Computer Methods and Programs in Biomedicine 225 (2022) 107016

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

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

Enabling Gait Analysis in the Telemedicine Practice through Portable and Accurate 3D Human Pose Estimation

Enrico Martini[a], Michele Boldo[a], Stefano Aldegheri[a], Nicola Valè[b], Mirko Filippetti[b], Nicola Smania[b], Matteo Bertucco[b], Alessandro Picelli[b], Nicola Bombieri[a,*]

[a] Department of Computer Science, University of Verona, Italy
[b] Neuromotor and Cognitive Rehabilitation Research Center (CRRNC) - Department of Neurosciences, Biomedicine and Movement Sciences, University of Verona, Italy

ARTICLE INFO

Article history:
Received 21 January 2022
Revised 24 June 2022
Accepted 7 July 2022

Keywords:
Markerless Gait Analysis
3D Human pose estimation
Edge Computing
Telemedicine
Embedded systems
Portable Gait Analysis systems

ABSTRACT

Human pose estimation (HPE) through deep learning-based software applications is a trend topic for markerless motion analysis. Thanks to the accuracy of the state-of-the-art technology, HPE could enable gait analysis in the telemedicine practice. On the other hand, delivering such a service at a distance requires the system to satisfy multiple and different constraints like accuracy, portability, real-time, and privacy compliance at the same time. Existing solutions either guarantee accuracy and real-time (e.g., the widespread OpenPose software on well-equipped computing platforms) or portability and data privacy (e.g., light convolutional neural networks on mobile phones). We propose a portable and low-cost platform that implements real-time and accurate 3D HPE through an embedded software on a low-power off-the-shelf computing device that guarantees privacy by default and by design. We present an extended evaluation of both accuracy and performance of the proposed solution conducted with a marker-based motion capture system (i.e., Vicon) as ground truth. The results show that the platform achieves real-time performance and high-accuracy with a deviation below the error tolerance when compared to the marker-based motion capture system (e.g., less than an error of 5° on the estimated knee flexion difference on the entire gait cycle and correlation $0.91 < \rho < 0.99$). We provide a proof-of-concept study, showing that such portable technology, considering the limited discrepancies with respect to the marker-based motion capture system and its working tolerance, could be used for gait analysis at a distance without leading to different clinical interpretation.

1. Introduction

Human pose estimation (HPE) is a key component for human motion analysis from RGB images and videos [1]. Among the many application fields, such as, sport performance analysis, human computer interaction, and action recognition, there is a growing interest in applying such a computer vision technology for the analysis of pathological gait detection [2–6]. The interest is mainly due to the fact that, differently from other solutions like multi-camera motion capture systems, multiple inertial measurement units (IMU), force plate, and pressure insoles, HPE does not require participants to wear markers or specialized sensors and can be rapidly applied without complicated setup.

Thanks to the advances in the convolutional neural networks (CNNs), there have been significant improvements in the accuracy of the 3D HPE. Different techniques rely on the training of end-to-end CNNs to predict the 3D human pose from each image (i.e., video frame) [7]. Other techniques achieve comparable accuracy by estimating the 2D pose first and then lifting the 2D pose to view-invariant 3D pose [8].

On the other hand, long latencies of the image processing (i.e., execution time of the *inference* phase) lead the system to work at low frequency and, as a consequence, to a sensible decrease in the overall estimation accuracy [9]. As both direct (3D) and indirect (2D + Depth) methods are computationally demanding, they require well-equipped and advanced architectures (i.e., multi-core CPUs supported by a GPU accelerator) to achieve accuracy and *real-time processing*.

* Corresponding author.
E-mail addresses: enrico.Martini@univr.it (E. Martini), michele.Boldo@univr.it (M. Boldo), stefano.Aldegheri@univr.it (S. Aldegheri), nicola.Vale@univr.it (N. Valè), mirko.Filippetti@univr.it (M. Filippetti), nicola.Smania@univr.it (N. Smania), matteo.Bertucco@univr.it (M. Bertucco), alessandro.Picelli@univr.it (A. Picelli), nicola.Bombieri@univr.it (N. Bombieri).

https://doi.org/10.1016/j.cmpb.2022.107016
0169-2607/

One of the main challenges is to adopt markerless gait analysis through CNN-based software (SW) applications in telemedicine. Telemedicine comprises medical care and professionals that adopt information and communication technologies in order to provide assessment, diagnosis, goal setting, treatment education, and monitoring through remote devices [10,11]. As to motion analysis, telehealth approaches are useful to provide evaluations in the management of conditions affecting movement (for example orthopedic or neurologic diseases, including, fractures, amputations, stroke, Parkinson disease, multiple sclerosis) and their follow-up. Furthermore, telehealth technologies for motion analysis may be useful for rehabilitation in order to assess body function limitations due to posture or movement alterations as well as to monitor execution and effects of physical exercises in patients suffering from disabling conditions.

Nevertheless, delivering gait analysis at a distance through cameras and HPE software requires the system to satisfy, beside accuracy and real-time, also portability and privacy compliance at the same time [12]. Addressing all these non-functional constraints in a seameless way requires the HPE software to execute on Internet-of-Things (IoT) devices *at the edge* [13], by which the video streams (i.e., the sensitive information) is elaborated close to the input sensor (i.e., the camera), while only the process results can be stored or sent over the communication network (i.e., Ethernet, WiFi) [14].

In this context, IoT devices for edge computing are *resource-constrained architectures*, in which memory and computing capability are bounded to guarantee portability, energy efficiency, and low-power consumption. State-of-the-art software solutions for HPE either do not run on these devices due to memory limitations or they achieve very low working frequencies (i.e., far from real-time)[1], which can lead to very low accuracy in the kinetic gait analysis. Although some solutions have been proposed to apply HPE on mobile devices [16], they achieve real-time performance at the cost of coarse pose estimation (i.e., not adapt for clinical gait analysis) due to memory or computing power limitations [17].

In this work, we address the problem of satisfying multiple and different non-functional constraints like real-time computing, accuracy, privacy compliance, and portability for remote gait analysis. We present a portable embedded system (see Fig. 1(a)) that implements a real-time accurate 3D HPE SW application with a low-cost camera and a low-power computing device.

The main contributions of the work, which are the key elements of the proposed system, are the following:

- We reimplemented a *2D* HPE engine to take advantage of the heterogeneous computing elements (i.e., CPU and integrated GPU) of an off-the-shelf portable computing device (i.e., NVIDIA Jetson board) to achieve real-time computation *at the edge*.
- We combined the inference phase with different image elaboration blocks to improve the accuracy while guaranteeing real-time and reduced memory footprint of the software. We selected a set of spatio-temporal filters and an interpolation procedure to extend the HPE from 2D to 3D and we implemented them thorough optimized parallel code to take advantage of multi-level parallelism on the device.
- We present a quantitative analysis of the proposed platform accuracy in extrapolating both the 3D positions of human joints and the joint angles. We present an extended comparison with: (i) an infrared marker-based motion capture system (i.e., Vicon), and (ii) the OpenPose software application executed on a high-performance computing server to evaluate the applicability of the platform for gait analysis.

[1] The most widespread HPE software, OpenPose [15], elaborates 6.3 frames per second over the 60 frames per second of the input camera on the most powerful NVIDIA embedded platform (NVIDIA Xavier AGX).

The article is organized as follows. Section 2 presents the analysis of the related work. Section 3 presents the proposed portable HPE platform. Section 4 reports the experimental results. Section 5 presents the discussion of the obtained results, while Section 6 is devoted to the conclusions.

