided an experimental analysis and future research direction in deep learning. While it is important to examine deep learning methods in isolation to identify the best methods according to the solution, it is also important to consider traditional appearance-based and statistical models for certain types of applications. Therefore, studying and reviewing traditional and deep learning methods can provide insights into method selection based on hardware and applications.

2.4. Applications-Based

Recent reviews have looked into detection-based multiple-object tracking [24], data association methods [25], long-term visual tracking [26], and methods used in ship tracking [27]. Dai et al. [24] introduced a taxonomy of multiple-object tracking and provided a detailed summary of the results of algorithms on popular datasets. Liu et al. [26] reviewed long-term tracking algorithms while describing existing benchmarks and evaluation protocols. Rocha et al. [27] reviewed datasets and state-of-the-art algorithms for single and multiple-object tracking with the view of applying them to ship tracking. Furthermore, they provided insights into developing novel datasets, benchmarking metrics, and novel ship-tracking algorithms. These reviews are focused on specific applications, such as single- or multiple-object tracking, and provide direction for research in their respective fields.

2.5. Trend in Reviews

Different approaches, such as appearance models, data association, and long-term tracking, were reviewed from previous reviews over the last ten years. A summary of reviews works on object tracking is provided in Table 1. Figure 2 shows the number of reviews covering different areas of object tracking from 2013 to 2023. A trend is noticed in Figure 2 where there is a peak of interest in object tracking in 2022, with five papers, out of which three focus exclusively on deep learning methods. The exclusive nature of the literature surveyed in recent reviews necessitates a comparative evaluation of the different approaches. Also, hardware equipment and hardware constraints in the application require investigating different types of sensors and their corresponding methods, applications, and scopes. Furthermore, based on an overview of the object tracking field, guidelines, and recommendations for the methods will contribute to the decision-making process for specific applications. Therefore, this survey aims to investigate different sensor equipment, datasets, approaches and methods, and object tracking applications in computer vision.

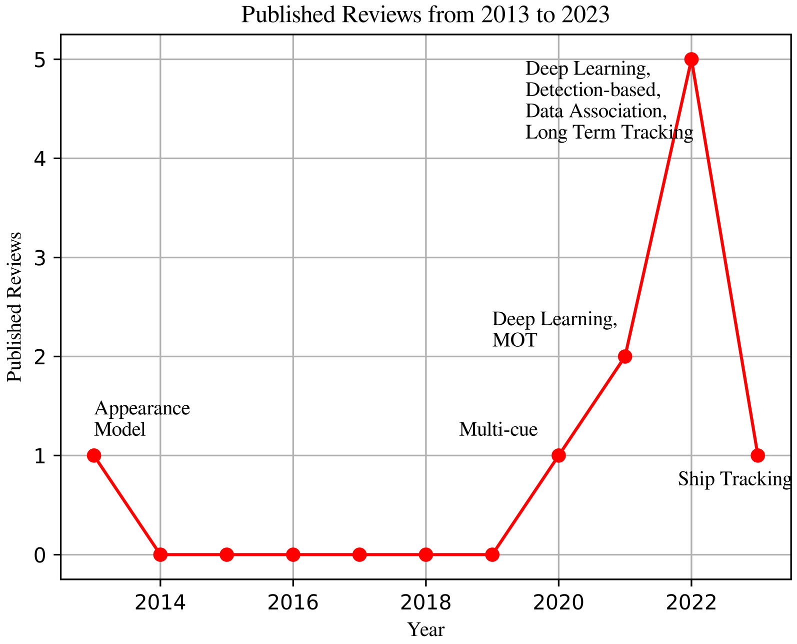

Figure 2. Trend in reviews from 2013 to 2023.

Table 1. Summary of the review works on object tracking.

Paper	Year	Topic	Main Contributions
[17]	2013	Appearance models in visual object tracking	• Review of visual representation according to their feature-construction mechanism. • Existing statistical modelling schemes for tracking by detection.
[19]	2020	Multi-cue-based visual tracking	• Categorisation of multi-cue object tracking based on the exploited appearance model into traditional architecture and deep learning-based tracker.
[20]	2021	Multiple object tracking in deep learning approach.	• Categorisation of previous MOT algorithms into 12 approaches and discussion of the main procedures for each category. • A review of the benchmark datasets and standard evaluation methods for evaluating MOT.
[21]	2021	Deep learning approaches in real-time multiple-object tracking	• Review of deep learning-based online MOT methods and networks that rank highest in the public benchmark.
[22]	2022	Deep learning frameworks for change detection	• Model design-based categorisation of the existing approaches. • Presentation of empirical analysis of evaluation settings for deep learning. • Future directions for change detection.
[23]	2022	Deep learning-based visual multiple-object tracking algorithm for autonomous driving	• Detailed review of object tracking methods: tracking by detection (TBD), joint detection and tracking (JDT), and transformer-based tracking.
[24]	2022	Detection-based video multiple-object tracking	• Taxonomy based on the MOT problem. • Summary of the results of 40 algorithms on popular datasets.
[25]	2022	Data association in multiple-object tracking	• Review of data association techniques via uniquely defined similarity functions and filters for multiple-object tracking. • Taxonomy of data association methods.
[26]	2022	Long term visual tracking	• Thorough review of long-term tracking, summarising the long-term tracking algorithms from framework architectures, and utilisation of intermediate tracking results' perspective.
[27]	2023	Ship tracking	• Review of datasets and state-of-the-art tracking algorithms for single- and multiple-object tracking. • Provides insights for developing novel datasets, benchmarking metrics, and novel ship-tracking algorithms.
Ours	2024	Object tracking in computer vision	• Systemic literature review on hardware usage, datasets, image processing and deep learning methods, and application areas. • Recommendations and guidelines for selecting sensors, datasets, and application methodologies based on their advantages and limitations. • Taxonomy for the sensor equipment and methodologies. • Research questions and future scope to address unresolved issues in the object tracking field.

3. Sensor Equipment

The development and implementation of object tracking methods begin with the sensor input. The choice of sensor equipment depends upon different constraints of the problem, such as depth requirement [10,28,29], tracking objects from multiple viewpoints [30], or intercepting the object following a certain trajectory [4]. Based upon the different problem constraints, different types of vision sensors such as monocular, stereo, depth-based camera, and hybrid vision sensors are used. Figure 3 shows the taxonomy of sensor equipment studied in the literature. The following sections categorise the research based on the types of vision sensors.

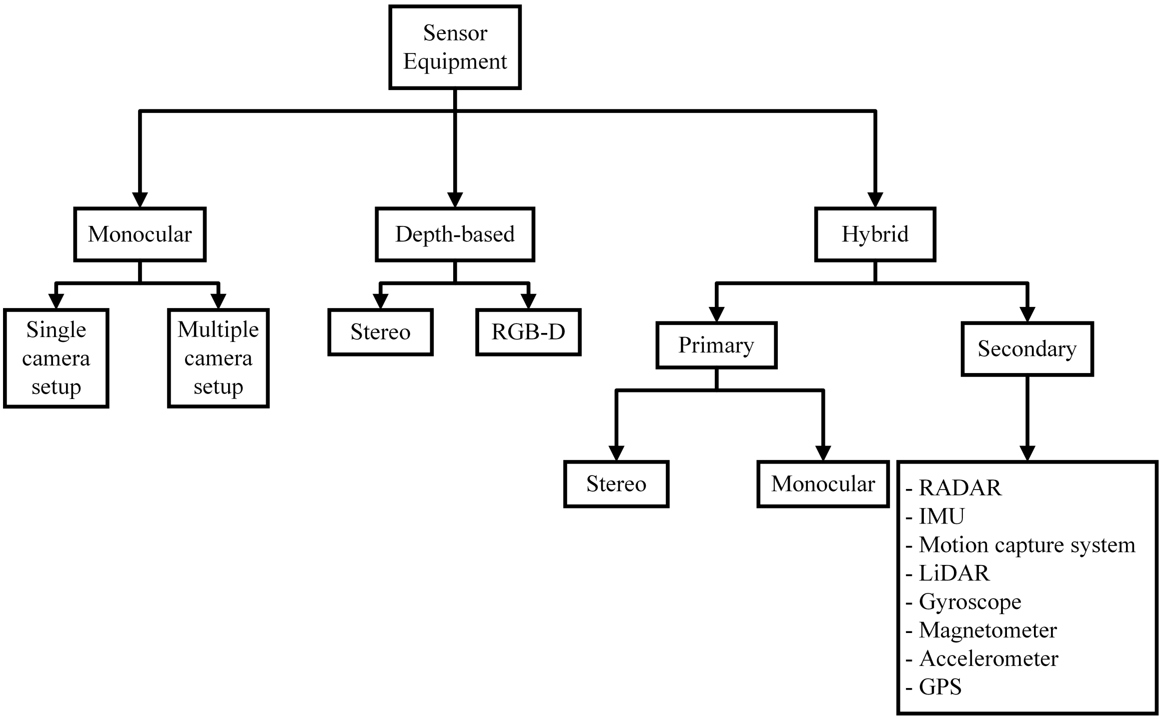

Figure 3. Taxonomy of sensor equipment.

3.1. Monocular Cameras

Monocular cameras are widely used in object tracking. A monocular camera refers to a single camera in a computer vision system, where the system relies on extracting information from a single image form the camera. While it is difficult to estimate the depth from a single image, some researchers incorporate multiple monocular cameras with the principles of stereoscopy that give the 3D position of the target object [30]. Considering the advantages and limitations of monocular vision, different methods are developed based on the information available from the single image or a modified system that incorporates multiple monocular cameras [30], eventually becoming uncalibrated stereo vision [31]. Since the cost and availability of cameras are important considerations in some applications, monocular cameras become a suitable option.

The camera setup is important for developing application-specific datasets. Kwon et al. [4] used a monocular camera to acquire images from a moving camera. Their approach for using a monocular camera was to derive homography matrices in estimating the pose of a target in six DOFs. Their proposed methods were to be used in a missile application, where the camera of the missile tracks a target missile as a moving object for interception. Their approach for overcoming depth and size information was to use the image sequences from the moving camera on the missile. The motion estimator used these images to estimate the rotational and translation motion of the free-moving target. Their research focused on deriving homography matrices for estimating the motion of a moving target using a monocular camera, and a practical simulation was designed. However, the performance of their methods depended upon accurate feature matching. Thus, any high-resolution monocular camera could be used to apply their methods.

Zarrabeitia et al. [16] used a single and two monocular cameras to detect the trajectories of a water droplet. Two monocular cameras allowed them to construct a stereo system for 3D trajectories. Yan et al. [32] used four fixed monocular cameras for handover problems in computer vision to track a skater as the skater escapes the field of view (FOV) of one camera to another. Gionfrida et al. [13] used a single monocular camera to capture the participant's images to develop a markerless hand motion capture system. They developed the ground truth for the hand movement with a marker-based approach using an eight-camera Qualisys motion capture system. They compared the motion obtained from a markerless monocular camera system with the ground truth. Huang et al. [33] developed a setup consisting of an overhead crane trolley, a camera, a spherical marker, a computer with a GUI connected to a motion control system, and a vision computer to process images and track the motion of a payload. The setup was designed in the lab, but it had the potential to be applied on outdoor overhead handling cranes.

The monocular camera setups have a unique application that solves a particular problem; however, the methods developed using these setups often require some modifications if the constraints of the problems change. The advantage of constructing a monocular camera setup is that multiple camera views can be used, which helps detect depth and address occlusion. Furthermore, multiple cues become accessible in the image by using different types of monocular cameras, such as infrared and RGB, on a setup. However, the disadvantage of such a system could be that a thorough calibration must be performed. Also, the delay in sequentially triggering multiple monocular cameras must be addressed since the data could be lost due to a delay in image capture in a dynamic environment. Knowing the capability and application is essential before selecting the appropriate camera system. Table 2 summarises the different types of camera systems used in literature with their depth estimation capability provided by the methods in the paper and their respective applications. Therefore, monocular camera setups are often developed when the problem has a unique requirement.

Table 2. Summary of monocular camera systems.

Paper	Camera System	Depth Estimation	Depth Estimation Method	Application
[4]	Moving camera	✓	Homography matrices	Missile interception
[16]	One or two cameras	✓	Stereo reconstruction	Bloodletting events (medical)
[32]	Four cameras	x	-	Tracking skaters (sports)
[13]	Single camera	x	-	Biomechanical assessment (Medical)
[33]	Single camera	x	-	Overhead crane

3.2. Depth-Based Cameras

Depth-based cameras provide images of the scene along with depth information. Stereo and RGB-D (RGB-Depth) cameras are the two types of depth-based cameras used in the object tracking literature. A stereo camera system comprises two or more monocular cameras, often as a single unit such as Bumblebee2 [10,28,29] or built from multiple monocular cameras [30]. RGB-D cameras such as Microsoft's Kinect sensor collect RGB images and depth information using an infrared (IR) projector and camera based on the principle of structured light [34]. Object tracking methods are developed by setting up the depth-based camera [12,28] or by using a public dataset [35] as in the case of monocular camera data. Since depth information is vital for machines to interact with their environment and know the location of the object in the real world, it is important to consider different depth-based camera setups for object tracking.

Stereo cameras are widely used in applications where depth measurement is required. Garcia et al. [36] developed a prototype of a stereo camera by using two static low-cost cameras. That stereo camera could be overhead in different urban environments with constant lighting. With the constraint of constant lighting conditions, the system was designed to track the movement, size, and height of the people passing under the camera. The system could be adjusted to operate at different heights depending on the urban environment by adjusting the system parameters to comply with the average height of the people and the camera location from the ground. Chuang et al. [11] used a stereo camera with six LED strobes, batteries, and computer housing for underwater operation. Their camera could have 4-megapixel images, and the data transfer rate was five frames per second using an Ethernet cable. Hu et al. [37] used two AVT F-504B cameras to construct a binocular stereo camera mounted on a tripod. They calibrated the camera using the calibration toolbox [38] in MATLAB. Yang et al. [15] used a binocular stereo placed in front of a person to collect data for hand gestures. Sinisterra et al. [29] mounted a Bumblebee2 stereo camera on top of an unmanned surface vehicle that was used for chasing a moving marine vehicle. Busch et al. [2] mounted their stereo camera on a manipulator arm attached to a drone for tracking tree branch movement. During the experimental procedures, they placed the stereo camera in front of the tree branch on an actuation system capable of performing sway action. Wu et al. [39] also developed a stereo camera mounted on a quadcopter with an NUC computer to detect and track a target. Richey et al. [12] used a stereo camera to track breast surface deformation for medical applications. Their setup consisted of an optical tracker, ultrasound, guidance display, and pen-marked fiducial points on the skin whose ground truth was collected by an optically tracked stylus. The depth information measured with the help of the stereo-matching process helps in the respective applications. Czajkowska et al. [14] used a stereo camera setup and a stereoscopic navigation system called Polaris Vicra to evaluate ground truth. Since a binocular stereo camera can be constructed by aligning two cameras or purchased as a single unit, the stereo setup is becoming popular when depth information is required.

RGB-D is another depth-based camera with an infrared projector and collector system to measure depth along with the RGB channels of the image [34]. The depth value relative to the position of the camera is collected for every pixel in the RGB-D camera. Kriechbaumer et al. [28] used RGB-D data for developing their methods; however, their methods were adapted to stereo later. Similarly, Rasoulidanesh et al. [40] used the RGB-D Princeton pedestrian dataset [41]. The use of RGB-D for tracking in the literature has been limited to public datasets developed using RGB-D cameras and in the indoor environment, as outlined by Kriechbaumer et al. [28]. An RGB-D camera has certain limitations when the object is far away, making it difficult for applications to track objects using drones [42]. Therefore, while RGB-D cameras have advantages in the indoor environment, they may not be suitable for outdoor applications due to their limited sensor range, which misses faraway objects.

Depth-based cameras are useful for localising the tracking object in a 3D space relative to the depth camera. Table 3 summarises the different types of depth-based vision sensors used in the surveyed literature. The table categorises cameras based on “Off the shelf” and “Constructed”. As the name suggests, off-the-shelf cameras are purchased as a single unit, while constructed cameras use different components, such as two monocular cameras, to construct a stereo camera. The advantage of using off-the-shelf products is that they often come with a software development kit that allows the user to use pre-built tools such as calibration, depth detection, disparity map, and point cloud map generation. The constructed camera would have an advantage where the problem constraint requires a custom baseline or camera lens, which may not be part of the off-the-shelf product. Furthermore, other aspects such as depth calculation methods, frames per second (FPS), and resolution play an important role in depth measurement accuracy and are often constraints on applications. Therefore, a depth-based camera has an advantage over a monocular camera as it provides all the information obtained from monocular (RGB image) and depth estimation capability.

Table 3. Summary of depth-based cameras.

