Rutinowski *et al.*
EURASIP Journal on Image and Video Processing (2024) 2024:8
https://doi.org/10.1186/s13640-024-00623-6

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Semi-automated computer vision-based tracking of multiple industrial entities: a framework and dataset creation approach

Jérôme Rutinowski[1*], Hazem Youssef[1], Sven Franke[1], Irfan Fachrudin Priyanta[1], Frederik Polachowski[1], Moritz Roidl[1] and Christopher Reining[1]

*Correspondence:
jerome.rutinowski@tu-dortmund.de

[1] Chair of Material Handling and Warehousing, TU Dortmund University, Dortmund, Germany

Abstract

This contribution presents the TOMIE framework (Tracking Of Multiple Industrial Entities), a framework for the continuous tracking of industrial entities (e.g., pallets, crates, barrels) over a network of, in this example, six RGB cameras. This framework makes use of multiple sensors, data pipelines, and data annotation procedures, and is described in detail in this contribution. With the vision of a fully automated tracking system for industrial entities in mind, it enables researchers to efficiently capture high-quality data in an industrial setting. Using this framework, an image dataset, the TOMIE dataset, is created, which at the same time is used to gauge the framework's validity. This dataset contains annotation files for 112,860 frames and 640,936 entity instances that are captured from a set of six cameras that perceive a large indoor space. This dataset out-scales comparable datasets by a factor of four and is made up of scenarios, drawn from industrial applications from the sector of warehousing. Three tracking algorithms, namely ByteTrack, Bot-Sort, and SiamMOT, are applied to this dataset, serving as a proof-of-concept and providing tracking results that are comparable to the state of the art.

Keywords: Warehousing, Computer vision, Object detection, Classification

1 Introduction

The continuous, real-time tracking of entities of interest plays a crucial role in industrial settings from production facilities to warehouses [1]. In light of future challenges, automated, vision-based tracking of industrial entities helps increase process transparency [2]. The application potentials for the industry are manifold. With emerging needs in digitization and automation, industrial entities need to be continuously tracked in real-time to increase the adaptability of logistics systems with different conditions in terms of layouts, conveyors, etc. The available, but still unused information of these entities could be leveraged to automate and efficiently design the subsequent interfaces in the process.

The adoption of object tracking in the industry would facilitate the creation of a future-proof, scalable, and flexible infrastructure for monitoring processes. Process steps that rely on manual object identification or scanning equipment could then be

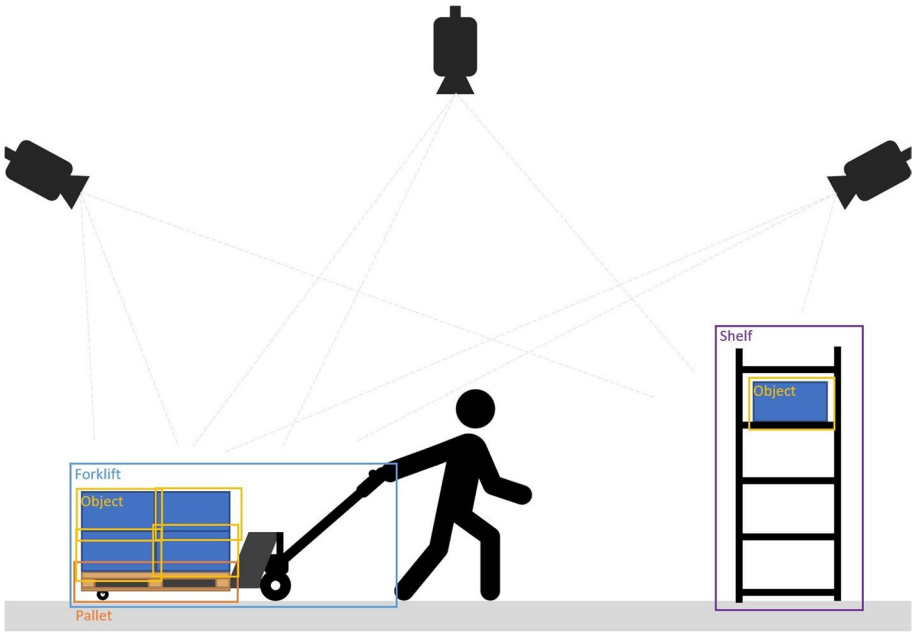


Fig. 1 Visualization depicting the RGB camera-based tracking of multiple industrial entities in a warehousing environment

eliminated and replaced by comparatively inexpensive cameras that function in an environment-agnostic manner. Given these requirements, our vision of a fully automated tracking of multiple entities in the industry can be articulated as follows. In an industrial environment, such as a warehouse, all entities should be continuously tracked, classified, and identified in real-time. As a consequence, their location, 6D pose, and identity are known at all times. This remains the case when multiple entities are present at once, might be in motion and might occlude one another. The sensors used for this purpose are comparatively inexpensive, do not need to meet a narrow set of criteria, do not need to be mounted in a very specific manner, and are easily obtainable. An example of one such sensor could be an RGB camera with a standard lens and resolution. The information that is inferred from the sensor data is used to monitor, optimize and increase the transparency of existing processes (e.g., in the form of a digital twin). Thanks to some of this information, novel processes might emerge. A visualization of this vision, put into practice in a warehousing scenario, might look like, can be seen in Fig. 1.

Besides the task of object tracking [2], research in the field of tracking concerning humans has also been performed [3–5]. We define the difference between (industrial) entities and (human) subjects in the sense that objects have simple and predictable movement patterns with only a brief and limited motion profile, if any. On the other hand, subjects, like humans, possess dynamic structures that are prone to self-occlusion, along with unpredictable and unrepeatable movement patterns. In our work, we only refer to object tracking, hence we take only industrial entities into account.

1.1 Problem statement

To put the herein-described vision of a fully automated tracking system into practice, the following challenges have to be addressed. Realistic scenarios, demonstrating the

movement of industrial entities throughout a common industrial environment, have to be chosen and planned. For this purpose, a viable data foundation, in the sense of entities that are commonly used in industrial settings, moved in a way in which they would be moved in the latter, has to be established. Out of these scenarios, a dataset has to be created. This dataset needs to contain annotated recordings, that can be used as trustworthy, ground truth training data for a computer vision algorithm. A set of such algorithms has to be selected and applied to the recorded data, and subsequently be compared to one another based on pre-defined evaluation metrics. Describing all these challenges, however, reveals the challenge that is at the core of this undertaking—the lack of a recording framework, that enables researchers to efficiently record and (semi-) automatically annotate data.

1.2 Goal of the contribution

The goals of this contribution are the following: we aim to provide a framework for the continuous tracking of industrial entities over a network of cameras. The provision of such a framework for the research community is motivated by the increase in efficiency and reduction of laborious annotation work entailed by it. We will describe the process of creating this framework in detail. Further, we aim to create a dataset with high-quality ground truth data, that can be used as a benchmark for subsequent research. This dataset will comprise multiple scenarios, that we will establish and describe in this contribution and that closely resemble industrial scenarios. We subsequently aim to apply a set of algorithms to the dataset, as to provide a proof-of-concept for our framework.

1.3 Structure and methodological approach

The next sections are structured as follows: Section 2 will outline and contextualize the related work on computer vision of tracking entities. This is followed by an explanation of the conducted experiments and used methodology in Sect. 3. Section 4 shows the corresponding results. Finally, in Sect. 5, the results are summarized and discussed, and an outlook is given on what further research in tracking industrial entities can look like. All in all, we want to provide a transparent approach on how state-of-the-art object tracking can be used as a benchmark for others. Our framework realizes tracking with a concrete approach that is also applicable to the industry and is a key element for practical application.

2 Related work

Computer vision-based tracking is a research field that has gained attention in recent years. The rapid developments of this field of study led to the emergence of numerous multi-object tracking algorithms and frameworks as well as datasets. Therefore, this chapter briefly presents the relevant literature related to camera-based object tracking techniques and frameworks, existing computer vision datasets, and methods of dataset creation. We also discuss existing tracking approaches in different application domains.

2.1 Camera-based object tracking

Camera-based object tracking involves detecting specific objects, estimating their motion paths, and maintaining individual identifications within the camera's view.

Multi-Object Tracking (MOT) is a widely applied computer vision concept, used in both single-camera and multi-camera systems across various applications. However, implementing MOT poses certain real-world challenges, i.e., long-term occlusion and the task of re-identification after the occlusion. Therefore, in this section, we review both systems to provide insights into their respective drawbacks and advantages.

2.1.1 Single-camera systems

The single-camera system is a fundamental system architecture for the development of tracking algorithms. Ciaparrone et al. [6] conducted a survey emphasizing the usage of Deep Learning (DL) in Multi-Object Tracking (MOT) for 2D data using the Single-Camera Tracking (SCT) technique. The survey discusses common steps in MOT algorithms for single-camera use, such as detection, feature extraction, affinity, and association, with a focus on implementing DL in these stages and evaluating them on an *MOTChallenge* dataset [7]. Typically, deep learning is applied to the initial stages, with limited use in affinity and association. From this survey [6], the authors emphasize three important parameters to deploy MOT algorithms: (i) the detection quality, (ii) Convolutional Neural Network (CNN) for feature extraction, and (iii) Single-Object Tracking (SOT) trackers. In terms of detection quality, appropriate detectors must be thoroughly selected to reduce the number of False Negatives (FN) in the Multi-Object Tracking Accuracy (MOTA) score. Currently, the best performing DL-based detector is Faster Region-based Convolutional Neural Network (RCNN) from [8]. In contrast, Single-Shot Detector (SSD) performs worse, as presented in [9, 10]. However, SSD was almost able to work in real-time (4.5 FPS), including the detection step.

