Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 51 (2020) 116–124

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021)
15-18 June 2021, Athens, Greece.

Low-cost Scalable People Tracking System for Human-Robot Collaboration in Industrial Environment

Matteo Terreran[a,*], Edoardo Lamon[b,c], Stefano Michieletto[a], Enrico Pagello[a]

[a] *Intelligent Autonomous Systems Laboratory (IAS-Lab), Dept. of Information Engineering (DEI), University of Padova, Padova, Italy*
[b] *Human-Robot Interfaces and Physical Interaction (HRI^2), Istituto Italiano di Tecnologia (IIT), Genova, Italy*
[c] *Robotics and Automation, Dept. of Information Engineering, Università degli Studi di Pisa, Pisa, Italy*

Abstract

Human-robot collaboration is one of the key elements in the Industry 4.0 revolution, aiming to a close and direct collaboration between robots and human workers to reach higher productivity and improved ergonomics. The first step toward such kind of collaboration in the industrial context is the removal of physical safety barriers usually surrounding standard robotic cells, so that human workers can approach and directly collaborate with robots. Anyway, human safety must be granted avoiding possible collisions with the robot. In this work, we propose the use of a people tracking algorithm to monitor people moving around a robot manipulator and recognize when a person is too close to the robot while performing a task. The system is implemented by a camera network system positioned around the robot workspace, and thoroughly evaluated in different industry-like settings in terms of both tracking accuracy and detection delay.

Keywords: People tracking; human-robot collaboration; low-cost industrial

1. Introduction

Industry 4.0 is the next industrial revolution aiming at the massive introduction of intelligent systems, leading to smart factories that assist people and machines in execution of their tasks. Among its design principles, this new paradigm goes beyond traditional factory robotics and envisions close/direct collaboration between robots and human workers that could lead to higher productivity and improved ergonomics because of the synergy between human intelligence and mechanical power. In the industrial context, robots are considered as a potential source of danger. Standard robotic cells have a fixed barrier to prevent human contact with machines. A first step toward human-robot collaboration is to remove any physical system separating the working environment of humans and robots, decreasing the amount of installation space and costs for safety barriers, but this leads to the issue of safety. In the simplest type of collaboration, a robot shares part of its workspace with workers: they work in the same space but not at the same time. When workers and robots operate simultaneously, accidental collisions between them must be avoided. In this work, we address human-robot collaboration and focus on guaranteeing safety of the human operators by developing a system to detect when a human enters the robot's workspace; in that case the machine is stopped until the worker leaves. Such system is based on a previous paper that track and monitor people in the working area of a robot manipulator as illustrated in Figure 1. Now, when a person gets too close to the robot and enters its reachable workspace, the system broadcast this information to the robot to take into account the presence and the position of the person. Based on such information the robot can react in a proper way to avoid collisions with the human operator. This simple behavior is designed to be robust and fast with the specific aim of demonstrating the performance of the people tracking algorithm with an industrial-like setting. The previous algorithm has been adapted to work for lost-cost embedded devices using several cameras to cover a large space populated by up to 10 people. The system has been developed within the framework of the EuRoC project for reconfigurable interactive manufacturing cells. We also investigate how to evaluate the performance of our system, proposing three different metrics to measure both tracking accuracy and detection delay.

* Corresponding author. Mail: matteo.terreran@dei.unipd.it

2351-9789
10.1016/j.promfg.2020.10.018

Metrics are computed comparing the tracking output of the system with ground truth data obtained with our novel annotation procedure. Together with tracking output data, we acquired RGB frames from one of the cameras in the camera network. Each frame is then annotated with bounding boxes around each person in the frame to get their position in the scene. Annotation has been speed-up by using Piotr's Matlab Toolbox annotation tool [7] that we extended to consider not only people position in the scene but also if they are or not inside the robot workspace. Summarizing, the work presents 3 main contributions: (i) a low cost people detection and tracking system working in real time, composed of several agents in a network, scalable and suitable for industrial scenarios; (ii) a monitoring system based on virtual barriers that identifies dangerous interactions between humans and robots and stops the robot movement as preventive measure; (iii) an annotation system to benchmark the performance of the proposed framework. The remainder of the paper is organized as follows. Section 2 reviews the works related to people tracking and people re-identification. In Section 3 the algorithms and solutions adopted to validate the results are described, while in Section 4 the proposed system is thoroughly evaluated in two different setups. Finally, in Section 5, conclusions are drawn and future directions of research identified.

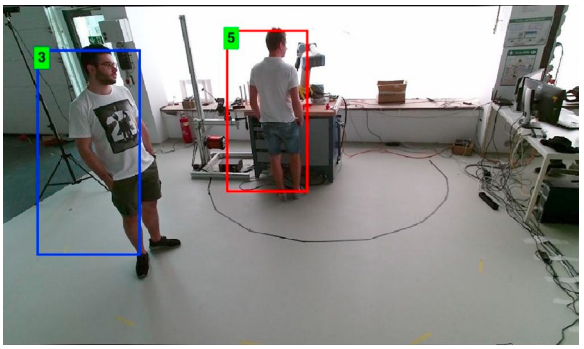

Figure 1. The proposed system in action. People inside the danger zone (shown in red) are correctly recognized by the system.