2. Related Work

Measuring gait variables using computer vision has been increasingly applied in the recent years to assess mobility and risks of fall [5,18] as well as to identify gait features in parkinsonism and other neurological diseases [19,20]. Quantifying gait pathology with commodity cameras increases access to quantitative motion analysis in clinics and at home and enables researchers to conduct large-scale studies of neurological and musculoskeletal disorders [21]. The same technology has been applied to measure the gait changes over time in hospitalized older adults with advanced dementia [22]. In [23], the authors combine expertise and perspectives from physical therapy, speech-language pathology, movement science, and engineering to provide insight into applications of pose estimation in human health and performance.

Several solutions have been also proposed to combine human pose estimation with CNN architectures for classification between normal and pathological gait in humans, with ability to provide information about the detected alterations from the extracted skeletal [24]. To achieve high accuracy, the majority of these solutions rely on OpenPose [15], an open source framework that uses a non-parametric representation (i.e., part affinity fields - PAFs) to associate body parts with individuals in the image.

Although all these solutions have demonstrated strong benefits and applicability in real-life situations, they are limited from the *two-dimensional* HPE. Some solutions adopt the Kinect sensor to estimate the 3D skeleton. In [25], the authors compared the accuracy of the new Azure Kinect DK and shown that such a sensor is highly accurate in tracking upper body movements. In [26], the authors applied Kinect v2-based gait analysis for children with cerebral palsy.

More advanced solutions have been recently proposed to implement 3D HPE with RGB cameras. An example is the mobile system presented by Guo et al. [6] to track humans and analyze their gait in canonical coordinates based on a single RGB-D camera. To alleviate the effects of viewpoint diversity, Wei et al. proposed a view-invariant 3D HPE module [9]. Zimmermann et al. demonstrated that HPE by leveraging both RGB and depth images performs better than using depth data alone [27].

In [22], the authors compared the accuracy of three different 2D HPE software (AlphaPose, OpenPose, Detectron) for gait analysis by considering a 3D motion capture system as golden model. They underlined that there are important opportunities to evaluate models capable of 3D pose estimation in video data, improve the training of pose-tracking algorithms for older adult and clinical populations, and develop video-based 3D pose trackers specifically optimized for quantitative gait measurement. A similar comparison between 2D HPE with Openpose and the Vicon motion capture system has been presented in [28]. The authors evaluated the software accuracy to measure pelvic segment angles, hip, knee, and ankle joint angles during treadmill walking and running. An orthogonal analysis of the 2D HPE accuracy has been conducted in [29]. The results underline that such a CNN-based method for extraction of gait parameters from video appears suitable for valid and reliable quantification of gait.

All the previous 3D HPE solutions requires well equipped and powerful computing architectures (i.e., high-performance computing servers with multi-core CPUs supported by GPU accelerators) to work in real-time. They do not run on resource constrained

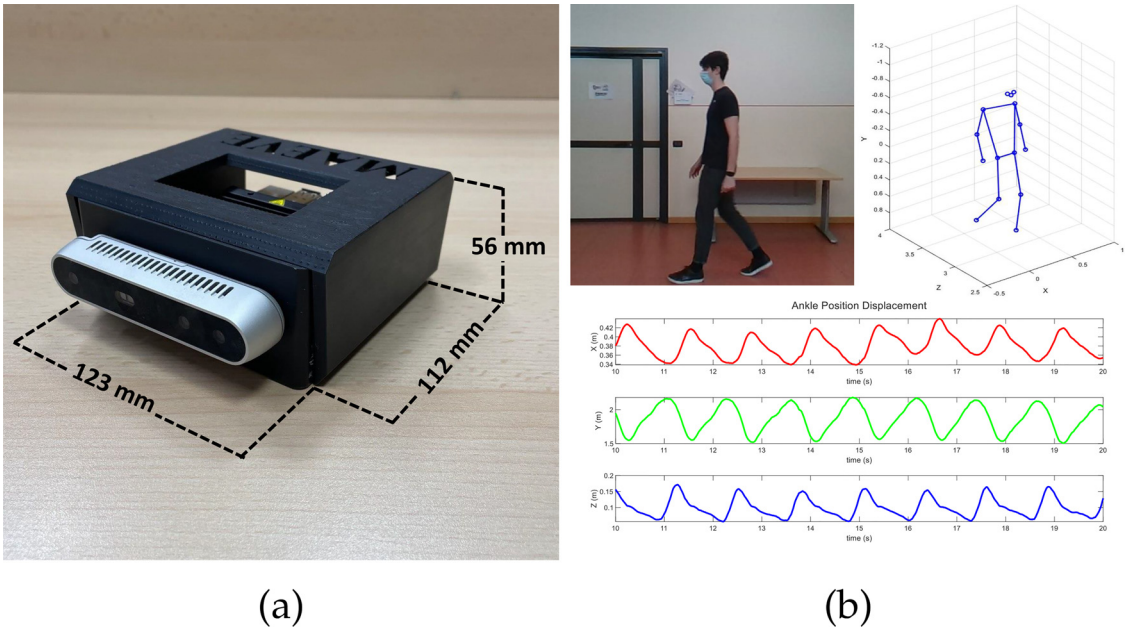


Fig. 1. (a): The proposed portable system (*MAEVE*) implemented with a 3D HPE software application, an RGB+Depth Intel Realsense camera, and an NVIDIA Jetson Xavier device. (b): Example of the output results represented by a 3D virtual skeleton (top) and a detailed temporal representation of the right ankle position (keypoint) in the three dimensions (bottom).

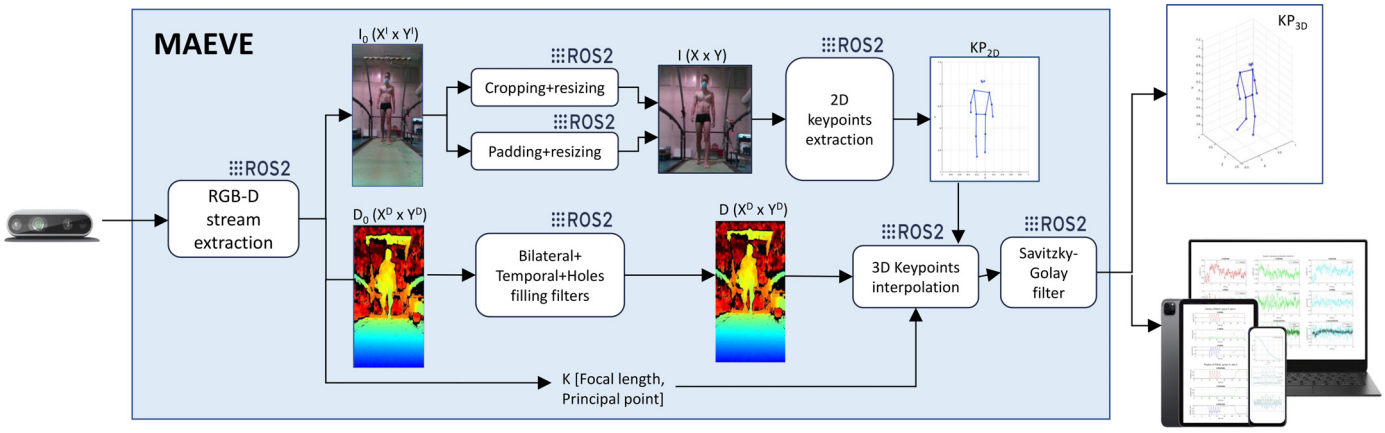


Fig. 2. Overview of the *MAEVE* platform

computing devices or they achieve very low accuracy due to sub-sampling of the input signal.

Some platforms have been proposed to apply *2D* HPE on mobile devices [16,17]. Nevertheless, they achieve real-time performance at the cost of low accuracy.

The platform proposed in this article implements real-time 3D HPE on a low-power low-cost portable platform. We provide an analysis of the HPE accuracy when used on a off-the-shelf embedded device to understand the reliability of such a marker-less solution for the analysis of pathological gait detection.

