Real-Time Multi-View 3D Human Pose Estimation using Semantic Feedback to Smart Edge Sensors

Simon Bultmann
Autonomous Intelligent Systems
University of Bonn, Germany
Email: bultmann@ais.uni-bonn.de

Sven Behnke
Autonomous Intelligent Systems
University of Bonn, Germany
Email: behnke@cs.uni-bonn.de

***Abstract*—We present a novel method for estimation of 3D human poses from a multi-camera setup, employing distributed smart edge sensors coupled with a backend through a semantic feedback loop. 2D joint detection for each camera view is performed locally on a dedicated embedded inference processor. Only the semantic skeleton representation is transmitted over the network and raw images remain on the sensor board. 3D poses are recovered from 2D joints on a central backend, based on triangulation and a body model which incorporates prior knowledge of the human skeleton. A feedback channel from backend to individual sensors is implemented on a semantic level. The allocentric 3D pose is backprojected into the sensor views where it is fused with 2D joint detections. The local semantic model on each sensor can thus be improved by incorporating global context information. The whole pipeline is capable of real-time operation. We evaluate our method on three public datasets, where we achieve state-of-the-art results and show the benefits of our feedback architecture, as well as in our own setup for multi-person experiments. Using the feedback signal improves the 2D joint detections and in turn the estimated 3D poses.**

I. Introduction

Accurate perception of humans is a challenging task with many applications in robotics and computer vision. It is a prerequisite e.g., for safe navigation and anticipative movement of robots in the vicinity of people and can enable human-robot interaction or augmented reality scenarios.

In this work, we address the task of 3D human pose estimation in allocentric world coordinates from a calibrated multi-camera setup. Most state-of-the-art methods [24, 23, 7, 26, 9] follow a two-step approach: First, 2D pose detections are generated for each available view (cf. Fig. 1 bottom). Second, detections from multiple views are fused into a 3D human pose estimate and post-processed using a skeleton model (cf. Fig. 1 top-left). Many recent methods focus more on accuracy than efficiency. They are thus difficult to employ in real-world scenarios with real-time constraints.

We propose a novel architecture for real-time multi-view 3D human pose estimation using distributed smart edge sensors for 2D pose estimation. Each camera view is interpreted locally using an embedded inference accelerator. The 2D human poses are streamed over a network to a central backend, where data association, triangulation and post-processing are performed to fuse the 2D detections into 3D skeletons. Furthermore, we propose a semantic feedback channel from backend to smart edge sensors. The allocentric 3D pose estimate is backprojected into the respective local views where it is combined with the joint detections. Thus, global context information can be incorporated into the local 2D semantic model of the individual smart edge sensor, which improves the pose estimation result.

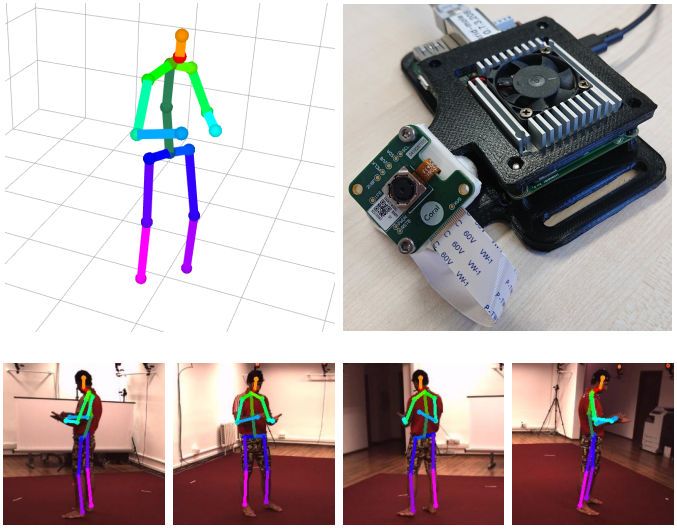


Fig. 1: Multi-view 3D human pose estimation using smart edge sensors: Sensor board with attached camera (top-right). 2D pose detections from four views of the H3.6M dataset [15] (bottom). Estimated 3D human skeleton (top-left).

The use of distributed smart edge sensors has several advantages over the centralized approaches more common in literature. As the images are processed directly on the sensor boards, raw images are not sent to the backend and only the 2D pose information has to be transmitted over the network. This significantly reduces the required communication bandwidth and furthermore mitigates privacy issues, as the abstract semantic information contains no personal details. Moreover, using a dedicated inference accelerator for each camera lessens hardware requirements on the backend side, which, in a centralized architecture, can quickly become the bottleneck. On the other hand, using an embedded sensor platform poses challenges, as the employed vision models need to meet the limitations of the hardware. For this, we propose a lightweight 2D pose estimation model for efficient image processing locally on the sensors, on the edge of the network.

In summary, the main contributions of our work are:

- a new real-time method for multi-view 3D human pose estimation dividing the computation between smart edge sensors performing image analysis locally for each camera view and a backend fusing the semantic interpretations of individual views and using a computationally efficient skeleton model to incorporate prior knowledge,
- a novel 3D / 2D feedback architecture enabling bidirectional communication on a semantic level between sensors and backend, and
- an extensive evaluation of the proposed approach on the single-person H3.6M dataset [15], the multi-person Campus and Shelf datasets [2], and in own multi-person experiments in a less-controlled environment.

II. Related Work

Human pose estimation from multi-camera input has been investigated for many years in the computer vision and robotics communities. Early works [2, 3, 5] use manually designed image features, such as HOG descriptors [8], for 2D part detection and combine multiple views using a graph-based body model. With the increasing success of deep-learning methods, more recent approaches [24, 23, 9] employ 2D convolutional neural networks (CNNs) for human joint detection [6, 30] and recover the 3D pose using variants of the Pictorial Structures Model (PSM) [2, 5]. In these approaches, the body model consists of a graph with 3D joint locations as nodes and pairwise articulation constraints on the edges. While the PSM body model recovers 3D poses accurately, it is computationally very expensive and generally not real-time capable, due to a large volumetric grid used as discrete state space for optimization. In our work, we also employ a graph-based body model but use a fast iterative optimization scheme [18], achieving real-time operation.

Qiu et al. [24] present an approach for cross-view fusion to improve the estimated 2D poses of individual camera views. 3D poses are recovered using an offline recursive PSM implementation with a processing time of several seconds per frame [26]. In our work, we take up the idea of across-sensor viewpoint fusion but propose a different formulation. Qiu et al. [24] implement the fusion between perspectives on a purely 2D basis, using epipolar constraints. Hence, a 2D joint in one view will be associated with all features on the corresponding epipolar lines of other views, which can be ambiguous. In contrast, we implement a semantic 3D / 2D feedback channel from backend to sensors based on reprojection of the estimated 3D skeleton into the individual camera views.

