17th International Conference on Intelligent Autonomous Systems (IAS), Zagreb, Croatia, June 2022.

3D Semantic Scene Perception using Distributed Smart Edge Sensors

Simon Bultmann and Sven Behnke

Institute for Computer Science VI, Autonomous Intelligent Systems,
University of Bonn, Friedrich-Hirzebruch-Allee 8, 53115 Bonn, Germany
bultmann@ais.uni-bonn.de, https://www.ais.uni-bonn.de

Abstract. We present a system for 3D semantic scene perception consisting of a network of distributed smart edge sensors. The sensor nodes are based on an embedded CNN inference accelerator and RGB-D and thermal cameras. Efficient vision CNN models for object detection, semantic segmentation, and human pose estimation run on-device in real time. 2D human keypoint estimations, augmented with the RGB-D depth estimate, as well as semantically annotated point clouds are streamed from the sensors to a central backend, where multiple viewpoints are fused into an allocentric 3D semantic scene model. As the image interpretation is computed locally, only semantic information is sent over the network. The raw images remain on the sensor boards, significantly reducing the required bandwidth, and mitigating privacy risks for the observed persons. We evaluate the proposed system in challenging real-world multi-person scenes in our lab. The proposed perception system provides a complete scene view containing semantically annotated 3D geometry and estimates 3D poses of multiple persons in real time.

Keywords: Semantic scene understanding, intelligent sensors and systems, visual perception, sensor fusion

1 Introduction

Accurate semantic perception of 3D scene geometry and persons is challenging and an important prerequisite for many robotic tasks, such as safe and anticipative robot movement in the vicinity of people as well as human-robot interaction. In this work, we propose a system for 3D semantic scene perception consisting of a network of distributed smart edge sensors. It provides a complete scene view containing semantically annotated 3D geometry and estimates 3D poses of multiple persons in real time.

We build upon our previous work on real-time 3D human pose estimation using semantic feedback to smart edge sensors [2]. While this existing pipeline is able to track poses of multiple persons in real time, it lacks modeling of other aspects of the scene, i.e. 3D geometry, object detections, and surface categorization. Semantically annotated 3D geometry, however, is required to explain and predict interactions between persons and objects in the scene, and

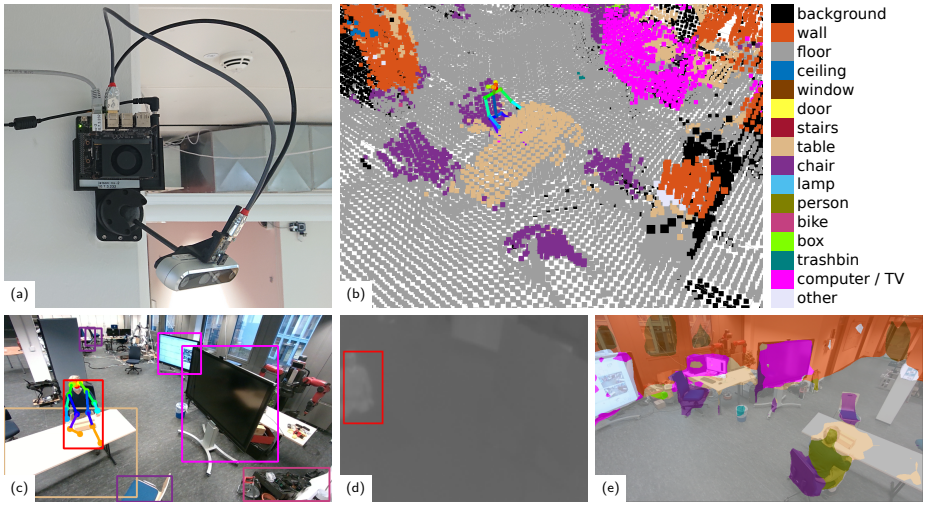

Fig. 1: Semantic perception with distributed smart edge sensors: (a) developed sensor node, (b) 3D semantic scene model with 3D human skeleton, (c) RGB and (d) thermal detections, (e) semantic segmentation. Person detections in red and skeleton keypoints colored by joint index. Occluded joints are marked in orange. CNN inference runs online on distributed sensors and semantic information is aggregated into an allocentric 3D scene model on the backend including 3D geometry (e.g., furniture, walls, floor) and 3D human pose.

to handle occlusions. Temporal aggregation and fusion of semantic point clouds from multiple sensor perspectives further leads to a consistent and persistent 3D semantic scene model with the field of perception not being limited by the measurement range or occlusions of a single sensor.