2. Related works

Detection and tracking of people in real-time are a key information for several applications. The main objective is to identify the presence of people in complex real world scenes with occlusions and cluttered or even moving backgrounds. Algorithms can rely on different technologies to achieve the task of detecting human in the environment depending on the scope. Considering visual data enables a wider set of applications and they have been largely studied as main or exclusive source of information. In particular, people detection is at the basis of various important and more general tasks like surveillance of sensible areas [23], domotics and smart homes [27], assistance in activities of daily living [5], pedestrian interaction for advanced driving and autonomous cars [9], analysis of sport activities [15]. Many works exist about people detection and tracking by using RGB images or range data only [3][12]. Using depth information could help reducing problems related to occlusions, varying in the scene illumination, and crowded environments [11][24]. Depth data can be calculated by using a stereo approach, but the computations needed for creating the disparity map impose limitations to the maximum frame rate achievable, especially when further algorithms have to run in the same CPU. A low-cost alternative is represented by the recently introduced RGB-D sensors such as the Microsoft Kinect. These sensors combine appearance and distance information by providing in real-time both high resolution RGB images and robust depth data with the drawback of a range limited to some meters and reduced usability in direct sunlight. Once the data source is

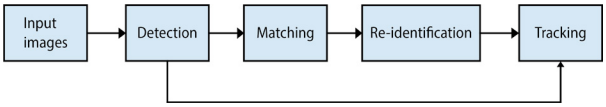

Figure 2. Process pipeline usually applied in people detection and tracking algorithms.

defined, the process pipeline is usually based on the steps summarized in Figure 2. The detection uses the data captured from the sensors in order to estimate in a static way the presence of people in the scene. In order to do that, there are different approaches, like background subtraction [22] [8] or dense scanning of RGB-D images [19] [13]. Actual and past detection data are used in a stochastic dynamic model which is able to improve the detection result and predict future values. Classical tracking approaches are Kalman [10] [14], bayesian [12], or particle [1] filters. The re-identification technique used in order to recognize if a detected person was tracked previously could be short or long term. Short term recognition is based on global features like color and shape histograms [6] or local features like 2D/3D keypoints detector [16] or skeleton keypoints [21], while long term recognition is based on skeleton lengths [2], face and point cloud shape [25]. Recent alternatives use end-to-end approaches relying on the direct estimation of people presence from a single image, and therefore removing the pipeline described so far [20] [4]. While all the previously cited approaches lead to people detection and tracking, not all of them could be applied in industrial scenarios. The long and complicated calibration procedures used in [8] [14] [25] cannot be easily used in real settings. In other cases, they are not easily scalable to large spaces like for [6] [10] [22]. Another limit is related to the real-time capabilities that are difficult to achieve when working with dense RGB-D data ([2] [16] [19]) or they can be obtained only with high-performance costly hardware ([20] [4] [21] [24]).

Detection and tracking are crucial elements for human-robot collaboration (HRC), since the robot must be capable to perceive what is happening around it in order to identify potential dangerous situations for the human workers. In [18] the distance between the robot and the human worker is constantly monitored by using an RGB-D camera and skeletal tracking. In [26] a multi-camera system is used to perceive the working area of a robot by tracking worker's hand. Such approaches are feasible when either small workcells are considered or just one hu-

man worker needs to collaborate with the robot. For large workcells, many workers may be operating close to the robot and people tracking systems are more suitable to correctly track all the people in the work area in the fastest way. Similar capabilities are available in commercial products like SafetyEYE[1], having the advantage to be ready for integration in production line and designed for applications up to Cat. 3 of EN ISO 13849-1:2008, SIL2 of IEC 61508, PL d of EN ISO 13849-1 and applications in accordance with DIN EN 61496. On the other hand, our system is built upon a camera network of multiple cameras with no restriction on the camera models, which offers a scalable solution, increase the area monitored by the system and help reducing problems related to occlusions. As a drawback, our system is not yet ready for production integration and we relied on the use of lightweight collaborative robots to comply to international safety standards, using the camera network for improving the safety level.

In a previous work [17], we presented a suitable scalable and distributed multi-camera people tracking system able of supporting a heterogeneous set of 3D sensors (e.g. Asus Xtion, Microsoft Kinect One, Mesa SwissRanger). In this paper, we improved the previous work by using the tracking information in order to model the interaction between human and robot. The first objective is to improve the safety of the human in an usually hostile environment by avoiding collisions. The 3D centroids and 2D bounding boxes defining the position of people in the scene has been extended to 3D solids to consider the actual space occupied by each person. Different shapes have been considered to better understand when people is entering a dangerous area. We develop a set of GUI tools in order to define the danger zone around the robot and annotate frame to compute valuable metrics to measure the performance of the system with the aim of meeting real industrial requests from end-users in the EuRoC project.

3. Methods

We developed our system building upon a previous work. In particular, we rely on OpenPTrack [17] to track people in the scene and on its user-friendly calibration procedure to quickly calibrate the camera network system. On top of that, we add the possibility to define the area around the robot, which we named danger zone, with respect to the coordinate system of the camera network. In such a manner the system can monitor the people moving in the area covered by the camera network, and detect when a person gets too close to the robot manipulator entering the danger zone. We also proposed a pipeline to manually annotated ground truth data needed to evaluate the performance of the system.

