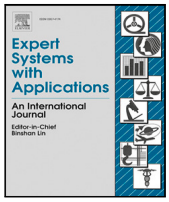
Expert Systems With Applications 252 (2024) 124089

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Real-time multi-camera 3D human pose estimation at the edge for industrial applications

Michele Boldo [a,*], Mirco De Marchi [a], Enrico Martini [a], Stefano Aldegheri [b], Davide Quaglia [a], Franco Fummi [b], Nicola Bombieri [b]

[a] Department of Computer Science, University of Verona, Verona, Italy
[b] Department of Engineering for Innovation Medicine, University of Verona, Verona, Italy

ARTICLE INFO

Keywords:
Distributed HPE
Edge devices
Human–machine interaction
Industrial applications

ABSTRACT

There is an increasing interest in exploiting human pose estimation (HPE) software in human–machine interaction systems. Nevertheless, adopting such a computer vision application in real industrial scenarios is challenging. To overcome occlusion limitations, it requires multiple cameras, which in turn require multiple, distributed, and synchronized HPE software nodes running on resource-constrained edge devices. We address this challenge by presenting a real-time distributed 3D HPE platform, which consists of a set of 3D HPE software nodes on edge devices (i.e., one per camera) to redundantly extrapolate the human pose from different points of view. A centralized aggregator collects the pose information through a shared communication network and merges them, in real time, through a pipeline of filtering, clustering and association algorithms. It addresses network communication issues (e.g., delay and bandwidth variability) through a two-levels synchronization, and supports both single and multi-person pose estimation. We present the evaluation results with a real case of study (i.e., HPE for human–machine interaction in an intelligent manufacturing line), in which the platform accuracy and scalability are compared with state-of-the-art approaches and with a marker-based infra-red motion capture system.

1. Introduction

Human pose estimation (HPE) is a key component for human motion analysis from images and videos (Liu, Chen, Zhao, Zhang and Zhang, 2021; Liu et al., 2018; Liu, Liu, Yang, & Zhang, 2024)

Among the many application fields, such as sports performance evaluation, gait analysis, and rehabilitation (Guo, Deligianni, Gu, & Yang, 2019), there is a growing interest in applying such a computer vision technology in human–robot interaction systems (Lim et al., 2021). On the one hand, recent advances in camera sensors and convolutional neural networks (CNN) architectures (Liu et al., 2022; Liu, Wang, Yang & Wang, 2021; Liu et al., 2023) have led to sufficiently high accuracy of the estimated 3D human poses (Zhang et al., 2023) even for such critical systems (Kidzinski et al., 2020; Pavlakos, Zhou, Derpanis, & Daniilidis, 2017). Some techniques implement end-to-end CNNs and take advantage of sensors that integrate RGB and depth information (e.g., Intel RealSense, Microsoft Kinect, StereoLab Zed) to estimate the pose in 3D. Other techniques achieve comparable accuracy by estimating the 2D pose first from an RGB sensor and then lifting the 2D pose to a view-invariant 3D pose (Chen, Xu, & Zou, 2023; Fang, Xu, Wang, Liu, & Zhu, 2018; Palermo, Moccia, Migliorelli, Frontoni, & Santos, 2021). On the other hand, all these approaches, which are based on a single camera point of view, suffer from field occlusions when applied to real industrial contexts (Dong et al., 2021). In standard working scenarios, both obstacles and other people are often between the subject and the camera sensor for long periods of time, and therefore, the subject often hides body parts from the camera. All this leads to inaccurate estimations of the whole body poses, which makes the HPE software unsuitable for real industrial scenarios. HPE based on multiple points of view (i.e., multi-camera networks) has shown great potential to solve such occlusion limitations. In the domain of multi-camera Human Pose Estimation (HPE), different methodologies are employed. Some systems utilize multiple views with 2D estimated poses, which are subsequently merged to generate a 3D reconstruction (Chen, Ai, Chen, Zhuang, & Liu, 2020). Conversely, others utilize separate 3D systems for each view and then aggregate the resulting contributions (Tu, Wang, & Zeng, 2020). Nevertheless, it requires one HPE software node per camera and strict synchronization among the generated data flows. At the state of the art, HPE platforms based on multi-camera generally target the triangulation

* Corresponding author.
E-mail addresses: michele.boldo@univr.it (M. Boldo), mirco.demarchi@univr.it (M. De Marchi), enrico.martini@univr.it (E. Martini), stefano.aldegheri@univr.it (S. Aldegheri), davide.quaglia@univr.it (D. Quaglia), franco.fummi@univr.it (F. Fummi), nicola.bombieri@univr.it (N. Bombieri).

https://doi.org/10.1016/j.eswa.2024.124089
Received 27 October 2023; Received in revised form 28 March 2024; Accepted 20 April 2024
Available online 23 April 2024

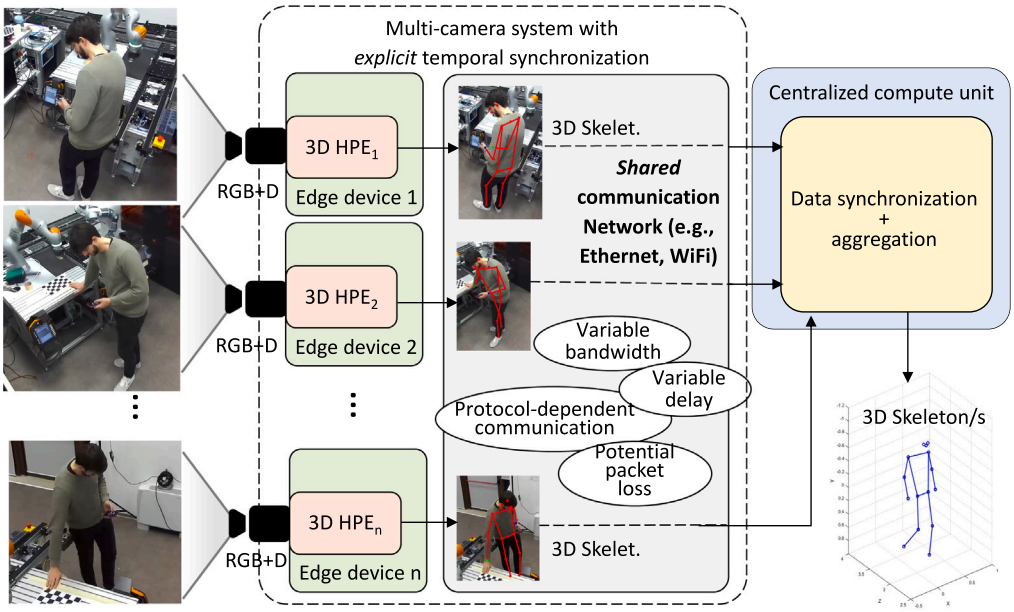


Fig. 1. Multi-camera system with 3D HPE offloaded on edge devices connected to a remote server performing synchronization and aggregation.

of many 2D poses to build the 3D pose. Since both the 2D HPE and the triangulation software are centralized, they assume (i) the 2D pose is always generated for each point of view, (ii) the synchronization among all the multiple 2D poses at each time frame is implicit, (iii) there are no constraints on the cameras-HPE network bandwidth, and (iv) there are no constraints on the computational resources (Carraro, Munaro, Burke, & Menegatti, 2019). Since all these constraints together cannot be guaranteed in real industrial environments, centralized solutions at the state of the art cannot guarantee enough accuracy for human–robot interaction systems. We propose a different approach where multiple instances of 3D HPE software run on a distributed set of edge devices (one instance per camera), and the results are collected by an aggregator unit on a centralized server. This architecture has never been considered in the literature as it implies that:

1. Since each single HPE node runs on a resource-constrained device, it achieves low accuracy (i.e., not enough for human–machine interaction).
2. Since the communication network is subject to bandwidth and delay variability as well as packet loss, the 3D poses generated in real-time by each node easily get out of synchronization before reaching the central aggregator unit.
3. The aggregation phase cannot work in case of multi-person in the scene, as the person detection implemented by the multiple HPE nodes are not synchronized.