For the feature extraction stage [6], the best-performing method, GoogLeNet [11], is applied to the datasets of MOT15 [12], MOT16, and MOT17 [13]. Approaches that do not use appearance (whether they are deep or conventional methods) typically perform worse. Visual features alone, however, are insufficient to compute affinity; many of the better-performing algorithms additionally include other characteristics, particularly motion features. The integration of SOT to the private MOT detectors along with DL is considered to generate well-performing online trackers.

Authors of [14–16] have investigated a DL approach for affinity using the MOT16 [13] dataset. Both works of [14, 16] demonstrate the reliable similarity measures to support person re-identification after occlusions and can reach the highest MOTA score of 49.3%. The survey also mentions that few have used DL to enhance the association process from the classical association, like the Hungarian algorithm, such as Recurrent Neural Network (RNN) [17], deep Multi-Layer Perceptron (MLP) [9], and Reinforcement Learning (RL) [18]. However, the usage of DL to directly guide the association algorithm and to perform tracking is still at its starting stage.

The Simple Online and Real-Time Tracking (SORT) [19] algorithm is regarded as the foundation for the online and real-time application of MOT. This approach implements Kalman Filter (KF) as the basic prediction of the tracklet bounding box between frames and the constant-velocity model as the motion model. One of the limitations of SORT is that it accumulates error estimation of the entity position over time due to obstacles or non-linear motion. To overcome this issue, the BoT-SORT tracker [20] was developed by combining the benefits of camera-motion correction, motion and appearance

information, and a more precise Kalman filter state vector. In addition, this tracker provides a novel, straightforward, and compelling technique of Intersection over Union (IoU) and re-identification through cosine-distance fusion, to obtain stronger correlations between detections and tracklets. The authors [20] further integrate BoT-SORT into the novel Byte-Track [21], which uses the backbone of the high-performance detector YOLOX. Both BoT-SORT and Byte-Tracker are evaluated using the datasets from the MOT17 and MOT20 challenges. The trackers outperform all current trackers in the MOTChallenge, with the results from the MOT17 test set, which are 80.2 IDF1 (the ratio of correct detections to the average number of ground truth and calculated detections), 65.0 HOTA (Higher Order Tracking Accuracy), and 80.5 MOTA.

2.1.2 Multi-camera systems

The aforementioned survey results demonstrate that SCT is a promising solution for MOT tasks, but it has limitations due to occlusions over longer time spans [22–24]. To address occlusion challenges, a Multi-Target Multi-Camera Tracking (MTMCT) approach is proposed by several contributions [25–28]. MTMCT combines perspectives from networked cameras to detect and track entities. This approach involves SCT for each camera, generating tracklets from the detection step and associating tracked targets using camera clustering to obtain Multi-Camera Tracks (MCTs) in a high-dimensional space as the final output [24, 26–30].

Zhang et al. [27] introduce a challenging benchmark for MOT on pedestrians that comprises two main modules: intra- and inter-camera tracking. Their dataset is recorded from non-overlapping video recordings from six to eight cameras with a resolution of 640×480 px. Intra-camera tracking generates tracklets for each individual camera that utilize the SCT algorithm. SCT's output is then forwarded to the inter-camera module where the data association takes place in the Multi-Target Multi-Camera Tracking (MTMCT) system. Tracking Length (TL), Crossing fragments (XFrag), and Crossing ID-switches (XIDS) are three possible evaluation metrics. For scenarios 2–6, TL results (percentage of the correctly tracked object) vary from 70 to 80%, XFrag results (number of times for a linked pair of tracks) range from 29 to 42 links, and XIDS results demonstrate from 23 to 44 tracks that lack a link to the ground truth trajectories.

A survey about intelligent multi-camera video surveillance is carried out by the authors of [26]. Their work introduces key technologies: multi-camera calibration, computation of camera network topology, multi-camera tracking, object re-identification, and multi-camera activity analysis. The survey looks at ways of estimating 3D camera calibration, including intrinsic and extrinsic parameters, common ground plane, automatic calibration, and two cameras with substantial overlap. The survey also emphasizes the topology of multi-camera networks which explains the handover of objects and computation of the topology. There are multiple methods for topology computation, including correspondence-based, correspondence-free, and topology inferred by non-overlapping camera networks. The section goes fairly in-depth into the ideas of inter-camera tracking based on multi-camera calibration, inter-camera tracking with appearance cues, and solving correspondence views across multiple cameras.

Specker et al. [28] define an occlusion-aware MTMCT approach for vehicle tracking and re-identification that enhances both SCT and Multi-Camera Tracks (MCTs)

operation. Furthermore, the authors adopt the global feature learning model from [31] to handle vehicle re-identification. To improve the resulting accuracy, a multiple re-identification network is applied. The SCT setup introduces an occlusion handling strategy and additional modules for filtering faulty detections. These steps can be achieved using temporal information from tracks. The MCTs setup uses a novel pipeline that includes a scene model, filtering of tracks, re-identification distance calculation, and hierarchical clustering. The hierarchical cross-camera clustering based on vehicle re-identification features is adapted from works of [32, 33] to merge the multi-camera tracks by leveraging topological and temporal constraints of the tracks of each camera in the network. The authors [28] propose that to decrease the negative influence of overlapping vehicles, one should improve re-identification by excluding boxes in the background or with occlusion.

2.2 Computer vision datasets

Successful deployment of DL-based computer vision applications relies on relevant and high-quality datasets [34]. Nowadays, datasets are aimed to encompass diverse and specific use cases and current trends tend to be dominated by outdoor applications, i.e., *MOTChallenge* (MOT15 [12], MOT16, MOT17 [13], MOT20 [35]), KITTI [36], MS COCO [37] (Common Objects in Context). The *MOTChallenge* dataset is a popular framework containing a large collection of multiple people-tracking datasets in dense pedestrian scenarios and the evaluation benchmark for various tracker algorithms.

The *MOTChallenge* uses different metrics to evaluate the performance of MOT methods. Standard evaluation metrics include multi-object tracking accuracy (MOTA) [38], higher order tracking accuracy (HOTA) [39], Identity F_1 Score (IDF1) [40], and Identity switches (IDs) [21]. Metrics differ in their consideration of the causes of errors. The IDs metric counts the number of swapped object identities during tracking. The MOTA metric combines three sources of errors and is defined as follows:

$$\mathrm{MOTA} = 1 - \frac{\sum_t (\mathrm{FP}_t + \mathrm{FN}_t + \mathrm{IDs}_t)}{\sum_t \mathrm{GT}_t}, \tag{1}$$

where t is the current frame and GT the total number of visible objects [13].

Alongside the TP, FP, FN, and TP measures, the HOTA metric considers the classification of associations. Given a TP c, the set of True Positive Associations (TPAs) is the set of TPs with the same ground truth and predicted identities as c[39]. The HOTA metric with a localization threshold α is defined as

$$\mathrm{HOTA}_\alpha = \sqrt{\frac{\sum_{c \in \mathrm{TP}} A(c)}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}}, \tag{2}$$

$$\text{with } A(c) = \frac{|\ \mathrm{TPA(c)}\ |}{|\ \mathrm{TPA(c)}\ | + |\ \mathrm{FNA(c)}\ | + |\ \mathrm{FPA(c)}\ |}. \tag{3}$$

The IDF1 Score considers the assignment of objects to their ground truth identities and is defined as

$$\mathrm{IDF}_1 = \frac{2\mathrm{TP}_{\mathrm{ID}}}{2\mathrm{TP}_{\mathrm{ID}} + \mathrm{FP}_{\mathrm{ID}} + \mathrm{FN}_{\mathrm{ID}}}. \tag{4}$$

An autonomous-driving related dataset is demonstrated in KITTI [36], which specifies various traffic scenarios. The published dataset contains 6 h of video from the cameras and sensor measurements which are captured at 10–100 Hz readings. Moreover, MS COCO [37] (Common Objects in Context) datasets contribute to providing daily life scenes with over 80 object classes and 200,000 labeled images. Despite large datasets, MS COCO does not cover industry-related computer vision applications. MVTec ITODD [41] accommodates realistic industrial setups for 3D object detection and pose estimation. The dataset consists of 28 asset classes that are sorted in more than 800 scenes and labeled using approximately 3500 rigid 3D transformations as the ground truth [41], i.e., engine parts, metal plates, bearings, injection pumps, etc. Luo et al. [42] present a benchmark dataset for industrial tools (ITD) to identify different types of tools at the level of usage. This dataset is aimed to accurately forecast how a robot would interact with various industry settings. ITD includes more than 11,000 hand-labeled RGB images in eight tool categories with 24 general industrial tools in total as well as their multi-perspective views of every tool. Regardless of various scenario views, this dataset only focuses on small industrial tools such as safety goggles, wrenches, screwdrivers, etc.