To enable perception of these additional characteristics of the scene, the sensor network is extended with updated smart edge sensors with higher compute capabilities and greater flexibility w.r.t. the employed vision CNNs, as shown in Fig. 1. This enables to run object detection and semantic image segmentation together with human pose estimation on the sensors in real time. RGB-D cameras estimate 3D scene geometry and thermal cameras increase the person detection performance in low-light conditions. Semantic information from detections and image segmentation is fused into the point cloud computed from the depth image and 2D human joint detections are augmented with the depth measured at the keypoint location. Semantic point cloud and human poses are communicated to a central backend, where they are fused into an allocentric 3D metric-semantic scene model. Only the semantic information is sent over the network; the raw images remain on the sensor boards, significantly reducing the required bandwidth, and mitigating privacy issues for the observed persons.

The semantic point clouds from multiple viewpoints are aggregated into an allocentric map of 3D scene geometry and semantic classes on the backend. The map is further updated via ray-tracing to account for moving objects. 3D human

poses are estimated in real time in the scene via multi-view triangulation. The allocentric 3D human poses are projected into the local camera views and sent back to the sensors as semantic feedback [2], where they are fused with the local detections. The 3D scene geometry enables to compute occlusion information for each joint in the respective camera view. This information is included into the semantic feedback from backend to sensors, improving the local scene model of each sensor by incorporating global context information. Unreliable, occluded joint detections can be discarded, and the local model is completed by the more reliable semantic feedback reprojected from the global, fused 3D model.

We evaluate the proposed system in experiments with challenging real-world multi-person scenes. In summary, our contributions are:

- The development of a smart edge sensor platform based on the Nvidia Jetson Xavier NX development kit and an RGB-D and thermal camera, running efficient vision CNN models for object detection and semantic segmentation together with human pose estimation on-device in real time;
- Temporal multi-view fusion of semantic point clouds from individual sensors into an allocentric semantic map of 3D scene geometry;
- The integration of multiple instances of the proposed novel sensor nodes into a network of distributed smart edge sensors for real-time multi-view 3D human pose estimation using semantic feedback [2], complementing the feedback from backend to sensor with occlusion information for human joints in the respective camera views, computed via ray-tracing through the estimated 3D scene geometry.

We make our implementation for both sensor boards[1] and backend[2] publicly available.

2 Related Work

Lightweight Vision CNNs for Embedded Hardware. Convolutional neural networks (CNNs) set the state-of-the-art for image processing and computer vision. On systems with restricted computational resources, like mobile embedded sensor platforms, however, lightweight, efficient models must be employed to achieve real-time performance. A popular approach is to replace classical backbone networks such as ResNets [10] with MobileNet [20, 11] or EfficientNet [23] architectures, as the main computational load of CNN inference often lies in the backbone feature extractor. These architectures decrease the number of parameters and the computational cost significantly, e.g., by replacing standard convolutions with depthwise-separable convolutions.

For object detection on embedded devices, single-stage architectures such as SSD [13] or YOLO [17], which use predefined anchors instead of additional region proposal networks, were shown to be efficient. In our work, we employ the

[1] https://github.com/AIS-Bonn/JetsonTRTPerception
[2] https://github.com/AIS-Bonn/SmartEdgeSensor3DScenePerception

recently proposed MobileDets [26], that are optimized for embedded inference accelerators using the SSD architecture with MobileNet v3 backbone.