To overcome these limitations, we propose a two-level synchronization approach customized for real-time multi-person 3D HPE (see Fig. 1). It aims at aggregating the contribution of the multiple and distributed 3D HPE nodes by supporting the multi-dimensional information received in real-time by the central unit (i.e., single/multi-person from single/multi-camera) and the communication issues (i.e., pose loss, pose obsolescence). With respect to the state of the art, the main novel contributions of this work are:

- A real-time distributed 3D HPE architecture in which the inference software is customized for resource-constrained edge devices. This improves the system's scalability by overcoming the constraints on the network bandwidth and on the computational resources.
- A centralized two-levels aggregation mechanism that synchronizes the 3D poses in real-time and merges keypoints belonging to multiple persons by compensating for time-varying communication delays.
- It quantitatively measures the accuracy of the 3D HPE poses with an infra-red marker-based motion capture system as ground-truth in an industrial environment. To the best of our knowledge, such an accuracy evaluation has never been done in the past.

2. Related works

Human pose estimation has been largely applied in human-cyber–physical systems. In Moghaddam and Piccardi (2014), the authors addressed the problem of adopting sequential probabilistic models for human action recognition. They proposed two methods for one-off initialization of hidden Markov models, which achieve a sound trade-off between accuracy and training time.

In Zhang, Liao, Paz, and Christensen (2022), the authors proposed to use the skeleton generated by state-of-the-art HPE platforms to identify human heads. As an alternative to wearable sensors (Zhu, Han, & Yi, 2022), they demonstrated that HPE software can reduce missed faces and better protect the identity information of human subjects.

In Lee, Cho, Liu, Cho, and Helal (2015), the authors investigated on the automated understanding and recognition of human activities and behaviours in a smart space. They proposed a context-activity-action nexus and showed how the approach combines modelling and visualization of actions with context and activity simulation. In Proia, Carli, Cavone, and Dotoli (2022), the authors reviewed control techniques for safe, ergonomic, and efficient human–robot collaboration in the digital industry.

Different solutions focused on human pose estimation based on multiple cameras to deal with occlusions and field-of-view limitations. A class of them rely on a topology of distributed 2D RGB cameras. Each camera sends, through the network, the input frames to a centralized module, which first extrapolates multiple 2D poses of the subject (one per view) and, then, it triangulates them to build the final pose in 3D (e.g., Chen et al. (2020)). In Remelli, Han, Honari, Fua, and Wang (2020) authors proposed a similar centralized approach by which the 2D keypoint positions are estimated through a *learned camera-independent representation* of the 3D poses. Then, through a GPU-optimized triangulation algorithm, they lift the 2D keypoints to 3D.

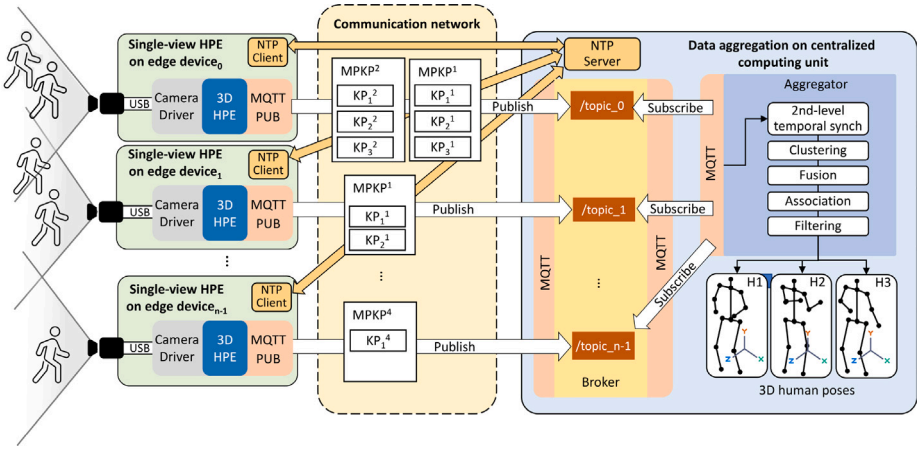


Fig. 2. Overview of the BeFine architecture.

Another class of approaches relies on multiple 2D cameras, and for each camera, the 3D pose is directly extrapolated by a centralized inference application through a CNN directly trained for 3D data (Tu et al., 2020). In Yeung, Kwok, and Wang (2013) authors proposed a methodology that takes advantage of multiple *RGB-D* sensors (Microsoft Kinect) to build 3D human poses. Their focus is on the pose fusion algorithm, which takes advantage of the hard constraints of the inter-joint distances.

In Li, Liu, Tian, Zhu, and Wang (2018) authors introduced the information weighted consensus filter (ICF), which increases the accuracy of body joint estimation through a data fusion algorithm. The algorithm works off-line by evaluating the sequences of generated keypoints. In Liu, Liu, Tian, and Ji (2019), the authors implemented the distributed 3D HPE through a *dynamic hybrid consensus filter* to fuse the human pose data. The method is based on a distributed aggregation and does not require a centralized master node for merging data.

In Carraro et al. (2019) authors proposed a system for estimating 3D human poses through a network of 3D RGB-D cameras. Each node estimates the 3D pose by first performing the inference on the RGB image through a 2D CNN. Then each node interpolates the results with the depth information provided by the sensor to extrapolate the third dimension of each keypoint. The main focus of the work is on the association algorithm of the different views.

All these approaches either assume no constraints in computational resources or no communication interferences between cameras, HPE inference applications, and the aggregation phase. In general, they either suffer from scalability if centralized or from sensitivity to temporal de-synchronization caused by real communication networks.

3. The BeFine platform

Fig. 2 shows the architecture of the proposed real-time distributed system (BeFine), which consists of a set of edge devices and a centralized aggregation unit connected by a communication network. Each edge device consists of an RGB-D camera and a heterogeneous embedded board running an inference application for single-view real-time 3D HPE. The devices are synchronized both spatially and temporally to guarantee common references for the data aggregation (Section 3.1). Local HPE information with timestamps is sent over the network through a standard communication protocol (Section 3.2) towards the centralized aggregation unit. The centralized aggregation unit implements data merge through a pipeline of filtering, clustering, fusion, and association algorithms (Section 3.3). The platform aims at supporting both single and multi-person pose estimation ($H1$, $H2$, and $H3$ in Fig. 2) in real-time.

3.1. Single-view 3D human pose estimation at the edge

We developed a resource-constrained edge device version of Human Pose Estimation (HPE) based on the 2D HPE method introduced in NVIDIA AI IoT (2020) (referred to as *TRTPose*). We re-implemented the inference application in C++ and CUDA to maximize the utilization of the heterogeneous CPU+GPU architecture present in off-the-shelf edge computing devices (i.e., NVIDIA Jetson boards). The application retrieves the 2D keypoints from the RGB image captured by the RGB-D camera, representing the human body joints. Then, using the depth matrix extracted from the RGB-D sensor, the 2D key points are associated with their depth information, indicating the distance of each point from the camera in 3D space. After this operation, the RGB image and the depth matrix are discarded. To extract the 3D coordinates in the space of each keypoint, the node implements a back-projection of each 2D keypoint (kp_i^{2D}) into the 3D space based on the pinhole camera model (Hartley, 2004). The result is a set of *3D keypoints* (KP_p^t) for each video frame with a timestamp (t), representing the joints of the human body for each person (p) identified in the scene:

$$KP_p^t = \{kp_j^t \ : \ j = 1...|CNN\_kps|\} \tag{1}$$

where $|CNN\_kps|$ is the total number of keypoints detected by the adopted CNN for each frame. The timestamp generated by the edge device is synchronized with the global reference time, as detailed in the next section. We assume a common frame rate among the camera sensors.