Several recent methods with a focus on computational efficiency have been proposed [7, 26]. In these approaches, 3D pose estimation is based on direct triangulation of 2D joint detections without usage of an expensive body model. Chen et al. [7] propose a fast iterative triangulation scheme but assume 2D pose detections as given. Remelli et al. [26] consider the whole pipeline including 2D keypoint estimation but use a fully centralized approach while our method employs distributed sensors for 2D pose estimation.

Naikal et al. [21] proposed a system for human joint detection and action recognition using a network of smart cameras transmitting only abstract image features to a central processing station. However, at the time, no CNN-based vision models were available for pose estimation on mobile devices, limiting the performance of their framework. Furthermore, their communication channel is purely feed-forward—no feedback for viewpoint fusion is implemented.

Research interest in computer vision models that run efficiently also on mobile and embedded devices has significantly increased in recent years. MobiPose [32] investigates human pose estimation on smartphone SoCs without dedicated inference accelerators, using motion vector-based tracking. Xiao et al. [30] propose a simple CNN architecture consisting only of a feature extractor and a deconvolutional head but use a standard ResNet backbone [12]. Popular lightweight backbone architectures include EfficientNet [27] and MobileNets [14, 13]. These architectures greatly reduce the number of parameters w.r.t. standard CNN feature extractors like ResNet, e.g., by replacing convolutions with depthwise-separable convolutions. Moreover, tensor processors for inference acceleration, like the Google Edge TPU [31], can be employed to efficiently run a CNN vision model within a limited size and energy budget. For compatibility with the Edge TPU, weights and activations of the model need to be quantized to 8-bit integer values using a quantization scheme as proposed by Jacob et al. [16].

To the best of our knowledge, the proposed framework is the first approach for real-time 3D human pose estimation using multiple smart edge sensors which perform 2D pose estimation on-device and incorporate global context via semantic feedback from the backend.

III. Method

An overview of our proposed approach is given in Fig. 2. We consider scenarios where N calibrated cameras with known projection matrices $\boldsymbol{P}_i$ perceive a scene with one or several individuals from multiple viewpoints. Our method is described for the single person case in the following. Extensions to handle multiple persons are described in Sec. III-E.

2D locations of a fixed set of J human joints $\{\boldsymbol{u}_i^j\}_{j=1}^J$ in camera view i, corresponding confidence values c_i^j and covariance matrices $\boldsymbol{\Sigma}_i^j$ are calculated directly on the respective smart edge sensor board using the vision model described in Sec. III-A. The 2D pose information is then transmitted over a network to a central backend, using the robot operation system (ROS) [25] as middleware for communication.

The clocks of sensors and backend are software-synchronized and each 2D pose message includes a timestamp representing the capture time of the corresponding image. Sets of N corresponding messages, one for each view, are determined based on these timestamps, and raw 3D poses are recovered via triangulation as detailed in Sec. III-B.

A skeleton model (cf. Sec. III-C), incorporating prior information on the typical bone-lengths of the human skeleton, is then applied and outputs the final estimated 3D pose.

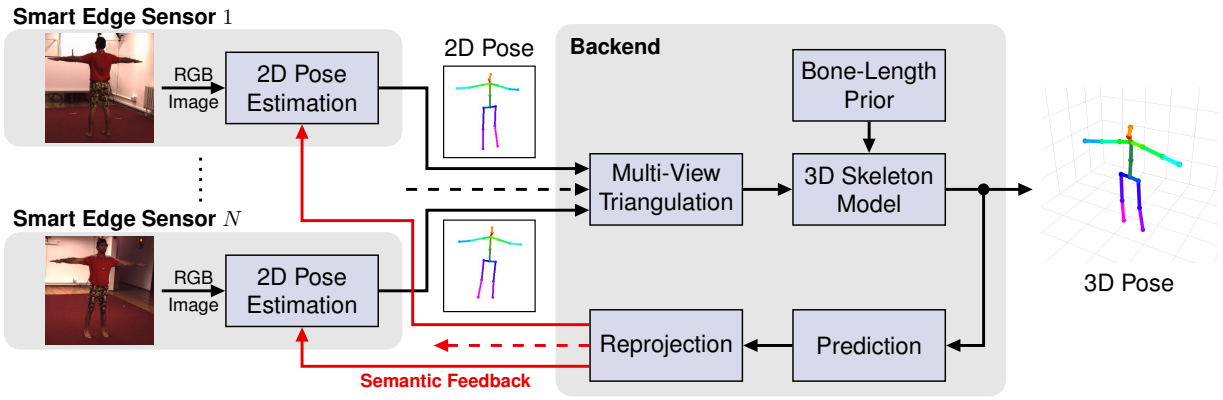


Fig. 2: Overview of the proposed pipeline for 3D human pose estimation using smart edge sensors and semantic feedback. Images are analyzed locally on the sensor boards. Semantic pose information is transmitted to the backend where multiple views are fused into a 3D skeleton. The 3D pose is reprojected into local views and sent to sensors as semantic feedback.

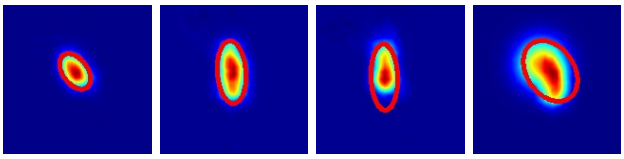


Fig. 3: Heatmaps and derived covariances (3σ ellipses).

A semantic feedback channel from backend to sensors is implemented as described in Sec. III-D, which enables each individual view to benefit from the fused 3D information. For this, first, a prediction step is performed to compensate for the pipeline delay. Second, the predicted 3D skeleton is reprojected into each camera view and sent to the sensors where it is incorporated into the local 2D pose estimation.

A. 2D Human Pose Estimation on Smart Edge Sensor

The smart edge sensor platform employed in this work (cf. Fig. 1 top-right) is based on the Google Edge TPU Dev Board [10], equipped with an ARM Cortex-A53 quad-core processor, the Edge TPU inference accelerator and 1 GB of shared RAM. A 5 MP RGB camera is connected to the board via the MIPI-CSI2 interface.

We adopt the CNN architecture of Xiao et al. [30] for 2D human pose estimation, consisting of a backbone feature extractor and three transposed convolution layers to extract heatmaps from image features. To achieve real-time performance on the mobile sensor platform, we exchange the ResNet backbone used by Xiao et al. [30] with the significantly more lightweight MobileNetV3 feature extractor [13]. Furthermore, for execution on the Edge TPU, the model is quantized for 8-bit integer inference using post-training quantization [16] as implemented in the TensorFlow ML framework [1]. In multi-person scenarios, a detector is also run on the sensor boards to provide person crops for the pose estimation network. It is based on the Single Shot Detector (SSD) architecture [20], also using the MobileNetV3 backbone.