The DeepLab v3+ architecture [5] for semantic segmentation uses elements of MobileNets, such as depthwise-separable convolutions, for efficiency on embedded hardware and shows state-of-the-art performance on large, general datasets. We employ a DeepLab v3+ model with MobileNet v3 backbone in our work.

For human pose estimation, OpenPose [4] set a new standard by detecting body parts of multiple persons in an image and associating them to individuals via Part Affinity Fields (PAFs). This bottom-up approach scales well with the number of person detections. Top-down approaches, on the other hand, first detect individuals and then estimate body keypoints for each single-person crop. These approaches achieve higher accuracy and better scale-invariance, as the person detections are interpolated to a fixed input resolution before pose inference. Xiao et al. [25] propose an efficient CNN architecture consisting of a backbone feature extractor and deconvolutional layers. We adopt this architecture and replace the ResNet backbone with MobileNet v3 for better efficiency on embedded hardware.

Semantic Mapping. Semantic information about the environment is a prerequisite for many high-level robotic functions. For this, semantic mapping systems build an allocentric model of 3D scene geometry with semantic class information.

SemanticFusion [14] builds semantic maps from RGB-D camera input using surface elements (Surfels), where a Gaussian approximates the local point distribution. Pixel-wise class probabilities are obtained from the color image via semantic segmentation and fused into the map using a Bayesian approach assuming independence of individual measurements.

Dengler et al. [8] proposed an object-centric 2D/3D map representation for real-time service robotics applications, using RGB-D data as input. A geometric segmentation of small objects in the point cloud is obtained via Euclidean clustering. Stückler et al. [22] fuse probabilistic object segmentations from multiple RGB-D camera views into a voxel-based 3D map using a Bayesian framework.

Recently, Bultmann et al. [3] proposed a framework for online multi-modal semantic fusion onboard a UAV combining 3D LiDAR data with detections and semantic segmentation of 2D color and thermal images. The semantic point clouds are aggregated into an allocentric voxel-based map using poses from LiDAR odometry to transform multiple viewpoints into a common coordinate frame.

3D Human Pose Estimation. 2D human joint detection, inferred by image CNNs as introduced above, provides the input for 3D human pose estimation. 3D poses are recovered from 2D keypoint detections from multiple, calibrated camera views via variants of the Pictorial Structures Model (PSM) [15, 9] or based on direct triangulation [6, 18]. The PSM approaches are computationally expensive, due to a large discrete state space used in the optimization, restricting them to offline processing. Triangulation-based approaches are more computationally efficient and enable 3D pose estimation for multiple persons in real time.

In previous work [2], we proposed a pipeline for real-time 3D human pose estimation using multiple calibrated smart edge sensors that perform 2D pose

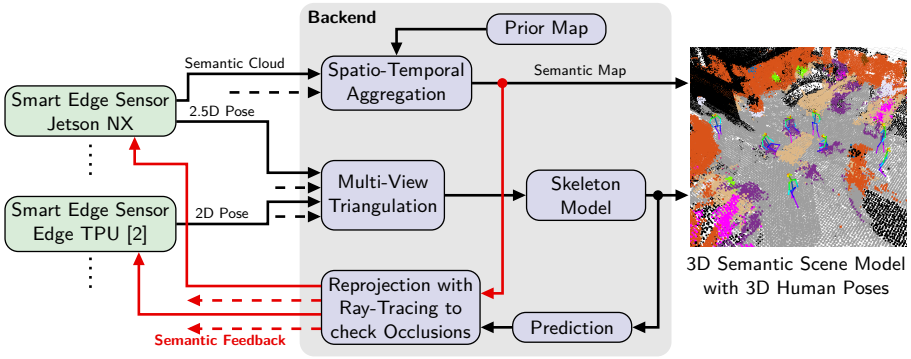

Fig. 2: Overview of the multi-sensor pipeline for 3D semantic mapping and human pose estimation: The Jetson NX smart edge sensors extend a sensor network from prior work [2] of nodes with lower compute capabilities. Semantic point clouds from multiple sensor views are aggregated into an allocentric 3D semantic map and 3D human poses are estimated in real time. The map is used to check reprojected joints for occlusion in the resp. sensor view via ray-tracing and this information is added to the semantic feedback sent to the smart edge sensors.

