Real-Time Object Tracking Using Event-Based Imaging in Cable-Driven Robotic Systems

Haris Karic[1], Risyadayana Rinaida Binti Razmi[2], Dirk Mohr[3], Ulrich Zwiers[4], Arockia Selvakumar Arockiadoss[5], and Daniel Schilberg[1]

[1] Institute of Robotics and Mechatronics, University of Applied Sciences Bochum, Bochum, Germany
[2] Faculty of Mechatronics and Mechanical Engineering Bochum, University of Applied Sciences Bochum, Bochum, Germany
[3] Department of Control Engineering and Machine Vision, University of Applied Sciences Bochum, Bochum, Germany
[4] Department of Engineering Mechanics, University of Applied Sciences Bochum, Bochum, Germany
[5] School of Mechanical Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India

Abstract. Event-based image processing represents an emerging research field within computer vision and is gaining increasing relevance in modern imaging technologies. In contrast to conventional frame-based approaches, neuromorphic image sensors record only positive and negative changes in light intensity at the pixel level. This enables data acquisition with temporal resolution in the microsecond range, thereby opening up novel approaches for the real-time capture and analysis of highly dynamic scenes. This Paper investigates event-based imaging (EBI) as a novel sensing technology for high-precision object tracking in a high-speed cable-driven robot. First, a fundamental overview of the operating principles and characteristic properties of event-based image sensors is provided, and their potential for capturing highly dynamic scenes is discussed. Subsequently, a systematic comparison with conventional frame-based image processing methods is conducted, particularly with respect to temporal resolution, latency, and robustness under real operating conditions. Building on these considerations, the development and implementation of an event-based sensor system for real-time position estimation in the cable-driven robot are presented. The performance of the system is analyzed and evaluated through experimental investigations, with a focus on rapid object detection and the precise capture of dynamic motion patterns.

Keywords: Event-Based Imaging; Neuromorphic Vision; Real-Time Object Tracking; Motion Detection; Event Stream Processing; Dynamic Vision Sensor; High-Speed Imaging; Machine Vision.

1 Introduction

In conventional camera systems used for motion or event monitoring, a large portion of the captured image data remains unchanged from frame to frame. Although the depiction of motion has continuously improved since the early days of imaging, when movement was represented as a sequence of still frames per second, the underlying capture technology has

largely remained unchanged and lacks significant innovation. Between individual frames, the camera is "blind," resulting in a loss of valuable information about moving objects. Furthermore, the captured images contain no temporally resolved data regarding these movements. Instead, static background elements are repeatedly recorded, which are irrelevant for motion analysis and merely contribute to unnecessarily large data volumes. Event-based image processing addresses these challenges by capturing only changes in light intensity at the pixel level. This ignores motionless areas of the image while detecting motion with extremely high temporal resolution in the microsecond range. This not only enables a drastic reduction in the amount of data to be processed, but also a higher sampling rate without restricting the field of view. In contrast to conventional image-based methods, which continuously capture frames, event-based imaging is based on the biological visual system. It captures only relevant changes, known as "events," as soon as they occur and transmits them in real time. The underlying concepts of event-based image processing were developed in the early 1990s to mimic the functionality of the human retina. The integration of neural networks into this technology opens a significant field of application for artificial intelligence. AI-supported implementations enable the generation and evaluation of events at the pixel level and, at the same time, create the basis for further applications in the field of machine learning. This results in more resource-efficient and, in particular, much better-suited processing of visual data for dynamic scenes. This paper provides a comprehensive overview of event-based imaging (EBI) technology and the algorithms used for EBI applications, comparing them with established conventional imaging methods. In addition, insight is provided into the research conducted at Bochum University of Applied Sciences, which investigated EBI technology for high-precision position detection in a high-speed cable-driven robot. The focus here was on developing a tracking program and evaluating the technology's performance under real operating conditions, particularly in rapid object recognition and the precise detection of dynamic processes. Building on this application, a real-time analysis was also implemented, the results of which were validated and referenced. The findings made a significant contribution to optimizing robot control, increasing the efficiency of motion tracking, and promoting the integration of intelligent sensor systems into industrial automation processes.

2 State of the Art

At the beginning of the paper, an overview of the scientific foundations of image processing is provided. The focus is placed on the key concepts and principles of image processing that are particularly relevant to this study. The theoretical fundamentals are explained in detail to establish a solid understanding and provide a strong foundation for further analysis. Subsequently, event-based imaging technology is introduced, including an explanation of its underlying mechanisms and a comparison with conventional frame-based imaging methods.

2.1 The Functions of Human Vision

Human vision is a fatigue-free, largely unconscious process of perception that enables the reception, processing, and interpretation of visual stimuli. Light stimuli are captured by specialized photoreceptors in the retina of the eye, converted into electrical signals, and transmitted to the brain via the optic nerve [1,2]. There, the information is integrated and interpreted into a coherent visual impression. Visual perception is therefore not based solely on the momentary reception of optical stimuli but is significantly supplemented by stored experiences and learned patterns.[3] This interplay of sensory perception and cognitive processing forms the basis for human understanding of visual environments and serves as a biological model for technical image processing systems. [1,2,4]

2.2 Technical Image Processing

Technical image processing is a key area of modern automation and information technology. It deals with the computer-aided capture, processing, and analysis of digital images to convert visual information into usable data. It is defined by the German Engineering Federation (VDMA) as follows:

"Technical (industrial) image processing is the technology of artificial vision. Cameras and computers give machines the ability to see, recognize, and make the right decisions. The data captured by the camera is evaluated by a computer, and the relevant information and results are passed on to the control system, which can then respond accordingly." [4]