The output heatmaps $\boldsymbol{H}_{\text{det}}$ of the pose estimation model are a multi-channel image with one channel per joint, encoding the confidence of a joint being present at the pixel location. 2D joint locations $\boldsymbol{u}^j = [u^j, v^j]^{\mathrm{T}}$ are inferred as global maxima of the resp. heatmap channel, as single person crops are processed. The value of the heatmap at the joint position gives the corresponding confidence c^j. Only joints with confidence above a minimum threshold are considered as valid detections.

The covariance matrices $\boldsymbol{\Sigma}^j$ are determined as proposed by Pasqualetto et at. [22]: Heatmap pixels with values above a threshold contribute to the empirical covariance with their x- and y-locations, weighted by the respective confidence:

$$\boldsymbol{\Sigma}^j = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_{yy}^2 \end{bmatrix} , \tag{1}$$

$$\sigma_{xy}^2 = \frac{1}{K} \sum_{k=1}^{K} c_k^j \left(x_k - u^j\right) \cdot \left(y_k - v^j\right) , \tag{2}$$

where K is the number of contributing pixels and the mean is replaced by the peak $\boldsymbol{u}^j$ to model a distribution about the detected 2D joint location. Some representative examples of heatmaps and extracted covariances are shown in Fig. 3. The uncertainty in the heatmaps, including their directionality, is well captured by the covariance ellipses. Note, that the dispersion for asymmetric heatmaps, as in the third example of Fig. 3, is overestimated by the proposed procedure.

B. Multi-View Fusion

The 3D position $\hat{\boldsymbol{x}}^j$ of each joint j is recovered from a set of 2D detections $\{\boldsymbol{u}_i^j\}_{i=1}^N$ via triangulation using the Direct Linear Transform (DLT) [11]. The relationship between 2D points $\boldsymbol{u}_i = [u_i, v_i]^{\mathrm{T}}$ from camera view $i \in \{1, \ldots, N\}$ and 3D point $\hat{\boldsymbol{x}}$ can be written as:

$$\boldsymbol{A}\tilde{\boldsymbol{x}} = \boldsymbol{0} , \tag{3}$$

with

$$\boldsymbol{A} = \begin{bmatrix} u_1 \boldsymbol{p}_{1,3}^{\mathrm{T}} - \boldsymbol{p}_{1,1}^{\mathrm{T}} \\ v_1 \boldsymbol{p}_{1,3}^{\mathrm{T}} - \boldsymbol{p}_{1,2}^{\mathrm{T}} \\ \cdots \\ u_N \boldsymbol{p}_{N,3}^{\mathrm{T}} - \boldsymbol{p}_{N,1}^{\mathrm{T}} \\ v_N \boldsymbol{p}_{N,3}^{\mathrm{T}} - \boldsymbol{p}_{N,2}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{2N \times 4} , \tag{4}$$

where $\tilde{\boldsymbol{x}} \in \mathbb{R}^4$ are the homogeneous coordinates of $\hat{\boldsymbol{x}}$ and $\boldsymbol{p}_{i,k}^{\mathrm{T}}$ denotes the k-th row of projection matrix $\boldsymbol{P}_i \in \mathbb{R}^{3\times4}$. According to the DLT algorithm [11], (3) is solved by a singular value decomposition (SVD) on $\boldsymbol{A}$, taking the unit singular vector corresponding to the smallest singular value of $\boldsymbol{A}$ as solution for $\tilde{\boldsymbol{x}}$. Finally, $\tilde{\boldsymbol{x}}$ is divided by its fourth coordinate to obtain the 3D vector $\hat{\boldsymbol{x}} = \tilde{\boldsymbol{x}}/(\tilde{\boldsymbol{x}})_4$.

The above formulation (3) assumes that all 2D detections make a similar contribution to the triangulation. However, 2D joint positions cannot be estimated reliably on some views, e.g., due to occlusions, which in turn degrades the result. The reliability of a detection is expressed by the heatmap confidence value c_i and can be incorporated into the DLT by multiplying each row of $\boldsymbol{A}$ with the corresponding element of a weight vector $\boldsymbol{w}$, reformulating (3) as:

$$(\boldsymbol{w} \circ \boldsymbol{A})\, \tilde{\boldsymbol{x}} = \boldsymbol{0}\,, \tag{5}$$

with

$$\boldsymbol{w} = \left(\frac{c_1}{\left\|\boldsymbol{a}_1^{\mathrm{T}}\right\|}, \frac{c_1}{\left\|\boldsymbol{a}_2^{\mathrm{T}}\right\|}, \ldots, \frac{c_N}{\left\|\boldsymbol{a}_{2N-1}^{\mathrm{T}}\right\|}, \frac{c_N}{\left\|\boldsymbol{a}_{2N}^{\mathrm{T}}\right\|} \right) \tag{6}$$

and $\circ$ the Hadamard product, similar to the approach of Chen et al. [7]. The confidence values in $\boldsymbol{w}$ are divided by the L^2-norm of the corresponding row of $\boldsymbol{A}$ to compensate for the different image locations of the joints in each view.

To obtain the 3D joint position $\hat{\boldsymbol{x}}^j$ and its covariance $\hat{\boldsymbol{\Sigma}}_{3\mathrm{D}}^j$, deterministic samples are propagated through the triangulation according to the Unscented Transform [17]. Sigma points are generated from the mean vector $\boldsymbol{\mu}^j = [\boldsymbol{u}_1^{j\mathrm{T}}, \ldots, \boldsymbol{u}_N^{j\mathrm{T}}]^{\mathrm{T}}$ and the block-diagonal matrix containing the 2D covariances $\boldsymbol{\Sigma}_i^j$ extracted from each heatmap. Each set of samples is triangulated according to (5) and $\hat{\boldsymbol{x}}^j$ and $\hat{\boldsymbol{\Sigma}}_{3\mathrm{D}}^j$ are determined as sample mean and covariance of the resulting points using weights given by the Unscented Transform.

C. Skeleton Model

We employ a factor graph model [18] representing the tree structure of the human body, with 3D joint positions $\boldsymbol{x}^j$ as nodes connected by unary and pairwise factors on the edges.