3.2. Temporal alignment across the communication network

The platform addresses the temporal alignment among the edge devices and the aggregator node at two levels. At the first level, the platform implements a network time protocol (NTP) (Martin, Burbank, Kasch, & Mills, 2010) to synchronize the clock of each edge device with the global reference clock of the computing unit hosting the aggregator. This allows each set of keypoints KP_p^t, for each person and for each analysed video frame, to be marked with a globally synchronized timestamp. In the example of Fig. 2, considering the first camera, $\{KP_1^1, KP_2^1, KP_3^1\}$ is the set of keypoints of three people at the time of the first video frame and $\{KP_1^2, KP_2^2, KP_3^2\}$ is the set at the time of the second frame. For each edge device and for each video shooting, the keypoints and the timestamp are encapsulated into a message and sent through the TCP transport protocol. Each message generated by the device v represents a *multi-person keypoint set* $MPKP_v^t = \{KP_1^t, \dots, KP_N^t\}$, and contains the whole set of keypoints of all (N) individuals present in the scene at time frame t.

To guarantee platform portability and modularity, we adopted a standard MQTT-based publisher/subscriber approach for the delivery

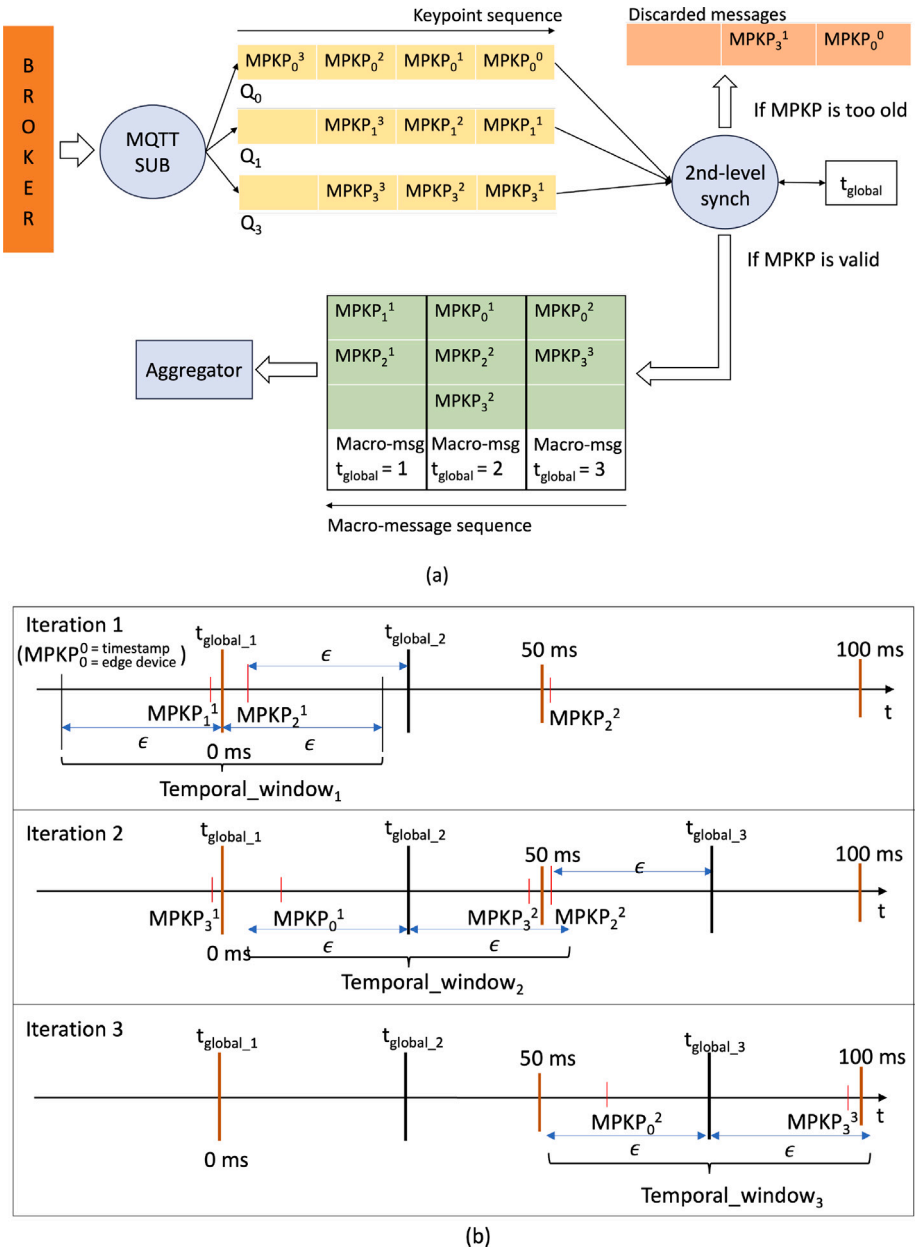


Fig. 3. Second-level temporal alignment (a) and example of t_{global} definition with the corresponding temporal window (b).

of the multi-person keypoint sets (see Fig. 2). The computing unit hosting the aggregator contains an *MQTT broker* that collects data from the edge devices. Each device acts as *MQTT publisher*, sending in real time a sequence of $MPKP_v^t$ messages on the corresponding *topic* (one per device). The aggregator acts as *MQTT subscriber* interested in all topics to collect data in real-time.

In real scenarios, the communication network is shared among different applications, and, as a consequence, it features variable load and potential packet losses, both translating into variable values of transmission delay (i.e., lost packets are re-transmitted by TCP inside MQTT). This may lead to a non-periodic arrival of messages from the same edge device and a misalignment among the messages referring to the same time but coming from different devices. The aggregator addresses both issues through a second-level time alignment.

The block diagram in Fig. 3(a) depicts the main idea. The asynchronous thread (`MQTT SUB`) in the aggregator subscribes to all MQTT topics and, in real time, forwards each message $MPKP_v^t$ to the corresponding FIFO queue Q_v (one queue per topic). MQTT allows us to assume that messages entering the corresponding queue are ordered incrementally by the timestamp. The first objective of the queues is to compensate for fluctuations of the inter-arrival time between messages arriving from the same camera. A second thread (`2nd-level synch` in the figure) implements Algorithm 1 to group $MPKP_v^t$ messages with similar timestamp into a *macro-message* and to discard obsolete messages. Given a reference timestamp (t_{global}) and a temporal tolerance ϵ, a macro-message contains all messages with timestamp in the temporal window $t_{global} \pm \epsilon$ ms. The maximum value of ϵ is half the period of the edge camera rate. By design, each macro-message can contain up to one message per queue (i.e., one per edge device). Too old messages are discarded since they are obsolete (rows 12–13). New messages outside the temporal window are left into the queues (rows 14–15). The thread sends the macro-message to the last stage of the aggregator and starts the generation of a new temporal window. To do that, it updates the reference timestamp (t_{global}) by adding half frame period to the value of the timestamp of the newest message in the previous macro-message (row 20). It updates t_{global} with a frequency up to twice the frame rate.