estimation on-device. Semantic pose information is transmitted to a central backend where multiple views are fused into a 3D skeleton via triangulation and an efficient, factor graph-based skeleton model. The fused allocentric 3D joint positions, after motion prediction to compensate for the pipeline delay, are reprojected into local views and sent back to the sensors as semantic feedback, where they are fused with the detected keypoint heatmaps. This enables the sensors to incorporate global context information into their local scene view interpretation. The pipeline delay is estimated as the difference of the timestamps of the current detection and the latest received feedback message on a sensor and updated using a moving average filter.

We build upon this work and extend the sensor network with new smart edge sensor nodes with significantly increased computational power and RGB-D cameras that enable the perception of 3D geometry. In addition to human pose estimation, object detection and semantic image segmentation are computed on the sensor boards and fused via 3D projection into a semantic point cloud. Semantic point clouds from multiple sensor views are fused into a sparse voxel hash-map with per-voxel full semantic class probabilities on the backend. The semantic map is used to obtain occlusion information for person keypoints in the local camera views, which is added to the semantic feedback to increase the robustness of the pose estimation pipeline.

3 Method

Figure 2 illustrates the proposed multi-sensor pipeline for 3D semantic perception and human pose estimation combining two types of smart edge sensors.

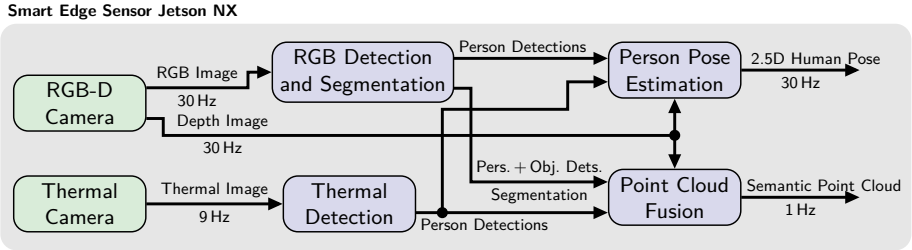

Fig. 3: Smart edge sensor semantic perception system overview. Human poses are estimated in real time, while the semantic point cloud of the static or slowly moving scene geometry is output at a lower frequency to save compute resources.

The proposed Jetson NX sensors are integrated into a sensor network from prior work [2], consisting of nodes based on the Google Edge TPU with lower compute capabilities and RGB image-only 2D human pose estimation, without local depth estimation. We consider a calibrated camera network, with known projection matrices from sensor to world coordinates, where the sensors are software-synchronized via NTP. Semantic mapping is only performed with the here proposed Jetson NX sensor nodes, while data from both sensor types is combined for 3D human pose estimation.

An overview of the proposed approach for semantic perception onboard each Jetson NX smart edge sensor is given in Fig. 3. We detail individual components of the data processing on each sensor board, as well as the fusion of multiple sensor views for 3D mapping and 3D human pose estimation in the following.

3.1 Smart Edge Sensor Hardware

We developed smart edge sensors based on the Nvidia Jetson Xavier NX developer kit[3] (cf. Fig. 1 (a)). They are equipped with a 6-core ARM processor, 384 CUDA cores, and 8 GB of RAM. The Jetson NX embedded system achieves a CNN inference performance of 21 trillion operations per second (TOPS), a significant increase compared to the 4 TOPS of the sensor platform employed in our previous work [2]. For visual perception, we connect an Intel RealSense D455 RGB-D camera and a FLIR Lepton 3.5 thermal camera to the Jetson NX board.