Synthetic-based industrial object datasets are, e.g., created in the research work of [43, 44]. The authors of [43] develop both real-world and synthetic data of industrial metal or reflective objects that are arranged as multi-view RGB images with 6D object pose labels. The real-world objects' dataset contains 600 scenes with 31,200 RGB images and the synthetic data provide 42,600 synthetic scenes containing 553,800 images. The twin resemblance of synthetic and real-world datasets including a controlled environment facilitates simulation-to-real-world research. In this manner, computer vision-based simulations with scalable scenarios can be conducted. Akar et al. [44] propose synthetic datasets of industrial objects for object detection applications. The datasets are generated as 200,000 photo-realistic generated images with precise bounding box annotations that are categorized as 8 industrial objects in 32 scenarios. The warehouse environment model as well as the datasets are rendered using NVIDIA Omniverse. The goal of synthetic datasets is to automatically generate datasets for real-world multiple object detectors from genuine camera feeds.

The Logistics Objects in Context (LOCO) [34] dataset presents an indoor environment dataset for warehousing logistics. However, the LOCO dataset does not contain timestamps for the recorded image streams which renders it unsuitable for object-tracking algorithms. This type of logistics or industry-related dataset is rare to encounter in research [45–47]. The authors [34] intend to accelerate computer vision-based research for logistics by emphasizing the creation of objects and scenes of warehousing entities and privacy protection of image acquisition. The LOCO dataset has 39,101 images comprising 151,428 annotated logistics entities such as pallets, pallet trucks, and forklifts.

2.3 Dataset creation methods

The creation of industry-related datasets is the topic of this subsection. Obtaining and marking such datasets in an industrial environment can be difficult due to factors

Table 1 Comparison of industrial-based datasets creation setups

Dataset	Acquisition tool	Camera type	Resolution [px]	Evaluation
MVTec ITODD [41]	3 Cameras 3D Camera	Grayscale Stereo	8 MP	PP3D PP3D-E PP3D-E-2D S2D RANSAC
Industrial Metal Objects [43]	JAI GO-5000-PGE mvBlueFOX3 RealSense L515 RealSense D415 Rico Theta S	RGB Grayscale RGB, LiDAR RGB, IR Stereo 360^{o} Camera	2560 x 2048 4064 x 3044 1920 x 1080 1920 x 1080	MSSD
ITD [42]	Kinect 2.0	RGBD	1024 x 575	FR-CNN R-FCN YOLOv3 SSD
LOCO [34]	MS Kinect v2 Intel Realsense D435 SJCAM SJ-4000MS LifeCam HD-3000 Logitech C310	RGBD RGBD RGB RGB RGB	1920 x 1080 1920 x 1080 1920 x 1080 1280 x 800 1280 x 800	YOLOv4608 YOLOv4tiny FR-CNN
Synthetic Object Dataset [44]	NVIDIA Omniverse	Renderer Software	720	FR-CNN SSD

such as it being time-consuming, susceptible to human mistakes, and constrained by various privacy and security regulations [34, 43, 44]. Therefore, using a semi- or fully-automated pipeline for the dataset creation should be considered. All setups of the related industrial dataset papers are summarized in Table 1.

Semi-manual annotation for the 3D images of the industrial objects is adapted in MVTec ITODD [41]. For each object, three types of scenes are captured: (i) those with only one instance of the object and no extra items, (ii) those with multiple instances of the object and no extra items, and (iii) those with both multiple instances of the object and additional clutter. The individual scene is recorded once using a 3D industrial camera, and twice using grayscale cameras: one scene with a randomly projected pattern and another one without a random pattern. Both grayscale and 3D cameras are located on top of the shelf setup and calibrated previously with regard to their relative position to the object. The recorded object is positioned on a calibrated turn's movements under the cameras that allow the multiple scenes to be captured automatically. In this manner, the ground truth of 3D object poses is transferred directly for every rotation. Instead of using a rounding box as the correctness measure, the authors [41] implement 3D pose-based evaluation. The datasets are evaluated using 3D pose-based methods: Shape-Based 3D Matching (S2D), Point-Pair Voting (PP3D), Point-Pair Voting with 3D edges (PP3D-E), Point-Pair Voting with 3D edges and 2D refinement (S2D), and RANSAC. Although S2D outperforms other methods when estimating the image results, a majority of the results are false positives. PP3D-E performs the prediction well with a top-1 detection rate of 68% with the given threshold of 5% but the running time is higher (by 0.1 s) which must be improved for industrial use.

The Industrial tool dataset (ITD) [42] is gathered utilizing a Kinect 2.0 sensor that can generate 30 RGBD frames per second, featuring a resolution of 1024×575 px, as

well as 512 × 424 px depth frames. To collect the data, the tools are positioned within a distance range of 1–5 m from the camera. The tools are placed in their typical positions and industrial settings, while the camera is positioned at the same point of view as that of the worker's eyes. The worker walks smoothly around the target tool while maintaining a consistent focus on it. The labeling process is conducted manually by experts. Each worker is tasked with identifying the name of the tool, the category it belongs to, and its potential usage. The task requires a total of approximately 200 h to complete. The performed evaluations demonstrate that cluttered backgrounds and inconsistent ambient lighting impact tool detection. Moreover, the performance suffers from the worker's motion-induced visual blur. To achieve the industrial requirements, the refinement of detection methods is necessary.

The dataset for industrial metal objects, described in [43], is recorded in two parts—real-world and synthetic data. An industrial grasping robot, the Fanuc M20ia, is equipped with the data acquisition setup listed in Table 1 (except the 360^o camera) to record multi-view images of various scenes in the real-world. The real-world scene is captured by each camera from 13 different viewpoints to obtain 6D poses of each object. Six different metal objects with different lighting setups are also considered during the recording. In addition, the objects are recorded in three different types of carriers: metal plates, small bins, and cardboard boxes. The labeling of 6D poses from object models is carried out semi-manually using a proprietary tool. The synthetic datasets are generated by mimicking real-world scenes, i.e., poses, lighting, models, and textures on Unity for which the virtual environment uses an HDRI environment map. This map is constructed by the captured images from a 360^o camera using different types of exposures. Finally, all real-world and virtual scenes are generated as the dataset containing subfolders for each camera ID and individual subfolders corresponding to each CAD model of the respective object. To evaluate the labeling performance, de Roovere et al. calculate the pose errors using Maximum Symmetry-Aware Surface Distance (MSSD).

A full synthetic dataset for warehousing environments is rendered in NVIDIA Omniverse based on the Universal Scene Description (USD) method [44]. Akar et al. employ Material AI tools to transform the captured images from real-world cameras and material scanners into realistic virtual models. The scene recording setups are emulated as authentic factory representations that have many assets and instances. For each scene recording, the randomized locations and rotations are assigned to the camera to capture the scene's randomness from diverse perspectives. Subsequently, synthetic image generation is initiated to automatically and accurately annotate the images in each scene up to the pixel level. FRCNN ResNet50 surpasses the SSD DL model in terms of detecting stillages, transport robots, dollies, and pallets with the Average Precision (AP) metric at 0.5 are 69.90%, 89.93%, and 48.60%, respectively. The recordings of the LOCO [34] dataset are captured using different types of cameras with diverse fields of view and resolutions in a real warehousing environment. The cameras are set up on a mobile unit with a special arm, thus enabling the re-adjustment of the camera's point of view. The mobile unit moves around the warehouse while changing the cameras' perspectives. The captured images are recorded and stored with a 1 Hz frequency. The LOCO annotator uses the backbone of the COCO annotator with additional features, such as an automated bounding box tool and new hotkeys. To ensure the privacy of the warehouse workers in

the dataset, Mayershofer et al. utilize a neural network to automatically perform pixelization of all detected faces during the annotation phase. The evaluated models exhibit a lower performance compared to the COCO benchmark, with an mAP at 0.5 ≈20–40% on the LOCO benchmark.

3 Methodology

Due to the existing deficiency in the publicly available object tracking datasets in the logistics and industrial domains, we collect a custom dataset and annotate it in a semi-automated fashion. The following section describes our dataset recording procedure, our dataset structure, and the annotation process. The word entity is used in this work to refer to the recorded objects. This excludes commonly used references in the literature such as object pose estimation, object tracking, and object detection.

3.1 Planning and execution of the dataset recording

We derive two situations from the warehousing sector that represent processes occurring in actual industrial use cases, namely a goods reception scenario and a block storage scenario. To ensure realistic circumstances, two different loading degrees of the pallets were recorded. In the first stage, only empty pallets are moved. The second stage involves fully loaded pallets. As to ensure a realistic environment, we use six different industrial entities (small load carriers, pallets, barrels, cardboard boxes, forklifts, and a mesh box, as shown in Fig. 2).

Pallets of different types were used, including Euro pallets, CHEP pallets, and hygiene pallets. The entities were handled with two manual pallet trucks. The selection of entities is inspired by DIN 55405 and DIN EN 13698-1 [48, 49].