3. Method

Fig. 2 shows an overview of the proposed platform (*MAEVE*). It consists of communicating and concurrent elaboration nodes that implement a pipeline of computer vision primitives and inference-based applications. The pipeline allows for an efficient processing of the frame sequence (i.e., video stream), which are generated by a RGB-D cameras and elaborated, in real time, on the heterogeneous computing elements of the embedded boards (i.e., CPU and GPU). The result is a set of *3D keypoints* (KP_{3D}), one set per frame, representing the joints of the human body.

The infrastructure has been implemented compliant to the standard ROS2 [30] to allow for the integration of post-processing modules such as keypoints elaboration and monitoring in the embedded device or, through the network, on mobile phones, tablets or laptops.

3.1. RGB-D stream extraction

The first node implements a synchronized reading of inputs from the cameras (i.e., RGB and Depth) and the input distribution to the heterogeneous computing elements of the edge device (CPU and GPU). The node reads an image ($I_0(X^I \times Y^I)$ in Fig. 2) from the RGB camera, which consists of a 3-channel matrix (8-bit pixels for the Intel RealSense D415 camera adopted in our experimental results) and a depth image from the D camera ($D_0(X^D \times Y^D)$), which represents the depth information (16-bit pixels for the adopted depth camera). The node also combines color and depth information with the intrinsic parameters of the input camera, which consist of focal length (f_x, f_y) and principal point coordinates

(pp_x, pp_y). They are required for the 3D keypoint interpolation, as exaplined in Section 3.4.

3.2. Bilateral, temporal, and holes filling filters

One of the main problem for the 3D accuracy of the system is the intrinsic noise generated by the depth camera [31]. In *MAEVE*, the *noisy* depth information undergoes three filtering algorithms. The filters have been selected and implemented to improve the overall result accuracy without compromising the real-time computing performance:

- *Spatial edge-preserving filter (bilateral)*. Since the platform targets human bodies (i.e., smooth surfaces), we selected a low-pass 2D edge-preserving filter. *MAEVE* implements the algorithm proposed in [32] for CPU, which complexity is linear in time with negligible memory footprint.
- *Temporal filter*. Since HPE is not characterized by very sharp pose variations in short-term time-frames, we selected a temporal-consistent filter to improve the depth data persistency. The implementation is based on the algorithm proposed in [33], which elaborates per-pixel values by using depth information of previous frames. The filter implemented by *MAEVE* performs a single pass on the data by adjusting the depth values and updating the tracking history. When the pixel data is missing or invalid, the filter uses a persistency mode by which a missing value is rectified with stored data if the pixel was valid in two out of the last 4 frames.
- *Holes filling filter*. It aims at rectifying missing data in the resulting image by using information from the neighboring pixels (similarly to the approach proposed in [34]). In *MAEVE*, the filter selects the neighboring pixel as the farthest from the sensor. It is implemented and run through parallel threads on the CPU cores.

3.3. 2D keypoints extraction

This node extracts the 2D *keypoints* from the RGB image, which represent the key joints of the human skeleton. We started from the framework proposed in [35], which implements 2D HPE as a Jupyter Notebook demo. We reimplemented such a CNN-based inference application in C++ and CUDA to take fully advantage of the GPU accelerator of the device. The application relies on a pre-trained CNN (MSCOCO dataset), which can be selected by the user between *Resnet*18 and *Densnet*18 in [35]). Both models provide the same set of 18 keypoints (blue squares in Fig. 3), while they provide slightly different accuracy and real-time performance (see Section 4). Each keypoint is represented as $kp_j = [u_j, v_j, c]$, where $[u_j, v_j]$ is the 2D pixel coordinates and c represents the prediction probability of the joint.

To guarantee real-time performance, differently from the state-of-the-art approaches (e.g., Openpose [36]), the 2D keypoints extraction node does not implement the sliding window process to support rectangular input frames [36]. The node directly implements joint recognition and supports only quadratic images (i.e., 224x224 px with ResNet, 256x256 px with DensNet CNN). MAEVE supports rectangular frames by implementing cropping/padding (i.e., depending on the image orientation) of the input images. The implementation is in CUDA and allows for two levels of parallelism. The first at thread level, by which GPU threads perform the image manipulation in parallel. The second is at node level, by which cropping/padding and resizing are run in parallel to the bilateral, temporal, and holes filter computation.

It is important to note that the 2D HPE proposed in [35] does take advantage of the GPU. Nevertheless, this is not enough as the original core software design and communication protocols are not suited to achieve the aimed real-time performance and accuracy. We translated the whole CNN model from Pytorch to TensorRT and the corresponding interface from Python in C++. This allowed us:

- To re-implement the CPU-GPU communication protocol at the basis of the inference step. We conducted an extended profiling activity to identify the best CPU-GPU communication between standard copy (always adopted in the original framework) and *zero-copy*. Zero copy allows data (RGB images) to be exchanged between CPU and GPU through shared memory and pointers, with no need of physical copies. On the other hand, the zero-copy mechanism switches off both CPU and GPU caches and this often eludes the advantages of such a virtual copy [37]. We conducted a profiling phase to identify when and where to adopt one communication mechanisms or the other.
- To have the complete control of the pipeline tasks across the heterogeneous computing elements (i.e., CPU and GPU) and to implement the pipeline with different levels of parallelism (i.e., on the CPU threads and between CPU and GPU beside the GPU threads).

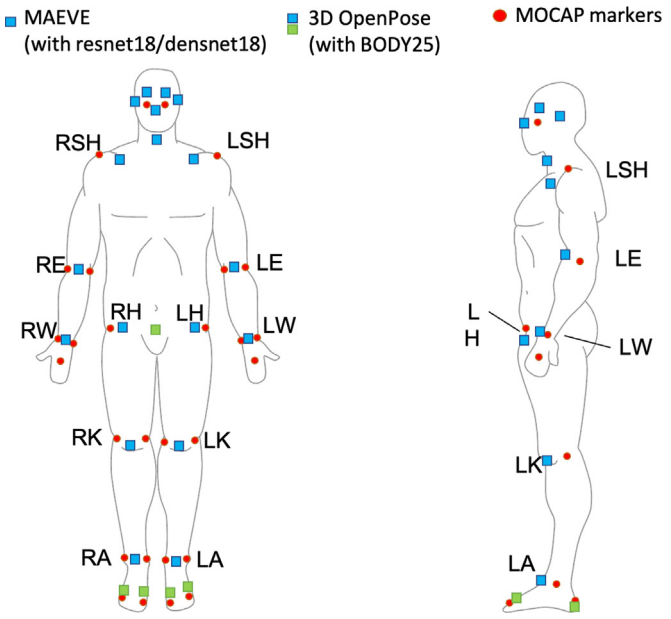


Fig. 3. Comparison of MAEVE/Openpose keypoints, and the Mocap markers. The *BODY_25* keypoint set consists of the blue+green points in the front view + the two points representing the heels in the lateral view.

3.4. 3D keypoint interpolation and filtering

To extract the 3D coordinates in space of each keypoint, this node combines the 2D coordinates (in pixels) received from the 2D keypoint extraction node (KP_{2D}) with the filtered depth matrix ($D(X^D \times Y^D)$). The node implements the alignment of the RGB and depth cameras by using the extrinsic parameters of the two sensors. In particular, the node aligns the depth image to the RGB image through a parallel OpenMP implementation. This allows such a computation to run in parallel on the CPU cores of the embedded platform.

The node then implements a back-projection of the 2D points onto the 3D space as $kp_j = [X_j, Y_j, Z_j]$ based on the pinhole camera model [38] by using the intrinsic parameters of the sensors (K). The final complete information of each keypoint consists of the 3D coordinates in space, the 2D coordinates in pixels and the prediction probability of the keypoint (joint).