An overview of our proposed system is depicted in Figure 3, summarizing the main steps involved. The first step is the calibration of the whole system, hence both the camera network system using the calibration pipeline provided in [17] , and the

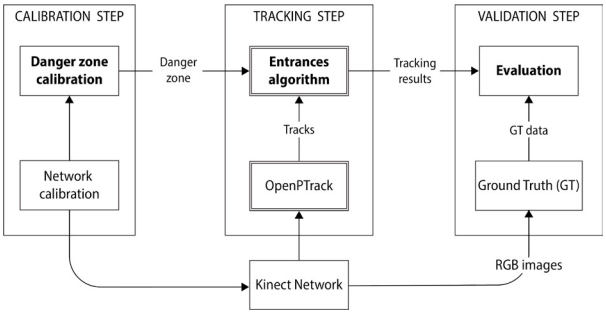

Figure 3. Overview of our proposed system, describing the main steps and the relation between them.

position of the danger zone with respect to the camera network coordinate system. The second step regards the online working of the system: the software runs on the calibrated camera network tracking all the people moving in the area covered by the cameras. For each person tracked by the system, the framework provides as output a track representing the position of the person inside the camera network over time. Each track is given as input to the entrance algorithm which analyzes if the track is inside the danger zone, whose position and width is given by the former calibration step. In the last step, tracking data and RGB images from one of the cameras are recorded to evaluate the system later on. RGB images are required to obtain a ground truth with which compare the tracking results after a manual annotation phase. Each component is described in detail in the remainder of this section.

3.1. Camera network system

OpenPTrack [17] allows to perform people tracking within a network of RGB-D sensors by distributing people detection and centralizing the tracking process; each sensor is directly attached to a computer which analyzes the data stream and performs people detection. Only the detections are sent through the network, in order to be merged at the tracking level after being referred to a common reference frame by means of calibration data, describing the pose of each camera within the network. Calibration data are obtained by means of a calibration procedure with checkerboards developed in ROS[2], the Robot Operating System. RGB-D data from each sensor are converted to point cloud and processed by the detection module running on the corresponding computer.

The point cloud is filtered and the points belonging to the ground removed; note that ground plane equation is computed in the network calibration procedure. The remaining points are divided in 3D clusters. A HOG-based people detection algorithm is applied to the projection onto the RGB image of the 3D clusters (extended to the ground) and also a Support Vector Machine (SVM) in order to keep only those clusters that are more likely to belong to people. The resulting output is a set of detections that are then passed to the tracking module.

[1] https://www.pilz.com/en-INT/eshop/00106002207042/SafetyEYE-Safe-camera-system

[2] https://www.ros.org

The tracking module is a centralized algorithm which receives detections from all over the network and performs data association every time a new set of detections arrives. This is done as a maximization of a joint likelihood composed by motion information and people detection confidence. An Unscented Kalman Filter (UKF) is used to predict people positions along the two world axes (x, y) using as motion term the Mahalanobis distance between tracks predicted positions and detection positions. Finally, the people detection confidence is also used for robustly initializing new tracks when no association with existing tracks is found. For further details about the people tracking system, please refer to [17].

3.2. Danger zone calibration

We denote as danger zone the reachable workspace of the robot manipulator, that is the volume the robot can reach during its movements. For simplicity, we assumed that the danger zone is approximately cylindrical with a circular base and centered on the base frame of the robot manipulator. With this assumption, we can describe the danger zone with only two parameters: the radius of the base circle and the position of its center with respect to the camera network coordinate system. With the term danger zone calibration we mean the problem of estimating the values of such parameters with respect to the camera network coordinate system. The world frame of such coordinate system is not a point in the real scene, but depends on the last checkerboard pose used during the camera network calibration. Therefore, is not possible to measure directly the position of the danger zone center and we needed to developed a calibration procedure which can be easily adopted in various scenarios. First, we drew on the floor the danger zone perimeter with some tape: we stretched out the robot arm and moved it around, keeping it parallel to the ground, while drawing on the floor the projection of the end effector. Drawing the whole danger zone it is not mandatory, only few points are sufficient; we chose to draw the whole perimeter in order to facilitate the manual annotations later on.

Consider now a circle of radius equals to the total length of the robot arm, centered in the world frame of the camera network coordinate system. This circle is defined in 3D coordinates with respect to the camera network coordinate system, but we can easily project this circle onto the image plane of one of the cameras thanks to the former camera network calibration. In particular, we sample the circle in 8 points and project each point onto the camera image plane; interpolating then those points gives an ellipsoid in the RGB image seen by the camera, as depicted in Figure 4. Using the RGB image provided by the camera, in which are visible both the real danger zone perimeter and the projected circle, it is possible to modify the danger zone parameters (i.e radius and center position) to make it overlap the line (or the points) drawn on the floor.

The procedure described has been implemented with a ROS node which let the user calibrate the danger zone for the real scenario at hand in a graphical way. The RGB image from one of the cameras in the network is shown, together with some trackbars which indicate the values for the danger zone parameters. Moving the trackbars, the user can modify the value of the danger zone parameters and the circle projected onto the image plane of the camera is updated according to the new values to give the user a visual feedback, as in Figure 4. When the projected circle overlaps the points drew on the floor with the tape, the calibration is terminated and the parameters values of the danger zone are saved to be used later on by the algorithm for people entrances detection.