Paper	Type	Off the Shelf	Constructed	Camera	Depth Calculation Method	Application	FPS	Resolution
[36]	Stereo	x	✓	Two static cameras	Epipolar geometry	Pedestrian tracking	30	320 × 240
[11]	Stereo	✓	x	Cam-trawl	Stereo triangulation	Tracking fish	5	2048 × 2048
[37]	Stereo	x	✓	AVT F-504B	Epipolar geometry	Pedestrian tracking	25.6	1360 × 1024
[29]	Stereo	✓	x	Bumblebee2	Stereo matching using SAD	Tracking ship	15	320 × 240
[2]	Stereo	✓	x	ZED	3D point cloud	Tree branch tracking	30	1920 × 1080
[39]	Stereo	✓	x	Mynteye	Stereo matching	Air and ground target tracking	25	752 × 480
[12]	Stereo	✓	x	Grasshopper	Stereo matching	Fiducial tracking for surgical guidance	5	1200 × 1600
[28]	Stereo	✓	x	Bumblebee2	Stereo triangulation	Autonomous ship localisation	8.2	1024 × 768
[40]	RGB-D	✓	x	KinectV2	Time of flight	Pedestrian tracking	30	1920 × 1080

3.3. Hybrid Sensors

In applications with uncertainties in vision data collection, additional sensors whose data can complement that of the vision data are used. These sensor setups are classified as hybrid sensors as they incorporate multiple sensors, which is important in the development of the method. Cesic et al. [10] mounted a stereo camera and radar on a moving vehicle in urban scenarios. Similarly, Ram et al. [43] also used radar and a monocular camera for autonomous cars, while Feng et al. [5] used a combination of monocular camera with an inertial measurement unit (IMU). Persic et al. [3] used a combination of stereo, monocular, and motion capture systems, monocular and radar, and monocular and LiDAR systems mounted on a car for autonomous driving. Kriechbaumer et al. [28] based their system on a platform on a survey vessel consisting of a Bumblebee2 stereo camera, an inertial measurement unit (IMU) fused with tri-axial MEMS gyroscope, accelerometer and magnetometers, a GPS receiver, a 360-degree prism, and a total station, which is an equipment used for land surveying. Contrary to detecting targets using drones, Zheng et al. [42] developed a panoramic stereo camera system on the ground to detect flying drones. Their platform comprised four stereo cameras mounted on a stand with a computer, IMU, router, and GPS module. The IMU and GPS were located on the ground node and used to measure the attitude and position of each sensing node in a global coordinate frame. Since the KITTI [35] dataset consists of different types of sensors, the research in [1,5,8,9,44] using this dataset also fit under hybrid sensors with the primary goal of localising a vehicle. Table 4 summarises the sensors based on primary sensors and a vision sensor along with the secondary sensor that complements the primary sensor. From the applications of different methods, hybrid sensors are used where the risk and uncertainties are high, such as in autonomous vehicles and drones. Therefore, for outdoor applications, combining vision sensor data with other sensor data to create a hybrid system is beneficial for high-risk applications.

Table 4. Summary of hybrid camera systems.

Paper	Primary Sensor	Secondary Sensors	Application
[10]	Stereo camera	• RADAR	Autonomous driving
[43]	Monocular camera	• RADAR	Autonomous driving
[5]	Monocular camera	• IMU	Autonomous driving
[3]	Stereo camera	• Motion capture systems • RADAR • LiDAR	Moving target tracking
[28]	Stereo camera	• IMU • Gyroscope • Accelerometer • Magnetometer • GPS	Autonomous ship tracking
[42]	Stereo camera	• IMU • GPS	Drone tracking

3.4. Recommendations for Sensor Selection for Applications

The sensor equipment is the first step to consider based on the type of object tracking application. The correct selection process for the sensor equipment is essential as it relies upon the capabilities of the sensor. Table 5 summarises the category of papers reviewed in the literature in this section. While application plays an important role in selecting a sensor type, other constraints, such as computing and hardware cost, must also be considered. This subsection aims to summarise, compare, and suggest guidelines for selecting sensors.

Monocular cameras, such as webcams, are accessible and less expensive than depth-based cameras. A high-resolution webcam can provide more details in terms of pixel density. However, the higher the resolution, the higher the computation cost to process the images. Furthermore, monocular cameras cannot provide depth information in the scene, but the depth information can be obtained using multiple monocular cameras [16] or a moving camera [4] along with the principles of stereography.

From the insights derived from the literature review, the following guidelines can be used to determine when monocular cameras are sufficient:

- If the tracking application does not require depth information.
- If the system does interact with its environment, such as tracking in sports [32], a biomechanical assessment [13], or observing pedestrian movements, a monocular camera is sufficient.
- If depth information is required, uncalibrated stereo methods can be used with either a moving camera [4] or multiple monocular cameras [16].

Depth-based cameras are more expensive compared to monocular cameras. The advantage of using depth-based cameras such as stereo cameras or RGB-D is that they provide depth information about objects relative to the position of the camera. This is beneficial information for localising a target object in the 3D space. Off-the-shelf depth-based cameras often have the advantage of proprietary software or a software development kit (SDK) provided by the manufacturer. The software provides functionality such as camera calibration, disparity map generation, and point cloud generation. An SDK often comes with the option of multiple programming languages, which provides pre-built code packages. These camera code packages, with features such as depth detection and point cloud generation, can be integrated within projects without the need to develop code from scratch for the camera input processing. Some of the functionalities of the SDK, such as real-time point cloud generation, often require high computer hardware specifications such as a GPU [2]. However, alternative software libraries such as OpenCV can be used to develop methods that do not require GPUs for image processing.

The following guidelines are recommended for selecting depth-based cameras for applications:

- Depth-based cameras are ideal if the depth information of the target object is needed.
- Stereo cameras are better than RGB-D ones in outdoor settings since an RGB-D camera relies on structured light, which may not be suitable for outdoor environments.
- RGB-D cameras are a better option than stereo cameras for indoor applications as the depth accuracy will be higher due to the structured light.
- A constructed stereo setup is a better option for a custom baseline, and the focal length of the lens is required for applications such as in panoramic stereo systems [42].

Hybrid sensors provide additional data for the overall application. For highly critical applications, such as autonomous vehicles, more data that can benefit the dynamic system, such as a moving vehicle in a dynamic environment, are essential. Sensors like IMUs, gyroscopes, and accelerometers can help maintain the stability of the dynamic system, while GPS helps localise it in 3D space. It is important to consider the stability of autonomous vehicles, their localisation in the environment, and other moving objects such as pedestrians and other vehicles.

The following are the recommendations for deciding on a hybrid system:

- Hybrid sensors are the best choice for a dynamic system interacting with a dynamic environment such as an autonomous vehicle [5,10,28,43].
- GPS as an additional sensor with the camera helps localise the camera system in the real world, thereby allowing the localisation of target objects.
- An IMU, accelerometer, and gyroscope provide additional data that can help the control system of the dynamic system for stability while tracking objects.

Table 5. Categorisation of papers based on the vision sensors.

Vision Sensor	Papers
Monocular	[4,13,16,32,33]
Depth-based	[2,6,11,12,14,15,29,36,37,39]
Hybrid	[3,5,10,28,42,43]

4. Datasets

Datasets are essential for evaluating methods and setting standards which cover a wide variety of scenarios. A diverse dataset is helpful to develop methods that can be evaluated before they are deployed in real-world systems. Some public datasets such as HumanEVA [45] and KITTI [35] cover various data catering to specific applications. In contrast, some others [7,42,43,46] develop their datasets for general tracking applications. Researchers who create an in-house dataset are looking for specific scenarios for their applications. The dataset is used for machine learning and deep learning methods to train a classifier for detection and tracking. Therefore, the availability of a dataset is essential for benchmarking the methods and training a machine learning or deep learning model to accomplish the tasks.

4.1. Object Tracking Datasets in Autonomous Vehicles

Research on autonomous driving has significantly increased in the past few years [47]. The KITTI dataset [35] is widely used for benchmarking the methods in autonomous driving applications. The KITTI dataset consists of high-resolution colour and greyscale stereo images, laser scans, GPS, and IMU data. Several researchers [1,5,8,9,44] developed their object tracking methods using the KITTI dataset in the application of autonomous driving. Deepambika and Rahman [9] also used the DAIMLER dataset [48], a pedestrian dataset, to evaluate their methods for autonomous driving. The DAIMLER dataset consists of stereo images captured from a calibrated stereo camera mounted on a vehicle in an urban environment. The pedestrian cutout is comprised of 24-bit PNG format images, float disparity maps, and ground truth shapes.

The Multivehicle Stereo Event Camera (MVSEC) dataset [49] is another stereo image dataset for event-based cameras developed for autonomous driving cars. The MVSEC dataset consists of greyscale images along with IMU data. The stereo camera was constructed from two Dynamic Vision and Active Pixel Sensors (DAVIS) cameras. A Visual Inertial (VI) sensor [50] was mounted on top of the stereo camera. This setup was mounted on a motorcycle handlebar along with GPS. A Velodyne LiDAR system was used to get the ground-truth depth information.

HCI [51] is a synthetic dataset comprising 24 designed scenes with the ground truth of a light field. The dataset comprises four images for three scenes: stratified, test, and training. These scenes consist of patterns and household images with their ground truth. They provide an additional 12 scenes with their ground truth in the dataset, which is not used for official benchmarking. Shen et al. [7] created their dataset for developing their methods by building on the HCI dataset for a potential application in autonomous driving. An autonomous driving dataset is often accompanied by additional sensor data such as GPS, IMU, and stereo camera images. Autonomous navigation is treated as an object tracking problem, and the dataset's availability can help benchmark the methods before deploying them for autonomous cars to avoid dynamic obstacles by tracking them in real time.

4.2. Single-Object Tracking Datasets

Single-object tracking (SOT) is the research area where a single object, as opposed to multiple objects, is the subject of the tracking. There have been different versions of Visual Object Tracking (VOT) datasets from its inception in 2013, with the latest being VOT2022 [52] as a part of the VOT Challenge. The VOT dataset consists of monocular images and is used to benchmark the methods for visual object tracking. Unlike MOT datasets, VOT datasets are for single object tracking.

In VOT2022 [52], the following evaluation protocols were used:

- **Short-term tracker** :
  - Target is localised and reported in each frame.
  - For the target that goes out of frame or gets occluded, there is no target re-detection from these trackers.
  - The information on the target object is not retained when the object is occluded.
- **Short-term tracking with conservative updating**:
  - Similar to the short-term tracker, the target is localised in each frame, and there is no re-detection of the target.
  - Tracking robustness is increased by a selective updating of the visual model based on the estimation confidence.
  - The tracking reliability relies on the confidence estimation, which is based on the object detection confidence, thereby performing a detection operation when the tracking estimation confidence is low.
- **Pseudo-long-term tracker**:
  - When the target position is predicted to be "not visible" due to occlusion or when the target is out of the image frame, it is not reported.
  - There is no explicit tracking re-detection, which means that when the object is occluded, the detection failure is reported, and there are no further efforts to search the object in the image frame.
  - There is an internal mechanism to identify tracking failure where the failure could be due to low confidence in the estimation, object detection, or both.
- **Re-detecting long-term tracker**:
  - Target position is not reported when the target prediction is "not visible".

  - Unlike a pseudo-long-term tracker, there is an explicit search over the image frame when the object is lost during tracking.
  - Object detection techniques can be employed to detect the object in the entire image frame.
  - Upon re-detection, the tracking is continued from the new location.

Object Tracking Benchmark (OTB) [53] is another single-object tracking dataset. OTB-50, consisting of 50 difficult target objects out of 100 targets from OTB [53], was used by Yan et al. [32] to evaluate their trackers. OTB has annotations consisting of 11 attributes: illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, and low resolution [53]. The Rigid Pose dataset [54] is a single-object tracking dataset created synthetically. Along with tracking, the dataset can also be used to evaluate methods for occlusion. The dataset consists of four objects from public KIT object model data [55]. These object models are placed on the image and manually manipulated to record the trace, which is used as ground truth.

Zhong et al. [56] used the Rigid Pose dataset for their evaluation. Furthermore, the ACCV14 dataset [57], an RGB-D dataset, was used for their evaluation. The Princeton [41] dataset is an RGB-D dataset used by Rasoulidanesh et al. [40] for evaluating their method for tracking the object along with depth. The Princeton dataset comprises 100 video clips with RGB and depth information and manually annotated bounding boxes as ground truth. Microsoft's Kinect 1.0 sensor was used for data collection with a depth range between 0.5 and 10 m. The Princeton dataset consists of three types of targets, with each scene having a different level of clutter in the background and occlusion.

HumanEva [45] is a multi-view synchronised motion capture dataset consisting of 40,000 frames for each camera. The HumanEva dataset is a pose estimation dataset of four human subjects performing six predefined actions. The ground truth for the motion was captured with ViconPeak, a commercial motion capture system.

Web crawling to download publicly available images on different websites has become more relevant [58]. The Stanford Cars Dataset [59] uses 16,185 images of 196 classes of cars. This dataset was used by Mdfaa et al. [46] to train a classifier for the moving-object class such as a car, and the Describable Textures Dataset (DTD) [60] was used for the non-moving class, such as buildings, in their application of tracking using a drone in a simulated urban environment. Stanford's car images dataset [59] was collected by web crawling popular websites. Then, a deduplication process was applied using perceptual hashing [61] to ensure distinct images belonged to a class. Then, Amazon Mechanical Turk was used to crowdsource the annotations. The DTD [60] consists of 5640 texture images annotated with 47 describable attributes. Like the Stanford dataset, DTD was also downloaded online instead of collecting images in the lab. Although both the Stanford and describable texture datasets are not developed for object tracking, they were used by Mdfaa et al. [46] for training a classifier that would be used for tracking by a detection approach. To evaluate their tracking methods, they used Visual Object Tracker (VOT) benchmarks [62–65]. Thus, a large dataset was available for training.

4.3. Multiple-Object Tracking Datasets

Multiple-object tracking (MOT) is a method in which multiple objects are tracked simultaneously in a given scene. Several datasets have been developed to benchmark the methods where multiple objects are present in a crowded environment. Pedestrian tracking is one such example where the video from a CCTV can be tracked over time. However, any problem in detecting and tracking multiple objects can be classified as an MOT-based problem. MOT [66] is a widely used dataset for evaluating multiple object problems. The MOT dataset, a part of MOTChallenge, has had several versions (MOT15 [67], MOT16 [68], MOT17 [68], and MOT20 [69]) over the years. The images in these datasets are a collection of images from publicly available datasets with standardised annotations. Luo et al. [70] reviewed the MOT tracking methods that outlined the collection of different MOT datasets.

The evaluation metrics are different for multiple object tracking. MOT20 [66] provided the following evaluation metrics:

- **Tracker to target assignment**:
    - No target re-identification.
    - Target object ID is not maintained when the object is not visible.
    - Matching is not performed independently but by a temporal correspondence in each consecutive video frame.
- **Distance measure**:
    - The Intersection over Union (IoU) is used to detect similarity between target and ground truth.
    - The IOU threshold is set to 0.5.
- **Target-like annotations**:
    - Static objects such as pedestrians sitting on a bench or humans in a vehicle are not annotated for tracking; however, the detector is not penalised for tracking these objects.
- **Multiple-Object Tracking Accuracy (MOTA)**:
  MOTA combines three sources of error: false negatives, false positives, and mismatch error.

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \tag{1}$$

    - t is the video frame index.
    - GT is the number of ground-truth objects.
    - FN and FP are false negatives and false positives, respectively.
    - $IDSW$ is the mismatch error or identity switch.
- **Multiple-Object Tracking Precision (MOTP)**:
  MOTP is the measure of localisation precision, and it quantifies the localisation accuracy of the detection, thereby providing the actual performance of the tracker.

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{2}$$

    - c_t is the number of matches in frame t
    - $d_{t,i}$ is the bounding box's overlap of target i with the ground truth object
- **Tracking quality measures**:
  Tracking quality measures how well the object is tracked over its lifetime.
    - The target is mostly tracked for successful tracking for at least 80% of its lifetime.
    - The target is mostly lost for successful tracking of less than 20% of its lifetime.
    - The target is partially tracked for the rest of the tracks.

Caltech's Pedestrian [71] dataset consists of a video recorded from a car comprising low-resolution images and occluded pedestrians. Wang et al. [72] used the first 1000 frames of the Caltech dataset for their Centretown sequence. Caltech's dataset consists of 10 h of video in traffic in an urban area taken from a vehicle. The dataset consists of 250,000 images along with 350,000 bounding boxes with labels and 2300 unique pedestrian annotations. Caltech's dataset also considered occlusion in their annotation, where they annotated the image frame with a bounding box even when the object was occluded. Three sequences were included in the data. MOT challenges keep improving upon their datasets by including different conditions in the image dataset for future development of MOT methods.