Finally, the platform implements an optimized version of the Savitzky-Golay filter [39] to smooth the sequence of the 3D keypoints.

Although the pipeline to extrapolate 3D keypoints from RGB-D data is well established, a platform that implements the pipeline for real-time computation in low-cost portable devices is still missing. In particular, the existing solutions either achieve high accuracy but require high-performance computing platforms, or they achieve very low accuracy. With a motion capture system as ground through, we studied the best trade-off among several combinations of CNN models, image filters, and their corresponding (parallel) implementations for heterogeneous architectures (i.e., multi-core CPUs and GPUs) to reach enough accuracy for motion analysis at a distance. Section 4 shows the obtained results.

4. Experimental results

We evaluated *MAEVE* in terms of real-time computing performance on different off-the-shelf embedded boards. In every configuration, the input camera consists of an Intel RealSense D415 1920x1080 RGB frame + 1280x720 depth frame running at 60 frames per second (FPS). We then evaluated the *MAEVE* accuracy in two ways. First, by comparing the absolute 3D coordinates of the joint keypoints extrapolated by *MAEVE* and the marker coordinates extrapolated by an infra-red motion capture system during the gait cycles. Then, by comparing the angular displacements for the joint flexion extrapolated by the HPE systems.

4.1. Results of real-time computation

We compared the runtime computing performance of *MAEVE* (both with the Resnet18 and Densnet 18 CNNs for the 2D keypoint extraction) with 3D *Openpose*. For *Openpose*, we adopted the BODY25 CNN model since the fastest and most accurate CUDA version for GPUs, as suggested by the authors [40]. Since the cropping/padding and resizing nodes are not necessary with *Openpose* (see Section 3.3), they have been turned off to save CPU bandwidth.

We evaluated the HPE software (*MAEVE* and *Openpose*) on three different embedded devices: An Nvidia Jetson Nano (4-cores CPU, 128-cores GPU, 4 GB RAM), an Nvidia Jetson TX2 (6-cores CPU, 256-cores GPU, 8 GB RAM), and an Nvidia Jetson Xavier AGX (8-cores CPU, 512-cores GPU, 32 GB RAM).

Table 1 reports the runtime performance of the HPE nodes and of the overall HPE application running on the different embedded devices. As expected, the 2D keypoint extraction is the heaviest node of the platform and strongly characterizes the overall performance of the 3D HPE platform.

Openpose does not run on the Nano and TX2 devices due to memory limitations. In the most powerful embedded device (i.e., Xavier AGX), *Openpose* achieves very low working frequency and, as a consequence, it leads to sensible subsampling of video frames (i.e., 6.3 FPS vs. 60 FPS of the input cameras).

With *MAEVE*, the Resnet CNN model provides performance higher than Densnet at the cost of slightly lower accuracy in the final 3D HPE (see the accuracy evaluation sections). The communication overhead marginally impacts on the overall performance. The depth filters, which are of paramount importance for the overall accuracy, are the second heaviest node of the platform. However, since they are implemented to run on the board concurrently to the Cropping/Padding+resizing and 2D keypoints extraction, they are not in the critical path of the system execution and their latency is hidden in the overall system performance. *MAEVE* leads to sensible subsampling when run on the very low-power and low-cost Jetson Nano. It achieves real-time performance (i.e., 60 FPS) on the Jetson Xavier.

Table 1
Runtime 3D HPE nodes and system performance.

HPE Framework	3D Openpose			MAEVE (densenet)			MAEVE (resnet)		
Nvidia Jetson	Nano(ms)	TX2(ms)	Xavier(ms)	Nano(ms)	TX2(ms)	Xavier(ms)	Nano(ms)	TX2(ms)	Xavier(ms)
RGBD Stream Extraction	11.3	4.3	2.8	11.3	4.3	2.8	11.3	4.3	2.8
Cropping/Padding + Resizing	3.2	1.8	1.1	3.2	1.8	1.1	3.2	1.8	1.1
Depth Filters	64.5	37.8	23.6	64.5	37.8	23.6	64.5	37.8	23.6
2D Keypoints Extraction	-	-	152.1ms	208.5	70.8	10.5	90.4	37.6	7.6
3D Keypoints Interpolation	4.1	1.9	1.3	4.1	1.9	1.3	4.1	1.9	1.3
Savitzky-Golay Filter	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
Communication overhead	4.7	4.1	3.9	4.7	4.1	3.9	4.7	4.1	3.9
Overall runtime	-	-	160.1 (6.3 FPS)	231.8 (4.3 FPS)	82.9 (12.1 FPS)	19.6ms (50.9FPS)	113.1 (8.8 FPS)	49.7 (20.1 FPS)	16.7 (59.9 FPS)

4.2. Spatial accuracy of the 3D keypoint coordinates

We evaluated the accuracy of *MAEVE* first by comparing the keypoints provided when run in real-time on the Jetson AGX with those extrapolated by an 8-camera marker-based motion capture system (Mocap) (MX 13, VICON, Oxfordshire, UK). Through the Mocap, kinematic data were collected at 100 *Hz* and reflective markers (14 *mm* in diameter) were placed over the following bony landmarks bilaterally: Cheekbones, acromion, medial and lateral epicondyle of the humerus, radial and ulnar styloid processes, dorsum of hand, greater trochanter, lateral and medial epicondyle of the femur, lateral and medial malleolus ankle, heel, tip of the big toe and the 5th metatarsal of the foot.

Considering the experimental setting of this study and its aims, we decided to use the marker set showed by Fig. 3 in order to estimate joint positions as the midpoint between the lateral and medial landmarks of wrist, elbow, knee and ankle. This was because the goal was to compare at best the *MAEVE* and Mocap evaluations by positioning reflective markers as near as possible to the keypoints estimated by the portable system. Data were digitized and processed using Nexus (VICON, Oxfordshire, UK). Measures derived from the Mocap were filtered by a lowpass digital Butterworth filter (4th-order, cutoff frequency 15 *Hz*)

In addition, it is worth noting that in a previous study [41] no significant test-retest differences were observed in the sagittal plane knee joint center location among different techniques: a) femoral epicondyle, b) femoral condyle, c) tibial ridge, d) plug-in-gait, and f) functional. The femoral epicondyle technique defined the joint center position as the radius of the lateral femoral epicondyle marker + 0.5 $\times$ knee width. In the same manner, the femoral condyle technique defined the joint center position as the radius of the lateral femoral condyle marker + 0.5 $\times$ knee width.

A group of five healthy adults (three males, two females; 26.8 ± 3.6 years; 1.77 ± 0.1 m height; 75.2 ± 17.1 kg weight), were asked to walk on a treadmill at 4.0 km/h for 35 seconds. We collected data from both the Mocap system and the RGB-D systems simultaneously. The RGB-D camera (Intel Realsense D415) was placed 2.30 m away from the center of the treadmill, recording the coronal plane of the subjects. All measurements were performed consecutively, so that the setup did not change between each acquisition. To ensure a complete temporal synchronization between the video cameras and the Mocap, we implemented the solution proposed in [42], by which an electrical signal was connected to a led lamp. The (manual) change of the electrical source was recorded by the Mocap system and, simultaneously, turned on the led lamp positioned in the field of view of all RGB sensors.

For a correct matching of the 3D coordinates, we considered the keypoints/markers mapping proposed in Fig. 3, which shows the 18 keypoints common to the three RGB-D-based systems (*MAEVE* with Resnet, *MAEVE* with Densnet, and 3D *Openpose*) thorugh blue squares. The green squares represent the additional keypoints extrapolated by *Openpose*. The red dots represent the position of the Mocap markers.