Specialized algorithms are used to perform tasks such as quality control, object recognition, position determination, and process monitoring. Rapid advances in computing power, storage technology, and algorithmic development have made technical image processing a key technology in numerous fields of application, from industrial manufacturing, robotics, and medical technology to transportation systems and intelligent automation.[3] It thus forms an essential basis for digital transformation and the further development of autonomous and adaptive systems. [4]

2.3 Image Noise in Image Processing

Like all electronic systems, digital image sensors are inherently affected by electronic noise. This manifests itself in random brightness or color deviations that can impair image quality and thus the evaluability of the recorded data. The strength of these interference signals is defined by the signal-to-noise ratio (SNR), which in image processing is described as the ratio between the average gray value of an image and the standard deviation of the image intensities. Image noise can basically be divided into additive and non-additive noise. Additive noise is modeled as a linear superposition of the ideal signal and a noise signal, while non-additive noise has more complex, non-linear relationships. The latter can be quantified by analyzing the correlation of neighboring pixels using a covariance matrix to distinguish between correlated and uncorrelated noise. [5] Physical causes include, in particular, photon noise, which is due to the quantum nature of light, and thermal noise, which results from the thermal motion of electrons in the semiconductor material. [4] Such noise components are reduced both by physical measures, such as increasing the illumination intensity or adjusting the integration time, and by mathematical signal processing methods, in particular filtering methods that optimize the data stream and achieve an improved signal-to-noise ratio. The following graphs in Figure 1 illustrate how various approaches, such as the use of filters, can reduce the data stream and adjust the noise sources. [4,6]

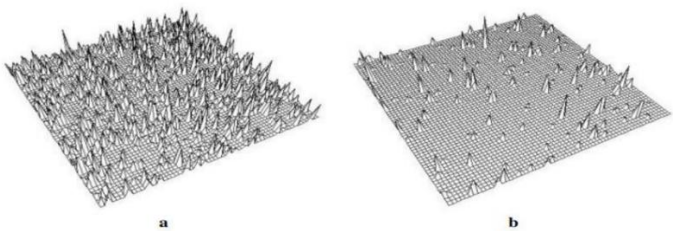

Fig 1. a) Image with strong noise, b) Resulting image after filtering. Source: A. Erhardt, Introduction to Digital Image Processing

2.4 Lighting in Image Processing

A key factor in the quality and reliability of image processing systems is the selection of lighting that is adapted to the conditions of use. Optimal lighting ensures homogeneous and temporally stable illumination of the entire relevant image area (region of interest, ROI) while reducing the need for subsequent image enhancement. The selection of suitable lighting systems is based on key criteria such as intensity, homogeneity, stability, spectral composition, and polarization properties of the light. Various types of lighting are used in industrial image processing. Natural daylight is usually unsuitable for reproducible measurement and analysis procedures due to its strong fluctuations in intensity and color temperature. Incandescent lamps provide a continuous spectrum, but their use is also limited due to mains frequency influences, interference, and high heat generation. Fluorescent tubes offer more uniform illumination and lower heat generation when operated with frequency rectifiers, but they are spectrally limited. Light-emitting diodes (LEDs) have established themselves as the standard light source in modern image processing.[7] They offer precise controllable light intensity, high energy efficiency, low heat generation, and a long service life. Due to their narrow wavelength range, they can be specifically adapted to specific applications. Laser illumination expands the spectrum to include high-precision, coherent light sources that are characterized by high radiation power, monochromaticity, and a variety of shapes (e.g., lines, grids, dot patterns).[7] They enable precise structure and contour detection, but require special safety and stability measures due to the potential risk to the eyes and temperature-sensitive aging. [4]

2.5 Object Classification

In technical image processing, object classification is based on quantitative features, whereas the human visual system can perceive and interpret both qualitative and quantitative characteristics. Digital image processing systems identify objects by measurable parameters that are mathematically defined and combined into feature vectors. These vectors represent points in a multidimensional feature space, where similar objects form clusters. Each cluster corresponds to a specific object class, with its position and extent indicating the degree of similarity among the objects it contains. The classification of new objects is performed by comparing their feature vectors with existing clusters. If no corresponding clusters exist, a training phase is initiated, during which the system learns from representative prototypes for each object class. Typical quantitative features include geometric and topological attributes such as length, width, area, and shape relationships. Applications of this approach include optical character recognition (OCR), automated cell counting in biological samples, and industrial quality assurance, for example, in the visual inspection of manufactured components.

2.6 Motion Detection

Motion detection is a key technique in technical image processing that is based on the recognition and analysis of changes within successive image sequences. The aim is to identify moving objects and determine their movement characteristics. Pixel or gray value changes between successive frames are evaluated to extract relevant motion information. Detection can be performed both on a global level, for example, to recognize scene movements, and on a local level to precisely track individual objects. Various algorithmic approaches are used to capture and analyze movements. Differential methods are based on calculating gray value differences between successive images. Changes in intensity indicate movements within the scene. This method is particularly effective in constant lighting conditions and with clearly

defined object contours. Correspondence methods, on the other hand, identify characteristic features or pixels that are recognized in successive frames. This allows motion vectors and object trajectories to be determined precisely. Another approach is the direct analysis of motion in space-time, in which image sequences are viewed as a three-dimensional data structure, with time forming the third dimension.[8] In this space, motions are interpreted as orientations or inclinations of gray-value structures, enabling continuous and temporally coherent capture of dynamic scenes as shown in Figure 2. [9]

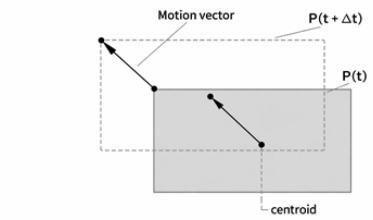

Fig 2. Image of motion vectors and corresponding points. Source: R. Schmid, Industrial Image Processing