Algorithm 1 Algorithm for time alignment in the aggregator

Require: $MPKP_i^t$ ordered in each queue
$Q \leftarrow \{Q_1, ..., Q_n\}$
$\epsilon \leftarrow \frac{1}{2 \times cameraFPS} \times 1000$ ▷ in milliseconds
t_{global} initialized
while $TRUE$ **do**
  $MacroMsg \leftarrow \{\}$
  $\tilde{t} \leftarrow \{\}$
  for all $queue_i \in Q$ **do**
    for all $MPKP_i^t \in queue_i$ **do**
      if $t_{global} - \epsilon \leq t \leq t_{global} + \epsilon$ **then**
        $MacroMsg \leftarrow MacroMsg \cup \{MPKP_i^t\}$
        $\tilde{t} \leftarrow \tilde{t} \cup \{t\}$
      else if $t \leq t_{global} - \epsilon$ **then**
        delete $MPKP_i^t$ from $queue_i$
      else if $t \geq t_{global} + \epsilon$ **then**
        continue
      end if
    end for
  end for
  Send_$MacroMsg$_to_aggregator()
  $t_{global} \leftarrow max(\tilde{t} \cup \{t_{global}\}) + \epsilon$
  sleep for ϵ
end while

As a consequence, the temporal windows are overlapped (see Fig. 3(b)) and the number of messages discarded due to late arrival is minimized.

Fig. 3(b) shows an example in which the frame rate is set to 20 Hz (50 ms period). The example depicts three instants, each one showing the messages arrived in the aggregator in any queue, which are considered for the generation of a macro-message. In the first instant, $MPKP_1^1$ has arrived from edge device 1, as well as both $MPKP_2^1$ and $MPKP_2^2$ from the device 2. $MPKP_1^1$ and $MPKP_2^1$ are grouped in the first macro-message, while $MPKP_2^2$ is left on the queue for the next iteration. In the second instant, besides $MPKP_2^2$, three new messages have arrived. All of them are grouped in the new macro-message except $MPKP_3^1$ since no longer useable for the new aggregation.

3.3. Data processing in the aggregator

The aggregator aims at generating one and only one 3D skeleton for each person present in the scene. To do that, it merges, in real time, the information coming from the different edge devices. The information is composed of a sequence of macro-messages, where each macro-message contains the set of keypoints representing one of the following configurations:

- Single person from single view;
- Single person from multiple views;
- Multiple people from single view;
- Multiple people from multiple views;
- A combination of the previous.

For each macro-message, the aggregator first applies a *clustering* algorithm to associate skeletons belonging to the same person taken from different cameras. This allows the system to understand whether multiple sets of keypoints in the macro-message belong to a single or different subjects (Section 3.3.1). The result is a number of clusters, one per human subject identified in the scene. The aggregator implements a *fusion* step to merge the keypoints of each cluster (Section 3.3.2). This allows the system to generate one single 3D skeleton per person. Then, the aggregator implements an algorithm of temporal *association* to guarantee consistency in the association of skeletons to people across different video frames (Section 3.3.3). This aims at avoiding switches between skeleton-person associations due to the dynamism in the scene in case of more than one person, which does not maintain the order of people identification along the frames. Finally, the identified 3D skeletons undergo a filtering step to denoise the keypoint information (Section 3.3.4).

3.3.1. Clustering

The input is a macro-message, which contains $MPKP$ messages corresponding to the same time interval and that describes the 3D position of the person (or people) keypoints in the different views. We assume that all the keypoints have already been rototranslated and mapped into a common coordinate system between all views thanks to the (standard) spatial calibration phase. If a person is taken by more than one camera, the macro-message contains more than one set of keypoints KP. The clustering step implements a density-based spatial clustering of applications with noise algorithm (DBSCAN) (Ester, Kriegel, Sander, & Xu, 1996) to select the keypoint sets for clustering. The algorithm identifies the clusters (one per person) that contain a sufficiently high density (3D space proximity) of keypoints to represent a person. Each single keypoint Kp^t is tagged with a person identifier (Kp_p^t where p is the identifier).

The algorithm relies on two parameters to identify the clusters, the first is the maximum *distance* that two samples should have in the same cluster to be considered of the same person. The second is the *minimum number of samples* that can compose a cluster. The clustering module calculates the *distance* by considering the topological constraints of the human body to reduce outliers.

3.3.2. Fusion metrics

For each macro-message, this module merges multiple sets of keypoints of the same human subject. Different metrics are at the state of the art for keypoint fusion (e.g., voting scheme Hong & Kim, 2018, etc.). They differ from the way they balance the contribution of each view to each keypoint and how they identify outliers. The fusion algorithm is out of the scope of this work and any solution in literature can be adopted in the proposed platform. It implements a basic algorithm that relies on the coordinate median of each keypoint kp.

3.3.3. Association

The result of the previous step is a set of non-redundant keypoints for each person in the scene, which is the input for the temporal association. This module implements such an assignment problem through a variant of the Munkers algorithm (Virtanen et al., 2020). We defined the cost function for the optimization as follows: for each couple of consecutive temporal instants t_i and t_{i-1}, the cost is the euclidean distances of all centroids of the keypoint sets representing the people present in the scene at time t_i with those of the people present in the scene at time t_{i-1}. It computes the centroids through the median of all the subject keypoints.

3.3.4. Filtering

The aggregator implements temporal filtering to correct errors due to the intrinsic approximation of the human poses performed by the system from the edge devices to the aggregator. These are due to the neural network and camera accuracy, depth extrapolation in particular (see Section 3.1), and to the network interference and the consequent packet loss. The system implements a *Gaussian filter* on the aggregated data. Since this filter acts on a temporal window of keypoints, to run such filtering in real time, the module implements a sliding window by which data is first accumulated and then filtered.

4. Experimental results

We evaluated *BeFine* in terms of accuracy, scalability, and robustness to the network interference.

Table 1
Performance of edge devices and transmission network. The proposed system outperforms the RTSP methodology in Max Framerate and minimizes the information loss due to time sync.

Setup		Edge: Performance per node			Network	
	Cam (#)	Workload %CPU;%GPU	Mem. (MB)	Max rate (FPS)	Used BW (kbps)	Frames/KPs lost due to time sync
RTSP	1	34.5;21.7	484	**19**	2452	0/405 **(0.0%)**
	2	34.6;21.7	489	**19**	4869	17/791 **(2.1%)**
	3	33.4;21.7	483	**19**	7318	6/935 **(0.6%)**
	4	40.4;21.0	496	**19**	9770	120/821 **(14.6%)**
	5	39.3;27.8	490	**19**	12,000	536/701 **(23.5%)**
	6	/	/	/	/	/
BeFine	1	22.5;37.4	3349	**24**	108	0/448 **(0.0%)**
	2	22.7;37.4	3312	**22**	214	1/896 **(0.1%)**
	3	20.9;37.4	3051	**23**	333	2/1340 **(0.1%)**
	4	22.5;37.4	3310	**22**	442	1/1788 **(0.0%)**
	5	24.9;37.4	3051	**23**	550	3/2236 **(0.1%)**
	6	23.1;37.4	3010	**23**	659	3/2680 **(0.1%)**
	10	25.6;37.4	3366	**20**	1009	9/4468 **(0.2%)**

4.1. Experimental setup

Each edge device consists of an Nvidia Jetson Xavier NX board, equipped with 384 CUDA cores + 48 tensor cores GPU, hexa-core CPUs, and 8 GB RAM LPDDR4X unified memory. Each device is directly connected to a StereoLabs Zed2 RGB-D Camera (2K resolution). The centralized compute unit consists of a desktop PC equipped with an Nvidia RTX 2070 Super GPU, an Intel i5 7400 CPU, 8 GB DDR4 RAM, and Ubuntu 18.04 LTS operating system. Edge devices and the centralized unit are connected to a 1 Gb/s Ethernet switch (Netgear GS305E).