We also included *Openpose* in the accuracy comparison. Since it cannot correctly run in real-time in the portable device, we evaluated the best accuracy of *Openpose* by running the software off-line (i.e., to avoid any input frame dropping) on a high-performance computing server equipped with an Intel i5 7400 CPU, 2xNvidia RTX 2070 GPUs (SLI), 16GB DDR4 RAM, and Ubuntu 18.04 LTS operating system.

We measured the Euclidean distance between each keypoint and the corresponding Mocap marker (d in the rightmost side of Fig. 4) and then extrapolated the standard deviation of such a value. In case the Mocap relies on two markers to identify the joint center (i.e., wrist, elbow, knee and ankle), we considered the coordinates of the midpoint between the lateral and medial landmarks), as shown in the leftmost side of Fig. 4.

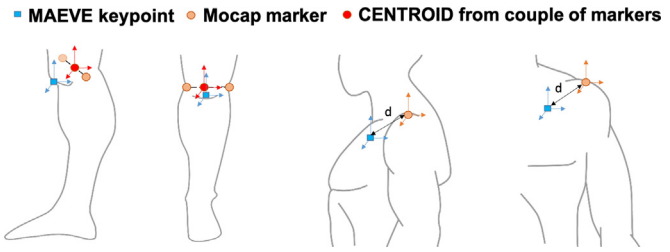


Fig. 4. MAEVE keypoints and Mocap markers positions.

Table 2
Median standard deviation of the euclidean distance between MAEVE 3D keypoints, 3D Openpose keypoints, and Mocap markers.

Keypoint	OpenPose 3D	MAEVE Densenet	MAEVE Resnet
LSH	1.275 cm	0.833 cm	1.550 cm
RSH	1.276 cm	1.368 cm	1.522 cm
LE	1.355 cm	1.528 cm	1.873 cm
RE	1.898 cm	1.582 cm	1.898 cm
LW	1.879 cm	1.903 cm	2.053 cm
RW	2.215 cm	2.210 cm	1.815 cm
LH	1.197 cm	1.016 cm	0.998 cm
RH	1.551 cm	1.522 cm	1.495 cm
LK	2.297 cm	2.643 cm	2.767 cm
RK	2.396 cm	2.240 cm	2.050 cm
LA	4.308 cm	4.340 cm	5.267 cm
RA	4.063 cm	3.863 cm	5.187 cm

Fig. 5 shows the average standard deviation referred to the subjects of the test. A Friedman non-parametric test for repeated measures did not show significant effects for the pose estimation methods ($\chi^2(2) = 3.10$, $p = 0.212$) on the standard deviation (SD), that is, not significant difference was found in the spatial accuracy among the three pose methods related to the joints coordinates measured with the Mocap. On average, the standard deviation was 2.1, 2.4 and 2.1 cm for *Openpose, MAEVE* with resnet and *MAEVE* with densenet respectively (see Table 2). Conversely, a significant effect was found for the keypoints ($\chi^2(11) = 117$, $p < 0.001$) on the standard deviation. The Durbin-Conover test for multiple comparisons reported lower standard deviations related to the most stationary keypoints during the gait cycle (shoulder, elbow and hip joints). The difference increases on the less stable joints (i.e., knees and ankles). In particular, the further the keypoints are away from the trunk, the higher is the relative distance between keypoints and the Mocap markers.

The experimental results in Fig. 5 represent the complete evaluation of the platform accuracy in measuring the 3D spatial position of the human joints depicted in Fig. 3 w.r.t. the MoCap. This is the first primary focus of the work, i.e., comparing keypoints given by a markerless platform (which allows for privacy-aware motion analysis in telemedicine) and the corresponding marker-based points given by the MoCap.

4.3. Accuracy of joint angle displacement

Starting from the 3D spatial position of the keypoints, we also extrapolated, geometrically, the joint angles. The aim was to evaluate the impact of the deviation between couples of 3D keypoints (marker-less vs. marker-based) on the extrapolation of joint angles. The joint angles were extrapolated geometrically from triples of keypoints (e.g., ankle, knee, and hip keypoints for the knee angle), according to [43]. The impact on other marker setups (e.g., plugin gait) are part of our current and future work.

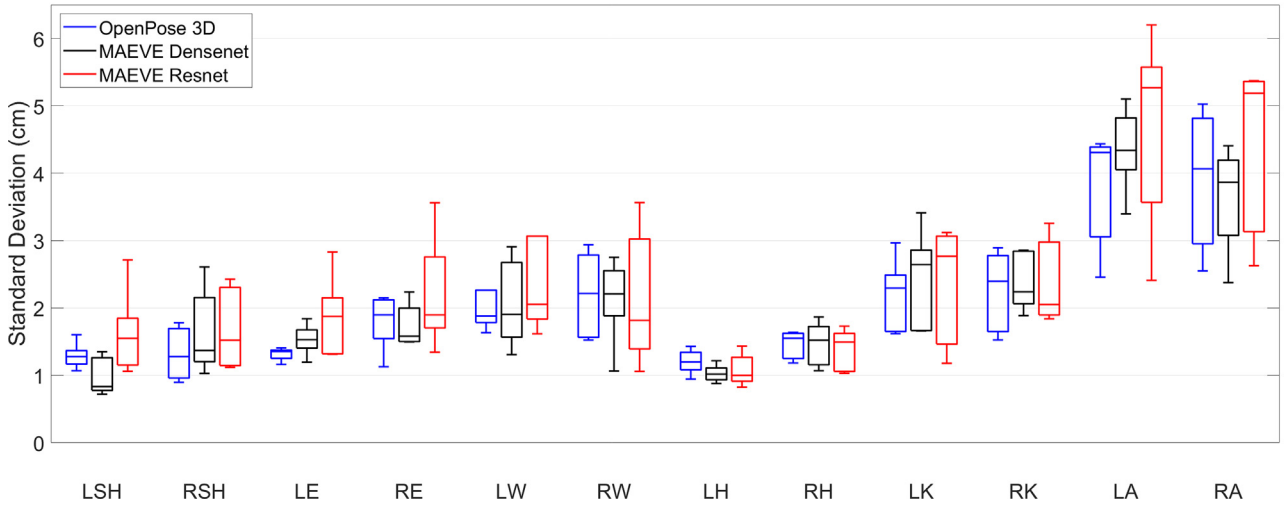


Fig. 5. Distribution of the standard deviation between MAEVE 3D keypoints, 3D Openpose keypoints, and Mocap markers.

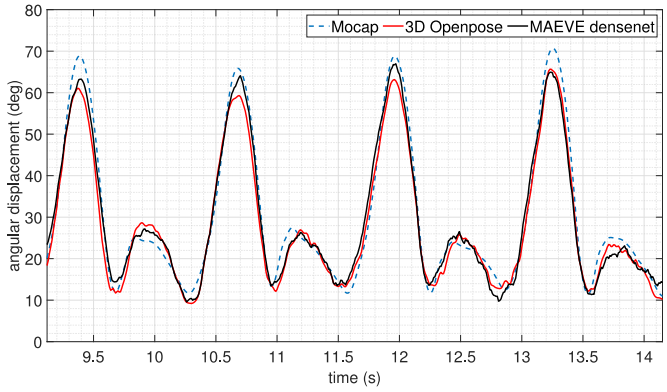


Fig. 6. Knee angular displacement of MAEVE (real-time on Jetson AGX), 3D Openpose (off-line on server), and Mocap.

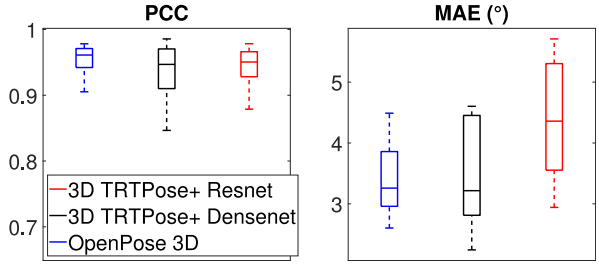


Fig. 7. Distribution of PCC and MAE between MAEVE 3D keypoints, 3D Openpose keypoints, and Mocap markers.