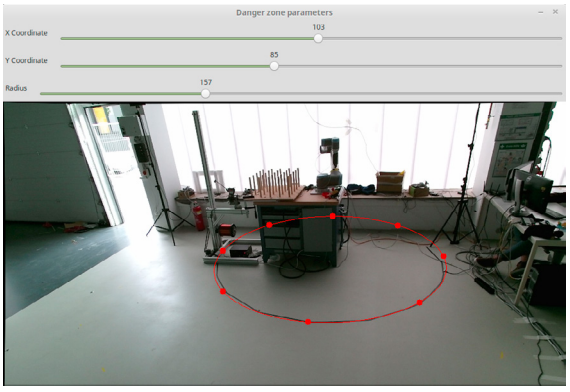

Figure 4. Danger zone calibration. A window let the user tuning the danger zone parameters until is aligned with the tape on the floor.

3.3. Entrance detection algorithm

Among the tracking data output by our framework, there are the centroid coordinates with respect to the world frame for each track found. Using this information and the danger zone parameters found with our calibration procedure, it is possible to determine when a person is inside the danger zone. As we assumed that the danger zone has a circular shape, a person is inside this area if the distance d between the track's centroid $\mathbf{O}_{TR}$ and the center of the danger zone $\mathbf{O}_{DZ}$ is lower than the danger zone's radius r_{DZ}, and outside otherwise.

Since humans can be modelled as points only as first approximation, we improved our algorithm by taking into account also the volume of a person. Therefore, we consider a bounding box around the track's centroid to take into account the body of a real person walking in the scene. Indeed, with the bounding box, we can detect people inside the danger zone when they just put a foot inside the area and not only when their centroid is inside. In such a manner, our algorithm is more suitable for safety purpose. Using a cylindrical bounding box of radius r_{BB} centered on the track's centroid, now we can consider a person as inside the danger zone when the bounding box starts to overlap the danger zone area. The bounding box's radius can be modified and adapted to the real scenario on hand. We find out that a reasonable value for our application is $r_{BB} = 0.20$ meters.

Using a circular bounding box, we are considering the motion in each direction with the same probability with respect to another, that is each direction has the same weight a priori. But actually, while walking people have a preferential direction along which they move, and it depends on the previous

positions (i.e. the trajectory is usually a continuous function). Therefore, our algorithm may overestimate the person's volume and consider a person as inside when the human is walking close to the danger zone but not entering it. From this observation, we tried to use a bounding box with an ellipse as base, with the major axis aligned with the movement direction of the track. The biggest challenge of this approach is the robust computation of the ellipse orientation. We used previous tracking data for each track and compute the angle θ between the major axis of the ellipse and the $x-$axis of the world frame as

$$\theta = Atan2\left(y_{new} - y_{old},\ x_{new} - x_{old}\right)$$

where (x_{new}, y_{new}) are the actual centroid's coordinate w.r.t world frame, (x_{old}, y_{old}) are the previous centroid's coordinate w.r.t world frame and $Atan2$ is the function which computes the arctangent of the arguments, taking into account their signs for choosing the right value of angle. In this way, we are able to align the major axis of the ellipse with the direction of travel of each track. This approach relies on the availability of previous tracking data. During our first tests with this approach, we observed that frequently there were some losses and delays in the tracking data that led to a wrong computation of the ellipse orientation. On average using the ellipse was less robust than using a circle as base of the cylindrical bounding box, so at the end we decided to use a cylindrical bounding box with circular base.

3.4. Ground truth data annotation

To evaluate the performance of our system, it is mandatory to associate data with a set of accurate measurements with which compare the system output: the *ground truth*. The approach we choose is to record RGB frames from a camera which captures most of the scene in its Field of View (FoV). Then, for each

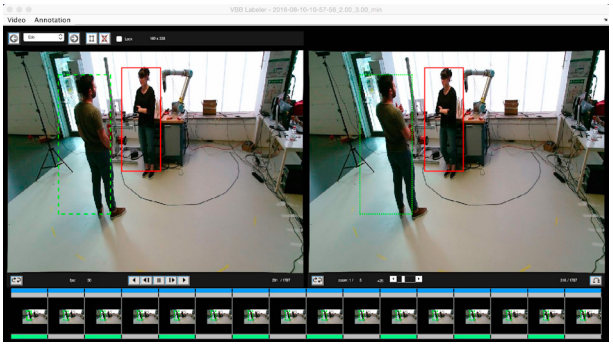

Figure 5. User interface of the annotation tool VBB Labeler.

frame, we annotate the position of every person in the working area using a rectangular bounding box as shown in Figure 5; the coordinates and the dimensions of those annotations will be our ground truth. To simplify the manual annotation phase, we used Piotr's Matlab Toolbox [7] and a Matlab labeling tool named "VBB Labeler", both from Caltech University. The first one defines a sequence file as a series of concatenated image frames with a fixed size header and provides Matlab routines for reading/writing/manipulating seq files. The latter uses a custom "video bounding box" (`vbb`) file format for annotations, it contains utilities to view seq files with annotations overlaid and provides an easy way to note each frame. A video bounding box (`vbb`) annotation is simply a Matlab structure and stores bounding boxes (`bbs`) of objects of interest. The primary difference from a static annotation is that each object can exist for multiple frames, that is a vbb annotation not only provides the locations of objects but also tracking information.