The dataset consists of 36 videos recorded in a smart manufacturing industrial line, in which up to three people play different working actions (e.g., human–robot arm interactions, walking, loading, moving), for a total amount of 18 min of recording.

For the accuracy evaluation of the HPE-based marker-less systems (i.e., both centralized and distributed), we used an 8-camera marker-based motion capture system (Mocap) MX 13, VICON, Oxfordshire, UK as ground truth. For the quantitative evaluation of the HPE accuracy, we measured the results in terms of average percentage of detected keypoints per person ($AvgKp$), mean-absolute error distance (MAE), and Pearson correlation (PCC) w.r.t. to the ground truth.

4.2. Analysis of accuracy and scalability

For the accuracy and scalability, we evaluated the multi-camera pose estimation results of the following configurations over an increasing number of cameras (i.e., viewpoints) and the corresponding HPE instances:

1. *CentralOpenPose*: A fully centralized approach with one of the most widespread 3D HPE (i.e., OpenPose Cao, Simon, Wei, & Sheikh, 2017) for each view and the aggregation phase presented in Section 3.3. Each view point consists of the only camera sensor, which is connected via cable to the centralized server. It does not include temporal synchronization among cameras since implicitly guaranteed. This allows us to evaluate the maximum accuracy of the multi-view 3D HPE software at the state of the art without the network interference.
2. *CentralTRTPose++*: A fully centralized approach as in (1), where the extended version of TRTPose presented in Section 3.1 replaces OpenPose. This allows us to evaluate the maximum accuracy and scalability of the proposed 3D HPE module without the interference of the network and communication protocol.
3. *RTSP*: A fully centralized approach as in (2) with the network and the corresponding communication protocols described in Section 4.1 between the cameras and the 3D HPE instances + aggregator system. The edge devices implement compression and transmission of the video using the real-time streaming protocol (RTSP). The depth extrapolation, 3D HPE of each video flow, and aggregation are centralized in the server. This allows us to evaluate the scalability of the approach by including the network constraints.
4. *BeFine*: The platform presented in Section 3.

Table 1 reports the performance of the software on the edge nodes with the *RTSP* and *BeFine* configurations.

With *RTSP*, since each edge device implements video stream compression and transmission, it supports a frame rate $\simeq 19\ FPS$. The high bandwidth and the nature of the RTSP protocol guarantee the transmission with no packet loss. Nevertheless, the arrival delay of packets increases with the increase of the view points. As a consequence, a sensible number of the transmitted frames are *discharged* by the system before the aggregation phase since temporally misaligned (up to 23.5% with a five camera network). With more than five cameras, the *RTSP* configuration runs out of memory in the centralized server, as detailed in the following.

With *BeFine*, the light 3D HPE software (*TRTPose++*) offloaded on the NVIDIA Jetson NX edge board supports a frame rate $\simeq 22\ FPS$. The data generated by each edge node consists of keypoints (i.e., one floating point $\times$ 3 dimensions $\times$ 25 human keypoints), which require a very limited bandwidth. The low data payload allows the packets to arrive with limited delays, which translates into a negligible number of discharged keypoints before aggregation. To evaluate the system's scalability, we included up to ten view points (i.e., cameras) in the system. However, since in this configuration, the highest workload is the aggregator procedure in the centralized compute unit, it is reasonable to suppose that the platform can be extended to a higher number of nodes.

Table 1 does not include information on the *Central OpenPose* and *Central TRTPose+* since they do not implement any software unit (for compression or HPE) at the edge.

Table 2 reports the performance of the software on the centralized server for all configurations. With *Central OpenPose*, the high accuracy of the OpenPose HPE requires high workloads on both CPU and GPU units, which limits the system's scalability. The centralized software achieves a limited frame rate with two cameras (i.e., lower than camera FPS) and runs out of memory with three cameras. This is due to the fact that it runs one OpenPose software instance per camera. The results obtained with one camera confirm the high workload and memory footprint of the *OpenPose* software, which prevent its application on the edge boards.

Central TRTPose+ supports a higher number of viewpoints and achieves higher FPS values. Nevertheless, scalability is limited to five

Table 2
Performance of centralized server. As the number of cameras increases, the framerate decreases. This becomes a problem for all methodologies considered (e.g. RTSP system with 5 cameras goes at a framerate of 3 fps). On the other hand, Befine, due to this distributed computation on edge devices, keeps the computing demand low on the centralized server, which acts exclusively as an aggregator.

Setup		Centralized compute unit: Performance		
	Cam (#)	CPU Workload %CPU;%GPU	GPU Mem. (MB)	HPE+Aggr (max FPS)
Central OpenPose	1	47.5;29.0	3146	17 FPS
	2	93.5;44.0	6292	11 FPS
	3	/	**Out-of-mem**	/
Central TRTPose++	1	39.6;18.0	1653	65 FPS
	2	58.9;33.0	3306	63 FPS
	3	80.9;50.0	4959	63 FPS
	4	85.1;59.0	6612	40 FPS
	5	93.9;59.0	8265	26 FPS
	6	/	**Out-of-mem**	/
RTSP	1	51.7;18.0	1540	8 FPS
	2	81.6;33.0	3080	8 FPS
	3	85.1;46.0	4620	5 FPS
	4	95.6;42.0	6160	6 FPS
	5	100;40.0	7700	3 FPS
	6	/	**Out-of-mem**	/
BeFine	1	16.2;0.0	Not used	233 FPS
	2	17.2;0.0	Not used	197 FPS
	3	16.9;0.0	Not used	200 FPS
	4	17.2;0.0	Not used	176 FPS
	5	17.0;0.0	Not used	170 FPS
	6	17.3;0.0	Not used	166 FPS
	10	18.9;0.0	Not used	153 FPS

Table 3
Accuracy comparison of the 3D HPE configurations (*Central OpenPose* as ground truth). The proposed system performs equal as the Centralized version (the version without transmission issues/latency).

Setup		Accuracy (optimal-cases:worst-cases of occlusions/poses)		
	Cam (#)	%Avg detected KPS	MAE w.r.t. Central OpenPose (cm)	PCC w.r.t. Central OpenPose (%)
Central TRTPose++	1	93.5	7.2:11.5	93.8
	2	99.2	4.5:8.9	97.5
	3	**99.5**	**4.9:8.4**	98.8
RTSP	1	93.7	8.1:12.4	89.5
	2	97.0	7.3:11.7	94.2
	3	**98.7**	**10.6:14.1**	97.1
BeFine	1	93.5	6.9:11.2	93.8
	2	99.2	4.5:8.9	97.5
	3	**99.5**	**4.9:8.4**	98.8

cameras, after that it runs out of memory due to the multiple TRTPose++ software instances.

RTSP suffers from scalability for the same reason and, in addition, it achieves lower FPS values as it implements image decompression before aggregation beside 3D HPE.

BeFine implements only the aggregation phase in the centralized server, and this allows the system to support very high FPS values regardless of the number of view points. The workload, which is entirely on the CPU, and the memory footprint are limited. This allows the aggregation unit to be offloaded on one of the edge devices (which is part of our future implementation).

Table 3 reports the comparison of accuracy achieved by the different configurations by considering *Central OpenPose* 3-cam as ground truth. In particular, it aims to evaluate the efficacy of the networked systems in addressing the occlusion limitations. To support three cameras, the results with *Central OpenPose* have been obtained offline.