To calculate motion vectors, contiguous image areas are identified that can be assigned to an object. The object and background are separated based on contrast and structural features. To achieve reliable results, corresponding points, such as distinctive pixels or centers of gravity of objects, must be clearly identified. These serve as the basis for determining the displacement between images and thus for precisely quantifying motion. [8]

2.7 Artificial Intelligence in Image Processing

Artificial intelligence (AI) has become an important component of technical image processing in recent years. It lets us automatically analyze complex visual info, spot patterns, and make decisions based on data-driven learning processes. John McCarthy defined AI as the science and technology of developing intelligent machines, especially those capable of solving tasks that require human intelligence.[11] In contrast to human intelligence, artificial intelligence is based on algorithmic processes that are trained using large amounts of data.[8] In image processing, AI is primarily used to recognize movements, classify objects, and identify structures in images, sequences, or videos. Supervised machine learning (ML) methods are used in industrial environments. These methods use annotated data sets to train models that learn from examples and can make general predictions based on them. Typical algorithms are convolutional neural networks (CNNs) and deep learning approaches, which are widely used due to their high accuracy and robustness. In supervised learning, models are trained with labeled training data so that they can match known inputs with the correct outputs and adjust their internal weightings. This process is data-intensive and requires high-quality, representative data sets. In contrast, unsupervised learning works with unlabeled data. Clustering and similarity analyses are used to identify structures, anomalies, and relationships within the data without first specifying a classification.[12] Neural networks, whose architecture is inspired by the human brain, play a central role here. They consist of several layers of artificial neurons that process information via weighted connections.[8] Input data is fed in via input layers, transformed in hidden layers, and finally converted into a result via the output layer. The weightings and activation functions determine how strongly a neuron responds to incoming signals and which information is passed on to the next layer.[13] Deep learning, a subcategory of neural networks with a large number of hidden layers, has become particularly well established in image processing.[8] Thanks to its deep network structures, it can independently extract complex features from large amounts of data and deliver highly accurate results. Deep learning models are capable of making

generalizations and achieving reliable results even with new, previously unknown images.[13] The main advantages of AI in image processing include its adaptability, robustness to noise, variability, and the ability to automatically analyze large amounts of data.[12] Nevertheless, challenges remain, particularly in the need for high-quality, diverse training data and the high computational effort required during model optimization. Careful parameterization and validation of the models is therefore crucial to ensure high accuracy and transferability of the results. [10]

2.8 Calibration of Imaging Systems

The precise calibration of cameras forms a central basis for applications in computer vision, especially for 3D reconstructions, object recognition, motion tracking, and augmented reality systems. The basis for this is the pinhole camera model, in which the transformation of points from 3D space into 2D image planes is described using intrinsic and extrinsic parameters [14]. The intrinsic parameters, summarized in the camera matrix, define the mapping of the camera coordinates to the image plane, while the extrinsic parameters describe the position and orientation of the camera in the world coordinate system [15]. For the projection of a 3D point Pw = (Xw, Yw, Zw) onto the 2D image plane p = (u, v), a combination of rotation and translation in a homogeneous transformation matrix is used, supplemented by the intrinsic camera matrix to ensure the correct mapping between world and pixel coordinates.[16] In real cameras, lenses cause additional geometric distortions that must be corrected. Radial distortions are caused by uneven refraction of light rays, while tangential distortions are due to slight misalignments of the lenses or sensors [14]. Thin prism distortions also model small manufacturing tolerances. Careful calibration takes these effects into account to enable accurate mapping of world points onto the image plane [17]. Classic methods are used to determine the camera parametersf. Conventional image-based methods cannot be directly applied to the calibration of event-based (EB) cameras, which deliver brightness changes per pixel as an asynchronous data stream. Modern approaches use illuminated calibration patterns, electronic displays, or learning-based image reconstruction methods to reliably determine the intrinsic and extrinsic parameters. [15]

2.9 Event-Based Imaging

Unlike conventional imaging methods, in which the incident light is captured completely at fixed time intervals and processed as an image frame, event-based cameras operate according to an asynchronous, event-driven measurement principle. Each pixel of the sensor continuously monitors the incident light intensity and only generates an event when a significant relative change in brightness is detected. This operating principle is based on the functioning of the biological retina and enables highly dynamic perception of visual changes. Technically, each pixel consists of a photodiode that is coupled to a contrast change detection unit and a logical processing unit. [19] The photons detected by the photodiode generate a photocurrent that is converted into a voltage. The logarithm of this voltage is then forwarded to the contrast detection unit. If the relative contrast change exceeds a defined threshold value, an event is triggered. [20, 21, 22] This principle allows immediate signal processing directly at the sensor level and significantly reduces redundant data. Each detected event is represented in a standardized numerical format and includes the pixel coordinates in the sensor array (x, y), the polarity of the brightness change, and a high-resolution timestamp. As shown in Figure 3, each time the photocurrent (red) exceeds a threshold, an event is generated (blue). [20] The polarity indicates whether a transition from dark to light (positive) or from light to dark (negative) was detected. The timestamp is typically specified in microseconds. The continuous event data stream can thus be described as a structured data

set with the attributes (x, y, p, t). For the analysis and presentation of event-based data, visualization in three-dimensional XYT space has proven effective, in which the temporal dimension is explicitly considered in addition to the spatial coordinates. This form of representation enables a precise mapping of the temporal-spatial continuity of events, avoids motion blur, and allows an extremely fast response to minimal changes in the scene. Typical fields of application for event-based cameras include autonomous vehicles, drone and UAV navigation, gesture recognition, and intelligent sensor systems. [20]