The annotation tool allows to efficiently browse and annotate a video in a minimum amount of time. Its most salient aspect is an interactive procedure where the annotator labels only a sparse set of frames and the system automatically predicts person positions in intermediate frames. Specifically, after an annotator labels a bounding box (BB) around the same person in at least two frames, BBs in intermediate frames are interpolated using cubic interpolation (applied independently to each coordinate of the BBs). Thereafter, every time an annotator alters a BB, BBs in all the unlabeled frames are reinterpolated. An example of the User Interface (UI) of this tool is depicted in Figure 5. At the bottom of the UI there is the sequence of frames composing the loaded sequence; frames mark as green are the one chosen to be interpolated. For every frame in which a given person is present in the scene, annotators must mark a BB that depicts the full extent of the person; if the person is occluded or partially occluded, annotators need to indicate this fact using the bar under the blue bar for each frame in which the person is occluded. The annotation tool described allows us to efficiently annotate a video but don't provide a way to indicate when a person is inside the danger zone or not. Therefore, we tried to adapt the existing tool in a simple and smart way. We saved two different `vbb` files where the first contains the annotations with the information about occlusions. The second one instead, contains the same annotations and uses the bar reserved for occlusions to point out if a person is inside or not.

3.5. Evaluation

Ground truth data are obtained by manual annotation of the frames from one camera with 2D bounding boxes, but we cannot directly compare them with the tracking data output by the system, since those are generally expressed as 3D coordinates with respect to the camera network coordinate system. Actually, among the tracking data of each track there are also 2D bounding boxes for each detected person but they are not expressed with respect to the same camera. Indeed, when the tracker merge the detections coming from the sensors, it produces a track data which contains information of its 2D bounding box with respect to the last camera which sent its detection. Because of that, we have not guarantees about which camera the track's bounding boxes are referred to and so we cannot compare them with the ground truth ones, which are computed with respect to the same camera. Therefore, we have to project the 3D information for each track with respect to the same camera

used for annotating the ground truth, which we denote Master camera from now on, and compute a 2D bounding box for each person detected. Each track published by the system contains the centroid coordinates with respect to the world frame and the estimated height of the person which the track corresponds to. Starting from that information, we built up a bounding box 2D with respect to the Master camera frame, using the same approach used in danger zone calibration based on 2D reprojection. So, assuming that the centroid has coordinates (x, y, z) and h is the estimated height, we consider two 3D points $\mathbf{C}_{top} = (x, y, h)$ and $\mathbf{C}_{bottom} = (x, y, 0)$. We project those points and the centroid on the Master camera frame; starting from the 2D points obtained we build up our bounding box 2D: the difference between y-coordinates of top and bottom will be the height of the bounding box and as width we consider half the height.

All this was added to a ROS node which implements the entrance algorithm and listens to the tracking results provided by the camera network, publishing a new message containing the information we need. So, for each track present in the tracking results, it computed if the person is inside the danger zone and a 2D bounding box as described earlier.

An issue which revealed to be not so trivial was how to associate, when possible, each ground truth track with its corresponding track in the tracking results and determine when it is not possible. For solving this problem we use an algorithm well-known in literature, the Munkres algorithm. This algorithm (also called *Hungarian algorithm*) is a combinatorial optimization algorithm that solves the assignment problem in polynomial time. The assignment problem requires to associate pairs of elements taken by two different sets; taking into account that every couple has an associated cost, we want to minimize the total cost. Formally, denoting by c_{ij} the associated cost of couple (i, j) and given a $n \times n$ matrix $C = [c_{ij}]$, we want to find a permutation ϕ of $\{1, \ldots, n\}$ that minimizes the cost function $\sum_{i=1}^{n} c_{i\phi(i)}$. In our case, we need to solve an assignment problem for each frame left after the selection. The two different sets for a single frame are represented by the person individuated in ground truth and the tracks given by the tracker algorithm. Denoting respectively by n and m the sizes of those sets, the cost matrix is a $n \times m$ matrix $C = [c_{ij}]$ where each element c_{ij} takes into account how good is the match for a couple of bounding boxes taken from the two sets. A reasonable choice for evaluating this match is to use a percentage computed as the ratio of overlap area to union area of the two bounding boxes; the closer this value is to 1, the better is the match between bounding boxes. Since we want to minimize the cost function $\sum_{i=1}^{n} c_{i\phi(i)}$ we choose as cost c_{ij} of each couple the quantity:

$$c_{ij} = 1 - \frac{A_{GT} \cap A_{TR}}{A_{GT} \cup A_{TR}}$$

where A_{GT} and A_{TR} are a couple of bounding boxes and i, j their indices in the corresponding set.

4. Experiments

The system presented up to this point is capable of tracking people inside the working area and detecting when a person enters the danger zone around the robot manipulator. We tested the whole system in two different indoor environments: a mock-up environment created in our laboratory and in the real working area in Stuttgart. In both testbeds, the camera network system was composed of 4 Kinect One[3]. Each Kinect One has been connected to a low cost embedded device capable of performing on the edge computing, namely an Nvidia Jetson TX1. The code in [17] has been adapted and compiled on the boards flashed with Ubuntu 14.04 and equipped with an external PCIe USB 3.0 card to avoid bandwidth limitations. Every Jetson TX1 board was part of a LAN to realize the camera network system. The total cost of the system is around 1500 euros considering the total amount a dedicated Ethernet network and the material (e.g. tripods, supports, adapters, cables) required for covering an area of 90 m^2 with limited occlusions.

4.1. Metrics

To evaluate the performance of our system, we propose three different metrics which measure tracking accuracy and precision of the whole system with respect to manually annotated ground truth. As required by the project in order to achieve industrial requests, for each metric we set a baseline and target value as summarized in Table 1. Baseline represents the minimum value to be achieved by our system, while target is the result we aim to achieve for the selected scenario. The first metric,

Table 1. Metrics considered and target values.