For the sake of space, we report the results obtained for two representative joint angles. First, the knee joint angle as (i) it is one of the most representative and clinically meaningful in the context of walking, and (ii) its extrapolation relies on one of the most sensitive 3D keypoints (i.e., the ankle keypoint). The magnitude and polarity of the angle was described such that when the knee was fully extended, it was described as 0 degrees flexion, and when the shank moved to a posterior direction relative to the thigh, the knee joint angle was said to be in flexion (knee angle > 0). Two vectors were generated from these points in order to get the orientation of the knee articulation. The hip keypoint was then defined in the knee reference system. In this way, the angle corresponding to the sagittal plane was derived through standard trigonometric formulas.

Fig. 6 shows the correlation among the knee angle extrapolated by *MAEVE* in real-time on the embedded board, *Openpose* off-line on the server, and the Mocap. The three traces match in almost the whole gait cycle, with the most markable difference in the knee flexion. To measure the correlation degree between curves, a cross correlation at 0 lag was computed between the knee angles calculated with Mocap and the pose estimation methods. There was a high similarity of the knee angles measured with Mocap and the pose estimation methods for all subjects (Table 3). The Friedman non-parametric test for repeated measures did not show significant differences for the pose estimation methods ($\chi^2(2) = 3.00$, $p = 0.223$) and body side factors ($\chi^2(1) = 3.00$, $p = 0.083$) on the correlation coefficient. To measure the correlation degree of the two curves, we used the Pearson Correlation Coefficient (PCC) as it is invariant under changes in location and scale. The estimation error is given by the Mean Absolute Error (MAE).

The Friedman test did not find significant difference for the pose estimation methods ($\chi^2(2) = 0.800$, $p = 0.670$) and the body side (left vs right) ($\chi^2(1) = 0.600$, $p = 0.439$) on the knee joint angle MAE (Fig. 7).

To sum up, Table 3 (upper box) reports the correlation (PCC) and MAE of the RGB-D systems w.r.t. the Mocap system in measuring the knee angle. We found that both the correlation and MAE values between *MAEVE* and Mocap are comparable to those between 3D *OpenPose* and Mocap. On the other hand, differently from 3D *Openpose*, *MAEVE* provides such values by running on the portable device in real-time.

Table 3 also reports the results obtained in measuring the hip angles (lower box). The results underline that, in general, the platform achieve more accurate measures of angles starting from more accurate keypoints. In the CNN model adopted in this version of the platform, the forefoot keypoints are not included. As a consequence, the platform cannot extrapolate the ankle angles. The extension of the model with a larger set of keypoints (including forefoots) and the corresponding evaluation is part of our future work.

Table 3
Value of Pearson Correlation Coefficient (PCC) and Mean Absolute Error (MAE) between MAEVE 3D keypoints, 3D Openpose keypoints, and Mocap markers.

Subject	Left Knee						Right Knee					
	OpenPose 3D		MAEVE Densenet		MAEVE Resnet		OpenPose 3D		MAEVE Densenet		MAEVE Resnet	
	PCC	MAE	PCC	MAE	PCC	MAE	PCC	MAE	PCC	MAE	PCC	MAE
s0	0.942	3.038	0.928	2.812	0.928	4.865	0.971	2.96	0.927	2.998	0.97	3.196
s1	0.899	3.378	0.91	4.452	0.797	5.71	0.936	3.932	0.956	3.43	0.941	3.853
s2	0.949	3.562	0.97	2.616	0.951	3.591	0.955	2.858	0.96	2.243	0.955	2.941
s3	0.905	4.489	0.906	4.256	0.827	5.627	0.969	2.603	0.968	3.003	0.92	3.553
s4	0.963	3.859	0.865	4.603	0.966	5.151	0.951	3.139	0.897	4.569	0.945	5.104
Subject	Left Hip						Right Hip					
	OpenPose 3D		MAEVE Densenet		MAEVE Resnet		OpenPose 3D		MAEVE Resnet		MAEVE Densenet	
	PCC	MAE	PCC	MAE	PCC	MAE	PCC	MAE	PCC	MAE	PCC	MAE
s0	0.955	3.719	0.936	2.546	0.920	4.472	0.937	2.686	0.906	2.303	0.983	3.787
s1	0.904	3.671	0.922	3.394	0.892	4.932	0.969	2.415	0.935	3.153	0.968	3.234
s2	0.932	2.351	0.905	2.377	0.934	3.041	0.988	2.011	0.905	2.969	0.937	2.054
s3	0.938	2.982	0.871	3.051	0.969	3.602	0.951	3.209	0.917	2.402	0.894	2.169
s4	0.968	2.968	0.902	2.266	0.931	2.077	0.915	2.379	0.953	2.195	0.912	2.424

5. Discussion

In our experimental analysis, the HPE software at the state of the art (i.e., *Openpose*) has shown to achieve very high accuracy at the cost of high computational load and memory footprint. This is due to the fact that, to achieve high accuracy, the core of the pose estimation framework requires a deep and convolutional neural network architecture. As a consequence, such an inference application either does not run on memory bounded portable devices or, when run on the most efficient and powerful devices (e.g., NVIDIA Jetson Xavier AGX), it achieves very low working frequency (e.g., it can process 6.3 video frames per second on the Jetson AGX). Even though each set of keypoints extrapolated by *Openpose* is accurate, the overall real-time results are very inaccurate due to subsampling of the input video frames. Fig. 8a shows, as an example, the effect of such a subsampling on the knee angle obtained by *Openpose* on the Jetson AGX and its correlation with the same trace obtained with the Mocap.

For the same reason, the Jetson Nano does not allow *MAEVE* to achieve enough accuracy *and* real-time computation (see Fig. 8b). Even though the Jetson TX2 device improves the supported working frequency (Fig. 8c), further investigation is required to identify which movements in the gait analysis can be evaluated without leading to different clinical interpretations.

In contrast, *MAEVE* achieves the best accuracy on the Jetson Xavier AGX device. In this configuration, it achieves real-time computation and supports the frame rate (60 FPS) of the input cameras (see Fig. 8d).

As far as it is concerned the spatial accuracy of the coordinates, the estimation of the keypoints with three pose methods (i.e., *Openpose* off-line on the server, *MAEVE Densnet* and *Resnet* in real-time on the Jeston AGX) showed comparable spatial dispersions of the Euclidian distance compared to the coordinates measured with the Mocap. On average, the standard deviation of the Euclidian distance was 2.2 cm, approximately 70% of the trials were < 2.5 cm. The larger values referred to the less stable joints (i.e., knee and ankle joints). Our results suggest a higher consistency of keypoints estimation compared to previous results during walking, countermovement jump and ball throwing movements[44]. The different spatial accuracy across keypoints (Fig. 5) is due to the different speeds of the keypoints during the gait cycle (e.g., hips vs. ankles). This suggests that such a high accuracy of the measures that include the ankles (e.g., knee angle) is guaranteed at medium walking speeds ($\simeq$ 4 Km/h). It could lead to less accurate measures in applications outside the gait analysis context.