Different datasets were used to evaluate the object tracking methods over different applications. A diverse dataset helps evaluate the methods in different scenarios, improving their potential for adaptability to different real-world circumstances. For the pedestrian tracking problem, the PETS2009 sequence [73] was used. The PETS2009 sequence consists of an image sequence and its ground truth from the footage recorded outdoors in different weather conditions of people performing different behaviours [73]. The PETS2009 dataset was used by Gennaro et al. [30] and Wang et al. [72] for pedestrian tracking application. The region-based object tracking (RBOT) [74] dataset is a monocular RGB dataset developed to determine the pose, such as translation and rotation, of the objects. These are known objects, and their pose is relative to the camera.

4.4. Miscellaneous Datasets

Different from the public datasets, some researchers create their in-house datasets. The reason for creating a dataset is either the unavailability of the data for an application or the application of their methods in a niche case where public datasets are insufficient.

Several datasets were developed using stereo or multiple cameras to detect the 3D location of an object. Zheng et al. [42] developed a stereo vision dataset for tracking unknown MAVs. Yan et al. [32] built a dataset of skaters where the movements of the skaters were tracked over four different monocular cameras as a part of the handover problem in computer vision. Busch et al. [2] collected a dataset using a stereo ZED camera of a pine tree branch. The pine tree branch was mounted on an actuator system to simulate the movement of the branch when capturing the images. Hu et al. [37] build a fully labelled dataset of seven sequence pairs and 20 objects using a calibrated binocular camera. They annotated their dataset with similar attributes to that of OTB [53]. Cesic et al. [10] developed a radar and stereo vision-based dataset for an application in autonomous driving and MOT. The data were collected by mounting the sensors on a car driving in the centre of a three-way street. Kriechbaumer et al. [28] collected more than 15,000 images on a 50 m long reach of the river for the application of tracking surface vehicles. Most of these datasets are either private or available upon request. The use of multiple cameras helps in the localisation and tracking of an object in 3D space.

Datasets developed on monocular cameras are also helpful in 2D tracking. These types of datasets are often accompanied by additional sensor data such as radar or IMU data. Ram et al. [43] created a dataset using a monocular camera and radar equipment for automotive target tracking. Gionfrida et al. [13] developed a labelled dataset for monocular 2D tracking. Garcia and Younes [75] developed a dataset with 8746 images of a mock drogue for the automatic refuelling application of unmanned aircraft. Monocular camera-based datasets are useful when the object's 3D information is not required. However, they are often accompanied by additional sensor data for 3D tracking.

The data collection process is not feasible for some applications, such as aerospace and different illumination conditions. Therefore, researchers create synthetic datasets generated using mathematical models or computer-generated designs. Kwon et al. [4] developed a simulated dataset based on a mathematical model for the applications of missile interception. Biondi et al. [76] developed simulated data by exploiting mathematical models of a smooth Keplerian motion of the target. The Keplerian motion of the target was assumed to describe the equation that provides the position of the centre of mass of the target object and chaser vehicle in the earth-centred inertial frame of reference. They also included the occlusion period in their dataset. While synthetic datasets are readily available to test different methods, they must be evaluated to ensure their authenticity for application.

4.5. Recommendations for Dataset Selection

There are several public datasets available for evaluating methods. The public datasets used for developing and testing object tracking methods are mentioned in Table 6. Developing more datasets by addressing the lack of diversity in current datasets is helpful for the research community in developing better methods.

While the two main categorisations of datasets are single-object tracking and multiple-object tracking, they are further categorised based on their applications. Different uncertainties must be taken into account for autonomous driving, such as self-localisation, safe navigation, obstacle avoidance, and pedestrian detection. Therefore, while autonomous vehicles can be classified as a multiple-object detection problem, they deserve their own category due to their complexity and the research area dedicated to the application of autonomous navigation. Since autonomous vehicles include a range of vehicles, such as automobiles, ships, and aerial vehicles, different datasets cater to each type of application. This dataset is often developed with the help of hybrid sensors because they can provide multiple types of data for high-risk operations.

Single- and multiple-object detection datasets are similar with one exception: their names suggest that they track single or multiple objects. The approach to developing the datasets for single and multiple objects differs from its application and evaluation metrics. Miscellaneous datasets do not fit in either the SOT or MOT categories and were developed by researchers to solve particular problems. The trackers developed for these datasets are limited to the application for which the datasets were developed.

The following are the recommendations for selecting the datasets:

- SOT datasets are sufficient for indoor environments where the tracker is focused on one object.
- MOT datasets are ideal for any outdoor applications where multiple objects are tracked, and their trajectories need to be remembered by the tracker.
- A dataset can be developed and annotated manually or crowd-sourced using platforms like Mechanical Turk [59].
- A simulated or synthetic tracking dataset such as Kwon et al.'s [4] can be developed for applications where the data collection process is not feasible.

Table 6. Datasets used for developing and evaluating object tracking methods.

Dataset	Description	Sensor Type	Data Type	Used by	Links $^{+}$
KITTI [35]	High-resolution colour and greyscale stereo images, laser scans, GPS, IMU	Stereo + hybrid	MOT	[1,5,8,9,44]	https://www.cvlibs.net/datasets/kitti/
PETS2009 [73]	RGB images from the real world with multiple synchronised cameras	Monocular	MOT	[30,72]	ftp://ftp.cs.rdg.ac.uk/pub/PETS2009/Crowd_PETS09_dataset/a_data/
RBOT [74]	Semi-synthetic dataset with 6-DOF pose tracking	Monocular	SOT	[77]	https://github.com/henningtjaden/RBOT
MVSEC [49]	Event-based stereo images with IMU and GPS data	Stereo + hybrid + event-based	MOT	[6]	https://daniilidis-group.github.io/mvsec/
VOT [62–65]	Visual object tracking dataset	Monocular	SOT	[46]	https://www.votchallenge.net/
MOT (MOT15 [67], MOT16 [68], MOT17 [68], and MOT20 [69])	Collection of publicly available dataset	Monocular	MOT	[78–80]	https://motchallenge.net/
Rigid Pose [54]	Synthetic dataset with varying objects, background motion, occlusions, and noise.	Stereo	SOT	[56]	http://www.karlpauwels.com/datasets/rigid-pose/
Princeton [41]	Video clips along with depth information with manually annotated bounding boxes.	RGB-D	SOT	[40]	http://tracking.cs.princeton.edu
DAIMLER [48]	Pedestrian dataset with a single object class	Stereo	MOT	[9]	http://www.gavrila.net/Datasets/Daimler_Pedestrian_Benchmark_D/daimler_pedestrian_benchmark_d.html
Caltech pedestrian [71]	Pedestrian dataset with ten hours of footage	Monocular	MOT	[72]	https://data.caltech.edu/records/f6rph-90m20
HumanEva [45]	Human subjects performing predefined actions	Monocular + motion sensor	SOT	[81]	https://github.com/mhd-medfa/Single-Object-Tracker

+ The links to the datasets were accessed on 27 February 2024.

5. Approaches and Methods

Computer vision problems are being addressed with two main approaches: classical image processing and deep learning. Since object tracking is also a computer vision problem, these two approaches address this problem. Object tracking problems in computer vision are often divided into two steps: first, the object of interest is detected and then tracked over a sequence of images. The tracking is further divided into different approaches, such as tracking by detection, where the target object is detected in each image frame, and joint tracking, where the detection and tracking happen simultaneously. The tracking can be performed only when the input is a sequence where the object is within the image frame. There are instances where the object disappears because it goes out of the field of view of the camera or is obstructed by other objects. Keeping track of these objects in the middle of the video when they partially disappear has created a class of problems called occlusion. Different filtering and morphological operations are performed in the image processing methods to develop a model for detection and tracking [11,15].

Deep learning models use training data to develop a classifier that detects and locates the object [82–84]. After detecting the objects, both approaches involve using statistical or data association methods to track them. Some researchers aim to develop an end-to-end deep learning model using attention mechanisms to learn a classifier that can track the objects [40].

Apart from tracking by detection, joint detection methods detect the object in a frame and connect the location of the object for every subsequent frame in the video sequence. Another approach is detection by tracking where the objects are located in the first frame of the video. Then, statistical methods predict the future location, and the confidence score is increased further by detection [8,15,44].

Figure 4 gives the taxonomy of the approaches and methods used for object tracking that classifies the approach and categorises the methods in each approach. The following subsections also highlight the strengths and limitations of each approach. This section categorises the methods that rely solely on image processing and deep learning detection methods. Each of the tracking procedures and type of problem, such as MOT and SOT, are outlined in each category.

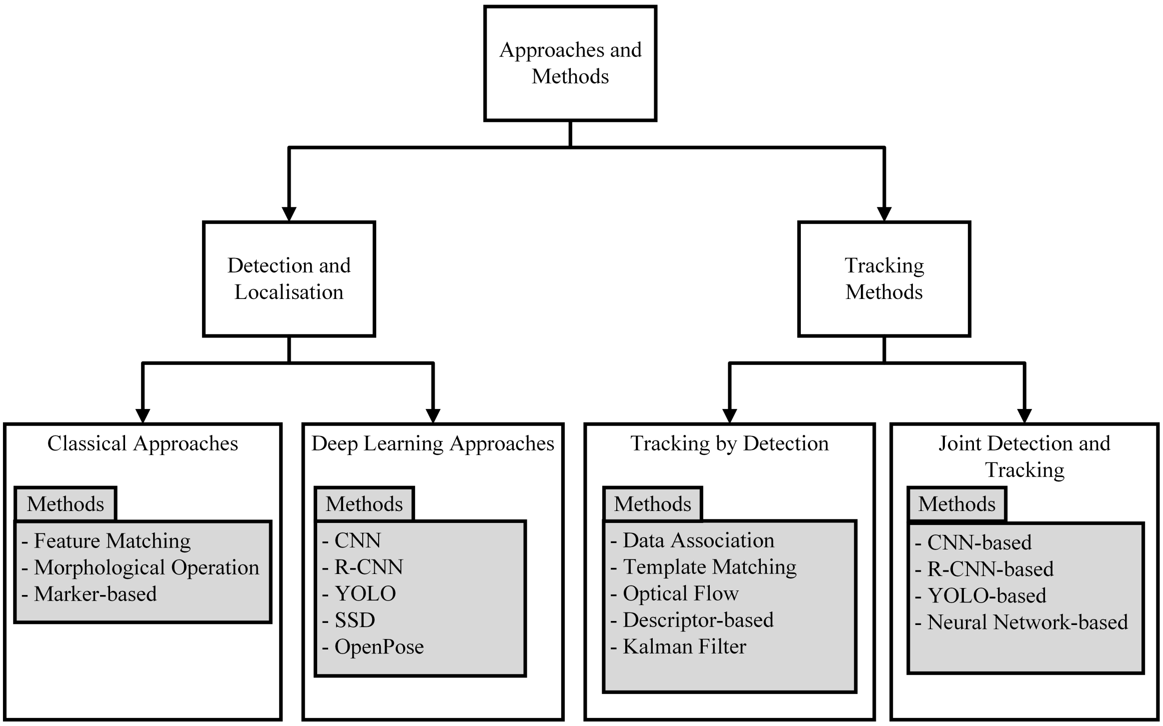

Figure 4. Taxonomy of approaches and methods for object tracking.

5.1. Detection and Localisation Methods

The first step in most tracking problems is detecting and localising the object. Detecting features and tracking those features using image processing has been an approach in many research studies for a long time. However, deep learning methods are becoming more prominent due to their higher accuracy and the use of end-to-end networks for localising and classifying objects. This section categorises and reviews the detection and localisation problems into image processing and deep learning approaches.

5.1.1. Classical Approaches

The classical approach encompasses the methods built using different image processing operations and algorithms. Since the operations and algorithms are tailored to fit the applications and datasets, no standard sets of operations are generalised for all the use cases. Furthermore, kernel size and threshold values are often empirically selected for different filtering and morphological operations in image processing [85]. Despite the tailored approach to solving the detection and tracking problem, some generalised steps are often used in many research approaches. However, researchers tweak the parameters to fit into their applications to find the optimal values that work with different operations and algorithms. The classical approach can be grouped by the methods that dominate these approaches. This paper further categorises the classical detection approaches into feature matching, morphological operation-based, and marker-based detection.

A. *Using feature matching*

Image matching deals with identifying features in the image and then matching them with the corresponding features on other images [86]. Kriechbaumer et al. [28] developed two algorithms for visual odometry for aquatic surface vehicles in a GPS-denied location. The first algorithm was based on image matching of sparse features [87] from the left and right input of the stereo camera along with consecutive stereo image frames where the input was a rectified greyscale image from a calibrated stereo camera. Additionally, a Kalman filter [88] was used for smoothing the estimated trajectory. The second algorithm was an appearance-based algorithm modified from the methods [89] developed for RGB-D cameras where the input of depth information was provided. Their experimental results were evaluated using ground-truth data collected using an electronic theodolite integrated with an electronic distance meter (EDM) and a total station, which is the equipment used in land surveying. Visual odometry enhances navigational accuracy on different types of surfaces. The position error with the feature-based technique was smaller than the appearance-based algorithm with a mean of ± 0.067 m, under the permitted limit of 1 m considered accurate. They performed a linear regression analysis that revealed that the error depended on the movement of the ship and the image features of the scene. Thus, the methods for environment surveying required further modifications depending on the type of application for river monitoring.

Jenkins et al. [90] developed methods for fast motion tracking by developing a fast compressive tracking method. They implemented a template matching technique using weighted multi-frame template matching and similarity metrics to detect the objects in consecutive video frames. They aimed to address problems such as occlusion, motion blur, and tracker offset. A bounding box with a confidence score was incorporated over the object detected with template matching over the image sequences. Overall, they developed a robust method to identify and keep track of the object in real time at an operating speed upwards of 120 FPS with minimal computation time. This was still dependent on the frame-by-frame template matching, and there was a potential of missed object detection in an image frame in case of occlusion.

Busch et al. [2] developed a method for detecting the branch of a pine tree by using the depth information from the stereo camera. They mounted the camera on a drone, and after calculating the depth of the features of the pine tree, they set a threshold of 0.6 metres to identify the ROI. The 0.6-metre threshold was arbitrarily selected as it would be the closest distance between the branch and the drone during the application. The distance threshold was used to generate a mask to isolate the ROI. They used a brute-force feature matching for the stereo matching operation from the

OpenCV [91] software library to calculate a 3D map of the tree branch to generate a point cloud of the branch. This detection approach was only limited to the pine tree branch detection.

B. *Morphological operation*

Morphological operations are a set of image processing operations that apply a structuring element that changes the structure of the features in the image. Two common types of morphological operations are erosion, where an object is reduced in size, and dilation, where the object is increased in size. A generalised way of approaching object tracking problems is tracking by detection. In tracking by detection, the focus is on detection operation in every image frame of a video sequence. Figure 5 shows a generalised diagram of tracking by detection, where the target object is detected, and the location information is stored and tracked for each video frame. The location of the object detected in each image frame of the video sequence is the tracking location of the object. Using stereo images, Chuang et al. [11] tracked underwater fish as an MOT problem. Their method included image processing steps such as double local thresholding, which includes Otsu's method [92] for object segmentation, histogram back-projection to address unstable lighting conditions underwater, the area of the object, and the variance of the pixel values within the object region. They developed a block-matching algorithm that broke the fish object down into four equal blocks and matched them using a minimum sum of the absolute difference (SAD) criterion. This detection process had too many morphological operations with varied parameters, such as kernel sizes and threshold values. Furthermore, the block-sized stereo-matching approach was innovative in reducing computation. However, it may not be a generalised solution to detect other aquatic life for applications in the fishing industry.

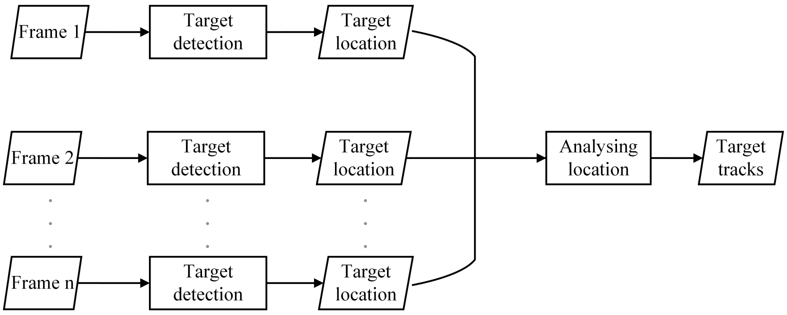

Figure 5. A generalised diagram of tracking by detection.

Yang et al. [15] developed a process for 3D character recognition with a potential for medical applications such as sign language communication or human–computer interaction in medical care by using binocular cameras. Their hand detection process involved converting the image from the RGB to YCbCr colour space and then applying morphological operations such as erosion [85] to eliminate small blobs not part of the hand. Then, they used Canny edge detection [93] to calculate the minimum and maximum distance of the edges in the image frame to determine the centre of the hand and then calculate the finger position, which would be the maximum distance from the centre. The tracking process relied on detecting the hand in each video sequence frame. The validity of hand gestures was determined by calculating the distance between the centre and the outermost feature. The distance value helped to know if the hand was not in a fist position and therefore, ready to be tracked. They further used stereo distance computing methods to track the feature in 3D space. Their method had several limitations, such as the hand needing to be the only skin exposed during the recording because if the face was visible, it would have been difficult to eliminate it during morphological operation, and it would have led to confusion regarding the location of the hand. Since the tracking relied upon detection, object location data were lost for any false negatives. The morphological operations could cause a loss of the exact location of the fingertip. Also, multiple processing stages in detection and tracking meant that the overall robustness of the system relied upon each stage working efficiently. Due to these reasons, there is a need for improvement in these methods for a robust implementation.