We define a pallet to be fully loaded if it is stacked with three layers of small load carriers on top of one another. In addition, entities such as barrels and cardboard boxes have been used and were not stacked. The first scenario, shown in Fig. 3 a and b, mimics an inbound material flow scenario that starts with an empty loading area, with the pallets set up to fill said area along the process. The dotted lines represent the spots that the pallets are placed in during this scenario. In the first stage, they are placed apart from one another while in the second scenario, they are placed more closely together. In the block warehouse scenario, shown in Fig. 3c and d, the recordings are with a block of pallets that is already set up. Subsequently, individual pallets are pulled out and moved outside of the field of view of the cameras. For this scenario, a 2×2 block of pallets has been used in the first stage, and a 3×3 one in the second stage.

In total seven recordings are performed, as shown in Fig. 4. Figure 4a shows scenario 1, stage 1, during which the pallets are arranged with a considerable distance between them. The inspiration for this scenario is that the two lanes that are built in this way could be found in the goods-receiving area of a warehouse, e.g., to unload trucks. The pallets are then unloaded, e.g., from a truck and are placed far apart to allow warehouse workers to inspect the newly arrived goods. In Fig. 4b, scenario 1, stage 1 with the closely placed pallets is shown. This scenario mirrors the loading process as it could be expected to be performed to load a truck. Figure 4c and d shows the first scenario in their second stage, i.e., with loaded pallets. Last, Fig. 4e,

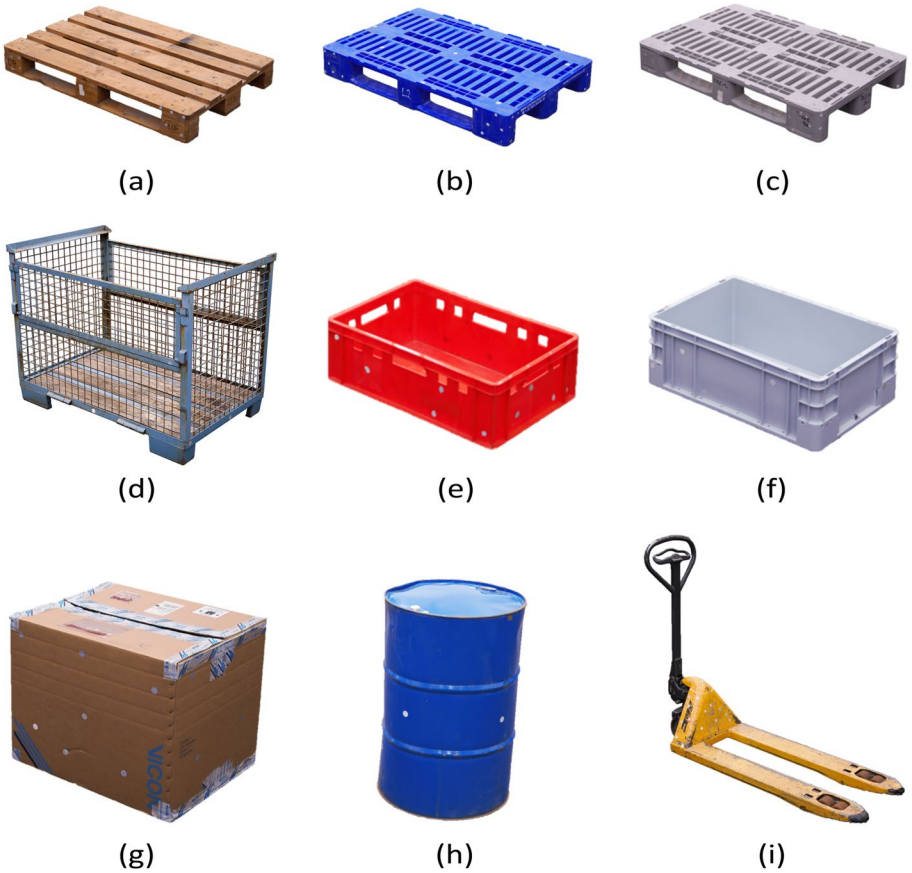


Fig. 2 Entities used in our recordings: **a** Euro pallet, **b** CHEP pallet, **c** Hygiene pallet, **d** Mesh box, **e** Red small load carrier, **f** Gray small load carrier, **g** Cardboard, **h** Barrel, **i** Forklift

f and g shows the second scenario, which mimics a block warehouse, in the above-mentioned stages. During the recording of these scenarios, varying lighting conditions were used.

3.2 Setup and data collection

The area that is used to record the data proposed in this work is a former warehouse that has been transformed into an applied research facility. Its recording space is covered by six monocular RGB cameras providing parallel video streams. The area is also covered by a marker-based motion capture system [45] comprising 52 infrared cameras. These cameras provide accurate poses of the tracked entities with respect to a common reference frame. This setup is shown in Fig. 5.

The dataset is collected by deploying industrial entities within the recording space, according to the configuration of the scenarios mentioned in section . The entities are moved around by human operators to simulate inbound and outbound operations, again according to the previously described scenarios. While doing so, a video stream is captured through the RGB cameras. Simultaneously, the ground truth pose information for all tracked entities is acquired through the motion capture system.

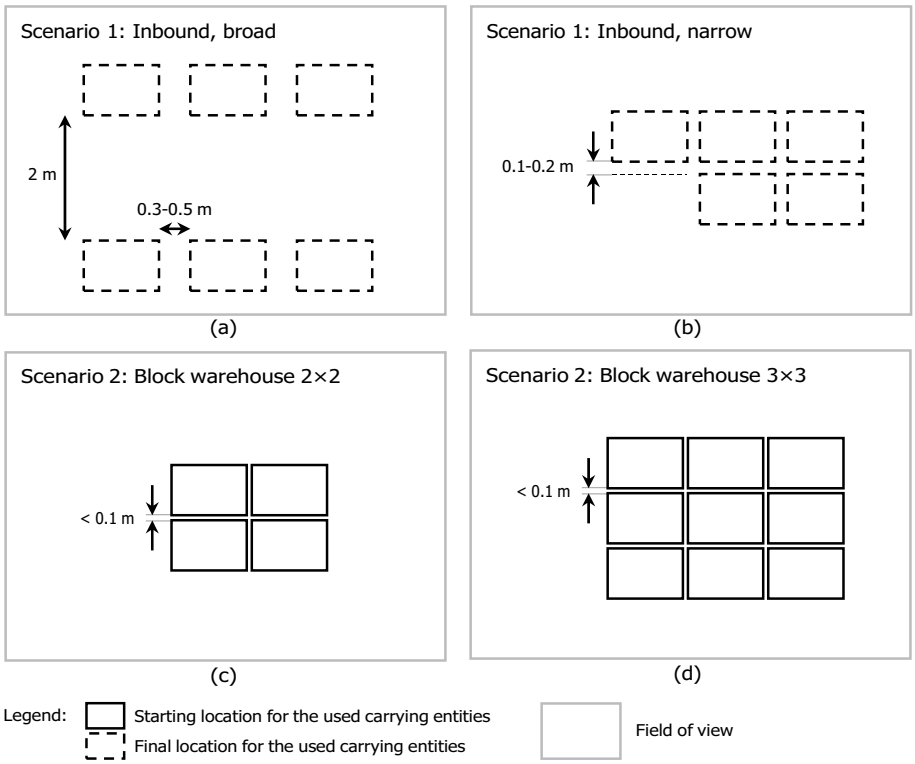


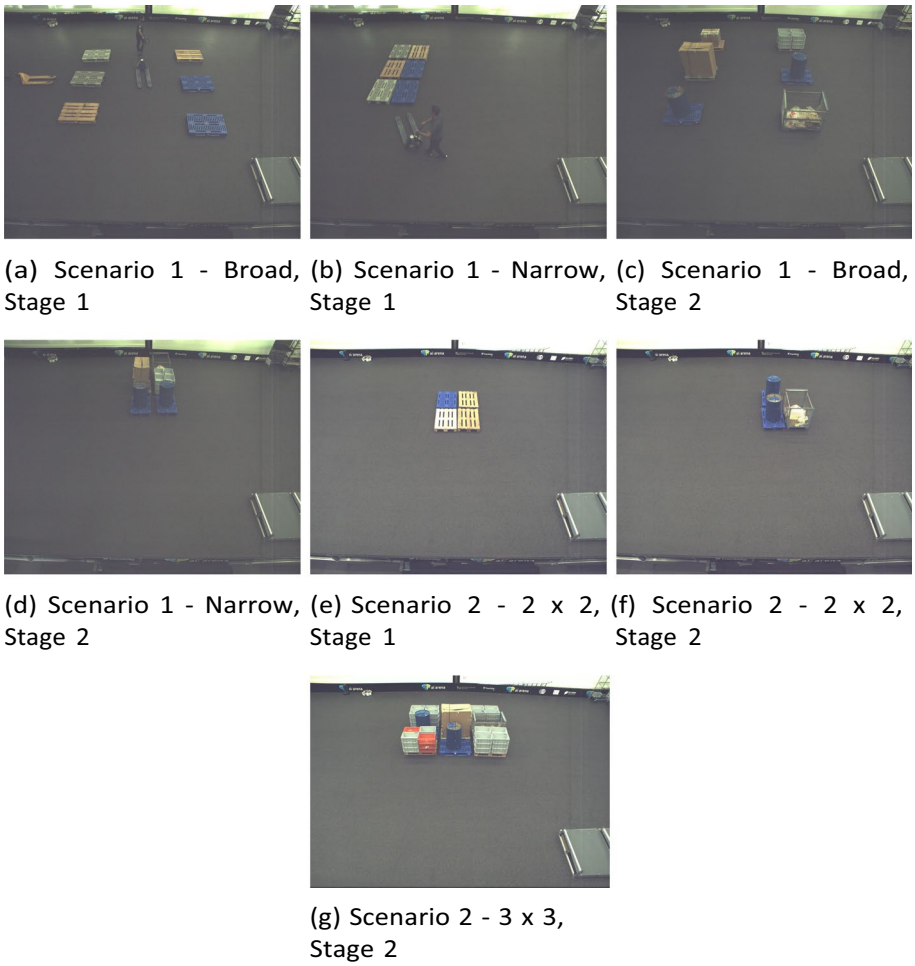
Fig. 3 Schematic illustration of scenario 1 (**a** and **b**) with its two degrees of pallet proximity and scenario 2 (**c** and **d**) with its two block warehouse pallet ordering structures