The unary constraints are given by the triangulated joint positions $\hat{\boldsymbol{x}}^j$ and covariances $\hat{\boldsymbol{\Sigma}}_{3\mathrm{D}}^j$ and follow a 3D Gaussian noise model:

$$f\left(\boldsymbol{x}^j\right) \sim \mathcal{N}\left(\boldsymbol{x}^j \,|\, \hat{\boldsymbol{x}}^j, \hat{\boldsymbol{\Sigma}}_{3\mathrm{D}}^j\right)\,. \tag{7}$$

The pairwise factors model typical limb-lengths of the human body and also follow a Gaussian noise model:

$$g\left(\boldsymbol{x}^j, \boldsymbol{x}^k\right) \sim \mathcal{N}\left(\left\|\boldsymbol{x}^j - \boldsymbol{x}^k\right\| \,|\, l_{j,k}, \sigma_{\mathrm{l}}\right)\,, \tag{8}$$

with $\left\|\boldsymbol{x}^j - \boldsymbol{x}^k\right\|$ the Euclidean distance between joints $\boldsymbol{x}^j$ and $\boldsymbol{x}^k$. $l_{j,k}$ and σ_{l} denote mean and standard deviation of the length of the corresponding limb determined from the statistics of the H3.6M dataset [15].

The final 3D human poses are obtained by optimizing the factor graph using the Levenberg-Marquardt algorithm and the *gtsam*-framework [18]. The optimization is initialized with the poses from the previous frame, predicted using a linear velocity model.

TABLE I: 2D Joint Detection Rate (JDR) (%) for different joint classes, feedback modes and training data.

Feedback	Training Data	Hips	Knees	Ankls	Shlds	Elbs	Wrists	**Avg**
w/o fb	H3.6M	99.2	96.1	90.3	93.3	93.3	89.1	95.1
w fb	H3.6M	**99.5**	97.6	96.1	97.2	96.5	**94.8**	97.5
w/o fb	COCO + H3.6M	99.3	97.1	96.9	98.9	96.2	92.8	97.6
w fb	COCO + H3.6M	99.3	**98.0**	**97.8**	**99.0**	**97.1**	**94.8**	**98.2**

TABLE II: 3D pose error (*mm*) for different joint classes, feedback modes and training data.

Feedback	Training Data	Hips	Knees	Ankls	Shlds	Elbs	Wrists	**Avg**
w/o fb	H3.6M	22.2	29.4	58.6	40.5	43.8	39.8	32.9
w fb	H3.6M	22.1	28.0	47.2	36.7	38.6	33.9	29.8
w/o fb	COCO + H3.6M	**19.2**	25.5	38.0	25.6	30.7	29.4	24.0
w fb	COCO + H3.6M	**19.2**	**24.9**	**36.9**	**25.5**	**29.9**	**28.3**	**23.5**

D. Semantic Feedback

To enable the local semantic models of each sensor to benefit from the globally fused 3D pose, a feedback channel from backend to sensors is implemented in our framework.

First, the motion of the 3D skeleton is predicted using a linear velocity model for each joint to compensate for the pipeline delay Δt. Second, predicted 2D joint positions $\{\hat{\boldsymbol{u}}_i^j\}$ and their image-plane covariances $\{\hat{\boldsymbol{\Sigma}}_i^j\}$ are determined by reprojecting the predicted 3D pose and its covariance extracted from the factor graph into each sensor view i using the projection matrix $\boldsymbol{P}_i$ and the Unscented Transform [17].

The reprojected feedback skeleton is sent to the smart edge sensors, where a feedback heatmap $\boldsymbol{H}_{\mathrm{fb}}$ is rendered to be fused with the detected heatmap $\boldsymbol{H}_{\mathrm{det}}$ of the current image crop. For each joint $\hat{\boldsymbol{u}}^j$, a 2D Gaussian blob is rendered in the corresponding heatmap channel according to the reprojected covariance matrix $\hat{\boldsymbol{\Sigma}}^j$.

The heatmaps are fused via weighted addition of detection, feedback, and their element-wise multiplication:

$$\boldsymbol{H}_{\mathrm{fused}} = s\left((1-\alpha-\beta)\,\boldsymbol{H}_{\mathrm{det}} + \alpha \boldsymbol{H}_{\mathrm{fb}} + \beta\,(\boldsymbol{H}_{\mathrm{fb}} \circ \boldsymbol{H}_{\mathrm{det}})\right), \tag{9}$$

with $\alpha + \beta < 1$. The scale s is set as $(1-\alpha-\beta)^{-1}$ to ensure that positive feedback always increases the joint confidence. The feedback gains α and β are important design parameters of our method. A sufficient weight must be accorded to the feedback to improve the raw detections, but too high gains can cause instability. Hence, the feedback gains are learned using a hyper-parameter search [4] optimizing the 3D pose error.

The above formulation models an arbitrary combination of additive and multiplicative feedback and can efficiently be executed on the embedded processor of the sensor board. Examples of the heatmap fusion are shown in Fig. 4. Through the feedback loop, evidence for joint occurrence from detection and feedback is combined in the fused heatmap, improving the accuracy of the joint locations and reducing their uncertainty.

E. Multi-Person Pose Estimation

To handle real-world scenes (cf. Sec. IV-C and IV-D), we extend our method to estimate the poses of multiple persons at

TABLE III: Evaluation result on H3.6M dataset: MPJPE 3D pose error (*mm*) per action type. 3D poses after application of the resp. post-processing or skeleton model are reported. $^+$ denotes using additional training data.

Method	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
Pavlakos et al. [23]	41.2	49.2	42.8	43.4	55.6	46.9	40.3	63.7	97.6	119.9	52.1	42.7	51.9	41.8	39.4	56.9
Tome et al. [29]	43.3	49.6	42.0	48.8	51.1	64.3	40.3	43.3	66.0	95.2	50.2	52.2	51.1	43.9	45.3	52.8
Qiu et al. [24]	28.9	32.5	26.6	28.1	**28.3**	29.3	28.0	36.8	42.0	**30.5**	35.6	30.0	28.3	30.0	30.5	31.2
Remelli et al. [26]	27.3	32.1	**25.0**	**26.5**	29.3	35.4	28.8	31.6	**36.4**	31.7	31.2	29.9	**26.9**	33.7	30.4	30.2
Ours, w/o fb	27.7	36.5	27.8	27.1	33.9	33.1	29.3	33.6	41.3	42.5	32.8	33.5	33.3	27.8	27.2	32.9
Ours, w fb	**27.1**	**29.9**	27.0	**26.5**	31.3	**28.9**	**27.1**	**29.8**	36.5	36.0	**30.8**	**29.3**	29.7	**27.3**	**26.3**	**29.8**
Qiu et al. [24]$^+$	24.0	26.7	23.2	24.3	24.8	**22.8**	24.1	28.6	32.1	26.9	31.0	25.6	25.0	28.1	24.4	26.2
Ours$^+$, w/o fb	**22.4**	24.3	22.4	**21.7**	24.6	24.7	22.4	**22.6**	26.8	28.4	25.0	23.1	**24.5**	22.0	21.5	24.0
Ours$^+$, w fb	**22.4**	**24.0**	**22.2**	**21.7**	**24.0**	23.9	**22.1**	**22.6**	**26.0**	**26.8**	**24.5**	**22.8**	24.6	**21.8**	**21.3**	**23.5**

a time. Person detections are associated across camera views based on the epipolar distance of their joints using the efficient iterative greedy matching proposed by Tanke et al. [28]. The rest of the pipeline is then run for each person observed in at least two views to compute 3D poses and feedback. A feedback skeleton is associated to its corresponding 2D detection based on the IoU overlap of their bounding boxes.