Deepambika and Rahman [9] developed methods for detecting and tracking vehicles in different illumination settings. They addressed motion detection using a symmetric mask-based discrete wavelet transform (SMDWT). Their system combined background subtraction, frame differencing, SMDWT, and object tracking with dense stereo disparity-variance. They used the SMDWT instead of the convolution or finite impulse response (FIR) filter method, as these lifting-based [94] methods are good in terms of computation cost. They used background subtraction and frame differencing, binarization and logical OR operations, and morphological operations for motion detection. Background subtraction allows the detection of moving objects from the present frame based on a reference frame. The output from the background subtraction and frame differencing was binarized for the thresholding operation to eliminate the noise in the image. Morphological operations could eliminate other undesired pixels. The next step was to obtain a motion-based disparity mask to extract the ROI for the object. Furthermore, the disparity map was constructed using SAD [95], a useful component for depth detection and stereo matching.

Czajkowska et al. [14] used a set of image processing steps to detect a biopsy needle and estimate its trajectory. They began by performing needle puncture detection. The detection algorithm applied a weighted fuzzy c-means clustering [96] technique to identify ultrasonic elastography recording before the needle touched the tissue. The needle detection was performed using the Histogram of Oriented Gradients (HoG) [97] detector.

C. *Marker-based*

Some detection methods use predefined markers. Markers are physically known objects the vision system has prior knowledge about. These markers are relatively easier to detect than markerless detection, which relies on feature extraction and comparison with the features of the target object. Huang et al. [33] developed a detection method for tracking the payload swing attached to an overhead crane. The payload detection was performed using the spherical marker attached to the payload. Similarly, Richey et al. [12] used a marker-based approach to detect breast surface deformations. Their marker-based detection approach used alphabets with specific ink colour and KAZE feature [98] detection for stereo matching. Using a marker-based approach reduces the computation cost in detection because the features to be detected in the image are known beforehand. However, the marker-based approach has certain problems, as object tracking only works for known objects in a controlled indoor environment. These methods are not ideal for tracking objects in the outdoor environment where the markers may be compromised due to external environmental factors such as wind or rain.

5.1.2. Deep Learning Approaches

Object detection uses a Convolutional Neural Network (CNN), a deep learning method. The primary use of CNNs in object tracking methods is to extract features for further template matching. Any deep learning methods capable of localisation and classifying the object in the image frame can be deployed in the object detection stage. This section investigates the different deep learning methods used to detect objects within the context of object tracking.

A. *R-CNN*

R-CNN [99] is an object localisation and classification method. R-CNN performs localisation and classification in two steps. First, different regions of the images are extracted and passed through a CNN for classification. If the object is detected in these extracted regions, it is localised in the image. Fast R-CNN [84] and its variants, such as Mask R-CNN [100] and Faster R-CNN [101] are other prominent object detection methods used within the context of object tracking for the detection stage. Meneses et al. [79] used R-CNN [99] to extract the detection features. Garcia and Younes [75] used Faster R-CNN [101] for object detection, where they trained the network on 8746 images of a mock drogue for its application to detecting a beacon. Li et al. [1] used Mask R-CNN [100] for object segmentation for segmenting vehicles in the application of autonomous driving. They developed the DyStSLAM method, which modified SLAM [102] to work in dynamic environments.

R-CNN [99] is beneficial for the localisation and classification of objects in an image. Detection windows of different sizes scan the image to extract small regions that are passed through the CNN for classification. This process ensures that different scales of objects are detected. However, the problem with this approach is that scanning multiple times over the images with different window sizes and passing each extracted region to classify the object is time-consuming. For the tracking-by-detection approach, the object detection process will be time-consuming for each image frame of a video sequence. Therefore, using R-CNN may not be ideal for real-time applications.

B. *Single-shot detection methods*

Single-shot detection methods such as Single-Shot Multibox Detector (SSD) [103] and You Only Look Once (YOLO) [82] can perform localisation and classification. These methods use default bounding boxes with different aspect ratios within the image to classify objects. The bounding boxes with higher confidence scores are responsible for object detection. YOLO [82] and its subsequent versions identified in the review by Terven et al. [104] have significantly improved object localisation, classification, pose estimation, and segmentation.

In the object detection for tracking, Aladem and Rawashdeh [8], Zhang et al. [80], Ngoc et al. [44], Wu et al. [39] used YOLOv3 [83], while Zheng et al. [42] used YOLOv5 [105]. Xiao et al. [78] used a Fast YOLO [106] network to localise a pedestrian object in each video frame and at the same time, they used the MegaDepth [107] CNN for the depth estimation.

The advantage of SSD [103] or YOLO [82] over R-CNN [99] is that both the localisation and classification process happen in a single pass through the CNN. Due to the single-pass detection, these methods are better than R-CNN for real-time applications. SSD and YOLO require a large dataset and computational power to train. Also, the detection is limited to the training images used to train the network. Therefore, it is important to consider if the target object class is present in the training dataset for these networks before deploying these methods for tracking.

C. *Other CNN methods*

Yan et al. [32] used CNN as a feature extractor and used these features in the template matching approach. Mdfaa et al. [46] used a CNN whose architecture was designed with the augmentation of SiamMask [108] and MiDaS [109] architectures where each of them was trained separately. ResNet18 [110] was used for binary classification, and two datasets, the Stanford Cars Dataset and Describable Textures Dataset (DTD) [60], were used for training. Gionfrida et al. [13] used OpenPose [111] to detect the hand pose for further tracking. DyStSLAM helps localise an autonomous vehicle by extracting dynamic information from the scene. The deep learning methods incorporated in detection are used or developed based on the applications. Faster detection methods are helpful when the applications are on a real-time system like autonomous driving. Thus, deep learning methods should be evaluated on these datasets with the development of new datasets. If the results are not accurate enough, they will motivate the development of new methods.

5.2. Tracking Methods

The tracking process takes place after object detection. The tracking method keeps track of the movement of the object over multiple video sequence frames. This subsection highlights the tracking methods based on the image processing framework, while identifying their strengths and weaknesses. Approaches towards tracking methods use the multi-step image processing approach or end-to-end deep learning methods. In image matching, the standard procedure is to identify the features of the object and match them in consecutive video frames. The image matching technique is often accompanied by data association methods that help to keep track of the object. The deep learning methods often use end-to-end networks trained on image sequences. Deep learning can also be a two-step approach where detection occurs before tracking, and the network tracks the features in the subsequent frames. The literature outlines the two approaches used for object tracking.

5.2.1. Tracking by Detection

Tracking-by-detection (TBD) methods involve detecting objects in each image frame without prior knowledge or estimation of their future state. The object is associated with the previous detection [23].

A. *Data association*

Data association is the process of using previously known information about the object pose, movement, and change in appearance and comparing it with the newly identified objects and tracking movements of the object [25]. Data association is one of the most used methods for tracking and it is often modified as per the specifications of the applications. Chuang et al. [11] developed tracking for low-frame-rate video to track live fish. Their method used stereo matching by dividing the fish object into four blocks of equal size. The four blocks were formed by taking four equal column widths of the object's bounding box. These blocks in each of the left and right images of the stereo were matched using the sum of absolute difference (SAD). The stereo-matching process was followed by feature-based temporal matching, where four cues, such as vicinity, area, motion direction, and histogram distance, were considered. They further modified the Viterbi data association used in single-target tracking to multiple tracking, using the Viterbi algorithm [112] for tracking. Since the video had low contrast and a low frame rate, the Viterbi data association process helped track the object in multiple frames.

Feng et al. [5] used 3D bounding boxes generated by an object detector [113]. These bounding boxes were the basis for a multilevel data association method and a geometry-based dynamic object classification method, enabling robust object tracking. The system also introduced a sliding window-based tightly coupled estimator that optimised the poses of the ego vehicle with the sensors mounted on it, IMU biases, and object-related factors that formed different features of the dynamic objects. This approach allowed for the optimisation of both the vehicle and object states. These tracking methods used visual odometry data for self-localisation and object detection to know the position of the object relative to the vehicle. Their approach required further development for tracking non-rigid objects and testing their methods in real-world applications.

Zhang et al. [80] proposed a Multiplex Label Graph based on graph theory. This graph was developed so that each node stored information about multiple detectors. A CNN generated these detectors from the Part-Based Convolution Baseline (PCB) [114] network that was trained on the Market-1501 dataset [115]. They treated the object tracking in the frame as a graph optimisation problem where the goal is to find the path of a detector in multiple image frames of a video sequence. To achieve this, they broke down the video frames into a group of images called "window" and detected the object within each successive frame in the window. They tested different window sizes on MOT16 and MOT17 [68] datasets and determined that a window size of 20 was the optimal value that increased tracking accuracy. Then, a data association was performed with certain threshold functions that identified whether the nodes in the successive frames were associated. The distance between the nodes in the successive frames checked that association.

B. *Template matching*

Template matching is a process of identifying small parts of the target image that match the features using cross-correlation methods to a template image of the object by scanning the target image [116]. Jenkins et al. [90] developed their methods to track different types of objects available in the tracking dataset [117]. For this purpose, they implemented a template matching technique using weighted multi-frame template matching to detect the objects in consecutive video frames. The weighted multi-frame template approach was tested using similarity metrics such as normalised cross-correlation and cosine similarity. The results of the similarity metrics showed a significant increase in accuracy on their chosen evaluation dataset. Overall, they developed a robust method to identify and keep track of the object in real time with minimal computation time. Tracking robustness depended upon frame-by-frame template matching, which may pose problems during the detection of any false negatives during the tracking stage.

Yang et al. [15] developed tracking methods for tracking the movement of hands in medical applications. The tracking process was performed by detection. They used hand gestures to automate the decision-making process regarding the beginning and end of the tracking process. They further used stereo-matching methods to compute the distance between the camera and the hand, allowing them to track the hand in 3D space. Their method relied on detection, which means that tracking information would be lost for any false negative detection.

Richey et al. [12] developed tracking methods for breast deformation while the patient was supine, and the video frames were collected using stereo cameras during the hand movement of the patient. The labelled fiducial points, with the alphabet written in blue ink on the breasts, were tracked over the video frame. The labels were propagated through a camera stream by matching the key points to previous key points. The features obtained from these fiducial points leveraged the ink colours and adaptive thresholding, which were tracked using KAZE [98] feature matching. The features were stored in order to be tracked over the sequences of images. This method relied upon detecting all 26 English alphabets written on the breast; therefore, a detection failure may disrupt the tracking process.

Zheng et al. [42] tracked drones from a ground camera setup. They proposed a trajectory-based Micro Aerial Vehicle (MAV) tracking algorithm that operated in two parts: individual multi-target trajectory tracking within each sensing node based on its local measurements and the fusion of these trajectory segments at a central node using the Kuhn–Mumkres [118] matching matrix algorithm. This research introduced an MAV monitoring system that effectively detected, localised, and tracked aerial targets by combining panoramic stereo cameras and advanced algorithms.

C. *Optical flow*

Optical flow deals with the analysis of the moving patterns in the image due to the relative motion of the objects or the viewer [119]. Czajkowska et al. [14] developed a tracking method for needle tracking. The detection step provided information about the position of the needle. The tracking of needle tips focused on the single-point tracking technique. Methods like Canny edge detection [93] and Hough transform [120] were used for the trajectory detection. To implement the tracking process in real time with low computation resources, they considered using the Lucas–Kanade [121] approach that helped solve the optical flow equation using the least square method. Finally, they used the Kanade–Lucas–Tomasi (KLT) [122] algorithm that introduces the Harris corner [123] features. Furthermore, the pyramid representation of the KLT algorithm was combined with minimum eigenvalue-based feature extraction to avoid missing the tracking point of the needle. The two paths used for tracking were helpful in addressing both cases of fully and partially visible needles with ultrasonic images. Their method had a low computational cost in tracking, so it could be used in real time.

Wu et al. [39] designed and implemented a target tracking system for quadcopters for steady and accurate tracking of ground and air targets without prior information. Their research was motivated by the limitations of existing unmanned aerial vehicle (UAV) systems that failed to track targets accurately in the long term and could not relocate targets after they were lost. Therefore, they developed a vision detection algorithm that used a correlation filter, support vector machines, Lucas–Kanade [121] optical flow tracking, and the Extended Kalman Filter (EKF) [124] with stereo vision on a quadcopter to solve the existing detection problems in UAVs. Their visual tracking algorithm consisted of translation and scale tracking, tracking quality evaluation and drift correction, tracking loss detection, and target relocation. The target position was inferred from the correlation response map of the translation filter. Based on the target position, the target scale was predicted by a scale filter [125]. Then, the drift of the target position was corrected with an appearance filter that detected if the target was lost and allowed the tracking quality evaluation, which had a similar structure to that of the translation filter. Furthermore, the tracking quality was evaluated by the confidence score, composed of the average peak-to-correlation energy (APCE) and the maximum response of the appearance filter. If the confidence score exceeded the re-detection threshold, the target was tracked successfully, and the translation and scale filters were updated. Otherwise, the SVM classifier was activated for target re-detection. They made improvements on the Lucas–Kanade [121] optical flow and Extended Kalman filter algorithms to estimate the local and global states of the target. Their simulation and real-world experiments showed that the tracking system they developed was stable.

D. *Descriptor-based*

Descriptors are the feature vectors of the object that capture unique features that help to classify a particular object [126]. Aladem and Rawashdeh [8] used the YOLOv3 detector as a tool to create an elliptical mask by using a bounding box to extract the features for a feature detector such as Shi–Tomasi's [127] for feature matching. The feature matching process was followed by Binary Robust and Oriented Features (BRIEF) [128] for matching between the consecutive frames. Their method was for the odometry data evaluated on the KITTI [35] dataset. There were certain limitations, such as losing the objects and being unable to detect them. When the same objects reappeared, they were classified as new objects. They suggested that using a Kalman filter [88] in the future would help to deal with the missing object problem during detection.

Ngoc et al. [44] used the features from YOLOv3 [83] for tracking. The features extracted within the bounding box of this object detector were used in the particle filter algorithm [129]. These particles were tracked in the subsequent frames of the KITTI dataset [35]. While solving this problem, they also focused on identifying multiple objects when the camera was in motion. They took a hybrid approach, using stereo and IMU data for target tracking. Their method also took into account the camera movement. Their method had a future scope of application in mobile robotics.

E. *Kalman Filter*

Kalman filtering is an algorithm that uses prior measurements or states and produces estimates for future states over a time period [88]. The Kalman filter has a wide range of applications where the future state estimate of the object of interest is required , such as guidance, navigation, and control of autonomous vehicles. Since the target object in a video sequence shows the same property of moving states where state estimates are required, the Kalman filter is applied in object tracking problems.

Busch et al. [2] tracked the movement of a pine tree branch. They tested different types of feature descriptors such as SIFT [130], SURF [131], ORB [132], FAST [133], and Shi–Tomasi [127]. Their results showed that FAST-SIFT and Shi–Tomasi combinations performed best at 1 m and a camera perspective of 0 degrees. These numbers indicated the optimal position and orientation of the camera on the drone for collecting the pine tree branch data. These features were further filtered and mapped to 3D space to create a point cloud. The principal component analysis method was used to detect the direction of the branch. A developed Kalman filter [88] was derived that improved the intercept point estimation of the pine tree branch, which was the point at 75 mm from the tip of the branch. This developed Kalman filter reduced the intercept point error, which was helpful when determining the intercept point as the sway parameter.

Huang et al. [33] developed a method where a Kalman filter initially predicted the target position [88]. The tracking ball area was obtained through mean shift iteration and target model matching. Since mean shift has problems with tracking fast objects, combining it with a Kalman filter offers stability in detection since a Kalman filter is useful in estimating the minimum mean square error in the dynamic system. Then, the minimum area circular method was integrated to identify the position of the tracking ball correctly and quickly. The recognition part was more robust when an auxiliary module that pre-processed the area determined by the mean shift iteration was proposed. Geometric methods obtained the swing angle for the ball mounted on the crane payload. Their method was tested on an experimental overhead crane with a swing payload setup. Therefore, the methods may need further modification when the vision tracking system is applied to an outdoor overhead traveling crane with background disturbances and unexpected outdoor environmental factors such as wind and illumination.