3.3 Data processing

The data collected by the motion capture system and the RGB camera system are processed on separate computers. The aim is to reduce the processing time necessary to request pose frames from the motion capture system and thus to increase the frames per second (FPS) of the streamed images from the RGB camera system. The frames from each of the six RGB cameras are collected on one computer along with their timestamps. The second computer collects information on entity IDs, entity poses, and timestamps from the motion capture system. The start and stop of collection from each of the systems are triggered manually. Each system's streams are synchronized in a post-processing phase.

In terms of hardware, six Genie Nano C2590 RGB cameras with a 2 MP resolution are used. The cameras are fitted with a Kowa LM8HC-SW lens with a 79.4×63.0 field angle. All six cameras are connected to a 10 Gigabit Ethernet switch, which passes the streamed data to a data collection computer via an optical fiber network connection. The motion capture system consisting of 52 cameras uses a mixture of Vicon Vero and Vicon Vantage cameras that are mounted on the ceiling and at different elevations in our research facility.

RGB camera settings such as brightness and white balance values were allowed to update periodically throughout the recordings. Illumination in the recording space was kept constant throughout each individual recording, changing in between recordings, and there was no significant color hue variation from the scene. Images were stored in raw *bmp* format and distortion was preserved.



(a) Scenario 1 - Broad, Stage 1 (b) Scenario 1 - Narrow, Stage 1 (c) Scenario 1 - Broad, Stage 2

(d) Scenario 1 - Narrow, Stage 2 (e) Scenario 2 - 2 x 2, Stage 1 (f) Scenario 2 - 2 x 2, Stage 2

(g) Scenario 2 - 3 x 3, Stage 2

Fig. 4 Frames taken from the two scenarios and their respective stages, used for our recordings

3.3.1 Synchronization

The recording of video streams is event-triggered for each camera. However, to guarantee an equal number of retrieved images from all cameras, simultaneous capturing is necessary. Synchronized, simultaneous capturing also has the advantage of preserving the instantaneous state of the scene. Recording in such a manner can facilitate performing hand-offs between the different perspectives for multi-camera tracking algorithms. This also has the advantage of enabling more accurate re-identification of entities from different viewpoints.

Simultaneous capturing is done for all cameras by triggering a single image capture on each camera followed by trigger locking to prevent further capturing. The software lock is released on all cameras simultaneously only when image retrieval on all cameras has ended. Thus, for each capturing trigger, the slowest camera determines the overall FPS of the system. An average of approximately 20 FPS per scenario is achieved.

Beyond achieving synchronization amongst the RGB cameras, it is necessary to synchronize between the RGB camera system and the motion capture system due to data capturing rate differences. During our experiments, the motion capture system

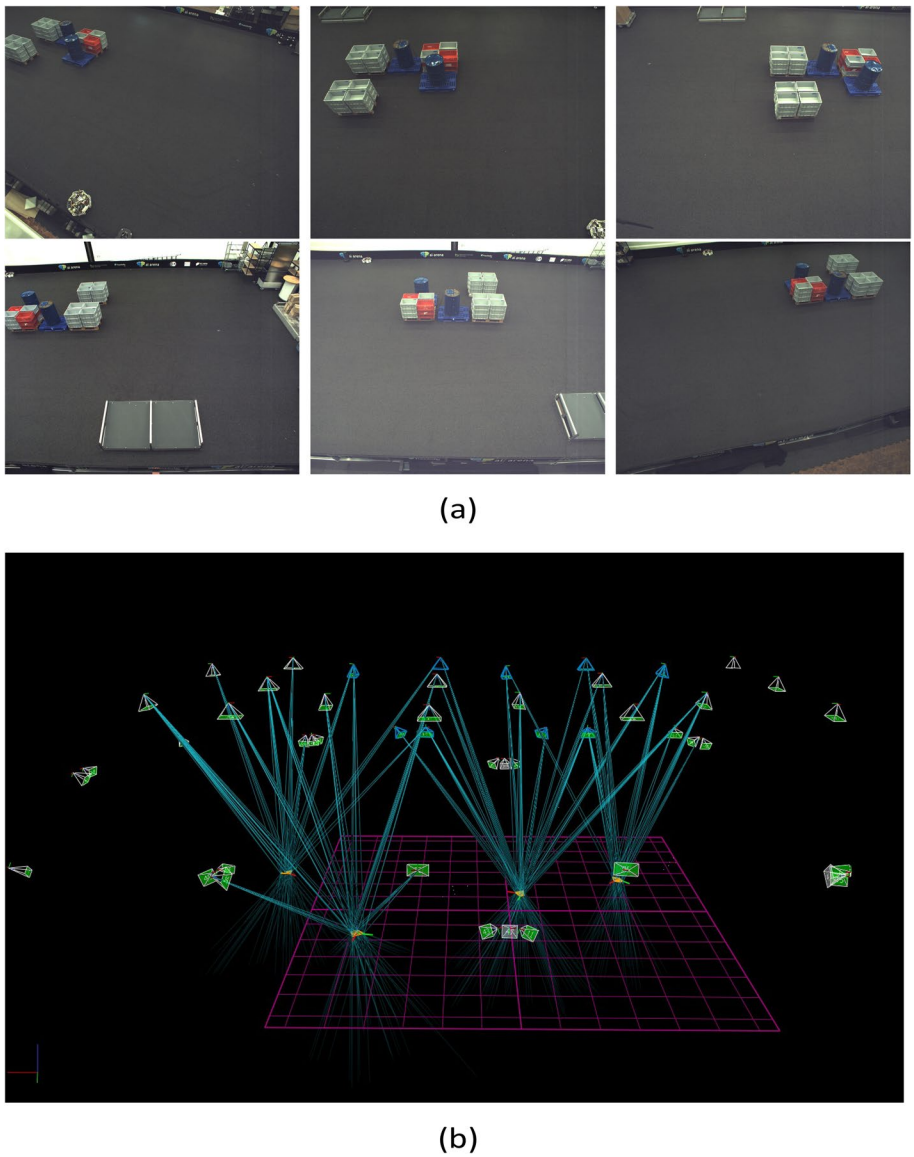


Fig. 5 Data collection setup. **a** The RGB images of the same scene as viewed from the six cameras, **b** entities as perceived in the motion capturing system (obtained for a different scene). Rays show the detected retro-reflective markers by the system

had a fixed pose update rate of 200 Hz. We match image frames to their respective poses based on the smallest timestamp difference between both instances. Since entities in the scene move at less than $1\frac{m}{s}$ and due to the high update rate of the motion capture system, pose differences between consecutive frames are insignificant. The synchronization between both streams is accomplished as a post-processing step.

3.3.2 Data structure

Since the currently available datasets for object tracking lack the combination of systems used in this work, we collect our data and process it into a custom data structure. The final annotation data structure of our custom dataset is shown in Table 2.

Table 2 Sample entries in post-processed annotated data

Image path	Entity name	Position	...Orientation	Delta time	Bounding box	Visible
...	...	...	...	...	...	...
camera_6/images/3.jpg	Pallet_9	[− 10672.35, 1815.89, 85.49]	...[0.0037, 0.0019, − 1.5481]	− 00.00088	[− 1, − 1, − 1, − 1]	0
camera_6/images/769.jpg	Forklift_2	[− 3142.96, − 1409.38, 239.16]	...[− 0.0035, − 0.0036, − 0.0014]	− 00.0037	[293, 0, 215, 339]	1
...	...	...	...	...	...	...