Metric	Target	Baseline
Metric I	$FN < 10\%$	$FN < 25\%$
Metric II	$FN < 2\%$	$FN < 5\%$
Metric III	Mean delay < 0.25 *sec*	Mean delay < 1.5 *sec*

Metric I, measure the frame-wise tracking accuracy by computing the percentage of False Negatives (FN). A False Negative occurs when the tracking algorithm does not recognize a person inside the scene. For each frame we have the number of people tracked by the algorithm and the number of people inside the scene from the ground truth manually annotated. As described in Section 3.5, we associate tracking and ground truth data according to the bounding boxes and, after the association step, we consider a correct detection if the ratio of overlap area to union area is greater than 50%. Let's denote N_f the total number of frames, we compute the metric as:

$$FN(\%) = \frac{\sum_{i=1}^{N_f} P_{scene}(i) - \sum_{i=1}^{N_f} P_{match}(i)}{\sum_{i=1}^{N_f} P_{scene}(i)} \cdot 100 \qquad (1)$$

[3] https://en.wikipedia.org/wiki/Kinect

where for each frame i, $P_{scene}(i)$ is the number of people in the scene and $P_{match}(i)$ is the number of correct detections. The baseline value of $FN < 25\%$, has been chosen according to state of the art performance in industrial environment with occlusions.

The second metric, *Metric II*, measure how many times a person is correctly detected as entering the danger zone within 2 seconds, as percentage of False Negatives. Compared to the previous metric, which depends on the camera network system accuracy, Metric II measures the accuracy of the algorithm to detect if a person is inside the danger zone or not. Therefore, instead of computing False Negatives for all frames and all people in the scene, to compute Metric II we need to look only at frames in which people enter the danger zone. From both ground truth and tracking data it is possible to compute how many people are inside the danger zone in each frame. Let's denote $P_{in}^{GT}(i)$ and $P_{in}^{TR}(i)$ as the number of people inside the danger zone at frame i for ground truth and tracking data respectively. Intuitively, P_{in} increases when a person enters the danger zone and decreases when the person leaves it, therefore the quantity $E(i) = P_{in}(i) - P_{in}(i-1)$ is positive for entrances, negative for exits and zero otherwise, that is when the number of people inside the danger zone does not change between frame $i-1$ and frame i. In this way we are able to highlight frames in which particular events occur by simply looking at the value taken by $E(i)$ for frame i. Starting from this observation, we compute the following quantities for each frame $i = 1 \dots N_f$,

$$E^{GT}(i) = P_{in}^{GT}(i) - P_{in}^{GT}(i-1)$$
$$E^{TR}(i) = P_{in}^{TR}(i) - P_{in}^{TR}(i-1)$$

and we consider a person as correctly detected if, given an entrance in the ground truth data, there is also an entrance in the tracking data within 2 seconds. That is, given the frame index i, such that $E^{GT}(i) > 0$ and an interval I, we search if there is a index $j \in I$ such that $E^{TR}(j) > 0$. The interval $I = [i_{min}, i_{max}]$ is chosen in order to cover an interval of two seconds, possibly centered on the timestamp of frame i. In general, it can be expected that the entrance view by the tracking algorithm happens later than that indicated in ground truth. But as explained before, the use of the bounding box can lead in certain cases to an overestimation and therefore the entrance in tracking data comes before the same entrance in ground truth. Because of that we decided to look before and after the timestamp of frame i. So we choose

$$i_{min} = max\{t_1, t_i - 1\}$$
$$i_{max} = min\{t_{N_f}, i_{min} + 2\}$$

where t_i is the timestamp of frame i expressed in seconds. In order to keep the interval length of 2 seconds, we put $i_{min} = (i_{max} - 2)$ when $i_{max} = t_{N_f}$. Let's denote $E_{correct}$ the sum of correct entrances detections and E^{GT} the sum of entrances in Ground Truth, that is the sum of only positive terms $E^{GT}(i)$ for $i = 1 \dots N_f$. Finally, we compute the metric required as

$$FN(\%) = \frac{E^{GT} - E_{correct}}{E^{GT}} \cdot 100 \qquad (2)$$

In the last metric, *Metric III* we measure with how much delay a person is correctly detected as entering the danger zone. To compute the previous metric we select only the frames in which an entrance occurs, that is all the frames i such that $E^{GT}(i) > 0$. For each of them we find out the corresponding frame j in which also the tracking algorithm detects the entrance within 2 seconds. The delay between them is simply the difference between the ground truth and tracking data timestamps, namely $d_i = t_j - t_i$. Therefore the third metric is computed as the mean of all d_i's obtaining the mean delay of the entrance detection algorithm. As illustrated before, in some cases the tracking algorithm moves up the entrance detection, leading to a negative value for the delay d_i; for this reason we compute the mean delay as the mean of the absolute value of each entrance detection delay d_i. Here, the baseline value chosen, mean delay < 1.5 *sec*, is lower than the typical delay in crowd counting.