5.2.2. Joint Detection and Tracking

Different from tracking by detecting, joint tracking methods are end-to-end trainable networks where tracking and detection are performed in a single network [23]. Different research groups have experimented with available CNN architectures, with more research literature being added. With the development of more methods, the deep learning approach can be further classified based on their methods. In this section, deep learning approaches for tracking are categorised based on CNN-based, R-CNN-based, YOLO, and other neural network-based methods. Deep learning methods for tracking are investigated by different reviews [21–23] that focus on MOT methods and their application for autonomous driving. In this subsection, the deep learning approach is classified based on the primary methods used for localisation for tracking by detection and joint tracking.

A. *CNN-based approaches*

Convolutional Neural Network-based approaches involve using deep learning methods for feature extraction to track these features in consecutive video frames. Rasoulidanesh et al. [40] developed a tracking method with an RGB and depth frame input. The spatial attention network extracted a glimpse from these data as the part of the frame where the object of interest was probably located. Then, the features of the object were extracted from the glimpse using a CNN with the first three layers of AlexNet [18]. The glimpse could extract two types of features: ventral and dorsal. The former extracted appearance-based features, while the latter aimed to compute the foreground and background segmentation. These features were then fed to an LSTM [134] network and fully connected neural networks to give a bounding-box correction. The bounding-box correction was fed back to the spatial attention section to compute the new glimpse and appearance for the next frame to improve object detection and foreground segmentation. They showed that adding depth increased accuracy, especially in more challenging environments. Their results showed that the depth-based models could perform accurate tracking with only depth information, without RGB.

Zhong et al. [56] used an encoder–decoder network. They proposed to combine a learning-based video object segmentation module with an optimisation-based pose estimation module in a closed loop. After solving the current object pose, they rendered the 3D object model generated on a computer to obtain a refined, model-constrained mask of the current frame. It was then fed back to the segmentation network for processing the next frame, closing the whole loop. To detect the occluded object, they designed a novel six-DOF object tracking pipeline based on a mutual guidance loop of video object segmentation along with six-DOF object pose estimation and combining learning and optimisation methods. They presented a robust six-DOF object pose tracker that could handle heavy occlusions. The experiments showed that their method could achieve competitive performance on non-occluded sequences and significantly better robustness on occluded sequences.

Yan et al. [32] developed a tracking method for the handover problem. They proposed a tracking algorithm that improved the tracking accuracy based on the MDNET [135], which is a multi-domain network. The target state in the initial frame of the video sequence was given, and the tracking was started. Then, the target handover began when the target crossed the field of view (FOV) line of the camera. The target feature extracted by a CNN was used for template matching. When the target handover was completed, the target was tracked in the next camera. In their research, they mainly improved the accuracy of target tracking and target handover. In terms of tracking, they improved on the original MDNET algorithm. In addition, they combined perspective transformation with features extracted by a CNN to realise the target handover.

B. *R-CNN-based approaches*

Meneses et al. [79] used R-CNN to extract features. The data association method used these features to track the object. They developed SmartSORT, which modelled the frame-by-frame association between new detections and existing targets as an assignment problem. They considered neural networks trained with the backpropagation algorithm as the regression model. Thus, given that the feature vector from R-CNN was related to the detection and the target, the regression model calculated their association cost. Once the regression model had computed every association cost, it optimally solved the assignment problem via the Hungarian method [136], which is an optimisation method that selects the best possible cost for a combination of activities, in this case, the tracking path over the frame of images.

Garcia and Younes [75] developed a tracking system that worked by capturing an image with a Kinect camera sensor, which acted as an input to a deep learning object detector using Faster R-CNN [101], which output the bounding box around each of the eight beacons on a drogue used to refuel an aircraft. Then, the navigation algorithms that used non-linear least squares and collinearity equations were used to find the position and orientation of the drogue, thereby allowing the aircraft to align with the beacon for refuelling. They performed their experiments on a mock drogue and verified their solution using the VICON motion tracking system. There were issues with the trained detectors with the inference time being too large. Also, they made several assumptions regarding using a mock drogue, and their image dataset was too small for training with limited augmentation.

C. *YOLO and other neural network-based approaches*

Mdfaa et al. [46] developed methods that used depth information and training data to train a Siamese network [137] to track an object. Since their application involved tracking a moving object using an aerial drone, they developed a system in which the drone kept following the object until it reached its location or the moving object stopped. In this type of tracking, there are two sub-tasks: identifying the tracked object and estimating its state, which is its position and orientation. The objective of the tracking mission is to automatically predict the state of the moving object in consecutive frames given its initial state. Their proposed framework combined

2D SOT with monocular depth estimation (RGB-D) to track moving objects in 3D space. Using this information, the Siamese network tracked the target object, which produced a mask, a bounding box, an object class, and an RPN score for the object.

Xiao et al. [78] used Fast YOLO [106] and MegaDepth [107] for detection and depth estimation. The results from these two networks were used as features for object detection and tracking using a Kalman Filter [88]. They proposed an algorithm that helped them track the pedestrian object in the video frame and developed data association rules regarding remembering the objects in case of occlusion. They developed a method that tracked the movement of multiple objects in 3D space on a video. However, their real-time tracking needed improvement for a dynamic system that interacts with the environment.

Yang et al. [6] developed the Self-Attention Optical Flow Estimation Network (SA-FlowNet) for applications on event-based cameras. SA-FlowNet independently uses crisscross and temporal self-attention mechanisms that help capture long-range dependencies and efficiently extract the temporal and spatial features from the event stream. Their proposed network used an end-to-end learning method to adopt a spiking-analogue neural network architecture. It gained significant computational energy benefits, especially for Spiking Neural Networks (SNNs) [138]. Their network architecture was based on a deep spike-analogue neural network architecture that combined event cameras for energy-efficient optical flow estimation. Their network could achieve higher performance and save energy consumption. It could also be used for object detection, motion segmentation, and challenging scenery tasks in dim light, occlusions, and high-speed conditions.

5.3. Recommendations for Approaches and Methods for Applications

The methods for object tracking in computer vision rely on object detection followed by tracking the detected object. The reliance on object detection before tracking ensures that object detection methods are studied and improved. This review outlines a detailed study of the detection methods incorporated into the object tracking literature over the last ten years.

Based on the insights gained from the literature survey and the identification of advantages and limitations of different methods as presented in Tables 7 and 8, the following recommendations are made for the selection of object detection methods:

- The classical approach is helpful when the target object can be identified by its geometry and where the computation resources and annotated datasets are limited to train a deep learning model.
- Deep learning approach in detection for tracking applications is helpful for objects with no standard geometry where the annotated dataset and computational resources are available.

The object tracking process involves keeping track of the detected objects over different video frames. Some methods detect objects in each video frame and then use association techniques to match the detection. This process of detecting objects in each image frame and later connecting the tracks is called tracking by detection (TBD). A different approach to tracking involves joint detection and tracking (JDT), where an end-to-end framework is used with estimation techniques to predict the objects in the next frame by using object features from the previous frame. Figure 6 shows a generalised diagram of end-to-end tracking using prior knowledge.

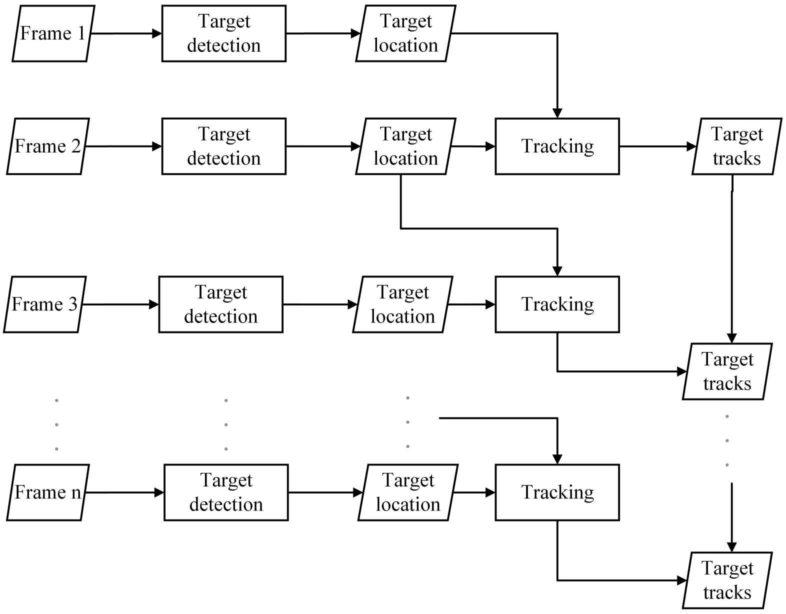

Figure 6. A generalised diagram of end-to-end tracking using prior knowledge.

Table 7. Summary of classical approaches for detection.

Paper	Key Methods	Advantages	Limitations
[28]	Sparse feature image matching, Kalman filter	Enhances navigational accuracy using visual odometry techniques, particularly useful in GPS-denied environments.	Relies on accurate feature matching and may not be ideal for objects without known feature geometries.
[90]	Template matching, weighted multi-frame template, confidence scoring	Provides a fast and robust method for object tracking in real-time video streams.	Template matching methods may not be suitable for different environmental conditions.
[2]	Depth-based feature matching, thresholding, point cloud generation	Effective for detecting specific objects in complex environments using depth information.	Limited to applications where depth information is available and may not generalise well to scenarios with different types of objects or backgrounds.
[9]	Morphological operations, wavelet transform, object tracking	Robust approach for vehicle detection and tracking in varying illumination conditions.	Accurate motion detection and further tests are required to address fast-moving uncertain objects.
[14]	Fuzzy clustering, HoG feature detection	Effective for detecting and tracking biopsy needles in medical applications.	Requires accurate needle puncture detection and feature extraction. Further tests are needed to ensure higher performance in scenarios with complex tissue structures or noisy ultrasound images.
[33]	Marker-based detection, geometric methods	Provides a reliable method for tracking payload swing in overhead cranes.	The methods were tested on a prototype in the laboratory setting, and the results of real-world data would confirm the robustness of the methods.
[12]	Marker-based detection, KAZE feature matching	Effective for detecting breast surface deformations using markers and stereo matching.	Using alphabets as markers sets the marker limits to 26 markers based on the English alphabet. A different marker identification system is required to overcome this limitation. Also, the method is suitable for detecting markers with a particular ink colour.
[11]	Stereo matching, block matching, Otsu's thresholding	Enables tracking of underwater fish using stereo image processing techniques.	The block stereo matching helps detect the fish. Morphological operations with arbitrary threshold values are used. The block-matching approach is not general enough to detect a variety of aquatic life.
[15]	Morphological operations, feature detection, stereo tracking	Provides a method for 3D character recognition and tracking using stereo vision.	The hand must be the only skin exposed during the recording because if the face is visible, it would be difficult to eliminate it during morphological operation, and it would lead to confusion regarding the location of the hand.

From the insights in terms of advantages and limitations of different methods and approaches presented in Tables 9 and 10, the following are the recommendations for the selection of tracking approaches:

- The tracking-by-detection method is useful to track multiple objects when the objects are not often occluded.
- Using data association methods is useful to track the trajectories of the target objects.
- Joint detection and tracking is useful when a dataset for tracking for a specific application and the computational resources are available to develop an end-to-end framework.

Table 8. Summary of deep learning approaches for detection.

Paper	Key Methods	Advantages	Limitations
[1,75,79]	R-CNN, Faster R-CNN for object detection, Mask R-CNN for object segmentation	Effective for object localisation, classification, and segmentation. Widely used in various applications like beacon detection and autonomous driving.	Time-consuming due to scanning multiple regions with different window sizes for each image frame and may not be suitable for real-time applications. Requires extensive training on target-specific datasets.
[8,39,42,44,78,80]	YOLOv3, YOLOv5, Fast YOLO for object detection	Performs localisation and classification in a single pass through a CNN; suitable for real-time applications. Efficient object detection for tracking without prior information.	Requires large datasets and computational power for training. Detection limited to classes present in the training dataset and may misclassify untrained class of object.
[32,46]	Custom CNN architecture for feature extraction, object detection	Combines deep learning features with traditional approaches. Incorporates multiple architectures for improved object detection performance.	Resource-intensive training process. Requires large datasets and computational power.
[13]	OpenPose for hand pose detection	Provides accurate hand pose detection for further tracking applications.	Dependent on the quality of the input data and the performance of the OpenPose model.

Table 9. Summary of tracking-by-detection methods.

Paper	Key Methods	Advantages	Limitations
[11]	Stereo matching, feature-based temporal matching, Viterbi data association	Effective for low-frame-rate video tracking, integrates stereo matching and feature-based matching for robust tracking.	Viterbi data association may introduce computational cost and may not perform optimally in scenarios with high object occlusions.
[5]	Multilevel data association, geometry-based dynamic object classification	Robust tracking based on 3D bounding boxes and dynamic object classification.	Further development is needed for tracking non-rigid objects and testing in real-world applications.
[80]	Multiplex Label Graph based on graph theory, CNN-based object detectors	Offers a novel approach to object tracking using graph optimisation techniques.	Computational complexity may be high, and optimisation parameters may require tuning for different scenarios.
[90]	Weighted multi-frame template matching	Robust template matching technique for real-time object tracking.	Relies on accurate template matching in consecutive frames, and it may suffer from computational complexity in scenarios with high frame rates.
[15]	Stereo matching, 3D tracking	Enables 3D tracking of hands in medical applications using stereo matching.	Tracking relies on accurate detection, may lose tracking information for false negative detections.
[12]	Feature extraction, fiducial tracking, KAZE feature matching	Tracks fiducial points on the breast for deformation analysis using stereo cameras.	Relies on accurate fiducial detection and may face challenges with detection in scenarios with complex backgrounds or lighting conditions.
[42]	Trajectory-based tracking, Kuhn–Mumkres matching matrix algorithm	Effective for tracking MAVs using panoramic stereo cameras and trajectory optimisation algorithms.	The method may face challenges with fast-moving objects or environments with limited visual cues.
[14]	Lucas–Kanade optical flow, KLT algorithm	Provides real-time needle tracking using optical flow and feature matching techniques.	Requires robust feature extraction and matching algorithms, and the accuracy may be affected in scenarios with rapid motion or complex backgrounds.
[39]	Correlation filter, SVM classifier, Lucas–Kanade optical flow, EKF	Stable and accurate target tracking system for UAVs using a combination of visual detection algorithms.	Complex algorithmic pipelines may introduce computational overhead and require fine-tuning for different UAV platforms or tracking scenarios.
[8]	YOLOv3 object detection, Shi–Tomasi feature matching, BRIEF descriptor	Efficient tracking using YOLOv3 features and robust feature matching techniques.	Relies on accurate object detection and feature matching, and robustness may be affected in scenarios with object occlusions or cluttered backgrounds.
[44]	YOLOv3 object detection, particle filter	Hybrid approach for object tracking using YOLOv3 features and particle filtering.	Parameter tuning may be required, and computational cost will increase in scenarios with large numbers of objects.

Table 9. *Cont.*

Paper	Key Methods	Advantages	Limitations
[2]	SIFT, SURF, ORB, FAST, Shi–Tomasi feature descriptors, Kalman filter	Provides accurate tracking of pine tree branches using a combination of feature descriptors and Kalman filtering.	Requires careful selection and tuning of feature descriptors and may face challenges in complex branch motion or occlusion scenarios.
[33]	Mean shift, Kalman filter, geometric methods	Effective for tracking crane-mounted objects using mean shift and Kalman filtering.	There is a possibility of reduced robustness in outdoor environments with unpredictable factors such as wind or lighting changes.

Table 10. Summary of joint detection and tracking methods.

Paper	Key Methods	Advantages	Limitations
[40]	Use of depth information for tracking accuracy enhancement	Improved accuracy, especially in challenging environments	Depth-based models may require additional hardware or sensors, increasing complexity and cost
[56]	Combination of video object segmentation and pose estimation in a closed loop	Robust tracking performance, particularly in handling occlusions	Complexity of closed-loop system may increase computational overhead
[32]	Integration of CNN features for template matching and perspective transformation	Improved accuracy for handover tracking tasks	The method is specific to handover tracking tasks and may not generalise well to other tracking scenarios
[79]	R-CNN features for frame-by-frame association	Accurate frame-by-frame association for tracking objects	Computational complexity may increase with the use of R-CNN features, potentially limiting real-time performance
[75]	Implementation of Faster R-CNN for object detection and navigation algorithms	Accurate object detection and navigation for aircraft refuelling	Issues with large inference time and limited training data may hinder real-world applicability
[46]	Integration of Siamese networks with depth information for 3D object tracking	Capability to track objects in 3D space, useful for applications like drone surveillance	Depth information may not always be available or reliable, limiting the applicability of the method
[78]	Usage of Fast YOLO and MegaDepth for pedestrian tracking	Efficient pedestrian tracking with consideration of occlusions	Real-time performance may be impacted by the computational demands of YOLO and MegaDepth networks
[6]	Introduction of SA-FlowNet for energy-efficient optical flow estimation	Reduced energy consumption and improved performance for object detection and motion segmentation	Specific to event-based cameras, may not be directly applicable to conventional camera systems

6. Applications

The main reason for developing different methods and datasets is to ensure they are applied to solve real-world problems. Each real-world scenario and problem is different, and each has its constraints. In object tracking using computer vision, each problem, depending upon the environmental conditions such as indoor or outdoor applications, available computational resources, and the cost of the system, can become a constraint. This section outlines the different domains in which the object tracking methods are applied. Table 11 categorises different papers based on their applications studied in this review. Some of the papers in Table 11 overlap the application domains, such as multiple-object tracking (MOT) application methods that can be applied to detect multiple pedestrians for surveillance applications. The following subsections are grouped by their primary applications, and Figure 7 shows the structure of the categorisation of the application.