3.2 Single-View Embedded Semantic Perception

Person and Object Detection. We employ the recent MobileDet architecture [26] for person and object detection. The RGB detector is trained on the COCO dataset [12] using *person* and 12 indoor object classes (e.g., *chair*, *table*, *computer / tv*), with an input resolution of 848×480 px. The same network architecture is used for the thermal detector, taking one-channel 8-bit gray-scale thermal

[3] https://developer.nvidia.com/embedded/jetson-xavier-nx-devkit

images at the camera resolution of 160×120 px as input. The thermal detector is trained on the ChaLearn IPHD dataset [7], with annotations for the *person* class only.

Person Keypoint Estimation. We adopt a top-down approach for person pose estimation on the smart edge sensors, where crops of single persons are analyzed by the keypoint estimation CNN. The CNN architecture of Xiao et al. [25] is the basis of our person pose estimation, but we exchange the ResNet backbone with the significantly more lightweight MobileNet v3 feature extractor [11]. We train the pose estimation network on the COCO dataset [12] using person keypoint annotations.

Person detections from RGB and thermal images are forwarded to the keypoint estimation CNN. The RGB-D depth thereby is used to project detections from the thermal camera to the color image. Redundant detections of the same person in both modalities are filtered via non-maximum suppression (NMS). Each person crop is then resized to the fixed 192×256 input resolution of the keypoint estimation CNN and inference is run for all crops together in batched mode. Batch processing gives a significant improvement in the scaling of inference time with the number of persons compared to previous work [2], where the embedded hardware only supported processing a single crop at a time (cf. Sec. 4.4).

The pose estimation model outputs multi-channel images, called *heatmaps*, encoding the confidence of a joint being present at the pixel location. As single-person crops are processed, 2D joint locations are determined as global maxima of the respective heatmap channel. The RGB-D range image is used to augment the 2D keypoints to a 2.5D pose representation. For each joint, the median depth of a 5×5 px region around the joint location is obtained from the depth image. The local depth estimate enables the projection of keypoints into three-dimensional space but often suffers from noise and occlusions, as is further analyzed in Sec. 3.5. The 2.5D pose estimate for each detected person is sent to a central backend, where multiple sensor views are fused into a coherent 3D pose representation. The pose estimation pipeline runs with the highest real-time priority on the sensor boards, to enable tracking of dynamic human motions. To save computational resources, the person detector is run only once per second and the crops are updated based on the keypoint estimations between detector runs.

Semantic Segmentation. We adopt the DeepLab v3+ [5] architecture with MobileNet v3 [11] backbone for semantic segmentation. We train the model on the indoor scenes of the ADE20K dataset [27] and reduce the labels to 16 classes most relevant for the intended indoor application scenarios (cf. Fig. 1). The input image size is set to 849×481 px during inference, fitting to the 16:9 aspect ratio of our camera. The semantic segmentation is run only once per second, similar to the person and object detector, to save computational resources.

3.3 Multi-Modal Semantic Point Cloud Fusion

We obtain a geometric point cloud by projecting the RGB-D range image into 3D. The point cloud is uniformly subsampled using a voxel-grid filter with 5 cm

resolution to reduce the amount of data, economizing computational resources, and later network bandwidth for transmission to the backend. Sparse outlier measurements are further removed by a statistical outlier filter, as implemented in the PCL library [19]. A point is deleted when the distance to its neighbors is outside an interval defined by the mean and standard deviation of the entire point cloud.

Semantic information from RGB and thermal detections, as well as RGB semantic segmentation, is fused into the point cloud using a projection-based approach as proposed by Bultmann et al. [3]. For this, the points are projected into the segmentation mask inferred from the RGB image. The semantic class scores $\boldsymbol{c}_{\text{segm}} \in \mathbb{R}^C$ are obtained from semantic segmentation via bilinear interpolation at the projected point location. A normalized probability distribution over the employed $C = 16$ classes is then approximated by applying the soft-max operation:

$$p_i = \sigma\left(c_i\right) = \frac{\exp c_i}{\sum_{j=1}^{C} \exp c_j}\,, \tag{1}$$