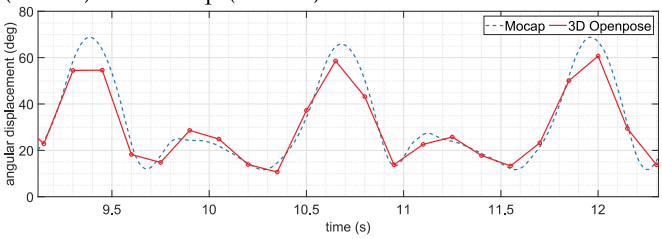
We then focussed in the accuracy of the platform in measuring the flexion-extension knee joint angle. Although this measure alone does not represent a full gait analysis evaluation, it is a key step towards for enabling the human motion measurements for gait analysis at a distance. The knee joint angle patterns computed from the keypoints estimated with our pose methods demonstrated a very strong correlation (on average, $\rho > 0.94$) with the knee angles time profiles measured with the Mocap. Moreover, a negligible estimation error in the knee angle profile was observed (less than $3.49^\circ \pm 1.05^\circ$) for all three pose methods in comparison to the ground truth knee joint angle. To sum up, the 3D pose estimation accuracy achieved by the three pose methods (i.e., *Openpose* off-line on the server, *MAEVE Densnet* and *Resnet* in real-time on the Jeston AGX) for measuring the keypoints position and the joint angles are comparable.

Previous work showed limited accuracy of markerless motion capture gait analysis compared to IMU or Mocap methods and recommended caution in using these systems in clinical movement analysis [44,45]. Specifically, D'Antonio et al. [45] found that the knee flexion estimation with an Openpose-based system was significantly different from a measure obtained with a previously validated IMU apparatus. However, they did not use depth cameras, and video acquisition was provided by simple webcams. Although we observed differences in the spatial accuracy of the 3D keypoints coordinates, our results showed negligible discrepancies in knee joint range of motion estimations between *MAEVE* and Mocap. It is conceivable that this should not lead to different clinical interpretations.

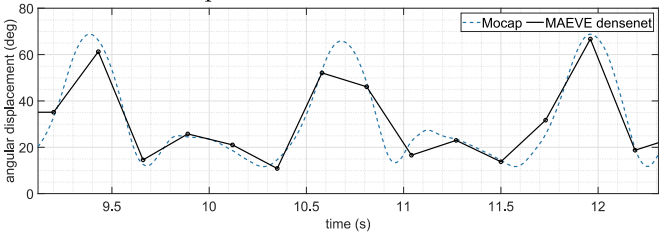
6. Conclusion

One of the main challenges of RGB-D based on 3D HPE is to achieve accuracy for clinical applications. Even more challenging is the HPE applied in the telemedicine practice, which requires the system to satisfy, beside accuracy, also portability, real-time, and privacy compliance constraints at the same time. In this article, we presented MAEVE, a 3D HPE that runs in real-time on a off-the-shelf portable computing board. The experimental results demonstrated that MAEVE achieves real time performance on such devices with comparable accuracy w.r.t. OpenPose run on a well equipped CPU+GPU server, and a deviation w.r.t. a Mocap system below the errore tolerance of such a marker-based system. The

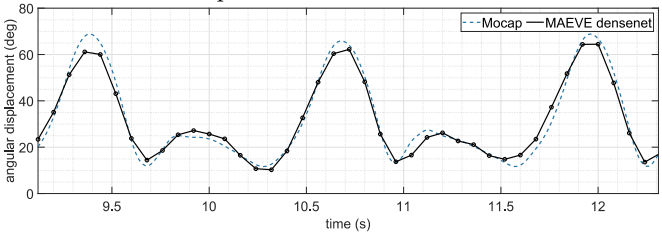
(a) Angular displacement of 3D Openpose on the Jetson AGX (6.3 Hz) and Mocap (100 Hz).



(b) Angular displacement of MAEVE on the Jetson Nano (4.3 Hz) and Mocap (100 Hz).



(c) Angular displacement of MAEVE on the Jetson TX2 (12.1 Hz) and Mocap (100 Hz).



(d) Angular displacement of MAEVE on Jetson Xavier AGX (59.9 Hz) and Mocap (100 Hz).

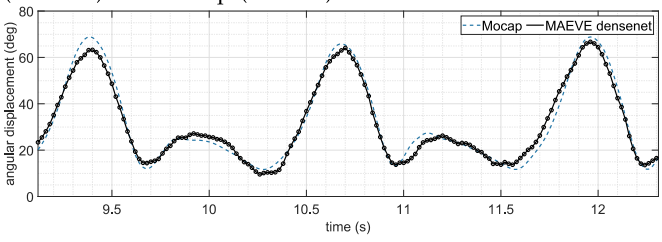


Fig. 8. Representative gait traces of MAEVE, 3D Openpose, and Mocap for the right knee joint angle.

preliminary results of this study lay the foundation for feasible applications of portable embedded systems for motion analysis, and serve as promising tools for home-based monitoring of gait and tele-rehabilitation aimed to improve walking ability and motor functions.

Declaration of Competing Interest

We would like to confirm no conflict of interest, financial or other, exists. This manuscript is entirely original, has not been copyrighted, published, submitted, or accepted for publication elsewhere.

References

[1] Y. Desmarais, D. Mottet, P. Slangen, P. Montesinos, A review of 3d human pose estimation algorithms for markerless motion capture, Computer Vision and Image Understanding 212 (2021), doi:10.1016/j.cviu.2021.103275.

[2] Y. Guo, X. Gu, G.-Z. Yang, Mcdcd: Multi-source unsupervised domain adaptation for abnormal human gait detection, IEEE Journal of Biomedical and Health Informatics 25 (10) (2021) 4017–4028.

[3] X. Gu, Y. Guo, F. Deligianni, B. Lo, G.-Z. Yang, Cross-subject and cross-modal transfer for generalized abnormal gait pattern recognition, IEEE Transactions on Neural Networks and Learning Systems 32 (2) (2021) 546–560.

[4] Y. Guo, R. Gravina, X. Gu, G. Fortino, G.-Z. Yang, Emg-based abnormal gait detection and recognition, in: Proceedings of the 2020 IEEE International Conference on Human-Machine Systems, ICHMS 2020, 2020, doi:10.1109/ICHMS49158.2020.9209449.

[5] K.-D. Ng, S. Mehdizadeh, A. Iaboni, A. Mansfield, A. Flint, B. Taati, Measuring gait variables using computer vision to assess mobility and fall risk in older adults with dementia, IEEE Journal of Translational Engineering in Health and Medicine 8 (2020), doi:10.1109/JTEHM.2020.2998326.

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

[7] G. Pavlakos, X. Zhou, K. Derpanis, K. Daniilidis, Coarse-to-fine volumetric prediction for single-image 3d human pose, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1263–1272.

[8] H.-S. Fang, Y. Xu, W. Wang, X. Liu, S.-C. Zhu, Learning pose grammar to encode human body configuration for 3d pose estimation, in: Proc. of AAAI Conference on Artificial Intelligence, 2018, pp. 6821–6828.

[9] G. Wei, C. Lan, W. Zeng, Z. Chen, View invariant 3d human pose estimation, IEEE Transactions on Circuits and Systems for Video Technology 30 (12) (2020) 4601–4610, doi:10.1109/TCSVT.2019.2928813.

[10] M. Abbas, D. Somme, R. Le Bouquin Jeanns, D-sorm: A digital solution for remote monitoring based on the attitude of wearable devices, Computer Methods and Programs in Biomedicine 208 (2021), doi:10.1016/j.cmpb.2021.106247.

[11] L. Yanicelli, C. Goy, E. Martnez, M. Herrera, Heart failure non-invasive home telemonitoring systems: A systematic review, Computer Methods and Programs in Biomedicine 201 (2021), doi:10.1016/j.cmpb.2021.105950.

[12] J. Calvillo-Arbizu, I. Romn-Martnez, J. Reina-Tosina, Internet of things in health: Requirements, issues, and gaps, Computer Methods and Programs in Biomedicine 208 (2021), doi:10.1016/j.cmpb.2021.106231.

[13] W. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, Edge computing: A survey, Future Generation Computer Systems 97 (2019) 219–235, doi:10.1016/j.future.2019.02.050.