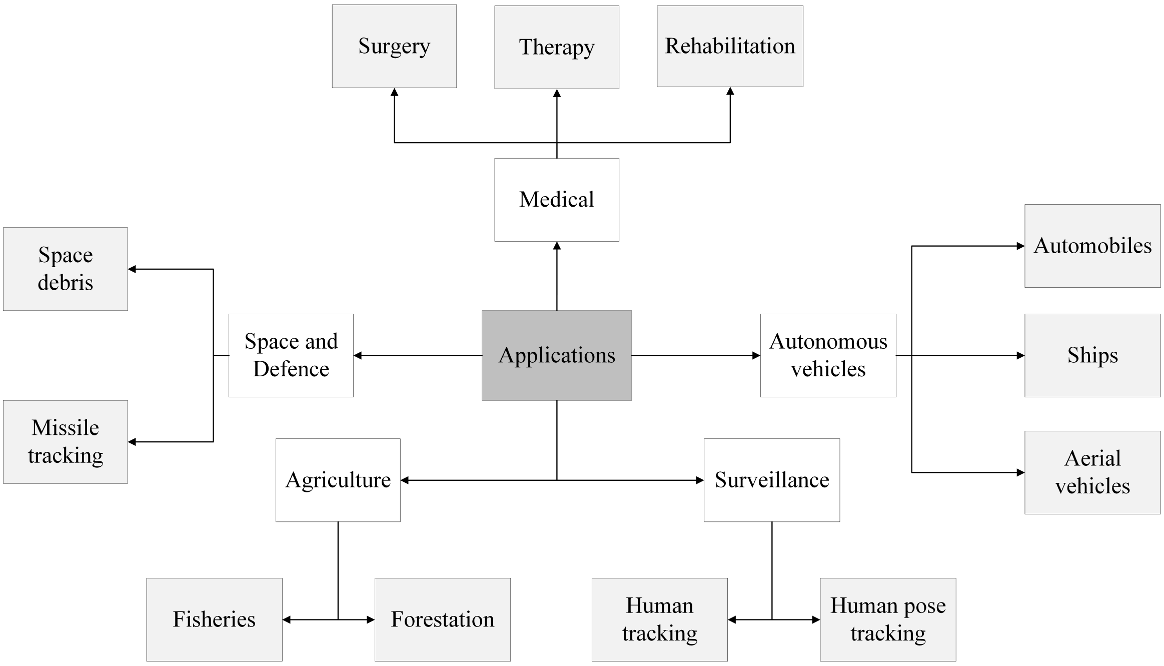

Figure 7. Structure of primary applications of object tracking.

6.1. Medical

Computer vision is preferred in medical applications where non-intrusive diagnoses are required. Non-intrusive diagnoses involve imaging and computational methods that elaborate results to help medical practitioners better diagnose patients. Richey et al. [12] used object tracking to track marked fiducial points for breast conservation surgery. Gionfrida et al. [13] used hand-pose tracking in the clinical setting to study hand kinematics using pose with a potential application in rehabilitation. Czajkowska et al. [14] developed processes for tracking a biopsy needle. Zarrabeitia et al. [16] applied their method for tracking 3D trajectories of droplets, which has a potential application in medicine for bloodletting events. Yang et al. [15] developed the 3D character recognition methods by tracking hand movement, which has an application in physical health examination and communicating using sign language. The results from object tracking provide insights into the operation procedure, providing greater details to the practitioners to make informed decisions. Thus, object tracking has a wider scope of application in numerous medical fields.

6.2. Autonomous Vehicles

An accurate object tracking solution is required in fields with a lot of dynamic movement, and autonomous driving is a primary example. Several types of research focus on detecting objects that could be observed in potential driving scenarios, thereby creating evaluation datasets of cars [35] and pedestrians [48] in the autonomous driving context. Different methods [1,3,5–10] have been proposed for applications in autonomous driving for detecting objects. Object tracking in autonomous driving involves detecting all moving objects, such as cars and pedestrians, from the sensor systems of the car. The datasets [35,49] collected for autonomous driving come with different attributes such as GPS, IMU, radar, and images. Yet, the scope of object detection for autonomous driving applications is limited to the few attributes in the dataset, such as radar, IMU, and images.

Similar to autonomous driving, water surface vehicle applications [28,29] face similar problem constraints. These attributes help detect objects and compute their trajectories in 3D space from the relative position of the vision system mounted on the vehicle. Knowing the movement of different objects around the autonomous vehicle, a future aim is to use this information for cruise control.

Autonomous aerial vehicles need to be aware of the dynamic environment around them. There are multiple applications in the field of aerial vehicles. Some applications track objects using sensors mounted on the aerial vehicle, while others track the flying aerial vehicle from the ground. Regarding tracking flying drones, Zheng et al. [42] applied their methods to develop a panoramic stereo to track rogue drones. Mdfaa et al. [46] developed a single-object tracker to be mounted on an aerial vehicle. Garcia and Younes [75] applied their method in automatically refuelling unmanned aerial vehicles using a drogue. Busch et al. [2] developed object tracking for the application of drones in agriculture. Wu et al. [39] applied target tracking on a quadcopter. The wide range of applications of unmanned aerial vehicles indicates that there are different niche cases to consider in aerial applications, which demand more datasets and methods. Figure 8 provides an overview of object tracking methods and their application to autonomous vehicles.

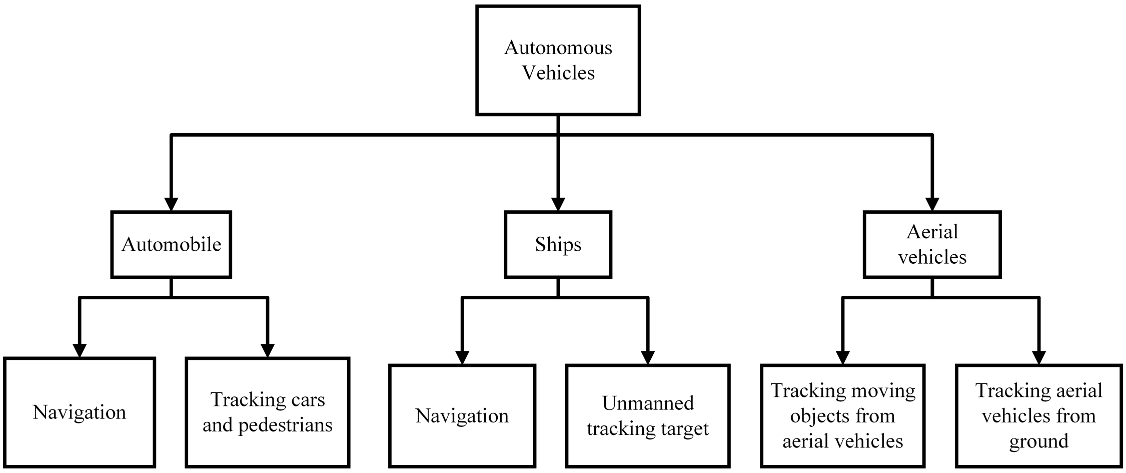
Autonomous Vehicles
Automobile
Ships
Aerial vehicles
Navigation
Tracking cars and pedestrians
Navigation
Unmanned tracking target
Tracking moving objects from aerial vehicles
Tracking aerial vehicles from ground

Figure 8. Overview of object tracking in autonomous vehicles.

6.3. Surveillance

Human movement tracking is one of the methods that is used in surveillance and sports. It is important to track the path of human movement in the scene and detect and track it over a longer period using multiple cameras. The application of human movement tracking also has to consider the problem of occlusion [56]. Yan et al. [32] tracked human skaters over multiple cameras to solve the object handover problem. Multiple methods [30,36,37,72,78,80,139,140] were developed for their applications in human pedestrian tracking. Along with human movement, pose estimation is another problem that fits well with action tracking. Different methods [13,77,81] were developed for pose estimation, which has applications in human action tracking and robotics [3,8]. The action tracking methods have different applications in surveillance, pose estimation, and robotics. Further development in these methods will have a wider scope for human–computer interaction problems.

6.4. Robotics

In robotic applications, a robot is an example of a dynamic system that interacts and manoeuvres itself autonomously within its environment. A robot needs to localise itself and the objects around it. Different sensors provide environmental input data to the robot, helping it accomplish its goals and operate safely without breaking itself, damaging nearby objects, or harming humans. Vision sensors on robots provide fine-grained data of the objects of interest, enabling the robots to perceive their surroundings. Busch et al. [2] used an object tracking method on aerial robots to investigate the movement of tree branches. Similarly, Wu et al. [39] also deployed a vision-based target-tracking method on aerial robots to track both ground and aerial objects. Therefore, using robots in object tracking applications is essential when the environment is too hostile or fast-paced for humans to operate, such as examining tree tops [2] or tracking aerial vehicles [39].

Persic et al.'s [3] method has an application in autonomous vehicles and robotics. Since their method focused on moving target tracking, it has a potential application in mobile or industrial robotics where there are different moving objects with higher uncertainty of object collisions. Similarly, Aladem and Rawashdeh [8] also developed their methods for safe navigation for mobile robots.

The field of robotics can benefit from object tracking as it allows the robots to perceive their environment while ensuring safe operation and preventing harm to humans. There is further potential for the application of object tracking methods in human–robot interaction, where the robots track human actions to work together to achieve a common goal.

6.5. Agriculture

Object tracking has potential in agriculture applications. Collecting information about plants and trees constantly swaying due to environmental factors such as wind and rain is important in agriculture. Busch et al. [2] applied object tracking to identify the swaying motion of a pine tree branch. Their motivation for developing tracking methods for tree branches was to allow researchers in the forestry industry to select trees for breeding, analyse genetics, and monitor plant diseases. The use of aerial vehicles with computer vision to examine tree branches outdates the use of ladders or manually climbing trees with a rope. In their application, they mounted their camera on an unmanned aerial vehicle with a manipulator arm to collect data on pine tree branches. Their proposed application has the potential to be used in the forestry industry to improve the efficiency of collecting tree data and thus maintain healthy forests.

Using an autonomous system in fishing is an important application in the fishing industry. Chuang et al. [11] developed methods for tracking live fish underwater. Tracking the movement of fish underwater is beneficial as it improves the efficiency of fishing operations. Knowing the positions of the fish, an autonomous system can deploy a trawl to catch fish. Furthermore, a computer vision system with object detection and tracking algorithms can lead to sustainable fishing techniques without damaging the ecosystem. Drawing inspiration from these applications, many more potential applications can be developed in agriculture using object tracking and computer vision.

6.6. Space and Defence

Object tracking has been applied to space and defence applications. Tracking space debris is an important application in the space industry. The damage caused by space debris could lead to the loss of space shuttles and human lives. Tracking space debris is essential for safer space flight, and thus, the space debris must be removed. Biondi et al. [76] developed their method to estimate the dynamic rotational state of space debris. Using computer vision to track space debris could lead to potential unmanned space missions to clear the space debris for safer space flights.

Defence applications are also using computer vision for object tracking tasks. Kwon et al. [4] developed a method for tracking and intercepting missiles with applications in defence technology. Their method aimed to solve the problem where both the target and the camera are moving. Thus, the method had potential applications in mobile robotics and unmanned aerial vehicles.

Garcia and Younes [75] developed methods for applications in autonomous aerial refuelling of aircraft. In the aerial refuelling task, a tanker aerial vehicle provides a refuelling probe to the drogue of the receiving aircraft and the refuelling is performed mid-air. In their research, their vision system, comprising a monocular camera on an unmanned aerial vehicle, used object detection to track the refuelling drogue in mid-flight and automatically refuel without human intervention. The refuelling task accounted for turbulence, and both the camera system and refuelling drogue were in motion.

The above-mentioned applications are reported based on computer simulation or experimental tests only. Further development will need to be conducted before they can be reliably deployed to real-world and critical applications.

Table 11. Categorisation of papers based on applications.

Application	Papers
Medical	[12–16]
Aerial vehicles	[2,39,42,46,75]
SOT	[33,40,46]
MOT	[11,44,76,79]
Human action tracking	[30,32,36,37,56,72,78,80,139,140]
Pose estimation	[13,77,81]
Autonomous driving	[1,3,5–10]
Aquatic surface vehicle	[28,29]
Robotics	[3,8]
Agriculture	[2,11]
Space/Defence	[4,76]

7. Discussion

Despite extensive research, object tracking using computer vision is still an active research area. The different solutions proposed to solve the tracking problem emerge from the constraints of the problem regarding resources and applications. The application of object tracking in different domains drives the development of the datasets, methods, and evaluation processes. Object tracking methods have several potential applications in different industries and research domains. The development of methods to address the problem constraints has evolved the approach from a set of image processing steps to using end-to-end deep learning models. While significant progress has been made in the last ten years in object tracking using computer vision, there is still room for improvement in addressing issues such as developing generalised procedures or frameworks, addressing lighting conditions, tracking fast-moving objects, and occlusion.

7.1. Methods

Despite the lack of a formal generalised procedure or framework for object tracking, the closest generalisation of procedure in the literature is first object detection and then object tracking. While this generalised tracking procedure is becoming more common, the dependency on multiple processing steps during the detection affects the overall robustness of the method. These image processing steps are developed iteratively, adjusting their parameters empirically or using statistical methods based on the results. When the algorithm receives the least error, it is ready for deployment. However, the method's accuracy is set based on the dataset upon which it was evaluated. Therefore, the two-step detection and tracking process can be combined into a single end-to-end deep learning framework.

Deep learning detection methods also incorporate an iterative process; however, since different architectures are already evaluated on a large and varied detection dataset with multiple classes, they become useful out of the box for detection. The object detection community is incrementally improving the detection method to be faster in real time [83]. Yet, these efficiency improvements come at higher computation costs. Classification and localisation can be performed simultaneously in real time with the detection architectures, such as YOLO [82] and subsequent versions. This dual functionality of deep learning methods to localise and classify in real time has led to a considerable leap in multiple-object tracking problems. However, in unique applications where the network was not trained to include a class of objects, the network needs to be trained either from scratch or using transfer learning [141] methods. Training a deep network requires computational resources; the image processing steps are preferred where such resources are unavailable. However, image processing methods in recent years have declined due to the availability of computational resources and pre-trained deep network architectures for detection. Apart from detection, very few methods use deep learning architecture for tracking. Tracking objects is still performed using estimation methods such as data association and Kalman filter. Using methods such as LSTM has helped create an end-to-end detection process in deep learning.

One of the important reasons for developing object tracking methods is for the machines to interact with their dynamic environment. This problem falls under the domain of ego-based problems where the sensors are mounted on machines such as robots or autonomous vehicles [5]. For ego-based problems, the objects are localised and tracked from the point of view of the machines. At the same time, the machines must also be able to localise themselves in the dynamic environment to function in a complex environment such as traffic or manufacturing. Therefore, there is a future scope for developing methods and procedures to adapt these vision systems on robots or autonomous vehicles to make an adaptive system in a dynamic environment.

Autonomous aerial vehicles such as unmanned drones are being used to track vehicles [39,46] and in the agricultural sector [2]. Since the range of vision sensors is limited, these drones often have to fly closer to the target, which can interfere with the object's natural state, such as vegetation, or distract humans in a crowded environment. Also, tracking drones from the ground station is an important application, and the distance from the ground station to the drone impacts the localisation and tracking of the drones [42]. Furthermore, in space applications for tracking debris, it is essential to track a fast-moving object at a faraway distance [76]. The range of measuring distance using a stereo camera depends upon the stereo camera parameters, such as the baseline between the two cameras. Zheng et al. [42] calculated the effective sensing range of the entire system of panoramic stereo reached 80 metres. Therefore, progress in increasing the current range of a state-of-the-art system will be significant progress in detecting faraway objects. Therefore, there is further scope for developing vision sensors and methods to track faraway objects.

7.2. Datasets

The applications of object tracking in diverse domains, from medical applications to autonomous navigation, have led to the creation of datasets catering to specific domains. The availability of the dataset ensures that all possible conditions of applications are considered. Since consistently testing on real-world applications can be expensive, the datasets can often simulate the real world to test the applications. In this case, the data can be manually collected from the real world or generated synthetically. However, if the methods are only evaluated on the dataset, it leaves further questions about their applicability in real-world dynamic situations.