IV. Evaluation and Experiments

We evaluate the proposed approach on three widely-used public datasets: The Human 3.6M dataset [15] and the multi-person Campus and Shelf datasets [2], as well as on own data from experiments in our lab. We make our C++ implementation publicly available at https://github.com/AIS-Bonn/SmartEdgeSensor3DHumanPose.

A. Dataset and Metrics

1) Human 3.6M: The Human 3.6M dataset [15] is a large-scale public dataset for single-person multi-view 3D human pose estimation. It contains 3.6 million frames of 11 different actors, captured by four synchronized cameras together with ground truth 2D and 3D poses.

We measure the 2D pose estimation accuracy as the percentage of correctly detected joints, the Joint Detection Rate (JDR). A joint is correctly detected when its distance towards the corresponding ground-truth annotation is smaller than a threshold. We set the JDR threshold to half the head size as proposed by Qiu et al. [24].

The 3D pose accuracy is measured by the Mean Per Joint Position Error (MPJPE) between estimated 3D joints $\boldsymbol{x}^j$ and ground truth 3D joints $\boldsymbol{x}^j_{\text{gt}}$: MPJPE $= \frac{1}{J}\sum_{j=1}^{J}\|\boldsymbol{x}^j - \boldsymbol{x}^j_{\text{gt}}\|$.

2) Campus and Shelf: The Campus dataset [2] consists of three people interacting outdoors, captured by three calibrated cameras. The Shelf dataset [2] consists of four people interacting and disassembling a shelf in a small indoor area, captured by five cameras. It is a more complex setting compared to Campus, as frequent occlusions occur between persons and with the shelf. The same evaluation protocol as in previous works [7, 9, 2, 3] is used, employing the 3D Percentage of Correct Parts (PCP) metric [5]. A body part is considered as correctly estimated if the average of the Euclidean distances of start and end point of the limb with the ground-truth is smaller than half the limb-length.

B. Evaluation on the H3.6M Dataset

1) Implementation Details: We adopt the network for pose estimation described in Sec. III-A and use two different training schemes: (i) training solely on H3.6M training data and (ii) pretraining the network on person keypoints from the COCO dataset [19] and finetuning on H3.6M. The input resolution is set to 256×256 pixels.

As is common practice in literature [24, 23, 29], we use subjects 1, 5, 6, 7, 8 for training and subjects 9 and 11 for testing. Input images are cropped using the provided ground-truth bounding box and evaluation is performed for every 5^{th} frame as subsequent frames are highly similar at the original frame rate of $50\,\text{Hz}$.

All four image streams are processed simultaneously, each on its own smart edge sensor board. The estimated 2D skeletons are transmitted to the backend, where they are triangulated, and the skeleton model is applied. We report evaluation results with and without using the proposed feedback channel. The parameters for the heatmap fusion (9) are determined as $\alpha = 0.15$ and $\beta = 0.75$.

2) Quantitative Results: Tab. I shows evaluation results for the accuracy of the 2D pose estimation calculated on the smart edge sensors, depending on the employed feedback mode and training data. Our experiments indicate that using the feedback channel (cf. Sec. III-D) significantly improves the JDR accuracy. The improvement is highest for the often-occluded wrist and ankle joints, $5.7\,\%$ resp. $5.8\,\%$ for the H3.6M-only model. For the better visible joint classes, detection is easier also without feedback and the improvement is smaller.

Pretraining the model on the COCO keypoint dataset generally improves performance, as the model trained on a larger and more varying dataset generalizes better to unknown scenes. For the stronger model, the gain from using the feedback signal is smaller, but still amounts to $2\,\%$ for the wrists which are the most difficult joints to detect.

The improved 2D joint detections in turn lead to more accurate 3D poses, as becomes apparent from the results in Tab. II, where 3D pose error is shown. As in the 2D case, the improvement from the feedback channel is more significant for the weaker model and highest for ankles and wrists, around $11\,\text{mm}$ resp. $6\,\text{mm}$ for the H3.6M-only network.

In Tab. III, the MPJPE 3D pose error is shown per action

TABLE IV: Average inference time and model size for H3.6M dataset (Values for Qiu et al. [24] taken from [26]).

	Qiu et al. [24]	Remelli et al. [26]	Ours
Inference Time	8.4 s	0.040 s	**0.024 s**
Model Size	2.1 GB	251 MB	**4 × 12 MB**

TABLE V: Ablation study of the impact of various components of our approach on MPJPE 3D pose error (*mm*).

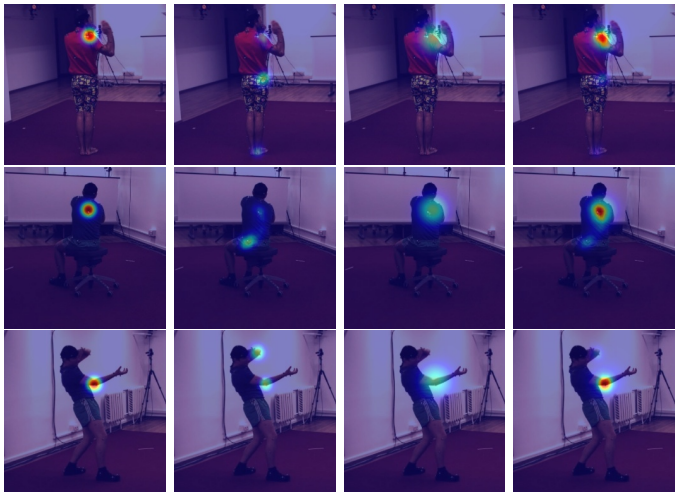
	MPJPE	skeleton model	heatmap covariance	add. feedback	mult. feedback
w/o model, w/o fb	38.08	-	-	-	-
w/o hm cov*, w/o fb	33.41	✓	-	-	-
w/o fb	32.94	✓	✓	-	-
w fb (add)	30.07	✓	✓	✓	-
w fb (mult)	30.10	✓	✓	-	✓
w fb	**29.82**	✓	✓	✓	✓

*skeleton model without directional heatmap covariances

category and compared to other approaches from literature. 3D pose errors after application of the resp. post-processing step or skeleton model are reported. The recent approaches [24] and [26] as well as our method provide significant improvements over the older methods [23] and [29]. Comparing the models trained on H3.6M only, the results of our approach using feedback are better than Qiu et al. [24] and Remelli et al. [26] for 10 of the 15 action categories and reduce the average error. The proposed semantic feedback channel is key to this improvement over the literature. When using additional training data, our method also achieves state-of-the-art results.