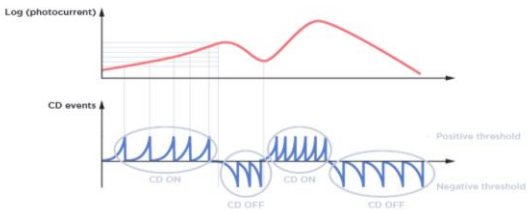

Fig 3. Representation of event generation. Source: Prophesee Documentation

(X, Y): the position of the pixel in the sensor array. Polarity (P): where P=1 indicates a CD-ON event (light change from dark to light) and P = 0 indicates a CD-OFF event (light change from light to dark). Timestamp (T): the time of the light changes, measured in microseconds (μs).[20]

2.9.1 Method Comparison

Conventional imaging systems capture scenes at fixed time intervals by acquiring full image frames at a predefined frame rate. This established approach is widely used in industrial, scientific, and security applications, providing high spatial resolution and detailed representations of static structures. However, frame-based acquisition inherently generates large amounts of redundant data, as unchanged or irrelevant scene content is repeatedly recorded. In addition, the camera remains temporally inactive between consecutive frames, which can result in information loss during fast motion or transient events. Consequently, conventional imaging is limited in applications that require high temporal resolution or continuous motion tracking. In contrast, Event-Based Imaging (EBI) is based on a fundamentally different principle. Instead of recording complete frames, event-based sensors operate asynchronously and respond only to changes in pixel intensity, registering brightness variations as discrete events. These events are processed in real time, producing sparse data streams that contain only information relevant to motion or scene changes.[21] This leads to a significant reduction in data redundancy while enabling extremely high temporal resolution, often in the microsecond range.

Building on this principle, the work of C. Willert and J. Klinner [21] at the German Aerospace Center (DLR) demonstrates how event-based approaches can be applied to flow measurement. Their study compares traditional techniques such as Particle Image Velocimetry (PIV) and Particle Tracking Velocimetry (PTV) with event-based methods for measuring particle motion in fluid flows.

The advantages of EBIV include high sensitivity to fast and transient phenomena, low data redundancy, and reduced computational and power requirements compared to conventional imaging systems. However, challenges remain, including limited spatial resolution, complex sensor calibration, and the need for specialized event-processing algorithms. [19, 22] Overall, both approaches are complementary; conventional imaging is well-suited for detailed, structured scenes with static or slowly moving objects, while event-based imaging provides superior performance in highly dynamic environments requiring high temporal accuracy and efficient data processing.

3 Research of the Bochum University of Applied Science

The objective of the research project conducted at Bochum University of Applied Sciences was the development of a real-time object tracking system capable of detecting moving objects with high precision and minimal latency. The starting point of the project was a cable-driven robot that moves a tool carrier plate throughout the workspace. The cable robot was controlled using a SpeedGOAT system, for which a position data acquisition system was to be developed as a feedback signal for validating the robot's position. For this purpose, an optical monitoring system was designed that operates reliably even at high movement speeds and accurately tracks the position of the carrier plate. In the subsequent sections, the development process of the tracking software for an event-based camera is systematically described. The focus is placed on the step-by-step implementation of the software architecture, including the conceptual design, implementation, and validation of the software solutions required to meet the defined project objectives and functional requirements. Finally, potential improvements and extension possibilities are discussed to provide an outlook on future optimizations and further developments.

An event-based camera from manufacturer Prophesee was used to implement the project. The core component of the camera is Sony's IMX636ES event-based image sensor, which was developed in collaboration between Sony and Prophesee. The camera is characterized by its compact dimensions of 30 × 30 × 36 mm and low weight of approximately 40 grams. The sensor has an HD resolution of 1280 × 720 pixels and is based on the principle of asynchronous, event-driven image capture [23, 24]. The camera comes with a software development kit (SDK) that supports implementation and integration into existing systems. Event-based cameras are particularly suitable for analyzing highly dynamic processes, such as capturing fast movements, vibration monitoring, object counting, and optical flow. Key areas of application also include autonomous systems, drone and vehicle navigation, gesture recognition, and intelligent sensor systems.[24] At the start of software development, a step-by-step approach was chosen to systematically increase the complexity of the application and gradually expand its functionality. To this end, three successive development stages were defined, all of which are based on the Basic Software Development Kit (SDK) of the event-based camera. Each development stage pursues a clearly defined goal and forms the basis for the subsequent expansion stage. In the first development stage, the focus was on pure generation and storage of event data. The aim was to capture and process the camera's event stream and output it in a suitable form to create a valid database for further analysis. The second stage of development expanded this approach to include an initial tracking application that could detect moving objects and visually represent their direction of movement using direction vectors. In the third and final development stage, complete object tracking was finally implemented, in which a bounding box is projected around each detected object, which remains continuously centered on the object's center of gravity. In addition, each tracked object is assigned a unique tracking ID so that even multiple objects can be reliably distinguished at the same time. The output of the tracking results is clearly defined and includes the object ID, the X and Y coordinates of the object center, and a timestamp. Additional functions have been integrated, including object size limitation, time control of recording, definition of regions of interest (ROI), and bias settings. The software application provides a flexible selection of the event source; either a recorded RAW file or the event-based camera can be used as the data source. This functionality is implemented in a dedicated class that initializes the corresponding event source based on user input. When using RAW data, a file path to the event file is specified, while in live operation, the camera is configured accordingly. In both cases, central parameters such as filter, bias, and ROI settings, as well

as configuration parameters of the tracking algorithms, can be defined to ensure efficient and application-specific data acquisition. The developed software is based on a pipeline-based architecture that enables efficient, modular, and clearly structured processing of event-based data. In this approach, processing is divided into successive stages, each of which performs specific tasks. Each pipeline stage can receive input data, process it, and forward output data to the next stage, whereby the data formats may differ between stages. [25, 26]