In the iterative development process, real-world scenarios may often not be considered, and the method may be more accurate than the dataset. Still, it may not perform well in real-world applications. The most widely used odometry dataset, KITTI [35], consists of different sensor data types that help localise autonomous driving. Researchers combine different object detection datasets and develop methods to cater to real-world applications in a dynamic driving environment. The methods are developed on simulated datasets since some applications are particular, such as space applications [4,76]. For such applications, it is difficult to obtain real datasets and to experiment on such systems, which is an expensive process. While the ground truths often consist of object location, it will be helpful to have additional ground truths about tracking in different situations, such as variations in illumination, at high speed, and with occlusions.

While it is important to develop vision sensors and methods for detecting and tracking faraway objects, developing the dataset for training a deep learning network and evaluating methods is equally important. For applications such as missile tracking or missile intercepting systems [4], collecting data can be a cumbersome process. An alternative in this situation is to generate a synthetic dataset that imitates the real-world application. However, this synthetic dataset needs to be validated before the methods and equipment are developed for the applications. Therefore, researching approaches to create synthetic datasets and evaluating their validity for complex applications such as faraway object detection can be an important research focus.

Several problems in object tracking incorporate the use of multiple cameras [30,32]. A class of problem that uses multiple cameras is the handover problem [32] in object tracking, where the object disappears from the field of view of a camera and appears in the field of view of the next camera. A large-scale dataset can be generated using multiple cameras with ground truths that track objects over multiple cameras.

8. Limitations and Future Work

As computer vision systems are being incorporated into different engineering domains, these systems' ability to interact with the dynamic world relies on tracking objects in real time. New problems are encountered in object tracking as new applications are investigated. While developing a generalised method is often the researchers' goal, addressing all the issues encountered in object tracking in one method is challenging. Therefore, the scope for developing methods in object tracking using computer vision is wide, and several areas can be further investigated to address each problem.

The literature review in this paper raised significant questions about the future scope of research. The research questions, along with recommendations, are outlined as follows:

Q1 Could an end-to-end deep learning approach be developed to detect, classify, estimate the pose, and track the object in a 3D space?
Recommendation: There is significant development in object detection and classification methods such as YOLO [82], R-CNN [99], and Fast R-CNN [84]. Since methods such as YOLO [105] can localise, classify, segment objects, and estimate object pose, it will be worth investigating if the additional feature of tracking can be incorporated in this deep learning framework over video frame sequence. A sequence of video frames could act as an input to these networks, and post-processing steps such as estimating the tracks and stereo matching can be incorporated to detect and track objects. Methods such as SA-FlowNet [6] use a sequence of images for event-based cameras to track objects over time. Spatial attention networks [40] address the tracking using a sequence of video frames for depth estimation using RGB-D sensors. These methods can be further investigated for both calibrated and uncalibrated stereo cameras for depth estimation using a deep CNN.

Q2 Could the range of 3D tracking for faraway objects be extended?
Recommendation: Object tracking is being incorporated in applications of aerial vehicles where the long-range for depth estimation is important. The current state-of-the-art system uses a DS-2CD6984F-IHS/NFC HIKVISION camera and achieves a tracking range of 80 metres using panoramic stereo on a ground station for drone detection [42]. The range may be enhanced by using cameras with a higher zoom factor to construct a similar panoramic system. However, it will be worth investigating whether changing the camera parameters will significantly impact the results using the same methods or if the current state-of-the-art method will require modifications to track faraway objects.

Q3 How can object tracking be implemented on adaptive systems in a dynamic environment?
Recommendation: Robotics is an example of an adaptive system where the robots are subjected to a dynamic environment with moving objects. In this environment, robots need to know the position of the moving objects relative to their position and estimate their location with respect to their trajectory to avoid a collision. This problem may be addressed by developing methods in robots that monitor their environment in real time. The tracking process used in the present methods is performed as a post-processing method where the entire video sequence is available. This creates a limitation in a real-time system, where future information about the environment is unavailable. A predictive tracking algorithm will be helpful for the robot to avoid collision with moving objects. Therefore, for applications in adaptive systems, object tracking accompanied with tracking prediction will have a wider scope for robotics application.

Q4 What improvements are required in the current datasets for object tracking?

Recommendation: The datasets currently used for object tracking, as highlighted in Section 4, were developed for their respective applications. Datasets such as KITTI [35] are specific for autonomous driving, which consist of not only stereo camera video data but also IMU, GPS, and laser scan data. Other datasets such as pedestrian tracking [48,71] were developed for surveillance applications. These datasets are specific to their applications, and their limitation is that they are not generalised enough for a wider application in multiple scenarios.

To develop a dataset for 3D object tracking, stereo camera data of diverse objects similar to ImageNet [142] or MS COCO [143] with their ground truth will provide a common ground to evaluate the performance of object tracking methods. Along with a wider range of object classes, this dataset should also consider the 3D position of the object with respect to the camera. Therefore, an object-tracking dataset may consist of the following attributes:

- Stereo camera video sequence;
- Object classes in each video frame;
- Object location with its bounding-box coordinates in each video frame;
- Ground truth for object tracks for each video sequence;
- Ground truth for object's 3D position relative to the camera.

Generating such a dataset may require extensive effort. However, some data collection processes could be automated, such as using ultrasonic sensors and structured light sensors such as RGB-D [34] to collect ground truth for distance where possible, and the annotation for the dataset could be crowd-sourced using Amazon Mechanical Turk as used by Stanford's dataset [59]. Therefore, there is a scope for developing methods and processes for data collection and benchmarking the dataset for object tracking in computer vision.

Q5 Should hybrid sensors be used for object tracking, or should object tracking completely rely on computer vision?

Recommendation: Having more sensor data when possible is always beneficial. In the case of the KITTI [35] dataset, multiple sensor data are available to the user. Since the application is focused on autonomous driving, using a variety of sensors helps this type of adaptive system make better decisions based on its dynamic environment.

There are systems where having more sensors could create an additional payload on the mechanical system. Aerial drones and industrial robots are examples of adaptive systems where the additional payload can create functional problems. Having a single vision sensor on these devices, such as a stereo or RGB-D camera, could reduce their weight, thereby reducing the additional power requirement for operation. In these situations, relying on computer vision is beneficial. Thus, there is a requirement for better methods that address the diverse scenarios where these systems are deployed.

9. Conclusions

Object tracking is still an ongoing research area, and there is no standardised approach to solving it. Many approaches are developed using different hardware, datasets, and application methodologies. This paper conducted a synthesised review to group these methods according to the hardware and datasets used, the methodologies adopted, and the application areas for object tracking.

In particular, we divided the literature according to the type of cameras used, such as monocular, stereo, depth, and hybrid sensors. The datasets were grouped according to their focused research applications, such as autonomous driving, single-object tracking, multiple-object tracking, and other miscellaneous applications. We also classified the existing literature according to the methodologies used. The application of object tracking is also grouped based on their area of focus, such as medical, autonomous vehicles, single-object tracking, multiple-object tracking, surveillance, robotics, agriculture, space, and defence.

The contribution of this review is the systemic categorisation of different aspects of the object tracking problem. This review highlighted the trends and interest in object tracking research over the last ten years, thereby contributing to the detailed literature review on hardware, datasets, approaches, and applications. Furthermore, tabulated information summarised different tools and methods to develop an object tracking system. A taxonomy was provided for the methods, while identifying the advantages and limitations of different approaches and methods. The review also recommended when the equipment, datasets, and methods can be used. Also, from the review of the literature, different research questions were identified with a recommended approach to address these questions.

Author Contributions: Conceptualisation, P.K. and G.F.; investigation, P.K.; data curation, P.K.; writing—original draft preparation, P.K.; writing—review and editing, P.K., G.F. and J.J.Z.; supervision, G.F. and J.J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analysed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

APCE	Average peak-to-correlation energy
CNN	Convolutional Neural Networks
DTD	Describable Textures Dataset
EDM	Electronic distance meter
FIR	Finite impulse response
FOV	Field of view
GUI	Graphical User Interface
HCI	Heidelberg Collaboratory for Image Processing
HoG	Histogram of Oriented Gradients
IMU	Inertial measurement unit
JDT	Joint detection and tracking
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
LiDAR	Light Detection and Ranging
MEMS	Micro-Electromechanical System
MOT	Multiple-object tracking
MVSEC	Multivehicle Stereo Event Camera
NUC	Next Unit Computing
R-CNN	Regions with CNN features
RBOT	Region-based object tracking
RPN	Risk Priority Number
SAD	Sum of absolute difference
SMDWT	Symmetric mask-based discrete wavelet transform
SNN	Spiking Neural Networks
SOT	Single-object tracking
SSD	Single-Shot Multibox Detector
TBD	Tracking by detection
VI	Visual Inertial
VOT	Visual object tracking
YOLO	You Only Look Once

References

1. Li, X.; Shen, Y.; Lu, J.; Jiang, Q.; Xie, O.; Yang, Y.; Zhu, Q. DyStSLAM: An efficient stereo vision SLAM system in dynamic environment. *Meas. Sci. Technol.* **2023**, *34*, 205105. [CrossRef]
2. Busch, C.; Stol, K.; van der Mark, W. Dynamic tree branch tracking for aerial canopy sampling using stereo vision. *Comput. Electron. Agric.* **2021**, *182*, 106007. [CrossRef]
3. Persic, J.; Petrovic, L.; Markovic, I.; Petrovic, I. Spatiotemporal Multisensor Calibration via Gaussian Processes Moving Target Tracking. *IEEE Trans. Robot.* **2021**, *37*, 1401–1415. [CrossRef]
4. Kwon, J.H.; Song, E.H.; Ha, I.J. 6 Degree-of-Freedom Motion Estimation of a Moving Target using Monocular Image Sequences. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 2818–2827. [CrossRef]
5. Feng, S.; Li, X.; Xia, C.; Liao, J.; Zhou, Y.; Li, S.; Hua, X. VIMOT: A Tightly Coupled Estimator for Stereo Visual-Inertial Navigation and Multiobject Tracking. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 3291011. [CrossRef]
6. Yang, F.; Su, L.; Zhao, J.; Chen, X.; Wang, X.; Jiang, N.; Hu, Q. SA-FlowNet: Event-based self-attention optical flow estimation with spiking-analogue neural networks. *IET Comput. Vision* **2023**, *17*, 925–935. [CrossRef]
7. Shen, Y.; Liu, Y.; Tian, Y.; Liu, Z.; Wang, F. A New Parallel Intelligence Based Light Field Dataset for Depth Refinement and Scene Flow Estimation. *Sensors* **2022**, *22*, 9483. [CrossRef] [PubMed]
8. Aladem, M.; Rawashdeh, S. A Combined Vision-Based Multiple Object Tracking and Visual Odometry System. *IEEE Sens. J.* **2019**, *19*, 11714–11720. [CrossRef]
9. Deepambika, V.; Rahman, M.A. Illumination invariant motion detection and tracking using SMDWT and a dense disparity-variance method. *J. Sens.* **2018**, *2018*, 1354316. [CrossRef]
10. Ćesić, J.; Marković, I.; Cvišić, I.; Petrović, I. Radar and stereo vision fusion for multitarget tracking on the special Euclidean group. *Robot. Auton. Syst.* **2016**, *83*, 338–348. [CrossRef]
11. Chuang, M.C.; Hwang, J.N.; Williams, K.; Towler, R. Tracking live fish from low-contrast and low-frame-rate stereo videos. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 167–179. [CrossRef]
12. Richey, W.; Heiselman, J.; Ringel, M.; Meszoely, I.; Miga, M. Soft Tissue Monitoring of the Surgical Field: Detection and Tracking of Breast Surface Deformations. *IEEE Trans. Biomed. Eng.* **2023**, *70*, 2002–2012. [CrossRef]
13. Gionfrida, L.; Rusli, W.; Bharath, A.; Kedgley, A. Validation of two-dimensional video-based inference of finger kinematics with pose estimation. *PLoS ONE* **2022**, *17*, e0276799. [CrossRef]
14. Czajkowska, J.; Pyciński, B.; Juszczyk, J.; Pietka, E. Biopsy needle tracking technique in US images. *Comput. Med. Imaging Graph.* **2018**, *65*, 93–101. [CrossRef]
15. Yang, J.; Xu, R.; Ding, Z.; Lv, H. 3D character recognition using binocular camera for medical assist. *Neurocomputing* **2017**, *220*, 17–22. [CrossRef]
16. Zarrabeitia, L.; Qureshi, F.; Aruliah, D. Stereo reconstruction of droplet flight trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 847–861. [CrossRef] [PubMed]
17. Li, X.; Hu, W.; Shen, C.; Zhang, Z.; Dick, A.; Van Den Hengel, A. A survey of appearance models in visual object tracking. *ACM Trans. Intell. Syst. Technol.* **2013**, *4*, 1–48 [CrossRef]
18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
19. Kumar, A.; Walia, G.S.; Sharma, K. Recent trends in multicue based visual tracking: A review. *Expert Syst. Appl.* **2020**, *162*, 113711. [CrossRef]
20. Park, Y.; Dang, L.M.; Lee, S.; Han, D.; Moon, H. Multiple object tracking in deep learning approaches: A survey. *Electronics* **2021**, *10*, 2406. [CrossRef]
21. Kalake, L.; Wan, W.; Hou, L. Analysis Based on Recent Deep Learning Approaches Applied in Real-Time Multi-Object Tracking: A Review. *IEEE Access* **2021**, *9*, 32650–32671. [CrossRef]
22. Mandal, M.; Vipparthi, S.K. An Empirical Review of Deep Learning Frameworks for Change Detection: Model Design, Experimental Frameworks, Challenges and Research Needs. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6101–6122. [CrossRef]
23. Guo, S.; Wang, S.; Yang, Z.; Wang, L.; Zhang, H.; Guo, P.; Gao, Y.; Guo, J. A Review of Deep Learning-Based Visual Multi-Object Tracking Algorithms for Autonomous Driving. *Appl. Sci.* **2022**, *12*, 10741. [CrossRef]
24. Dai, Y.; Hu, Z.; Zhang, S.; Liu, L. A survey of detection-based video multi-object tracking. *Displays* **2022**, *75*, 102317. [CrossRef]
25. Rakai, L.; Song, H.; Sun, S.; Zhang, W.; Yang, Y. Data association in multiple object tracking: A survey of recent techniques. *Expert Syst. Appl.* **2022**, *192*, 116300. [CrossRef]
26. Liu, C.; Chen, X.F.; Bo, C.J.; Wang, D. Long-term Visual Tracking: Review and Experimental Comparison. *Mach. Intell. Res.* **2022**, *19*, 512–530. [CrossRef]
27. Rocha, R.d.L.; de Figueiredo, F.A.P. Beyond Land: A Review of Benchmarking Datasets, Algorithms, and Metrics for Visual-Based Ship Tracking. *Electronics* **2023**, *12*, 2789. [CrossRef]
28. Kriechbaumer, T.; Blackburn, K.; Breckon, T.; Hamilton, O.; Casado, M. Quantitative evaluation of stereo visual odometry for autonomous vessel localisation in inland waterway sensing applications. *Sensors* **2015**, *15*, 31869–31887. [CrossRef] [PubMed]
29. Sinisterra, A.; Dhanak, M.; Ellenrieder, K.V. Stereovision-based target tracking system for USV operations. *Ocean Eng.* **2017**, *133*, 197–214. [CrossRef]