In Tab. IV, we compare the inference time per frame set (i.e., for a set of four images) and model size of our approach with the recent approaches [24] and [26]. The approach of Qiu et al. [24] is an offline method with a runtime of several seconds. The approach of Remelli et al. [26] achieves near real-time performance on a powerful desktop GPU. The runtime of our method is still about $40\,\%$ lower while running on efficient embedded sensor boards and a backend that doesn't require a GPU. Our pose estimation model, optimized for the Edge TPU inference accelerator, requires only $12\,\mathrm{MB}$ of memory, significantly less than the models of other approaches.

Results of an ablation study on the impacts of different parts of our proposed pipeline are shown in Tab. V. Using the skeleton model to post-process the raw 3D poses obtained by triangulation significantly improves the average MPJPE. Employing the directional covariances extracted from the heatmaps, instead of modeling uncertainties only by the confidence value, again reduces the error. The semantic feedback further improves the result, where the proposed combination of additive and multiplicative feedback is more efficient than using only a single type. The impact of the feedback signal for each action class can also be observed in Tab. III. It improves the results for all actions for the H3.6M-only trained model, with an average improvement of $3.1\,\mathrm{mm}$. When using the stronger pose estimation model trained on additional data, the average improvement amounts to $0.5\,\mathrm{mm}$. The feedback signal is more important when the raw pose estimates are less accurate but reduces the average 3D pose error in both cases.



(a) ground-truth (b) detected (c) feedback (d) fused

Fig. 4: Samples of the heatmap fusion approach for left wrist (Rows 1-2) and right elbow (Row 3): Detected heatmap (b) and feedback heatmap (c) are combined into the fused heatmap (d). In difficult situations such as occlusions (Rows 1-2) or left-right inversions (Row 3), using feedback results in a heatmap closer to the ground-truth (a).

3) Qualitative Results: In addition, we qualitatively show how the proposed feedback loop improves the pose estimation result. Fig. 4 shows three example situations, where the feedback heatmap helps to recover from incorrect or imprecise 2D joint detections. The images are overlaid with the respective heatmaps for a specific joint. In the first and second row, the left wrist of the actors is occluded by their body and the detected heatmap is very inaccurate. However, from the perspectives of other cameras, the joint is visible, and its 3D position can be estimated. This is reflected in the feedback heatmap which predicts the joint detection close to the ground truth location. The resulting fused heatmap, obtained by combining detection and feedback according to (9), permits to accurately estimate the respective joint despite the imprecise detection. In Row 3 of Fig. 4, a similar situation is shown, but for the right elbow, which here cannot be distinguished from the left elbow due to the challenging pose.

C. Evaluation on the Campus and Shelf Datasets

1) Implementation Details: To process multi-person scenes, a person detector is employed together with the pose estimation model (cf. Sec. III-A). The detector is trained for 130 Epochs on the person class of the COCO dataset [19], for input images of 640×480 px and achieves a mAP of $44.6\,\%$. The pose estimation network is trained for 140 epochs on COCO for person crops of 192×256 px. It achieves a mAP of $69.6\,\%$ in FP32-mode and $68.4\,\%$ in INT8-mode on the COCO

TABLE VI: Evaluation result on Campus and Shelf dataset: Percentage of Correct Parts (PCP) (%) and average run-time of 2D and 3D pose inference. '-' means offline pre-computation.

	PCP (%)				Inference Time	
Campus	Actor1	Actor2	Actor3	Avg	2D pose	3D pose
Belagiannis et al. [3]	83.0	73.0	78.0	78.0	-	1 s
Dong et al. [9]	97.6	93.3	98.0	96.3	-	105 ms
Chen et al. [7]	97.1	**94.1**	**98.6**	96.6	-	**1.6 ms**
Ours, w/o fb	98.8	93.4	97.5	96.6	**30 ms**	8.8 ms
Ours, w fb	**99.2**	93.6	98.3	**97.0**	**30 ms**	8.8 ms
Shelf	Actor1	Actor2	Actor3	Avg	2D pose	3D pose
Belagiannis et al. [3]	75.0	67.0	86.0	76.0	-	1 s
Dong et al. [9]	98.8	94.1	**97.8**	96.9	-	105 ms
Chen et al. [7]	**99.6**	93.2	97.5	96.8	-	**3.1 ms**
Ours, w/o fb	99.4	94.6	96.8	96.9	**40 ms**	20 ms
Ours, w fb	99.3	**95.7**	97.3	**97.4**	**40 ms**	20 ms

validation set using ground-truth detections. Note, that the generic detector and pose estimation networks are employed without any fine-tuning on the evaluated datasets.

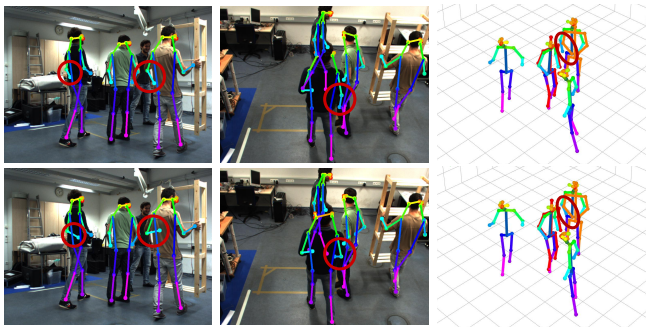
The three or five image streams of the respective dataset are processed simultaneously on the sensor boards. The entire image is passed to the detector and image crops of the detected persons are analyzed by the pose estimation network. To improve the processing speed, the detector is only run once per second. In between, the crops are determined based on the detections of the previous frame. This is necessary, as alternating between models is inefficient on the Edge TPU, as parameter caching cannot come into effect in this case [31].

On the backend, the estimated 2D poses are synchronized based on their timestamps and the framework is run in multi-person mode as detailed in Sec. III-E. The feedback delay amounts to one frame during dataset processing.

2) Quantitative Results: We report evaluation results of PCP score and runtime on the Campus and Shelf datasets in Tab. VI. and compare our method with other approaches: Belagiannis et al. [3] were among the first to propose 3D PSM-based multi-person pose estimation and exploit temporal consistency in videos. Dong et al. [9] propose to reduce the PSM state-space and exploit appearance information for data association. Chen et al. [7] propose a fast iterative triangulation scheme performing data association in 3D space.