The program sequence begins with the transfer of a start command via the command line, which the software uses to activate or deactivate the desired options. After initialization and configuration of the event source, data acquisition and processing begin. Depending on the selected application, predefined algorithms and filters are used, for example, for object tracking, feature extraction, or further analysis. These include time-based storage functions ("write-from" and "write-to") for the targeted recording of event data, as well as size filters ("min-size" and "max-size") that can be used to limit the objects to be tracked based on their size. Furthermore, additional filter mechanisms, including STC filters, noise filters, and bias filters, can be selected to increase the number of relevant events. [20, 25, 26] Interactive controls allow the user to adjust key parameters such as object size or update frequency during runtime, or to terminate the program in a controlled manner. The results generated can then be saved, visualized, or further processed via defined interfaces, as shown in Figure 4. The event data can either be saved in raw format without further processing or exported in various file formats such as HDF5, DAT, or CSV. [19, 20]

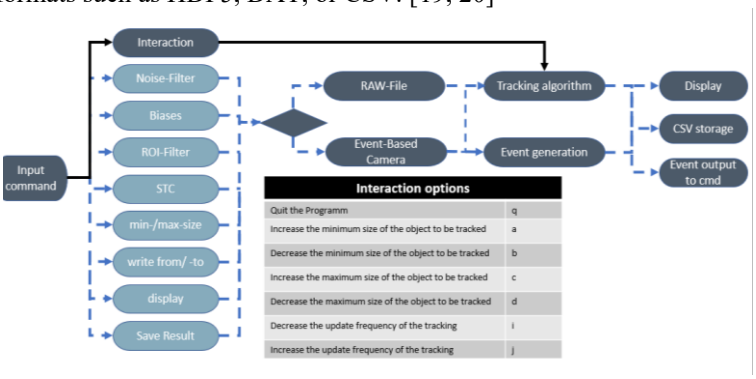

Fig 4. Flowchart representation of the final software application. Source: Image by Author

The programming language C++ was used for this project. Python is a suitable alternative for more advanced applications, particularly in the field of machine learning. Based on the selection of a suitable programming language and the resulting implementation flexibility, the software architecture described below was developed. This forms the basis for event-based object tracking and uses an asynchronous tracking algorithm that processes the recorded event data at fixed time intervals. [19, 20] Depending on the length of the time window under consideration, a variable number of tracking outputs can be generated. The algorithm is divided into four central processing steps: cluster formation, data association, tracker initialization, and tracking. In the cluster formation phase, events are grouped into object clusters. Two methods are available for this purpose: a grid-based method, in which the field of view is divided into elementary cells and active cells are combined into clusters based on an event threshold value, and an event-driven method that evaluates spatio-temporal distances between neighboring events. The relevant reference parameters are determined adaptively using a medoid shift approach. This is followed by data association, in which existing trackers are assigned to the clusters. This is done either using a distance-based method ("Nearest") or the intersection-over-union metric (IoU). If no clear IoU assignment is possible, a distance threshold is used as an alternative. In the initialization phase, motion

models are created, and new trackers are generated based on cluster information and event data. Various motion models are used for this purpose, including simple, smooth, and Kalman-based approaches. Two types of trackers are used for actual tracking: a Kalman tracker, which uses cluster features as observations and performs state predictions using prediction and update steps, and an ellipse tracker, which works on an event-based basis and continuously updates the object pose through weighted event contributions. The results are displayed via a visualization function that superimposes object boundaries and identification numbers over the event data. To synchronize events and tracking results, the input data is buffered, and demand-driven frames are generated so that the visualization matches the output frequency of the tracking results. The tracking results are then saved in a CSV file. The logged data includes the X and Y coordinates of the object center, the timestamp, and the object ID. A key challenge is the reduction of noise effects, which can lead to undesirable events in event-based cameras. [25, 26] Through iterative optimization of various filter options, a robust bias filter configuration was identified and integrated into the main program, achieving reliable and reproducible tracking quality. Another system-specific challenge arises from the fact that stationary objects do not generate events and therefore can no longer be detected. This problem could not be solved purely on the software side. As a hardware-based solution, a target with an integrated LED was developed that continuously generates events even when stationary, thus enabling permanent tracking. To achieve this goal, various filter options such as STC (Spatial Temporal Clustering), anti-noise filters, and bias filters were thoroughly investigated and subjected to extensive testing.[20] Through iterative adjustment and optimization of these filters, an optimal configuration of the bias filter was finally achieved, ensuring reproducible functionality and effective reduction of interference events. This optimized filter configuration was then implemented in the developed programming to enable robust and precise tracking of objects and minimize the distortion of tracking results by unwanted events. Structured storage enables downstream analysis, evaluation, or further processing of the data. A system-specific challenge arises from the fact that stationary objects do not generate events and therefore can no longer be detected. This problem could not be solved purely on the software side. A carrier plate with integrated LED was developed as a hardware-based solution, which continuously generates events even when stationary, thus enabling permanent tracking.