The Image Path refers to the relative image path with respect to each camera view. Images are converted to jpg format for efficient storage. Entity Name refers to the entity ID as retrieved via the motion capture system. It is worth noting that initially an entry is preserved for all entities in each captured image, regardless of their existence in the captured scene. During the annotation phase, as discussed in section , invalid projections of the entities' 3D models are removed. Position and Orientation are 3×1 vectors defining the relative pose of the entities in 3D space with respect to each camera. Position data are provided in mm and orientation data are provided in radians in intrinsic *XYZ* Euler format. The position is obtained with respect to the motion capture system's global reference frame. The reference frames of the motion capture system and the RGB camera system are unified to enable the calculation of the transformation chain generating the entity's relative pose. The entry Delta Time is the smallest calculated time offset between the capturing time of the RGB image and its corresponding pose. The Bounding Box is the 4×1 vector defining the pixel coordinates of the top left x and y coordinates, along with the width and height of the box. The Visible flag indicates whether an entity is perceived in the field of view of the respective camera. The flag is generated automatically as part of the post-processing step of the annotation pipeline used. This is accomplished by disregarding entity 3D model projections when rendered at their ground truth pose, as discussed in section. Bounding boxes that correspond to entities that are invisible in the relevant camera view are denoted with coordinates of −1. Invalid data from the motion capture system, such as those obtained when an entity is outside the system's region of operation, are filtered out in a post-processing step.

3.4 Annotation

To maximize image capturing throughput, we separate the data collection phase from the annotation phase. To facilitate the automated annotation of collected data, two pre-processing steps are required. First, for each industrial entity, an accurate 3D model has to be created. Second, the intrinsic and extrinsic parameters of the camera have to be calculated. In the annotation phase, the image annotations are generated by leveraging the VisPy visualization library [50]. For each collected pose, the relative position to each camera is calculated using the extrinsic camera parameters. Afterward, this relative position is used to calculate the 3D models' projection at the ground truth pose, collected from the motion capture system onto the image

plane using the camera parameters. The annotation pipeline fits bounding boxes to the 2D image projections of the 3D models at their obtained poses in the scene relative to the camera of interest.

The annotation pipeline comprises different phases. Initially, the RGB images and motion capture system poses are collected simultaneously. Then the reference frames of the motion capturing system and the RGB camera system are unified, and incoming streams from both systems are synchronized. This is followed by the main phase during which the relative transformations are calculated between the tracked entities of interest and each camera. Finally, the 3D models are projected at their calculated relative transformations where they are fitted with bounding boxes to generate the final image annotations.

4 Results

The herein presented TOMIE dataset includes a total of 112, 860 images and 640, 936 entity instances. In comparison to similar datasets, the number of captured images outnumbers the biggest dataset [13] by a factor of 4, while the number of captured entity instances is approximately 25% smaller.

The annotations were generated using a computer equipped with an Intel Core i9 that possesses 28 cores and 128 GB of RAM. The renderer deployed, VisPy [50], uses the onboard Nvidia Titan Xp GPU with 12 GB of VRAM throughout the annotation process. Samples of annotated images are shown in Fig. 6. We provide the source code for

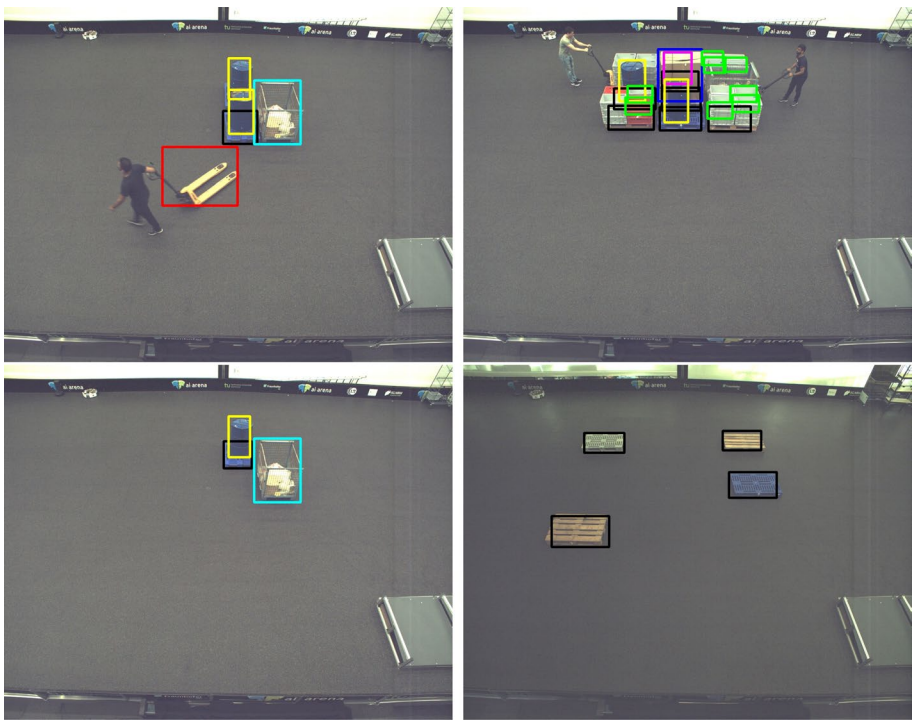


Fig. 6 Samples of annotated images from a single view and different scenarios from our custom dataset. Bounding box colors are unique to each entity class

Table 3 Dataset statistics per camera

Sequence	I	II	III	IV	V	VI
# instances	64,430	55,136	76,904	208,134	51,364	184,968
# frames	14,825	19,141	20,767	23,359	12,651	22,117
Annotation time (min)	1618	1388	1926	5209	1285	4637

Table 4 Dataset statistics per entity class

Entity	Barrel	Forklift	Pallet	Mesh Box	Cardboard Box	Load Carrier
# instances	55,492	87,914	305,498	33,452	57,672	100,908

Table 5 Average precision (AP) and average recall (AR) for bounding box estimation of industrial entities

Metric	AP^{val}	AR^{val}	$F1 - Score$	$AP^{val}_{.50}$	$AP^{val}_{.75}$	$AP^{val}_{.50:.95}$
Result	0.80	0.83	0.815	0.92	0.87	0.83

our automated annotation pipeline[1] for public usage as well as the source code for our data collection phase[2].

During the annotation process, an average of 1.5 s was spent on each object instance in the recording. This amounts on average to 9 s spent per image for the annotation of all visible entities. The annotation speed achieved through the use of automated annotation is significantly higher than comparable manual annotation, like the one described in [51]. Dataset statistics per camera and per entity are shown in Tables 3 and 4.

To evaluate how far our custom dataset can be used for training classifiers that achieve a performance sufficient for industrial applications, multiple experiments were conducted. For these experiments, three of the currently best-performing models for the MOT20 [35] dataset, namely ByteTrack [20], SiamMot [52], and Bot-SORT [20] were chosen. Publicly available and official implementations for all models were used during the evaluation. The ByteTrack and Bot-SORT models rely on YoloX [53] as a backbone for object detection. To this end, one YoloX model was pre-trained on our custom dataset to be used for both evaluation models. The average precision and recall of the resulting model were measured and are shown in Table 5. The resulting object detection results are visualized on some samples of our custom dataset in Fig. 7 as well.

All models were trained and evaluated on our custom dataset in accordance with their respective work. For evaluation, the CLEAR metrics [38], including MOTA, as well as IDF1, and HOTA were used. These metrics evaluate different aspects of the detection and tracking performance. The results are displayed in Table 6.

While the TOMIE dataset is composed of more data, the results show that the performance of the tracking algorithms does not match those of similar datasets. This

[1] https://github.com/FLW-TUDO/TOMIE-Dataset.

[2] https://github.com/FLW-TUDO/RGB-Camera-System.

Fig. 7 Samples of annotated images by the chosen object detector of different scenarios from our custom dataset

Table 6 Results on our validation data

Method	MOTA	IDF1	HOTA	IDs
BYTE-TRACK	0.654	0.641	0.564	778
BoT-SORT	0.672	0.667	0.58	569
SiamMOT	0.575	0.594	0.503	928

deficit could be the result of the change in observed entities compared to MOT20, as well as limitations in the dataset itself.

5 Conclusion and outlook

In this contribution, a novel framework and approach for the efficient computer vision-based tracking of multiple industrial entities was presented. Using a space of approximately 16 x 8 sqm in a warehousing environment, 52 infrared cameras and six RGB cameras mounted on the ceiling and railings of this warehouse, a tracking space was defined. In this space, six industrial entities, including small load carriers, pallets, barrels, cardboard boxes, forklifts, and a mesh box were tracked using reflective markers and tracking software using infrared tracking hardware. With this tracking setup, the herein presented TOMIE dataset was recorded, including 112, 860 frames worth of RGB images and annotation files that contain approximately 16 min of recordings, after data synchronization and filtration. The recordings were subdivided into distinct logistical scenarios, drawn from industrial applications (e.g., setting up pallets in lanes, to

be loaded into trucks). Three commonly used tracking algorithms, namely ByteTrack, SiamMot, and Bot-Sort, were applied to the herein-developed dataset, performing overall worse than on comparable state-of-the-art datasets.

While developing the recording setup, during the process of recording itself, and while evaluating the resulting data and its use, additional limitations and challenges were encountered.

5.1 Limitations and challenges

While setting up the camera network for recording, a major challenge arose while trying to mark the industrial entities in a way, in which they would be detectable and distinguishable for the infrared cameras. As previously described, the marking tape needed to be distributed along the faces of the entities in such a unique way, that they would be distinguishable by virtue of the resulting point cloud. When working with a limited amount of entities, that have large surface areas, this does typically not cause any trouble. However, applying the same approach to a multitude of entities, especially smaller ones (e.g., the small load carriers in our dataset), causes the infrared cameras to yield suboptimal tracking results.