[14] M. Sassi, M. Abid, Security and privacy protection in the e-health system: Remote monitoring of covid-19 patients as a use case, Smart Innovation, Systems and Technologies 237 (2022) 829–843, doi:10.1007/978-981-16-3637-0_58.

[15] Z. Cao, T. Simon, S. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1302–1310, doi:10.1109/CVPR.2017.143.

[16] J. Zhang, D. Zhang, X. Xu, F. Jia, Y. Liu, X. Liu, J. Ren, Y. Zhang, Mobipose: Real-time multi-person pose estimation on mobile devices, in: SenSys 2020 - Proc. of ACM Conference on Embedded Networked Sensor Systems, 2020, pp. 136–149.

[17] W. Yu, J. Choi, Human identification in health care systems using mobile edge computing, Transactions on Emerging Telecommunications Technologies 31 (12) (2020).

[18] S. Rupprechter, G. Morinan, Y. Peng, T. Foltynie, K. Sibley, R. Weil, L.-A. Leyland, F. Baig, F. Morgante, R. Gilron, R. Wilt, P. Starr, R. Hauser, J. OKeeffe, A clinically interpretable computer-vision based method for quantifying gait in parkinsons disease, Sensors 21 (16) (2021), doi:10.3390/s21165437.

[19] A. Sabo, S. Mehdizadeh, K.-D. Ng, A. Iaboni, B. Taati, Assessment of parkinsonian gait in older adults with dementia via human pose tracking in video data, Journal of NeuroEngineering and Rehabilitation 17 (1) (2020).

[20] A. Sabo, S. Mehdizadeh, A. Iaboni, B. Taati, Estimating parkinsonism severity in natural gait videos of older adults with dementia, IEEE Journal of Biomedical and Health Informatics 26 (5) (2022) 2288–2298, doi:10.1109/JBHI.2022.3144917.

[21] L. Kidzinski, B. Yang, J. Hicks, A. Rajagopal, S. Delp, M. Schwartz, Deep neural networks enable quantitative movement analysis using single-camera videos, Nature Communications 11 (1) (2020), doi:10.1038/s41467-020-17807-z.

[22] S. Mehdizadeh, M. Faieghi, A. Sabo, H. Nabavi, A. Mansfield, A.J. Flint, B. Taati, A. Iaboni, Gait changes over time in hospitalized older adults with advanced dementia: Predictors of mobility change, PLoS ONE 16 (11 November) (2021), doi:10.1371/journal.pone.0259975.

[23] J. Stenum, K.M. Cherry-Allen, C.O. Pyles, R.D. Reetzke, M.F. Vignos, R.T. Roemmich, Applications of pose estimation in human health and performance across the lifespan, Sensors 21 (21) (2021), doi:10.3390/s21217315.

[24] A. Rohan, M. Rabah, T. Hosny, S.H. Kim, Human pose estimation-based real-time gait analysis using convolutional neural network, IEEE Access 8 (2020) 191542–191550, doi:10.1109/ACCESS.2020.3030086.

[25] M. Antico, N. Balletti, G. Laudato, A. Lazich, M. Notarantonio, R. Oliveto, S. Ricciardi, S. Scalabrino, J. Simeone, Postural control assessment via microsoft azure kinect dk: An evaluation study, Computer Methods and Programs in Biomedicine 209 (2021), doi:10.1016/j.cmpb.2021.106324.

[26] Y. Ma, K. Mithraratne, N. Wilson, Y. Zhang, X. Wang, Kinect v2-based gait analysis for children with cerebral palsy: Validity and reliability of spatial margin of stability and spatiotemporal variables, Sensors 21 (6) (2021) 1–17, doi:10.3390/s21062104.

[27] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, T. Brox, 3d human pose estimation in rgbd images for robotic task learning, in: Proc. of IEEE

International Conference on Robotics and Automation, 2018, pp. 1986–1992, doi:10.1109/ICRA.2018.8462833.
[28] M. Ota, H. Tateuchi, T. Hashiguchi, N. Ichihashi, Verification of validity of gait analysis systems during treadmill walking and running using human pose tracking algorithm, Gait and Posture 85 (2021) 290–297, doi:10.1016/j.gaitpost.2021.02.006.
[29] A. Aberg, F. Olsson, H. hman, O. Tarassova, A. Arndt, V. Giedraitis, L. Berglund, K. Halvorsen, Extraction of gait parameters from marker-free video recordings of timed up-and-go tests: Validity, inter- and intra-rater reliability, Gait and Posture 90 (2021) 489–495, doi:10.1016/j.gaitpost.2021.08.004.
[30] Open Robotics, Robotic Operating System (ROS) 2, 2021, https://index.ros.org/doc/ros2/.
[31] O. Korkalo, T. Takala, Measurement noise model for depth camera-based people tracking, Sensors 21 (13) (2021), doi:10.3390/s21134488.
[32] E.S.L. Gastal, M.M. Oliveira, Domain transform for edge-aware image and video processing, ACM Transactions on Graphics 30 (4) (2011) 69:1–69:12. Proceedings of SIGGRAPH 2011
[33] S. Matyunin, D. Vatolin, Y. Berdnikov, M. Smirnov, Temporal filtering for depth maps generated by kinect depth camera, in: 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 3DTV-CON 2011 - Proceedings, 2011, doi:10.1109/3DTV.2011.5877202.
[34] Z. Xue, Y. Zhang, C. Cheng, G. Ma, Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression, Neurocomputing 376 (2020) 95–102, doi:10.1016/j.neucom.2019.09.074.
[35] NVIDIA AI IoT, Tensor RT Pose Estimation, 2020, https://github.com/NVIDIA-AI-IOT/trt_pose.
[36] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, Y. Sheikh, Openpose: Realtime multi-person 2d pose estimation using part affinity fields, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (1) (2021) 172–186.
[37] F. Lumpp, H.D. Patel, N. Bombieri, A framework for optimizing cpu-igpu communication on embedded platforms, in: Proceedings - Design Automation Conference, volume 2021-December, 2021, pp. 685–690, doi:10.1109/DAC18074.2021.9586304.
[38] R. Hartley, Multiple view geometry in computer vision, Cambridge University Press, Cambridge, UK New York, 2004.
[39] J.-H. Kim, G.-W. Jeung, J.-W. Lee, K.-S. Kim, Performance evaluation of a two-dimensional savitzky-golay filter for image smoothing applications, Lecture Notes in Electrical Engineering 382 (2016) 309–316, doi:10.1007/978-981-10-0740-8_35.
[40] CMU-Perceptual-Computing-Lab, Openpose, 2021, https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/include/openpose/flags.hpp.
[41] J. Sinclair, J. Hebron, P. Taylor, The test-retest reliability of knee joint center location techniques, Journal of Applied Biomechanics 31 (2) (2015) 117–121, doi:10.1123/JAB.2013-0312.
[42] R. Tanaka, H. Takimoto, T. Yamasaki, A. Higashi, Validity of time series kinematical data as measured by a markerless motion capture system on a flatland for gait assessment, Journal of Biomechanics 71 (2018) 281–285, doi:10.1016/j.jbiomech.2018.01.035.
[43] D. Winter, Biomechanics and motor control of human movement: Fourth edition, Biomechanics and Motor Control of Human Movement: Fourth Edition (2009) 1–370, doi:10.1002/9780470549148.
[44] N. Nakano, T. Sakura, K. Ueda, L. Omura, A. Kimura, Y. Iino, S. Fukashiro, S. Yoshioka, Evaluation of 3D Markerless Motion Capture Accuracy Using OpenPose With Multiple Video Cameras, Frontiers in Sports and Active Living 2 (May) (2020) 1–9, doi:10.3389/fspor.2020.00050.
[45] E. DAntonio, J. Taborri, E. Palermo, S. Rossi, F. Patan, A markerless system for gait analysis based on openpose library, in: Proc. of IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2020, pp. 1–6.