In terms of PCP score, our method largely outperforms the older method [3] and is on par with the recent approaches [9, 7]. The overall result is improved by our method using feedback for both Campus and Shelf dataset in comparison to the literature. The improvement is most significant for Actor2 of the Shelf dataset, whose arms are often severely occluded, which can be resolved by the semantic feedback signal.

In terms of processing speed, our method does not reach the high frame rates of Chen et al. [7] but achieves significant improvements over [9] and [3]. Furthermore, 2D poses are estimated online, at run-times of 30-40 ms per frame, while other methods use offline pre-computed keypoint detections. Our method is the only approach in the comparison providing a fully online multi-person pose estimation.



(a) Camera 2 (b) Camera 3 (c) 3D pose

Fig. 5: Evaluation on Shelf dataset: 2D pose detections and estimated 3D pose without (top) and with feedback (bottom). 3D annotations for Actor1 (red) and Actor2 (orange). Highlighted are improvements due to the feedback signal.

TABLE VII: Evaluation in own experiments with up to 16 cameras and 6 persons: Reprojection error (px) per joint class between detected 2D poses and fused 3D poses.

Feedback	Cams	Pers	Hips	Knees	Ankls	Shlds	Elbs	Wrists	**Avg**
w/o fb	4	1-4	5.4	4.6	5.0	2.8	4.0	5.2	4.2
w fb	4	1-4	4.4	**3.5**	**3.4**	**2.3**	**3.2**	**3.7**	**3.3**
w/o fb	16	6	5.4	5.2	6.4	3.9	5.1	6.4	5.1
w fb	16	6	**4.3**	3.8	4.7	3.4	4.0	4.9	4.1

3) Qualitative Results: Fig. 5 shows an exemplary scene of the Shelf dataset. The proposed semantic feedback improves the estimation of occluded wrist joints in 2D and 3D. Annotations for evaluation are only provided for two of the four actors in this scene.

D. Experiments in Multi-Person Scenes

We further evaluate the proposed framework in online experiments in multi-person scenarios in our lab.

1) Implementation Details: 16 sensor boards are mounted under the ceiling of our lab in a roughly 12×16 m area. The cameras face downwards towards the center and run at 30 Hz and VGA resolution. We conduct experiments with a subset of 4 cameras, similar to the setting of the H3.6M dataset, as well as with all 16 sensors to demonstrate the scalability of our method to large-scale camera systems. The same detection and pose estimation models as for the Campus and Shelf dataset are employed in the experiments and the pipeline runs in multi-person mode (cf. Sec. III-E).

2) Quantitative Results: To analyze the consistency of the online pose estimation, we evaluate the error between detected 2D poses and fused 3D poses reprojected into the camera views in Tab. VII. The reprojection error decreases for all joints when using semantic feedback, indicating that the locally estimated 2D poses are more consistent with the globally fused 3D poses through the proposed feedback architecture. The error is slightly higher with 16 than with 4 sensors, probably due to the more difficult camera calibration and synchronization in the large-scale setup.

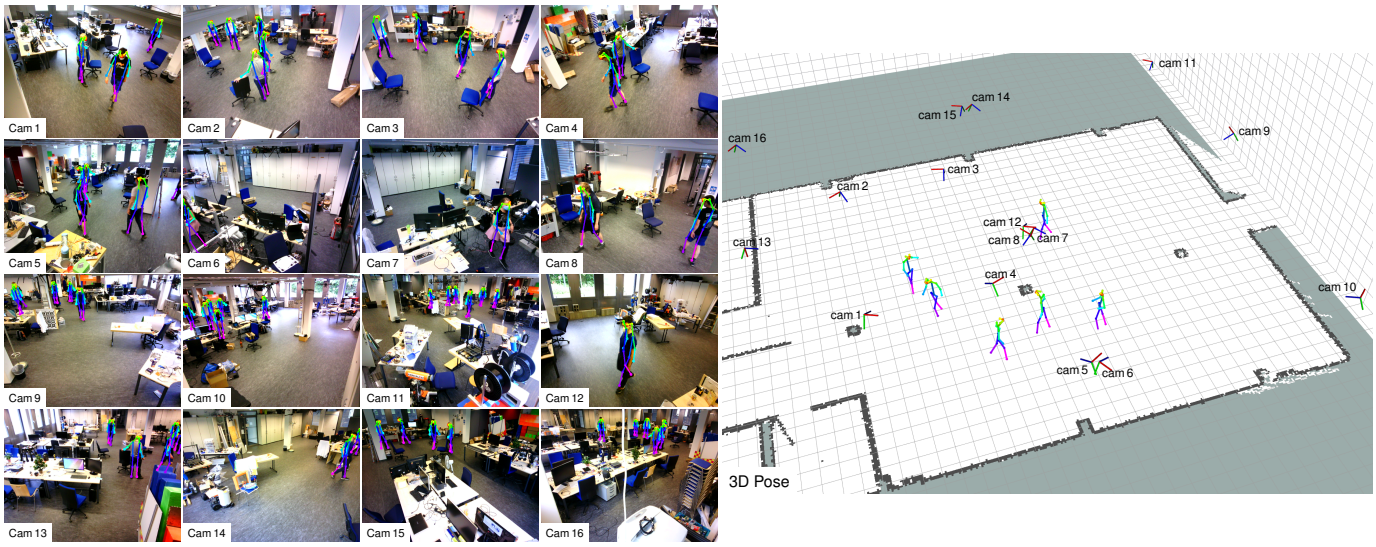


Fig. 6: Evaluation in multi-person scenarios: Estimated 3D poses reprojected into the corresponding camera image.

3) Qualitative Results: An exemplary real-world scene from the experiments conducted in our lab is shown in Fig. 6. The 16 camera views contain up to six persons and complex, cluttered backgrounds. Estimated 3D poses are reprojected onto the corresponding images to provide a visual evaluation. The reprojected skeletons closely fit the persons in the images, indicating that 3D and 2D poses are reliably estimated. People are reliably detected also at large distances to the cameras and occlusions by objects or other people can be resolved through the multi-view architecture and the semantic feedback.

The human poses are estimated online, in real time, and could directly be used, e.g., for human-robot interaction. Note, that the camera images are not transmitted during operation of our framework but are only shown for visualization. A video of the experiments is available on our website[1].

4) Run-time Analysis: The average processing time per image crop on the sensor boards consists of $4.5\,\mathrm{ms}$ for pose estimation on the TPU and $6\,\mathrm{ms}$ on the ARM-CPU for pre- and post-processing and sums to $10.5\,\mathrm{ms}$ per detected person. Once per second, the person detector requires additional $20\,\mathrm{ms}$ on the TPU. Up to three persons can thus be tracked at the full camera frame rate of $30\,\mathrm{Hz}$, six persons still at $15\,\mathrm{Hz}$.