In addition, the proximity of the entities that ought to be tracked to one another further complicated the tracking process. When the markers on the edges of one entity came too close to those of another, one or both entities tended to disappear in the tracking software, resulting in frames that provide users with no positional ground truth. However, both the ground truth and the realistic positioning of the entities in a way that resembles industrial applications are of importance.

Furthermore, the software used in the herein-presented tracking setup does not enable the tracking of human motion. The operators in the recorded tracking scenarios were therefore not tracked and came with no labeled ground truth in our dataset. The addition of such data might be of interest to researchers in the field of human activity recognition or person re-identification.

Once recorded, the data proved challenging to be interpreted for the purpose of frame-wise object detection, due to the use of multiple RGB cameras and the underlying ground truth being infrared camera-based. This is because the ground truth is calculated based on the markers on the given entity in combination with its 3D-rendered model. Using this set of data, no information is given on visual occlusion by other entities present in the recording. This results in the creation of 2D bounding boxes as a ground truth that are accurate in free space but would result in poor IoU results, when used with common object detection algorithms, which would only detect the non-occluded parts of the entities. In addition, when using more than one RGB camera, the notion of the term occlusion becomes even more complicated to deal with, as an entity that is occluded in one perspective might be entirely visible in another. This results in bounding boxes being created for entities that are entirely occluded in some perspectives, which would lead to an IoU of 0%, if the data were to be put to a test.

Subsequently, once an industrial entity were to be detected, the interest would lie in the classification and identification of said entity. While classification is in part feasible with the herein presented recording setup, the identification of specific entities,

analogous to the work presented in [46], would necessitate an altered sensor use. More specifically, this would entail the use of cameras at a level close to the ground and closer to the recorded entities, so as to capture their surface structure in more detail. This, however, might lead to further occlusions, due to camera positioning.

In addition, while handling the recorded data, synchronization problems occurred, in which the RGB and infrared frames were not overlapping as they should. The reason for this has yet to be further explored. In addition, the volume of the data that is generated using this recording setup is not to be underestimated. An efficient way of handling such large amounts of data is also of great importance, to increase the efficiency and applicability of our recording approach.

Looking back at the vision for a tracking system that was established at the beginning of this contribution, some limitations still persist. One such limitation are the above-mentioned occlusions, which occur in industrial scenarios that are uncontrollable. In addition, since this work was conducted in only a single recording environment, it is yet to be evaluated, whether the selected algorithms would perform similarly in another environment. It is also important to mention that the framework has not been used put to use in the industry thus far but has been tested (as can be seen by the results in the preceding section) in an industry-like setting. Real-time tracking of industrial entities is a widespread problem. We therefore selected the entities to be applicable to as many industries as possible. Pallets, small load carriers, barrels, and cardboard boxes are widely used and standardized. We hope to have thereby created the foundation for our vision of a holistic tracking system.

5.2 Follow-up research

Taking the limitations mentioned in the previous section and our results in general into account, we identified the following ways in which our contribution could be expanded upon:

The scenarios that were recorded could be expanded upon in terms of their diversity (i.e., different versions of the same scenarios or more scenarios to begin with) and their duration. Furthermore, the complexity of the scenarios could be increased by including a greater amount of industrial entities and a greater amount of entity classes, including human operators. The prediction accuracy per class could also further be analyzed. This value might vary per class due to class imbalances or other factors, such as differences in color (i.e., some object classes might stand out more than others due to the dark floor used in this particular example).

The way in which the industrial entities are marked with reflective tape could be analyzed once more, creating a system that would allow for a more reliable marking of a larger amount of entities. In doing so, reproducibility and result quality could be enhanced.

Finally, the tracking software that was used thus far could be replaced by a self-developed one, which could be tailored for a multi-camera setup. This tracking software might then be able to not only provide bounding boxes that would take occlusions into account but might also provide 3D bounding boxes, including information on the entity's orientation in space. The use of depth information (e.g., by virtue of RGBD cameras)

might be necessary to accomplish this task and as a next step, the deployment of our framework in the industry would be desirable to further test its real-life feasibility.

Abbreviations

CNN	Convolutional neural network
DL	Deep learning
FN	False negatives
FP	False positives
IoT	Internet of Things
KF	Kalman Filter
LSTM	Long short-term memory
MCTs	Multi-camera tracks
MLP	Multi-layer perceptron
MOT	Multi-object tracking
MOTA	Multi-object tracking accuracy
MOTP	Multi-object tracking precision
MT	Mostly tracked
MTMCT	Multi-target multi-camera tracking
RCNN	Region-based convolutional neural network
RGB	Red green blue
RL	Reinforcement learning
RNN	Recurrent neural network
SCTs	Single camera trackings
SORT	Simple online and real-time tracking
SSD	Single-shot detector
SOT	Single object tracking

Acknowledgements
In our experiments, we adopt real-world scenarios from the warehousing sector. As to ensure the validity of the herein presented scenarios, we had our colleagues at the Fraunhofer Institute for Material Flow and Logistics in Dortmund, Germany, evaluate them. We would specifically like to thank Jennifer Beuth, head of the department of warehousing logistics and IT planning at the Fraunhofer Institute for her support.

Author contributions
Jérôme Rutinowski and Hazem Youssef worked in equal proportions on the methodology, results, and overall conceptualization. Sven Franke created the scenarios, directed the recordings, and wrote the corresponding sections. Irfan Fachrudin Priyanta conducted the research for the related work section. Frederik Polachowski participated in the data analysis and provided technical support. Moritz Roidl and Christopher Reining provided assistance and supervision. All authors worked on the final manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL. This work is part of the project "Silicon Economy Logistics Ecosystem" which is funded by the German Federal Ministry of Transport and Digital Infrastructure. This work is part of the research of the Lamarr Institute for Machine Learning and Artificial Intelligence which is funded by the German Ministry of Education and Research.

Availability of data and materials
All recordings and additional data are accessible free of charge on Zenodo: https://zenodo.org/record/7849183.

Declarations

Competing interests
All authors confirm that there are no competing interests.

Received: 25 April 2023 Accepted: 6 March 2024
Published online: 22 March 2024

References

1. A. Frankó, G. Vida, P. Varga, Reliable identification schemes for asset and production tracking in industry 4.0. Sensors **20**, 3709 (2020). https://doi.org/10.3390/s20133709
2. L. Anuj, M. G. Krishna, Multiple camera based multiple object tracking under occlusion: a survey, in International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), pp. 432–437 (2017). https://doi.org/10.1109/ICIMIA.2017.7975652
3. W. Liu, Q. Bao, Y. Sun, T. Mei, Recent advances of monocular 2D and 3D human pose estimation: a deep learning perspective. ACM Comput. Surv. **55**, 1–41 (2023). https://doi.org/10.1145/3524497