Following successful programming and verification of the process, the next step is to calibrate the camera system. This involves adjusting the sensor parameters to the actual operating conditions to ensure precise measurements. Careful calibration forms the basis for the subsequent validation of the tracking system under realistic conditions. [15, 27] Correct positioning of the camera is crucial for precise image capture. In particular, the working distance to the object must be selected so that distortion, blurring, or other image quality reductions are avoided. The required working distance g was determined using the relation

$$g = \mathrm{f} * \left(1 + \frac{G}{B}\right)$$

where f is the focal length (8 mm), G is the object size (800 mm), and B is the image size, i.e., the sensor height (4.3 mm for the used 1/2.5” sensor). The calculation yields an optimal working distance of approximately 1.5 m (1496.37 mm) above the base of the system. After completing the system setup and programming, the camera had to be calibrated. This calibration is essential to accurately determine the position of the detected targets. For the intrinsic calibration, two software tools were used. Around 50 images of a blinking chessboard pattern were captured, while the pattern was moved in front of the camera from different angles and positions [28]. Afterwards, the calibration tools processed all information (Distance between Workspace and Camera, Focal length, and the camera middle point) and computed the final camera parameters automatically.

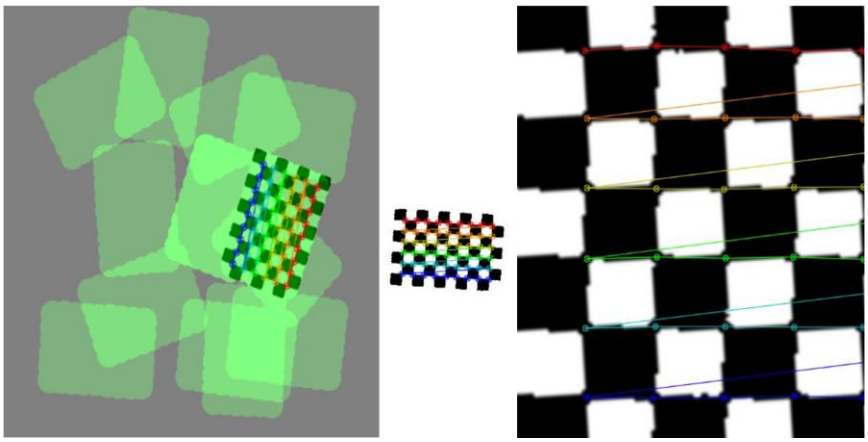

Fig 5. Intrinsic calibration process using a blinking chessboard pattern and Key Points Detection. Source: Image by Author

These transform pixel coordinates into proper camera coordinates. It allows the pixel measurements to be scaled to the real dimensions of the workspace, in this case, to millimeter units. After determining the intrinsic camera parameters, extrinsic calibration was performed. This involves determining the orientation of the camera in space relative to the observed object. Since the camera was mounted parallel to the recording surface, only rotations around the Z-axis were relevant, while rotations around the X- or Y-axis could be neglected. Multiple reference points equipped with blinking LEDs were recorded (see Figure 6) and subsequently analyzed. The evaluation revealed a systematic clockwise rotational offset of approximately 0.074 rad. [15, 16]

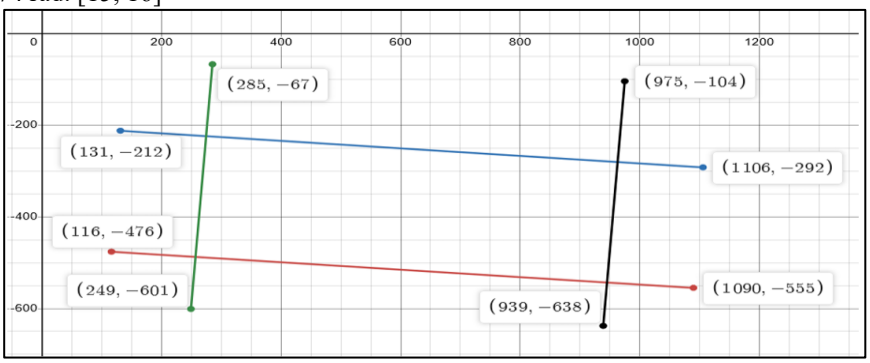

Fig 6. Position of LED in Camera Coordinates with Pixel units. Source: Image by Author

$$\theta_{\text{Green}} = \left|\tan^{-1}\left(\frac{285-249}{-67-(-601)}\right)\right| = 0.0673138757927\,\text{rad} \quad (1)$$

$$\theta_{\text{Black}} = \left|\tan^{-1}\left(\frac{975-939}{-104-(-638)}\right)\right| = 0.0673138757927\,\text{rad} \quad (2)$$

$$\theta_{\text{Blue}} = \left|\tan^{-1}\left(\frac{-212-(-292)}{1106-131}\right)\right| = 0.081867887924\,\text{rad} \quad (3)$$

$$\theta_{\text{Red}} = \left|\tan^{-1}\left(\frac{-476-(-555)}{1090-116}\right)\right| = 0.0809316663503\,\text{rad} \quad (4)$$

$$\theta = \frac{\theta_{\text{Green}} + \theta_{\text{Black}} + \theta_{\text{Blue}} + \theta_{\text{Red}}}{4} = 0.0743568264651\,\text{rad} \quad (5)$$

$$R = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

θ=0.0743568264651 radians

The determined rotation values are integrated into the extrinsic rotation matrix to correctly account for the spatial position and orientation of the camera. [15, 16]

4 Validation of the Implemented Tracking Application

After completing the development of the tracking application, a simulation environment was set up to validate the distance measurements using the event-based tracking system. The aim was to replicate the movement of the rope robot under controlled conditions and to verify the accuracy of the position and distance determination. For this purpose, two fixed positions were defined within the simulated workspace, with a real distance of 0.793 m between them. To achieve reproducible and controlled tracking results with the system based on event-based imaging (EBI) technology, these positions were connected via a linear guide and approached using a movable linear slide, as shown in Figure 7. Since the positions were temporarily stationary during the measurements, the movable linear slides were equipped with flashing LEDs to ensure reliable event generation. The LEDs were operated at a frequency of 155 Hz to generate sufficient tracking events even at a static position [29, 30].