4.2. Lab testbed

The space used as working area in our laboratory is depicted in Figure 6. It represents a very challenging environment due to different causes which negatively affect tracking performances, for example the sunlight coming through the windows or the metal closets which give many reflections. On the other hand, however, we could change the position of the cameras freely. We perform four experiments by changing the number

Figure 6. Experimental testbed in our lab. The red circle on the floor represents the danger zone obtained in the calibration step.

of people walking in the working area. In all the experiments the robot manipulator was moving, performing a repetitive trajectory while people were entering carefully the danger zone several times. The results obtained are summarized in Table 2.

Table 2. Results obtained in our laboratory testbed.

	Run 1	Run 2	Run 3	Run 4	mean
Metric I	24.45%	13.35%	6.67%	6.06%	12.63%
Metric II	0.00%	0.00%	0.00%	0.00%	0.00%
Metric III	0.361s	0.216s	0.088s	0.489s	0.289s

In all the experiments the system performs better than the baseline chosen according to the state of the art for people tracking algorithm. The value of 0.00% for Metric II in all the experiments shows that all the people entering the danger zone are correctly detected.

4.3. Stuttgart testbed

The testbed in Stuttgart was smaller than the one in our lab, as it is shown in Figure 7. A main difference with respect to previous testbed is that in our lab we were free to place the cameras where we wanted to, while in the Stuttgart testbed we had to comply with the limited setup and place the camera network system inside the assigned area.

This leads to some difficulties in setting up the camera network system, since a smaller area forces to place the cameras closer to each other, thus decreasing the overlap between the various field of view of the cameras and the tracking performance. Several attempts were necessary before finding an acceptable camera network configuration, in which the cameras are placed along two sides of the assigned area.

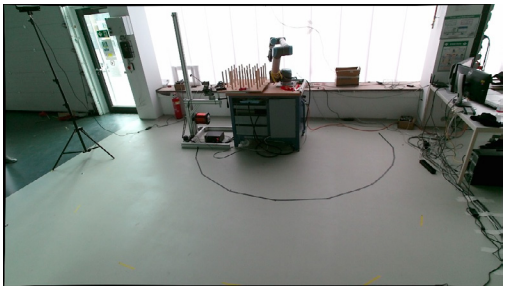

Figure 7. Experimental testbed in Stuttgart. The black tape on the floor represent the danger zone, with the robot manipulator at its center.

As required, we recorded a whole day of work and then an external supervisor selected 5 sequences of 1 minute each. Each sequence contains both the tracking results provided by the overall system and the RGB frames recorded by the camera directly connected to a master PC. We manually annotated the RGB frames corresponding to each sequence as previously described, and evaluate the tracking results according to the three metrics.

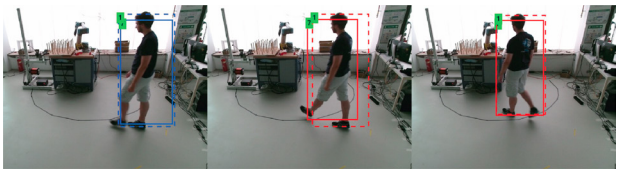

Figure 8. An example of the evaluation phase. In each frame the ground truth bounding boxes (dotted lines) are compared with the bounding box computed from the tracking data (solid lines). Blue and red colors represent detections inside or outside the danger zone respectively.

Table 3. Results obtained in Stuttgart testbed.

	Run 1	Run 2	Run 3	Run 4	Run 5	mean
Metric I	12.64%	0.77%	3.88%	0.96%	0.32%	3.91%
Metric II	0.00%	0.00%	-	0.00%	0.00%	0.00%
Metric III	0.217s	0.133s	-	0.422s	0.133s	0.226s

An example of the evaluation phase is depicted in Figure 8, while the results obtained in the evaluation phase are summarized in Table 3. As shown in Table 3, for each sequence the system performed better than the baseline and, in many case, it performed also better than the target as well. As for Metric II in which we obtained 0.00% for all the experiments demonstrating the high accuracy of our entrance detection algorithm. Note that in Run 3 there is no values for Metric II and III, since in the sequence selected by the external supervisor there were no people entering the danger zone.

5. Conclusions

In this work, we proposed a novel system based on a camera network system, capable of monitoring people in the working area around a robot manipulator. The system detects when a person enter the robot workspace and broadcast this information to the robot in order to avoid collisions by stopping its motion. The innovation of the work is threefold. First, a simple and flexible procedure to define the danger zone with respect the camera network coordinate system has been introduced. Second, we proposed a benchmark procedure to evaluate the system composed of 3 different metrics to measure tracking accuracy, accuracy in entrance detection and detection delay. Third, we developed an annotation tool and described in detail our evaluation pipeline to associate the tracking results with the ground truth annotations related to easy-to-capture RGB data. The proposed system has been evaluated in two different experimental setups, showing in both cases good performance in terms of tracking accuracy and detection delay. We consider this work as a solid starting point for safe human-robot interaction, providing clear and comparable metrics for understanding if people detection and tracking systems are capable of the robust and real-time performance for being introduced in demanding industrial scenarios. It contributed with an easy-to-use tool for ground-truth annotations working for generic applications. Compared to other approaches or commercial products, our system is a low-cost, flexible and scalable solution. Further industrialization of the current solution could lead to a reduced cost of the hardware. Anyway, more extensive considerations are necessary to understand the engineering cost to bring the Technology Readiness Level (TRL) 5/6 aimed during the EuRoC project to an industrial application (TRL 8/9). Particular attention is necessary to pair this system with not lightweight collaborative robots in order to deal with safety measures required by the ISO/TS standard 15066 for safe human-robot collaboration. For sure, this point represents a very interesting yet challenging future research direction. In the future, we also plan to further improve the system with skeletal tracking to better perceive human movements and predict their future actions, in order to allow the robot not to just stop, but coordinate with the human worker by slightly modifying its trajectory.