4. Y. Zhan, F. Li, R. Weng, W. Choi, Ray3D: ray-based 3D human pose estimation for monocular absolute 3D localization, in Computer Vision and Pattern Recognition (CVPR), pp. 13106–13115 (2022). https://doi.org/10.1109/CVPR52688.2022.01277
5. J. Wang, S. Tan, X. Zhen, S. Xu, F. Zheng, Z. He, L. Shao, Deep 3D human pose estimation: a review. Comput. Vis. Image Underst. (2021). https://doi.org/10.1016/j.cviu.2021.103225
6. G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, F. Herrera, Deep learning in video multi-object tracking: a survey. Neurocomputing **381**, 61–88 (2020). https://doi.org/10.1016/j.neucom.2019.11.023
7. P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, L. Leal-Taixé, Motchalllenge: a benchmark for single-camera multiple target tracking. Int. J. Comput. Vis. **129**, 845–881 (2021). **381**, 61–88 (2020). https://doi.org/10.1007/s11263-020-01393-0
8. F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, J. Yan, POI: Multiple object tracking with high performance detection and appearance feature, in European Conference on Computer Vision (ECCV) Workshops, pp. 36–42 (2016). https://doi.org/10.1007/978-3-319-48881-3_3
9. H. Kieritz, W. Hübner, M. Arens, Joint detection and online multi-object tracking, in: Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1540–15408 (2018). https://doi.org/10.1109/CVPRW.2018.00195
10. D. Zhao, H. Fu, L. Xiao, T. Wu, B. Dai, Multi-object tracking with correlation filter for autonomous vehicle. Sensors (2018). https://doi.org/10.3390/s18072004
11. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2015). https://doi.org/10.1109/CVPR.2015.7298594
12. L. Leal-Taixé, A. Milan, I. Reid, S. Roth, Motchallenge 2015: Towards a benchmark for multi-target tracking (2015). https://doi.org/10.48550/arXiv.1504.01942
13. A. Milan, L. Leal-Taixe, I. Reid, S. Roth, K. Schindler, MOT16: a benchmark for multi-object tracking. arXiv (2016). https://doi.org/10.48550/arXiv.1603.00831
14. S. Tang, M. Andriluka, B. Andres, B. Schiele, Multiple people tracking by lifted multicut and person re-identification. Conference on Computer Vision and Pattern Recognition (CVPR), 3701–3710 (2017). https://doi.org/10.1109/CVPR.2017.394
15. L. Chen, H. Ai, C. Shang, Z. Zhuang, B. Bai, Online multi-object tracking with convolutional neural networks. International Conference on Image Processing (ICIP), 645–649 (2017). https://doi.org/10.1109/ICIP.2017.8296360
16. L. Ma, S. Tang, M.J. Black, L.V. Gool, Customized multi-person tracker, in Asian Conference on Computer Vision (ACCV) (2018)
17. C. Ma, C. Yang, F. Yang, Y. Zhuang, Z. Zhang, H. Jia, X. Xie, Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking. in International Conference on Multimedia and Expo (ICME), pp. 1–6 (2018). https://doi.org/10.1109/ICME.2018.8486454
18. L. Ren, J. Lu, Z. Wang, Q. Tian, J. Zhou, Collaborative deep reinforcement learning for multi-object tracking, in European Conference on Computer Vision (ECCV), pp. 605–621 (2018). https://doi.org/10.1007/978-3-030-01219-9_36
19. A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in International Conference on Image Processing (ICIP), pp. 3464–3468 (2016). https://doi.org/10.1109/ICIP.2016.7533003
20. N. Aharon, R. Orfaig, B.-Z. Bobrovsky, BoT-SORT: robust associations multi-pedestrian tracking. arXiv (2022). https://doi.org/10.48550/ARXIV.2206.14651
21. Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, X. Wang, Bytetrack: multi-object tracking by associating every detection box (2022). https://doi.org/10.1007/978-3-031-20047-2_1
22. J.H. Yoon, M.-H. Yang, J. Lim, K.-j. Yoon, Bayesian multi-object tracking using motion context from multiple objects. Winter Conference on Applications of Computer Vision (WACV), 33–40 (2015). https://doi.org/10.1109/WACV.2015.12
23. M. Tiwari, R. Singhai, A review of detection and tracking of object from image and video sequences. Int. J. Comput. Intell. Res. **13**, 745–765 (2017)
24. L. Kalake, W. Wan, L. Hou, Analysis based on recent deep learning approaches applied in real-time multi-object tracking: a review **9**, 32650–32671 (2021). https://doi.org/10.1109/ACCESS.2021.3060821
25. M. Bredereck, X. Jiang, M. Körner, J. Denzler, Data association for multi-object tracking-by-detection in multi-camera networks. International Conference on Distributed Smart Cameras (ICDSC), 1–6 (2012)
26. X. Wang, Intelligent multi-camera video surveillance: a review. Pattern Recogn. Lett. **34**, 3–19 (2013). https://doi.org/10.1016/j.patrec.2012.07.005
27. S. Zhang, E. Staudt, T. Faltemier, A.K. Roy-Chowdhury, A camera network tracking (CamNeT) dataset and performance baseline, in Winter Conference on Applications of Computer Vision, pp. 365–372 (2015). https://doi.org/10.1109/WACV.2015.55
28. A. Specker, D. Stadler, L. Florin, J. Beyerer, An occlusion-aware multi-target multi-camera tracking system, in Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 4168–4177 (2021). https://doi.org/10.1109/CVPRW53098.2021.00471
29. A. Specker, L. Florin, M. Cormier, J. Beyerer, Improving multi-target multi-camera tracking by track refinement and completion, in Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 3198–3208 (2022). https://doi.org/10.1109/CVPRW56347.2022.00361
30. C. Liu, Y. Zhang, H. Luo, J. Tang, W. Chen, X. Xu, F. Wang, H. Li, Y.-D. Shen, City-Scale multi-camera vehicle tracking guided by crossroad zones, in Computer Vision and Pattern Recognition Workshops (CVPRW), vol. 3, pp. 4124–4132 (2021). https://doi.org/10.1109/CVPRW53098.2021.00466
31. S. He, H. Luo, W. Chen, M. Zhang, Y. Zhang, F. Wang, H. Li, W. Jiang, Multi-domain learning and identity mining for vehicle re-identification, in Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 582–583 (2020). https://doi.org/10.1109/CVPRW50498.2020.00299
32. H.-M. Hsu, T.-W. Huang, G. Wang, J. Cai, Z. Lei, J.-N. Hwang, Multi-camera tracking of vehicles based on deep features Re-ID and trajectory-based camera link models, in Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 416–424 (2019)

33. P. Kohl, A. Specker, A. Schumann, J. Beyerer, The MTA dataset for multi-target multi-camera pedestrian tracking by weighted distance aggregation, in Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1042–1043 (2020). https://doi.org/10.1109/CVPRW50498.2020.00529
34. C. Mayershofer, D.-M. Holm, B. Molter, J. Fottner, Loco: Logistics objects in context, in International Conference on Machine Learning and Applications (ICMLA), pp. 612–617 (2020). https://doi.org/10.1109/ICMLA51294.2020.00102
35. P. Dendorfer, H. Rezatofighi, A. Milan, J.Q. Shi, D. Cremers, I.D. Reid, S. Roth, K. Schindler, L. Leal-Taix'e, Mot20: a benchmark for multi object tracking in crowded scenes (2020). https://doi.org/10.48550/ARXIV.2003.09003
36. A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: the kitti dataset. Int. J. Robot. Res. **32**, 1231–1237 (2013). https://doi.org/10.1177/0278364913491297
37. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: common objects in context, in European Conference on Computer Vision (ECCV), pp. 740–755 (2014)
38. K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the CLEAR MOT metrics. EURASIP J. Image Video Process. (2008). https://doi.org/10.1155/2008/246309
39. J. Luiten, A. Osep, P. Dendorfer, P.H.S. Torr, A. Geiger, L. Leal-Taixé, B. Leibe, HOTA: A higher order metric for evaluating multi-object tracking. Int. J. Comput. Vis. (2020). https://doi.org/10.1007/s11263-020-01375-2
40. E. Ristani, F. Solera, R. Zou, R. Cucchiara, C. Tomasi, Performance measures and a data set for multi-target, multi-camera tracking, in European Conference on Computer Vision (ECCV) Workshops, pp. 17–35 (2016). https://doi.org/10.1007/978-3-319-48881-3_2
41. B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, C. Steger, Introducing MVTec ITODD-a dataset for 3D object recognition in industry, in Iinternational Conference on Computer Vision Workshops (ICCVW), pp. 2200–2208 (2017). https://doi.org/10.1109/ICCVW.2017.257
42. C. Luo, L. Yu, E. Yang, H. Zhou, P. Ren, A benchmark image dataset for industrial tools. Pattern Recogn. Lett. **125**, 341–348 (2019). https://doi.org/10.1016/j.patrec.2019.05.011
43. P. De Roovere, S. Moonen, N. Michiels et al., Dataset of industrial metal objects (2022). https://doi.org/10.48550/ARXIV.2208.04052
44. C. AbouAkar, J. Tekli, D. Jess, M. Khoury, M. Kamradt, M. Guthe, Synthetic object recognition dataset for industries, in SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), vol. 1, pp. 150–155 (2022). https://doi.org/10.1109/SIBGRAPI55357.2022.9991784
45. F. Niemann, C. Reining, F. MoyaRueda, N.R. Nair, J.A. Steffens, G.A. Fink, M. ten Hompel, LARa: creating a dataset for human activity recognition in logistics using semantic attributes. Sensors **20**(15) (2020). https://doi.org/10.3390/s20154083
46. J. Rutinowski, T. Chilla, C. Pionzewski, C. Reining, M. ten Hompel, Towards re-identification for warehousing entities—a work-in-progress study, in Emerging Technologies in Factory Automation (ETFA), pp. 501–504 (2021). https://doi.org/10.1109/ETFA45728.2021.9613250
47. J. Rutinowski, C. Pionzewski, T. Chilla, C. Reining, M. ten Hompel, Deep Learning Based Re-identification of Wooden Euro-pallets, in International Conference on Machine Learning and Applications (ICMLA) (2022)
48. DIN: DIN 55405:2014-12, Packaging—Terminology—Terms and definitions (2014)
49. DIN: DIN EN 13698-1:2004-01, Pallet production specification—Part 1: construction specification for 800 mm × 1200 mm flat wooden pallets (2004)
50. L. Campagnola, E. Larson, A. Klein, D. Hoese, Siddharth, C. Rossant, A. Griffiths, N.P. Rougier, L. van Dijk, K. Mühlbauer, et al., vispy/vispy: Version 0.9.5. Zenodo (2022). https://doi.org/10.5281/zenodo.5974509
51. B. Adhikari, J. Peltomäki, J. Puura, H. Huttunen, Faster bounding box annotation for object detection in indoor scenes. European Workshop on Visual Information Processing (EUVIP) (2018). https://doi.org/10.1109/EUVIP.2018.8611732
52. B. Shuai, A.G. Berneshawi, D. Modolo, J. Tighe, Multi-object tracking with siamese Track-RCNN (2020). https://doi.org/10.48550/ARXIV.2004.07786
53. S. Zhang, F. Wang, L. Songtao, G. Zheng, YOLOX: Exceeding yolo series in 2021 (2021). https://doi.org/10.48550/arXiv.2107.08430

Publisher's Note