The accuracy is expressed in terms of average detected keypoints, mean absolute error (MAE) of the Euclidean distance of the keypoints, and Pearson correlation w.r.t. the ground truth. As expected, all configurations suffer from occlusions with one camera (≃93.5%). The average of detected keypoints sensibly increases with two or three cameras. It is important to note that small percentages of non detected keypoints reflects on sensible lost of accuracy in the estimation of the overall skeleton position (see Fig. 4 and the following discussion).

The table shows that, by considering the MAE and Pearson correlation of the overall estimated skeletons, *Central TRTPose++* and *BeFine* achieve a comparable accuracy (i.e., below 5 cm optimal video capturing scenario, below 9 cm worst case scenario, 98.8% PCC). This proves the efficiency of the temporal synchronization and aggregation unit. The efficiency of this configuration with non optimal network conditions is presented in Section 4.3.

With *RTSP*, the low values of supported FPS (see Table 2) due to the bottleneck in the central server leads to a strong degradation of the aggregation results (from 11 to 14 cm average error with three cameras).

In terms of performance, we obtained similar results with the camera sensors set to lower resolutions (i.e., 1920 × 1080 and 1280 × 720, respectively) with higher working frequency (i.e., 30 FPS and 60 FPS, respectively).

Figs. 4, 5, and 6 report the accuracy evaluation of *Central OpenPose* (software golden model) and *BeFine* w.r.t. the marker-based infra-red multi-camera MoCap system (i.e., Vicon) as ground truth. Fig. 4 reports the results with one camera, which underlines the very low accuracy of the marker-less systems with a single point of view due to occlusions in real scenarios. The boxplots represent the error in cm of each keypoint w.r.t. the ground truth (segment extremes as minimum and maximum values, lower and upper sides of rectangles as first and third

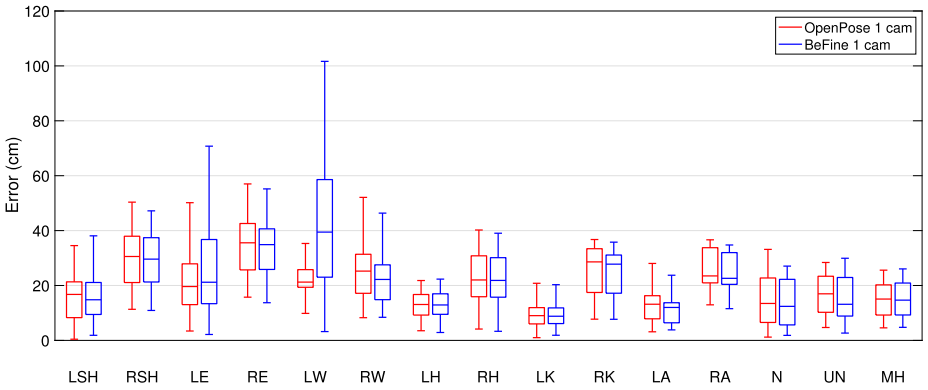


Fig. 4. Comparison between the accuracy in measuring the keypoints achieved with the marker-less systems (1 camera centralized OpenPose and 1 camera *BeFine*) and the marker-based MoCap system (Vicon). Keypoints (Left and Right): shoulder (SH), elbow (E), wrist (W), pelvis (H), knee (K), ankle (A), upper and lower extremities of the neck and the midpoint of the pelvis. Befine 1cam and Openpose 1cam perform similarly, with low accuracy due to occlusions.

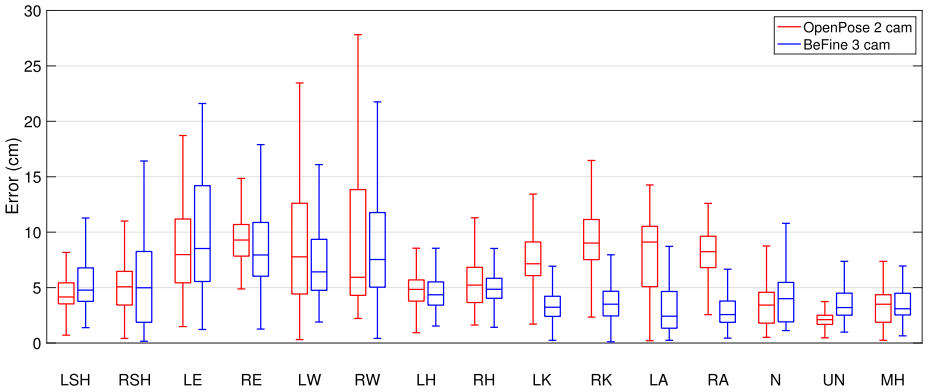


Fig. 5. Comparison between the accuracy in measuring the keypoints achieved with the marker-less systems (2 cameras centralized OpenPose and 3 cameras *BeFine*) and the marker-based MoCap system. Befine with 3 cameras perform better than Openpose with 2 cameras (the best configuration of Openpose for real-time purposes).

quartiles, and median in the rectangles). It is important to note that the accuracy of this comparison (1-cam) is affected only by the 2D pose estimator and the depth sensors. Since the proposed methodology is completely modular, an upgrade on these two parts (e.g. adopting a more accurate transformer-based pose estimator) has the effect of improving the quality of the whole system.

Fig. 5 shows the comparison results between *Central OpenPose* with two cameras (best setup to support the real-time pose estimation) and *BeFine* with three cameras. The comparison is representative to show that, although the *lighter TRTPose++* of *BeFine* is less accurate in each single view, it achieves a system level accuracy comparable to *Central OpenPose* by increasing the number of view points. Thanks to the additional camera support, *BeFine* achieves higher accuracy than *Central OpenPose* for more than one keypoints. It is important to note that the average error includes the bias given by the position of the markers in the human body for the MoCap and the position of the extrapolated keypoints of both the HPE systems. We measured an average Person correlation between keypoints and markers $\simeq 99\%$, which emphasizes the very limited difference between the human skeleton estimated by the HPE systems and the MoCap in dynamic scenes.

Fig. 6 shows the overall system accuracy achieved by *BeFine* with 10 cameras with optimal (few occlusions and reduced subject mobility) and suboptimal video capturing scenarios (many occlusions and fast subject mobility). The results are the average values obtained with the whole dataset, which includes a mix of single/multiple person in the scene with single/multi view points per person. Even considering the bias between marker and keypoint coordinates, it is evident how occlusions (mostly involved by a single view point per person in the dataset) lead up to an additional 9 cm average error in the keypoint estimation.

4.3. Analysis of the robustness to communication issues

In the previous tests, we considered a state-of-the-art local area network with a high-end Ethernet switch providing 1 Gb/s bandwidth and 1 ms end-to-end latency. We now compare the system efficiency in worst case scenarios where the available network bandwidth is lower and end-to-end latency as well as packet loss rate may be significant. We report the results obtained in two use cases:

- a mobile setup in a private area where edge devices communicate with the aggregator through WiFi;
- a public area setup where the aggregator is hosted on the cloud and edge devices transmit on a cellular network.

In both cases, the first step consisted of characterizing the global device-aggregator channel in terms of available bandwidth, packet loss rate due to impairments in the wireless link and end-to-end latency and its variability. To this purpose, we set up a real test bed with two Linux machines running Iperf3.[1] For the WiFi scenario, the Linux boxes were connected to a Ubiquiti UAP-AC-M access point. For the cellular scenario, we connected one Linux box to Oneplus7t cell phone through USB3 tethering while the Iperf3 server was installed in a Linux virtual machine on GARR cloud[2] with public IP address 90.147.167.187. First, several Iperf3 UDP sessions were launched to find a stable value for the available bandwidth, which was 25 Mb/s for WiFi and between 2 Mb/s and 10 Mb/s for the cellular setup depending on the signal quality. Then, in the cellular case, Iperf3 was used to generate a UDP flow

[1] http://software.es.net/iperf/.
[2] https://cloud.garr.it/.