30. Gennaro, T.D.; Waldmann, J. Sensor Fusion with Asynchronous Decentralized Processing for 3D Target Tracking with a Wireless Camera Network. *Sensors* **2023**, *23*, 1194. [CrossRef]
31. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004.
32. Yan, M.; Zhao, Y.; Liu, M.; Kong, L.; Dong, L. High-speed moving target tracking of multi-camera system with overlapped field of view. *Signal Image Video Process* **2021**, *15*, 1369–1377. [CrossRef]
33. Huang, J.; Xu, W.; Zhao, W.; Yuan, H. An improved method for swing measurement based on monocular vision to the payload of overhead crane. *Trans. Inst. Meas. Control* **2022**, *44*, 50–59. [CrossRef]
34. Zhang, Z. Microsoft Kinect Sensor and Its Effect. *IEEE MultiMedia* **2012**, *19*, 4–10. [CrossRef]
35. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
36. García, J.; Gardel, A.; Bravo, I.; Lázaro, J.; Martínez, M. Tracking people motion based on extended condensation algorithm. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2013**, *43*, 606–618. [CrossRef]
37. Hu, M.; Liu, Z.; Zhang, J.; Zhang, G. Robust object tracking via multi-cue fusion. *Signal Process* **2017**, *139*, 1339–1351. [CrossRef]
38. Bouguet, J.Y. Camera Calibration Toolbox for Matlab. 2022. Available online: https://data.caltech.edu/records/jx9cx-fdh55 (27 February 2024).
39. Wu, S.; Li, R.; Shi, Y.; Liu, Q. Vision-Based Target Detection and Tracking System for a Quadcopter. *IEEE Access* **2021**, *9*, 62043–62054. [CrossRef]
40. Rasoulidanesh, M.; Yadav, S.; Herath, S.; Vaghei, Y.; Payandeh, S. Deep attention models for human tracking using RGBD. *Sensors* **2019**, *19*, 750. [CrossRef] [PubMed]
41. Song, S.; Xiao, J. Tracking Revisited using RGBD Camera: Unified Benchmark and Baselines. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013. [CrossRef]
42. Zheng, Y.; Zheng, C.; Zhang, X.; Chen, F.; Chen, Z.; Zhao, S. Detection, Localization, and Tracking of Multiple MAVs with Panoramic Stereo Camera Networks. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 1226–1243. [CrossRef]
43. Ram, S. Fusion of Inverse Synthetic Aperture Radar and Camera Images for Automotive Target Tracking. *IEEE J. Sel. Top. Signal Process* **2023**, *17*, 431–444. [CrossRef]
44. Ngoc, L.; Tin, N.; Tuan, L. A New framework of moving object tracking based on object detection-tracking with removal of moving features. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 35–46. [CrossRef]
45. Sigal, L.; Balan, A.O.; Black, M.J.; Balan, A.O.; Black, M.J.; Black, M.J. HUMANEVA: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion. *Int. J. Comput. Vis.* **2010**, *87*, 4–27. [CrossRef]
46. Mdfaa, M.A.; Kulathunga, G.; Klimchik, A. 3D-SiamMask: Vision-Based Multi-Rotor Aerial-Vehicle Tracking for a Moving Object. *Remote Sens.* **2022**, *14*, 5756. [CrossRef]
47. Karangwa, J.; Liu, J.; Zeng, Z. Vehicle Detection for Autonomous Driving: A Review of Algorithms and Datasets. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 11568–11594. [CrossRef]
48. Flohr, F.; Gavrila, D. PedCut: An iterative framework for pedestrian segmentation combining shape models and multiple data cues. In Proceedings of the British Machine Vision Conference (BMVC), Bristol, UK, 9–13 September 2013.
49. Zhu, A.Z.; Thakur, D.; Ozaslan, T.; Pfrommer, B.; Kumar, V.; Daniilidis, K. The Multi Vehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2800793. [CrossRef]
50. Nikolic, J.; Rehder, J.; Burri, M.; Gohl, P.; Leutenegger, S.; Furgale, P.T.; Siegwart, R. A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 431–437. [CrossRef]
51. Honauer, K.; Johannsen, O.; Kondermann, D.; Goldluecke, B. A dataset and evaluation methodology for depth estimation on 4D light fields. In *Computer Vision–ACCV 2016, Proceedings of the 13th Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016*; Revised Selected Papers, Part III 13; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10113, pp. 19–34. [CrossRef]
52. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Kämäräinen, J.K.; Chang, H.J.; Danelljan, M.; Čehovin Zajc, L.; Lukežič, A.; et al. The Tenth Visual Object Tracking VOT2022 Challenge Results. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2023; Volume 13808, pp. 431–460. [CrossRef]
53. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
54. Pauwels, K.; Rubio, L.; Díaz, J.; Ros, E. Real-time Model-based Rigid Object Pose Estimation and Tracking Combining Dense and Sparse Visual Cues. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013. [CrossRef]
55. Kasper, A.; Xue, Z.; Dillmann, R. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *Int. J. Robot. Res.* **2012**, *31*, 927–934. [CrossRef]
56. Zhong, L.; Zhang, Y.; Zhao, H.; Chang, A.; Xiang, W.; Zhang, S.; Zhang, L. Seeing through the Occluders: Robust Monocular 6-DOF Object Pose Tracking via Model-Guided Video Object Segmentation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5159–5166. [CrossRef]

57. Krull, A.; Michel, F.; Brachmann, E.; Gumhold, S.; Ihrke, S.; Rother, C. 6-DOF Model Based Tracking via Object Coordinate Regression. In Proceedings of the Computer Vision—ACCV, Singapore, 1–5 November 2014; Springer International Publishing: Berlin/Heidelberg, Germany, 2015. [CrossRef]
58. Hwang, J.; Kim, J.; Chi, S.; Seo, J. Development of training image database using web crawling for vision-based site monitoring. *Autom. Constr.* **2022**, *135*, 104141. [CrossRef]
59. Krause, J.; Stark, M.; Deng, J.; Li, F.-F. 3D Object Representations for Fine-Grained Categorization. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2–8 December 2013. [CrossRef]
60. Cimpoi, M.; Maji, S.; Kokkinosécole, I.; Mohamed, S.; Vedaldi, A. Describing Textures in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014. [CrossRef]
61. Zauner, C. Implementation and Benchmarking of Perceptual Image Hash Functions. 2010. Available online: http://www.phash.org/docs/pubs/thesis_zauner.pdf (accessed on 27 February 2024).
62. Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Fernández, G.; Vojí, T.; Häger, G.; Nebehay, G.; Pflugfelder, R.; Gupta, A.; et al. The Visual Object Tracking VOT2015 challenge results 2015 IEEE International Conference on Computer Vision Workshop 2015 IEEE International Conference on Computer Vision Workshop. *Chin. Acad. Sci.* **2015**, *32*, 79. [CrossRef]
63. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Čehovin, L.; Vojír, T.; Häger, G.; Lukežič, A.; Fernández, G.; et al. The visual object tracking VOT2016 challenge results. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part II*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2016; Volume 9914, pp. 777–823. [CrossRef]
64. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Čehovin Zajc, L.; Vojír, T.; Bhat, G.; Lukežič, A.; Eldesokey, A.; et al. The sixth visual object tracking VOT2018 challenge results. In Proceedings of the Computer Vision—ECCV 2018 Workshops, Munich, Germany, 8–14 September 2018; Lecture Notes in Computer Science; Volume 11129, pp. 3–53. [CrossRef]
65. Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Pflugfelder, R.; Kämäräinen, J.K.; Zajc, L.C.; Drbohlav, O.; Lukezic, A.; Berg, A.; et al. The seventh visual object tracking VOT2019 challenge results. In Proceedings of the 2019 International Conference on Computer Vision Workshop, ICCVW 2019, Seoul, Republic of Korea, 27–28 October 2019; pp. 2206–2241. [CrossRef]
66. Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.; Roth, S.; Schindler, K.; Leal-Taixé, L.; Taixé, T. MOT20: A Benchmark for Multi Object Tracking in Crowded Scenes. *arXiv* **2020**, arXiv:2003.09003.
67. Leal-Taixé, L.; Taixé, T.; Milan, A.; Reid, I.; Roth, S.; Schindler, K. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv* **2015**, arXiv:1504.01942.
68. Milan, A.; Leal-Taixé, L.; Taixé, T.; Reid, I.; Roth, S.; Schindler, K. MOT16: A Benchmark for Multi-Object Tracking. *arXiv* **2016**, arXiv:1603.00831.
69. Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.; Roth, S.; Schindler, K.; Leal-Taixé, L.; Taixé, T. CVPR19 Tracking and Detection Challenge: How crowded can it get? *arXiv* **2019**, arXiv:1906.04567.
70. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T.K. Multiple object tracking: A literature review. *Artif. Intell.* **2021**, *293*, 103448. [CrossRef]
71. Dollar, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian detection: A benchmark. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 304–311. [CrossRef]
72. Wang, Z.; Yoon, S.; Park, D. Online adaptive multiple pedestrian tracking in monocular surveillance video. *Neural Comput. Appl.* **2017**, *28*, 127–141. [CrossRef]
73. Ferryman, J.; Ellis, A.L. Performance evaluation of crowd image analysis using the PETS2009 dataset. *Pattern Recognit. Lett.* **2014**, *44*, 3–15 [CrossRef]
74. Tjaden, H.; Schwanecke, U.; Schömer, E.; Cremers, D. A Region-Based Gauss-Newton Approach to Real-Time Monocular Multiple Object Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1797–1812. [CrossRef]
75. Garcia, J.; Younes, A. Real-Time Navigation for Drogue-Type Autonomous Aerial Refueling Using Vision-Based Deep Learning Detection. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 2225–2246. [CrossRef]
76. Biondi, G.; Mauro, S.; Pastorelli, S.; Sorli, M. Fault-tolerant feature-based estimation of space debris rotational motion during active removal missions. *Acta Astronaut.* **2018**, *146*, 332–338. [CrossRef]
77. Wang, Q.; Zhou, J.; Li, Z.; Sun, X.; Yu, Q. Robust and Accurate Monocular Pose Tracking for Large Pose Shift. *IEEE Trans. Ind. Electron.* **2023**, *70*, 8163–8173. [CrossRef]
78. Xiao, P.; Yan, F.; Chi, J.; Wang, Z. Real-Time 3D Pedestrian Tracking with Monocular Camera. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 7437289. [CrossRef]
79. Meneses, M.; Matos, L.; Prado, B.; Carvalho, A.; Macedo, H. SmartSORT: An MLP-based method for tracking multiple objects in real-time. *J. Real-Time Image Process.* **2021**, *18*, 913–921. [CrossRef]
80. Zhang, Y.; Sheng, H.; Wu, Y.; Wang, S.; Ke, W.; Xiong, Z. Multiplex Labeling Graph for Near-Online Tracking in Crowded Scenes. *IEEE Internet Things J.* **2020**, *7*, 7892–7902. [CrossRef]
81. Du, M.; Nan, X.; Guan, L. Monocular human motion tracking by using de-mc particle filter. *IEEE Trans. Image Process.* **2013**, *22*, 3852–3865. [CrossRef]

82. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
83. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
84. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
85. Soille, P. Erosion and Dilation. *Morphol. Image Anal.* **2004**, *2*, 63–103. [CrossRef]
86. Ma, J.; Jiang, X.; Fan, A.; Jiang, J.; Yan, J.; Lepetit, V.; Yan, J.; Jiang, X. Image Matching from Handcrafted to Deep Features: A Survey. *Int. J. Comput. Vis.* **2021**, *129*, 23–79. [CrossRef]
87. Geiger, A.; Ziegler, J.; Stiller, C. StereoScan: Dense 3d reconstruction in real-time. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden, Germany, 5–9 June 2011; pp. 963–968. [CrossRef]
88. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Fluids Eng. Trans. ASME* **1960**, *82*, 35–45. [CrossRef]
89. Steinbrücker, F.; Sturm, J.; Cremers, D. Real-time visual odometry from dense RGB-D images. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 719–722. [CrossRef]
90. Jenkins, M.; Barrie, P.; Buggy, T.; Morison, G. Extended fast compressive tracking with weighted multi-frame template matching for fast motion tracking. *Pattern Recognit. Lett.* **2016**, *69*, 82–87. [CrossRef]
91. Itseez. Open Source Computer Vision Library. 2015. Available online: https://github.com/itseez/opencv (accessed on 27 February 2024).
92. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern* **1979**, *9*, 62–66. [CrossRef]
93. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]
94. Hsia, C.H.; Guo, J.M.; Chiang, J.S. Improved Low-Complexity Algorithm for 2-D Integer Lifting-Based Discrete Wavelet Transform Using Symmetric Mask-Based Scheme. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 1202–1208. [CrossRef]
95. Kanade, T.; Kano, H.; Kimura, S.; Yoshida, A.; Oda, K. Development of a video-rate stereo machine. In Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, Pittsburgh, PA, USA, 5–9 August 1995; Volume 3, pp. 95–100. [CrossRef]
96. Szwarc, P.; Kawa, J.; Pietka, E. White matter segmentation from MR images in subjects with brain tumours. In *Information Technologies in Biomedicine, Proceedings of the Third International Conference, ITIB 2012, Gliwice, Poland, 11–13 June 2012*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7339 LNBI, pp. 36–46. [CrossRef]
97. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. [CrossRef]
98. Alcantarilla, P.F.; Bartoli, A.; Davison, A.J. KAZE features. In Proceedings of the Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Part VI 12; Springer: Berlin/Heidelberg, Germany, 2012; pp. 214–227. [CrossRef]
99. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
100. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. [CrossRef]
101. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
102. Leonard, J.; Durrant-Whyte, H. Mobile robot localization by tracking geometric beacons. *IEEE Trans. Robot. Autom.* **1991**, *7*, 376–382. [CrossRef]
103. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In *Computer Vision–ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part I 14; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2016; Volume 9905, pp. 21–37. [CrossRef]
104. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [CrossRef]
105. Jocher, G. YOLOv5 by Ultralytics. 2020. Available online: https://doi.org/10.5281/zenodo.3908559 (accessed on 1 October 2023).
106. Shafiee, M.J.; Chywl, B.; Li, F.; Wong, A. Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video. *arXiv* **2017**, arXiv:1709.05943. [CrossRef]
107. Li, Z.; Snavely, N. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
108. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H. Fast Online Object Tracking and Segmentation: A Unifying Approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
109. Ranftl, R.; Lasinger, K.; Hafner, D.; Schindler, K.; Koltun, V. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1623–1637. [CrossRef] [PubMed]

110. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
111. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 172–186. [CrossRef]
112. Forney, G. The viterbi algorithm. *Proc. IEEE* **1973**, *61*, 268–278. [CrossRef]
113. Li, P.; Zhao, H.; Liu, P.; Cao, F. RTM3D: Real-Time Monocular 3D Detection from Object Keypoints for Autonomous Driving. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12348, pp. 644–660. [CrossRef]
114. Sun, Y.; Zheng, L.; Yang, Y.; Tian, Q.; Wang, S. Beyond Part Models: Person Retrieval with Refined Part Pooling (and A Strong Convolutional Baseline). In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 11208, pp. 501–518. [CrossRef]
115. Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; Tian, Q. Scalable Person Re-identification: A Benchmark. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1116–1124. [CrossRef]
116. Brunelli, R.; Poggiot, T. Template matching: Matched spatial filters and beyond. *Pattern Recognit.* **1997**, *30*, 751–768. [CrossRef]
117. Wu, Y.; Lim, J.; Yang, M.H. Online Object Tracking: A Benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013. [CrossRef]
118. Munkres, J. Algorithms for the Assignment and Transportation Problems. *J. Soc. Ind. Appl. Math.* **1957**, *5*, 32–38. [CrossRef]
119. Horn, B.K.; Schunck, B.G. Determining optical flow. *Artif. Intell.* **1981** , *17* 185–203. [CrossRef]
120. Hough, P.V. Method and Means for Recognizing Complex Patterns. U.S. Patent 3,069,654, 18 December 1962.
121. Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence—Volume 2, San Francisco, CA, USA, 24–28 August 1981; IJCAI'81, pp. 674–679.
122. Tomasi, C.; Kanade, T. Detection and tracking of point. *Int. J. Comput. Vis.* **1991**, *9*, 3.
123. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 15–17 September 1988; Volume 15, pp. 10–5244.
124. Li, Q.; Li, R.; Ji, K.; Dai, W. Kalman Filter and Its Application. In Proceedings of the 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), Tianjin, China, 1–3 November 2015; pp. 74–77. [CrossRef]
125. Witkin, A.P. Scale-Space Filtering. In *Readings in Computer Vision*; Morgan Kaufmann: Burlington, MA, USA, 1987; Volume 2, pp. 329–332. [CrossRef]
126. Persoon, E.; Fu, K.S. Shape Discrimination Using Fourier Descriptors. *IEEE Trans. Syst. Man Cybern.* **1977**, *7*, 170–179. [CrossRef]
127. Shi, J.; Tomasi. Good features to track. In Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600. [CrossRef]
128. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary Robust Independent Elementary Features. In *Computer Vision—ECCV 2010, Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September* 2010; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; pp. 778–792. [CrossRef]
129. Mozhdehi, R.J.; Medeiros, H. Deep convolutional particle filter for visual tracking. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3650–3654. [CrossRef]
130. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
131. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded Up Robust Features. In Computer Vision–ECCV 2006, Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3951, pp. 404–417. [CrossRef]
132. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571. [CrossRef]
133. Rosten, E.; Porter, R.; Drummond, T. Faster and Better: A Machine Learning Approach to Corner Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 105–119. [CrossRef]
134. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
135. Nam, H.; Han, B. Learning Multi-domain Convolutional Neural Networks for Visual Tracking. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302. [CrossRef]
136. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist.* **2005**, *52*, 7–21. [CrossRef]
137. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 6–11 July 2015; Volume 2.
138. Gerstner, W.; Kistler, W.M. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*; Cambridge University Press: Cambridge, UK, 2002. [CrossRef]
139. Varga, D.; Szirányi, T.; Kiss, A.; Spórás, L.; Havasi, L. A Multi-View Pedestrian Tracking Method in an Uncalibrated Camera Network. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 184–191. [CrossRef]
140. Koppanyi, Z.; Toth, C.; Soltesz, T. Deriving Pedestrian Positions from Uncalibrated Videos. In Proceedings of the ASPRS Imaging & Geospatial Technology Forum (IGTF), Tampa, FL, USA, 12–16 March 2017; pp. 4–8.

141. Hosna, A.; Merry, E.; Gyalmo, J.; Alom, Z.; Aung, Z.; Azim, M.A. Transfer learning: A friendly introduction. *J. Big Data* **2022**, *9*, 102. [CrossRef]
142. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
143. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.