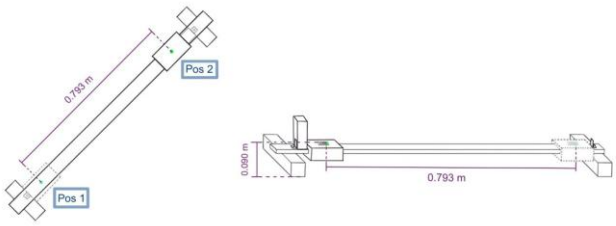

Fig 7. The structure Experimental Setup (Source: Image by Author)

To improve detection quality, suitable camera bias parameters were set to minimize noise interference and suppress false triggers. The following settings were used, among others: 100% bias_diff_off, 48% bias_diff_on, 48% bias_fo, 64% bias_hpf, and 74% bias_refr. In addition, the parameters for minimum and maximum object size were adjusted to the object to be tracked. This configuration enabled stable and reliable detection of small, static, flashing LEDs and formed the basis for validating the distance measurements. The measurements were carried out under four different lighting conditions (day/night, lights on/off) to evaluate the robustness and stability of the system under varying lighting conditions. Under three of these conditions, the measured pixel coordinates stabilized after a short time. Only during nighttime measurements without additional lightning did significant fluctuations occur, so these data sets were excluded from further evaluation. The results show that initial fluctuations in the pixel coordinates occur in the first microseconds of the measurement, which then transition into stable measured values. Under sufficient lighting conditions, both during the day and at night with the lights on, the coordinates remained stable throughout the entire measurement period. In contrast, nighttime measurements without lighting led to considerable instability. Under stable conditions, the Euclidean distance between the LED positions could be calculated reliably [24]. A detailed error analysis identified extrinsic camera parameters as the main cause of the initial deviations between calculated and actual distances [19, 25]. These results highlight the significant influence of lighting conditions on measurement stability. In particular, the absence of a constant light source led to increased variability in the measurement data, while sufficient illumination enabled consistent and reproducible results. During the stable measurement phases, the Euclidean distances between the positions were determined. After correcting the rotation angles in the X and Y directions, repeated measurements showed significantly improved agreement between calculated and expected distances, with deviations of less than

1 mm. The maximum measured distance between the stabilized positions 1 and 2 was 0. m, the minimum 0.7876596 m, which corresponds to a difference of 0.56572 mm. [31] Rounded to three decimal places, both measurements resulted in a distance of 0.788 m. These results confirm the high consistency and precision of the tracking data during the stable measurement phases.

5 Conclusion and Future Work

The studies conducted show that Event-Based Vision (EBV) is a highly efficient alternative for position tracking. With data rates of up to 130 MB/s, it achieves a high temporal resolution that surpasses that of conventional frame-based systems operating at 150 fps, while significantly reducing the computational load. At the same time, it was demonstrated that, after correcting extrinsic rotation components, a position accuracy with a deviation of less than 1 mm from the expected value is achieved. Furthermore, the RMS reprojection error of the camera calibration of 0.322 pixels confirms the high quality of the underlying mapping and calibration models. Furthermore, the system's temporal resolution enables an effective event rate of approximately 720 events/s for robust position feedback. Despite these positive results, the system has fundamental limitations. In particular, the so-called standstill problem represents a systemic limitation of DVS-based sensor technology, since no events are generated without changes in brightness, and stationary objects thus remain temporarily "invisible." Although this effect can be partially compensated for by active markers such as flashing LEDs (155 Hz), this requires additional hardware on the target object side. Furthermore, there is a trade-off in the filtering of event data: while strong filtering mechanisms effectively reduce noise, there is a risk of losing relevant events during fast motion sequences (over-filtering). Conversely, filters that are too weak can lead to erroneous reconstructions due to artifacts or so-called ghost events. Against this backdrop, further research questions arise regarding the robust handling of static scenes, the optimization of filtering strategies, and the integration of predictive tracking approaches to stabilize object identification across motionless phases. Building on these aspects, the next steps of the project will focus on implementing the developed tracking system into the existing smart factory infrastructure. The aim is to integrate the system into the real production environment and evaluate its practical applicability and performance under industrial conditions. At the same time, we are working on further developing the high-speed detection application. This involves expanding the X/Y position tracking so that velocity and acceleration can now also be calculated in real time from the position data. This additional information is crucial for optimizing the system for fast and dynamic movements. Another focus is on expanding data processing and interfaces. Specifically, this means developing and integrating connections to platforms such as SpeedGOAT, MATLAB/Simulink, and other real-time systems to enable seamless processing, analysis, and transfer of event-based data. Finally, validation of the tracking programs in dynamic test areas is planned. This will allow us to test the system's performance and reliability under realistic conditions and optimize it as needed. In summary, these steps form the basis for precise, fast, and robust real-time object tracking in industrial applications.

6 Acknowledgments

This work was funded by the Innovative University (IHS), an initiative of the Federal Ministry of Education and Research (BMBF) and the Joint Science Conference (GWK) as part of the THALESruhr transfer project "SMART Factory 4.0 for SMEs" at Bochum University of Applied Sciences.