References

[1] Aguirre, E., García-Silvente, M., Pascual, D., 2016. A multisensor based approach using supervised learning and particle filtering for people detection and tracking, in: Robot 2015: Second Iberian Robotics Conference, Springer. pp. 645–657.

[2] Almazan, E., Jones, G., 2013. Tracking people across multiple non-overlapping rgb-d sensors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 831–837.

[3] Brunetti, A., Buongiorno, D., Trotta, G.F., Bevilacqua, V., 2018. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. Neurocomputing 300, 17–33.

[4] Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A., 2019. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence .

[5] Cardinaux, F., Bhowmik, D., Abhayaratne, C., Hawley, M.S., 2011. Video based technology for ambient assisted living: A review of the literature. Journal of Ambient Intelligence and Smart Environments 3, 253–269.

[6] Choudhury, S.K., Padhy, R.P., Sa, P.K., Bakshi, S., 2019. Human detection using orientation shape histogram and coocurrence textures. Multimedia Tools and Applications 78, 13949–13969.

[7] Dollár, P., . Piotr's Computer Vision Matlab Toolbox (PMT). https://github.com/pdollar/toolbox.

[8] Eveland, C., Konolige, K., Bolles, R.C., 1998. Background modeling for segmentation of video-rate stereo sequences, in: Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231), IEEE. pp. 266–271.

[9] Geronimo, D., Lopez, A.M., Sappa, A.D., Graf, T., 2009. Survey of pedestrian detection for advanced driver assistance systems. IEEE transactions on pattern analysis and machine intelligence 32, 1239–1258.

[10] Hamuda, E., Mc Ginley, B., Glavin, M., Jones, E., 2018. Improved image processing-based crop detection using kalman filtering and the hungarian algorithm. Computers and Electronics in Agriculture 148, 37–44.

[11] Han, J., Shao, L., Xu, D., Shotton, J., 2013. Enhanced computer vision with microsoft kinect sensor: A review. IEEE transactions on cybernetics 43, 1318–1334.

[12] Li, P., Wang, D., Wang, L., Lu, H., 2018. Deep visual tracking: Review and experimental comparison. Pattern Recognition 76, 323–338.

[13] Liciotti, D., Paolanti, M., Frontoni, E., Zingaretti, P., 2017. People detection and tracking from an rgb-d camera in top-view configuration: review of challenges and applications, in: International Conference on Image Analysis and Processing, Springer. pp. 207–218.

[14] Liu, J., Liu, Y., Zhang, G., Zhu, P., Chen, Y.Q., 2015. Detecting and tracking people in real time with rgb-d camera. Pattern Recognition Letters 53, 16–23.

[15] Lu, W.L., Ting, J.A., Little, J.J., Murphy, K.P., 2013. Learning to track and identify players from broadcast sports videos. IEEE transactions on pattern analysis and machine intelligence 35, 1704–1716.

[16] Ma, A.J., Yuen, P.C., Saria, S., 2015. Deformable distributed multiple detector fusion for multi-person tracking. arXiv:1512.05990 .

[17] Munaro, M., Basso, F., Menegatti, E., 2016. Openptrack: Open source multi-camera calibration and people tracking for rgb-d camera networks. Robotics and Autonomous Systems 75, 525–538.

[18] Rosenstrauch, M.J., Pannen, T.J., Krüger, J., 2018. Human robot collaboration-using kinect v2 for iso/ts 15066 speed and separation monitoring. Procedia CIRP 76, 183–186.

[19] Spinello, L., Arras, K.O., 2011. People detection in rgb-d data, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE. pp. 3838–3843.

[20] Stewart, R., Andriluka, M., Ng, A.Y., 2016. End-to-end people detection in crowded scenes, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2325–2333.

[21] Sun, S.W., Kuo, C.H., Chang, P.C., 2016. People tracking in an environment with multiple depth cameras: a skeleton-based pairwise trajectory matching scheme. Journal of Visual Communication and Image Representation 35, 36–54.

[22] Supreeth, H., Patil, C.M., 2018. Efficient multiple moving object detection and tracking using combined background subtraction and clustering. Signal, Image and Video Processing 12, 1097–1105.

[23] Wang, X., 2013. Intelligent multi-camera video surveillance: A review. Pattern recognition letters 34, 3–19.

[24] Yan, Z., Duckett, T., Bellotto, N., 2017. Online learning for human classification in 3d lidar-based tracking, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 864–871.

[25] Yan, Z., Duckett, T., Bellotto, N., 2020. Online learning for 3d lidar-based human detection: experimental analysis of point cloud clustering and classification methods. Autonomous Robots 44, 147–164.

[26] Yang, S., Xu, W., Liu, Z., Zhou, Z., Pham, D.T., 2018. Multi-source vision perception for human-robot collaboration in manufacturing, in: 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), IEEE. pp. 1–6.

[27] Zabulis, X., Grammenos, D., Sarmis, T., Tzevanidis, K., Padeleris, P., Koutlemanis, P., Argyros, A.A., 2013. Multicamera human detection and tracking supporting natural interaction with large-scale displays. Machine vision and applications 24, 319–336.