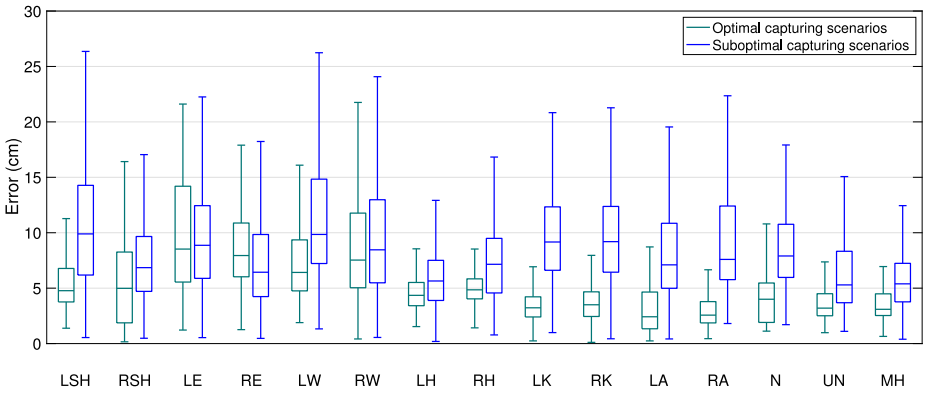


Fig. 6. Comparison between the accuracy achieved by 10 cameras BeFine with optimal (few occlusions and reduced subject mobility) and suboptimal (many occlusions and fast subject mobility) video capturing scenarios.

Table 4
Impact of communication issues on the BeFine accuracy with 10 views. If the bitrate exceeds the available bandwidth, packets are lost due to the congestion and then retransmitted, causing delay and subsequent error increase.

Frame rate (FPS)	Bandwidth (Mb/s)	Radio loss (%)	Latency (ms)	Bitrate (Mb/s)	Lost KPs due to delay (%)	Avg. detected KPs (%)	MAE w.r.t. unloaded network (cm)	PCC w.r.t. unloaded network (%)
15	25	/	/	1.5	0.1	99.6	0.0 ± 0.0	100.0
	10	/	/	1.5	0.1	99.6	0.0 ± 0.0	100.0
	5	/	/	1.5	0.1	99.6	0.0 ± 0.0	100.0
	2	/	/	1.5	0.1	99.6	0.0 ± 0.0	100.0
	10	5	/	1.6	0.3	99.6	0.0 ± 0.0	100.0
	10	10	/	1.7	0.7	99.6	0.1 ± 0.0	100.0
	10	15	/	1.6	1.0	99.6	1.4 ± 0.0	99.9
	10	25	/	1.7	5.5	99.6	3.2 ± 0.1	99.8
	10	/	15 ± 5	1.5	0.1	99.6	0.0 ± 0.0	100.0
	10	/	35 ± 1	1.5	0.1	99.6	0.0 ± 0.0	100.0
25	25	/	/	2.5	1.1	99.6	0.2 ± 0.1	100.0
	10	/	/	2.5	1.2	99.6	0.1 ± 0.0	100.0
	5	/	/	2.5	11.5	99.6	3.2 ± 4.3	99.8
	2	/	/	2.4	68.6	95.4	12.5 ± 11.5	91.8
	10	5	/	2.6	2.8	99.6	0.6 ± 3.0	99.8
	10	10	/	2.8	28.9	99.6	4.2 ± 6.9	99.5
	10	15	/	3.0	19.7	99.2	3.0 ± 6.6	99.8
	10	25	/	2.8	56.0	98.5	8.6 ± 11.8	97.9
	10	/	15 ± 5	2.5	4.5	99.6	0.4 ± 2.1	99.9
	10	/	35 ± 1	2.5	8.8	99.6	1.2 ± 3.5	99.9

with a bitrate equal to the available bandwidth, and the packet loss rate was measured. In this way, we measured the so-called *radio packet loss rate* due to radio signal problems and not to congestions in the wide area network. Latency was measured by using the PING utility between the two Linux systems. As expected, the WiFi latency was negligible, whereas in the cloud connection, the latency was between 15 ms and 35 ms.

The next step consisted of reproducing these network conditions in the original Ethernet network. In the network interface of the central unit, the Linux traffic control tools `tc`[3] and `tc-netem`[4] have been used to create a bottleneck to simulate a reduction of the available bandwidth, to drop packets according to a given probability and to introduce transmission delays. *BeFine* was tested with 10 simultaneous video inputs at 15 frame/s and 25 frame/s.

Table 4 shows the impact of the described communication issues on the *BeFine* accuracy. Columns 2 to 4 report the considered network conditions. The next columns describe the behaviour of *BeFine*. Column 5 reports the total bitrate generated by the 10 edge devices, which depends on the frame rate. If it exceeds the available bandwidth, packets are lost due to congestion, and the MQTT TCP re-transmits them leading to further delays. Column 6 reports the keypoints rejected by the aggregator because they are obsolete. Lost keypoints are the main cause of inaccuracy as witnessed by the last columns of the table. It is worth noting that the keypoint loss is more affected by radio packet loss and TCP re-transmissions due to congestions than by the network latency introduced in Column 4. This is one of the benefits of Algorithm 1, which considers inter-arrival delays to minimize the number of rejected messages.

The last 2 columns demonstrate that the proposed pose estimation platform keeps errors within reasonable limits except for bandwidth lower than 10 Mb/s or radio packet loss rate higher than 10%, which can be considered unusual conditions in forthcoming networks. Vice versa, an architecture that does not offload pose estimation at the edge is much less robust because of the huge amount of traffic sent over the channel.

In summary, the proposed method has been demonstrated to enhance the accuracy of 3D human pose estimation in intricate environments characterized by multiple occlusions, such as industrial plants. This improvement is achieved while minimizing latency, ensuring privacy, and reducing transmission network load. Our system, despite employing a lightweight neural network model for edge computing, demonstrates comparable performance to the Openpose model, which achieves superior results in the 1-cam scenarios. Considering this, as part of our future work, we intend to explore lightweight 2D pose estimators to enhance our results further.

[3] https://man7.org/linux/man-pages/man8/tc.8.html.
[4] https://man7.org/linux/man-pages/man8/tc-netem.8.html.

5. Conclusion

This article addressed the challenges related to the adoption of human pose estimation platforms in real industrial applications. It focused on the problem of occlusions, on the solution based on multi-camera and 3D HPE at the edge, and on the problem of data aggregation and synchronization. It presented *BeFine*, a platform that implements such a distributed 3D HPE system and a two-level synchronization mechanism to deal with delays of real communication networks. With an extended set of experimental results in real working scenarios, the article compared the accuracy achieved by the proposed platform with centralized approaches at the state of the art by using an infra-red motion capture system as ground truth. Furthermore, the work showed that the reduction of communication traffic due to a well-designed offloading enables the introduction of pose estimation in less ideal communication scenarios.