Funding code (FKZ) 13IHS273

References

1. Zeiss, Das menschliche Auge, 2023. [Online]. Available: https://www.zeiss.de/visioncare/besser-sehen/sehen-verstehen/das-menschliche-auge.html [Accessed: May 28, 2023].
2. Burger, W., Burge, M.J. (2022). Edge-Preserving Smoothing Filters. In: Digital Image Processing. Texts in Computer Science. Springer, Cham. https://doi.org/10.1007/978-3-031-05744-1_17
3. Jähne, B. (2024). Einführung. In: Digitale Bildverarbeitung. Springer Vieweg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-59510-7_1
4. R. Dudziak and D. Mohr, Skript zur Vorlesung Technische Bildverarbeitung, Hochschule Bochum, 2017.
5. M. Raffel et al., Particle Image Velocimetry. Cham, Switzerland:Springer, 2018.
6. M. Buchwitz, Handbuch zur industriellen Bildverarbeitung Qualitätssicherung in der Praxis, 3rd ed., 2017. [Online]. Available: https://www.vision.fraunhofer.de/de/publikationen/leitfaeden/band17.html.
7. L. Fermum, Laser als Beleuchtung, Jun. 4, 2023. [Online]. Available: https://www.visiondoctor.com/laser-beleuchtung.html. [Accessed: Jun. 4, 2023].
8. Jähne, B. (2024). Merkmale. In: Digitale Bildverarbeitung. Springer Vieweg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-59510-7_16
9. R. Schmid, Industrielle Bildverarbeitung: Vom visuellen Empfinden zur Problemlösung,
10. F. Weber, Künstliche Intelligenz für Business Analytics: Algorithmen, Plattformen und Anwendungsszenarien, 2020.
11. C. Willert and J. Klinner, “Event-based imaging velocimetry: An assessment of event-based cameras for the measurement of fluid flows,” 2022.
12. T. Nguyen, S. Roy, and J. Meunier, “SmithNet: Strictness on Motion Texture Coherence for Anomaly Detection,” 2022.
13. A. Frommknecht and I. Effenberger, KI-basierte Bildverarbeitung in der Qualitätssicherung, 2021. [Online]. Available: https://www.dgq.de/fachbeitraege/ki-basierte-bildverarbeitung-in-der-qualitaetssicherung/. [Accessed: Jun. 6, 2023].
14. E. Weisstein, “Euler's Rotation Theorem,” Wolfram MathWorld. [Online] https://mathworld.wolfram.com/EulersRotationTheorem.html. [Accessed: Sept., 2024].
15. OpenCV, “Camera Calibration and 3D Reconstruction,” OpenCV, Dec. 31, 2019. [Online]https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html. [Accessed: Aug. 22, 2024].
16. W. Burger, “Zhang's Camera Calibration Algorithm: In-Depth Tutorial and Implementation,” Hagenberg, Austria, 2016.
17. M. J. Domínguez-Morales, Á. Jiménez-Fernández, G. Jiménez-Moreno, C. Conde, E. Cabello and A. Linares Barranco, "Bio-Inspired Stereo Vision Calibration for Dynamic Vision Sensors," IEEE Access, vol. 7, pp. 138415-138425, 2019.
18. D. Scaramuzza, “Vision Algorithms for Mobile Robotics: Lecture 03 Camera Calibration,” ETH Zürich, 2020 https://rpg.ifi.uzh.ch/docs/teaching/2020/03_camera_calibration.pdf.

19. OpenCV, “OpenCV Tutorials,” OpenCV, 2024. Available: https://docs.opencv.org/4.x/d9/df8/tutorial_root.html. [Accessed: Sept. 27, 2024].
20. Y. Ren et al., “Event-based imaging of levitated microparticles,” 2022.
21. Prophesee, Metavision SDK - Documentation, 2022. [Online].Available: https://docs.prophesee.ai/stable/index.html.
22. J. Hidalgo-Carrió, G. Gallego, and D. Scaramuzza, “Event-aided Direct Sparse Odometry,” 2022.
23. G. Roos, “Eval kit combines Sony HD sensor and Prophesee event based Metavision sensing,” Electronic Products, 2022. [Online].Available: https://www.electronicproducts.com/eval-kit-combines-sony-hd-sensor-and-prophesee-event-based-metavision-sensing/.
24. D. O’Shea, “Prophesee kit aims to spur adoption of Sony’s new event based vision sensors,” Electronic Products, 2022. [Online]. Available: https://www.electronicproducts.com/eval-kit-combines-sony-hd-sensor-and-prophesee-event-based-metavision-sensing/.
25. Prophesee, “Biases,” Prophesee, 2024. [Online]. Available: https://docs.prophesee.ai/stable/hw/manuals/biases.html. [Accessed: Aug. 17, 2024].
26. Prophesee, “Metavision Training Videos | Sensor Settings - Bias Tuning & Region of Interest (ROI),” Aug. 11, 2023.[Online]. Available: https://www.youtube.com/watch?v=7c7n5DMuatY. [Accessed: Aug. 17, 2024].
27. R. Szeliski, Computer Vision: Algorithms and Applications, 2nd ed.Cham, Switzerland: Springer, 2022.
28. M. Muglikar, M. Gehring, D. Gehring, and D. Scaramuzza, “How to Calibrate Your Event Camera,” in Proc. IEEE/CVF Conf. Computer Vision, 2022.
29. M. J. Domínguez-Morales et al., “Bio-Inspired Stereo Vision Calibration for Dynamic Vision Sensors,” IEEE Access, vol. 7, pp. 138415-138425, 2019.
30. Prophesee, “Intrinsics Calibration (C++),” Prophesee, 2024. [Online].Available: https://docs.prophesee.ai/stable/samples/modules/calibration/intrinsics.html. [Accessed: Sept. 27, 2024].
31. A. Athri, "Camera intrinsics: Axis skew," 2019. [Online]. Available: https://blog.immenselyhappy.com/post/camera-axis-skew/. [Accessed 22 August 2024].