The backend processing on a desktop PC with an Intel i9-9900K CPU takes $10.7\,\mathrm{ms}$ in average per frame set for the 4-camera setup and $60.8\,\mathrm{ms}$ during the experiments with 16 cameras and six persons. Especially the computational load of multi-view triangulation grows with larger number of cameras.

Camera images and semantic feedback are processed asynchronously on the sensors during the online experiments, the frequencies of the feedback and feed-forward parts of the pipeline do not need to be balanced. The most recent feedback message not older than a threshold is used for a camera image. The average pipeline delay Δt including processing on sensors and backend as well as network and synchronization delays sums to $89\,\mathrm{ms}$ in the 4-camera setup and to $200\,\mathrm{ms}$ with 16 cameras. This delay does not limit the feed-forward frequency of pose inference due to the asynchronous parallel processing. The latency is compensated by the prediction step in the feedback channel (cf. Sec. III-D).

5) Network Bandwidth and Power Consumption: The network usage when processing a $30\,\mathrm{Hz}$ video stream only amounts to $15\,\mathrm{kB/s}$ per detected person, as only semantic skeletons are transmitted between sensors and backend. This is an over $99\,\%$ reduction of bandwidth compared to $27\,\mathrm{MB/s}$ when transmitting the raw VGA images. The power consumption of a sensor board was measured as approx. $7\,\mathrm{W}$ when running inference on the $30\,\mathrm{Hz}$ multi-person video stream.

V. Conclusion and Future Work

In this work, we proposed a novel method for real-time 3D human pose estimation using a network of smart edge sensors. Our main idea is to process each camera view on-device, transmit only semantic information to a backend where it is fused into a 3D skeleton, and implement a 3D / 2D semantic feedback channel which lets the local semantic models incorporate fused multi-view information. The pipeline is able to track up to three persons at $30\,\mathrm{Hz}$ and up to six persons at $15\,\mathrm{Hz}$, achieving real-time performance. It is evaluated on the H3.6M, Campus, and Shelf datasets where it is shown to achieve state-of-the-art results, as well as on own data in scenarios with up to 16 cameras and six persons.

In future research, we plan to use the estimated human pose information to enable safe human-robot interaction and anticipative robot behavior in a workspace shared with people. Mobile robots carrying a smart sensor board can participate in the network for collaborative perception and add further viewpoints. The semantic scene model could be extended to also include objects and scene geometry. Furthermore, using a more elaborate motion model in the prediction step could compensate better for the pipeline delay and improve the feedback signal, especially for fast motions.

[1] https://www.ais.uni-bonn.de/videos/RSS_2021_Bultmann

ACKNOWLEDGMENTS

This work was funded by grant BE 2556/16-2 of the German Research Foundation (DFG), a Google faculty research award, and Fraunhofer IAIS.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI)*, pages 265–283, 2016.

[2] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3D pictorial structures for multiple human pose estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1669–1676, 2014.

[3] Vasileios Belagiannis, Xinchao Wang, Bernt Schiele, Pascal Fua, Slobodan Ilic, and Nassir Navab. Multiple human pose estimation with temporally consistent 3D pictorial structures. In *Computer Vision - ECCV 2014 Workshops*, pages 742–754. Springer, 2015.

[4] J. Bergstra, D. Yamins, and D. D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Intl. Conf. on Machine Learning (ICML)*, page I–115–I–123, 2013.

[5] Magnus Burenius, Josephine Sullivan, and Stefan Carlsson. 3D pictorial structures for multiple view articulated pose estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3618–3625, 2013.

[6] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 43 (1):172–186, 2019.

[7] Long Chen, Haizhou Ai, Rui Chen, Zijie Zhuang, and Shuang Liu. Cross-view tracking for multi-human 3D pose estimation at over 100 fps. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3279–3288, 2020.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[9] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3D pose estimation from multiple views. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7784–7793, 2019.

[10] Google. EdgeTPU dev board. https://coral.ai/docs/dev-board/datasheet, 2020. Accessed: 2021-02-22.

[11] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[13] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for MobileNetV3. In *IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 1314–1324, 2019.

[14] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *preprint arXiv:1704.04861*, 2017.

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(7):1325–1339, 2014.

[16] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2704–2713, 2018.

[17] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[18] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The Intl. Journal of Robotics Research*, 31(2):216–235, 2012.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conf. on Computer Vision (ECCV)*, pages 740–755, 2014.

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European Conf. on Computer Vision (ECCV)*, pages 21–37, 2016.

[21] Nikhil Naikal, Pedram Lajevardi, and Shankar. S. Sastry. Joint detection and recognition of human actions in wireless surveillance camera networks. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4747–4754, 2014.

[22] Lorenzo Pasqualetto Cassinis, Robert Fonod, Eberhard Gill, Ingo Ahrns, and Jesus Gil Fernandez. CNN-based pose estimation system for close-proximity operations around uncooperative spacecraft. In *AIAA Scitech 2020 Forum*, page 1457, 2020.

[23] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3D human pose annotations. In *IEEE*

Conf. on Computer Vision and Pattern Recognition (CVPR), pages 1253–1262, 2017.

[24] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3D human pose estimation. In *IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 4341–4350, 2019.

[25] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. ROS: An open-source robot operating system. In *IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Software*, volume 3, page 5, 2009.

[26] Edoardo Remelli, Shangchen Han, Sina Honari, Pascal Fua, and Robert Wang. Lightweight multi-view 3D pose estimation through camera-disentangled representation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6040–6049, 2020.

[27] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural metworks. In *Intl. Conf. on Machine Learning (ICML)*, pages 6105–6114, 2019.

[28] Julian Tanke and Juergen Gall. Iterative greedy matching for 3D human pose tracking from multiple views. In *German Conf. on Pattern Recognition (GCPR)*, pages 537–550, 2019.

[29] Denis Tome, Matteo Toso, Lourdes Agapito, and Chris Russell. Rethinking pose in 3D: Multi-stage refinement and recovery for markerless motion capture. In *Intl. Conf. on 3D Vision (3DV)*, pages 474–483, 2018.

[30] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conf. on Computer Vision (ECCV)*, pages 466–481, 2018.

[31] Amir Yazdanbakhsh, Kiran Seshadri, Berkin Akin, James Laudon, and Ravi Narayanaswami. An evaluation of Edge TPU accelerators for convolutional neural networks. *arXiv:2102.10423 [cs]*, 2021.

[32] Jinrui Zhang, Deyu Zhang, Xiaohui Xu, Fucheng Jia, Yunxin Liu, Xuanzhe Liu, Ju Ren, and Yaoxue Zhang. MobiPose: Real-time multi-person pose estimation on mobile devices. In *ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, page 136–149, 2020.