CRediT authorship contribution statement

Michele Boldo: Conceptualization, Data curation, Investigation, Formal analysis, Software, Writing – original draft, Writing – review & editing, Visualization. **Mirco De Marchi:** Conceptualization, Data curation, Investigation, Formal analysis, Software, Writing – original draft, Writing – review & editing, Visualization. **Enrico Martini:** Conceptualization, Data curation, Investigation, Formal analysis, Software, Writing – review & editing, Visualization. **Stefano Aldegheri:** Conceptualization, Data curation, Investigation, Formal analysis, Software, Visualization. **Davide Quaglia:** Conceptualization, Methodology, Writing – review & editing. **Franco Fummi:** Conceptualization, Methodology, Writing – review & editing. **Nicola Bombieri:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This study was carried out within the PNRR research activities of the consortium iNEST (Interconnected North-Est Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.5 –D.D. 1058 23/06/2022, ECS 00000043). This manuscript reflects only the Authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

Cao, Z., Simon, T., Wei, S., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. of IEEE conference on computer vision and pattern recognition* (pp. 1302–1310).

Carraro, M., Munaro, M., Burke, J., & Menegatti, E. (2019). Real-time marker-less multi-person 3D pose estimation in RGB-DEPTH camera networks. *Advances in Intelligent Systems and Computing*, *867*, 534–545.

Chen, L., Ai, H., Chen, R., Zhuang, Z., & Liu, S. (2020). Cross-view tracking for multi-human 3D pose estimation at over 100 fps. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 3276–3285).

Chen, S., Xu, Y., & Zou, B. (2023). Prior-knowledge-based self-attention network for 3D human pose estimation. *Expert Systems with Applications*, *225*, Article 120213.

Dong, J., Fang, Q., Jiang, W., Yang, Y., Huang, Q., Bao, H., et al. (2021). Fast and robust multi-person 3D pose estimation and tracking from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*.

Fang, H.-S., Xu, Y., Wang, W., Liu, X., & Zhu, S.-C. (2018). Learning pose grammar to encode human body configuration for 3D pose estimation. In *Proc. of AAAI conference on AI* (pp. 6821–6828).

Guo, Y., Deligianni, F., Gu, X., & Yang, G.-Z. (2019). 3-D canonical pose estimation and abnormal gait recognition with a single RGB-D camera. *IEEE Robotics and Automation Letters*, *4*(4), 3617–3624.

Hartley, R. (2004). *Multiple view geometry in computer vision*. Cambridge, UK New York: Cambridge University Press.

Hong, S., & Kim, Y. (2018). Dynamic pose estimation using multiple RGB-D cameras. *Sensors (Switzerland)*, *18*(11), 1–17.

Kidzinski, L., Yang, B., Hicks, J., Rajagopal, A., Delp, S., & Schwartz, M. (2020). Deep neural networks enable quantitative movement analysis using single-camera videos. *Nature Communications*, *11*(1).

Lee, J. W., Cho, S., Liu, S., Cho, K., & Helal, S. (2015). Persim 3D: Context-driven simulation and modeling of human activities in smart spaces. *IEEE Transactions on Automation Science and Engineering*, *12*(4), 1243–1256.

Li, J., Liu, G., Tian, G., Zhu, X., & Wang, Z. (2018). Distributed RGBD camera network for 3D human pose estimation and action recognition. In *International conference on information fusion* (pp. 1554–1558). ISIF.

Lim, J., et al. (2021). Designing path of collision avoidance for mobile manipulator in worker safety monitoring system using reinforcement learning. In *IEEE ICSR* (pp. 94–97).

Liu, H., Chen, Y., Zhao, W., Zhang, S., & Zhang, Z. (2021). Human pose recognition via adaptive distribution encoding for action perception in the self-regulated learning process. *Infrared Physics & Technology*, *114*, Article 103660.

Liu, S., Gao, D., Wang, P., Guo, X., Xu, J., & Liu, D.-X. (2018). A depth-based weighted point cloud registration for indoor scene. *Sensors*, *18*(11), 3608.

Liu, G., Liu, T., Tian, G., & Ji, Z. (2019). Distributed human 3D pose estimation and action recognition. In *IEEE international conference on robotics and biomimetics, ROBIO 2019* (pp. 2316–2321).

Liu, T., Liu, H., Yang, B., & Zhang, Z. (2024). LDCNet: Limb direction cues-aware network for flexible human pose estimation in industrial behavioral biometrics systems. *IEEE Transactions on Industrial Informatics*, 1–11.

Liu, H., Liu, T., Zhang, Z., Sangaiah, A. K., Yang, B., & Li, Y. (2022). ARHPE: Asymmetric relation-aware representation learning for head pose estimation in industrial human–computer interaction. *IEEE Transactions on Industrial Informatics*, *18*(10), 7107–7117.

Liu, T., Wang, J., Yang, B., & Wang, X. (2021). NGDNet: Nonuniform Gaussian-label distribution learning for infrared head pose estimation and on-task behavior understanding in the classroom. *Neurocomputing*, *436*, 210–220.

Liu, H., Zhang, C., Deng, Y., Liu, T., Zhang, Z., & Li, Y.-F. (2023). Orientation cues-aware facial relationship representation for head pose estimation via transformer. *IEEE Transactions on Image Processing*, *32*, 6289–6302.

Martin, J., Burbank, J., Kasch, W., & Mills, P. D. L. (2010). *Network time protocol version 4: Protocol and algorithms specification: RFC 5905*, RFC Editor.

Moghaddam, Z., & Piccardi, M. (2014). Training initialization of hidden Markov models in human action recognition. *IEEE Transactions on Automation Science and Engineering*, *11*(2), 394–408.

NVIDIA AI IoT (2020). Tensor RT pose estimation. https://github.com/NVIDIA-AI-IOT/trt_pose.

Palermo, M., Moccia, S., Migliorelli, L., Frontoni, E., & Santos, C. P. (2021). Real-time human pose estimation on a smart walker using convolutional neural networks. *Expert Systems with Applications*, *184*, Article 115498.

Pavlakos, G., Zhou, X., Derpanis, K., & Daniilidis, K. (2017). Coarse-to-fine volumetric prediction for single-image 3D human pose. In *Proc. of IEEE CVPR* (pp. 1263–1272).

Proia, S., Carli, R., Cavone, G., & Dotoli, M. (2022). Control techniques for safe, ergonomic, and efficient human-robot collaboration in the digital industry: A survey. *IEEE Transactions on Automation Science and Engineering*, *19*(3), 1798–1819.

Remelli, E., Han, S., Honari, S., Fua, P., & Wang, R. (2020). Lightweight multi-view 3D pose estimation through camera-disentangled representation. In *Proc. of IEEE CVPR* (pp. 6039–6048).

Tu, H., Wang, C., & Zeng, W. (2020). VoxelPose: Towards multi-camera 3D human pose estimation in wild environment. *Lecture Notes in Computer Science*, *12346 LNCS*, 197–212.

Virtanen, P., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*, 261–272.

Yeung, K. Y., Kwok, T. H., & Wang, C. C. (2013). Improved skeleton tracking by duplex kinects: A practical approach for real-time applications. *Journal of Computing and Information Science in Engineering*, *13*(4).

Zhang, H., Liao, J.-Y., Paz, D., & Christensen, H. I. (2022). Robust human identity anonymization using pose estimation. In *2022 IEEE 18th international conference on automation science and engineering*. IEEE.

Zhang, X., Zhou, Z., Han, Y., Meng, H., Yang, M., & Rajasegarar, S. (2023). Deep learning-based real-time 3D human pose estimation. *Engineering Applications of Artificial Intelligence*, *119*, Article 105813.

Zhu, C., Han, F., & Yi, J. (2022). Wearable sensing and knee exoskeleton control for awkward gaits assistance. In *2022 IEEE 18th international conference on automation science and engineering*. IEEE.