obtaining $\boldsymbol{p}_{\text{segm}} \in \mathbb{R}^C$, with $p_i \in [0, 1]$ and $\sum_i p_i = 1$. If a projected point falls inside a detection bounding box in either thermal or color images, we further fuse the detector result with the semantic segmentation. We reconstruct the detection probability distribution $\boldsymbol{p}_{\text{det}}$ from the score for the detected class following the maximum entropy principle: The probability of the detected class p_{det} is given by the detector score and the remaining probability mass $1 - p_{\text{det}}$ is equally distributed over the remaining $C - 1$ classes. Both estimates are fused following the Bayesian update rule [14], assuming independence of segmentation and detection:

$$\boldsymbol{p}_{\text{fused}} = \frac{\boldsymbol{p}_{\text{segm}} \circ \boldsymbol{p}_{\text{det}}}{\sum_{i=1}^{C} p_{i,\text{segm}}\, p_{i,\text{det}}}\,, \tag{2}$$

with $\circ$ being the coefficient-wise product. For better numerical stability, we use the implementation of the Bayesian fusion in logarithmic form from [3].

As the detection bounding boxes are axis-aligned, border-effects have to be considered for non-rectangular or non-axis-aligned objects before detection fusion. Inclusion of all points projected into the bounding box in the fusion would falsely label points on the ground and in the background as the detected class. To alleviate this issue, the ground plane is removed and the remaining points are clustered in 3D Euclidean space by a distance threshold [3]. Only the clustered points are included into detection fusion.

The output semantic point cloud includes the class probability vector and the argmax class color per point (cf. Fig. 4 (a)) and is sent to the central backend over the network. It is computed at a reduced update frequency of 1 Hz on the sensors, as it is targeted to observe the static or slowly moving scene geometry. Thus, computational resources are kept free for the real-time estimation of dynamic human motions in the pose estimation pipeline.

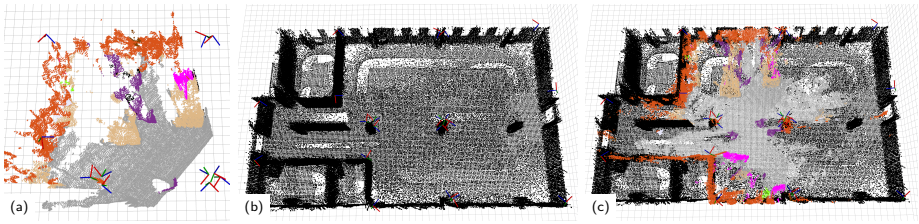

Fig. 4: 3D Semantic Mapping: (a) semantic point cloud of a single sensor, (b) prior map, (c) fused semantic map. The smart edge sensors send semantic point clouds of their resp. perspectives to the backend. Here, the fused map is initialized with a prior map and updated with the observations including semantic classes.

3.4 3D Semantic Mapping

Semantic point clouds from multiple calibrated camera perspectives are fused into an allocentric semantic map on the central backend, as illustrated in Fig. 4. For this, the 3D space is uniformly subdivided into cubic volume elements (voxels). We employ sparse voxel hashing [16] as a memory-efficient data structure.

For indoor environments, prior information on building structure is often easily available, e.g., via floor plans or 3D models. To incorporate this prior information, we initialize the scene model with a prior map of the empty building (Fig. 4 (b)). Here, the prior map was obtained from aggregated laser scans of the empty rooms, but it could also be replaced, e.g., by a floor plan with a fixed wall height or an architectural CAD model of the building.

To include semantic information and current observations into the map, we transform the semantic point clouds from individual sensors (Fig. 4 (a)) into global coordinates using the known camera calibration and bin the points into voxels of 10 cm side length. The semantic probabilities of all points falling into a voxel are fused probabilistically, using Bayes' rule [14], assuming independence between observations $P\left(l_i|X_k\right)$ for the semantic point cloud X_k with label l_i for class i:

$$P(l_i|X_{1:k}) = \frac{P\left(l_i|X_{1:k-1}\right) P\left(l_i|X_k\right)}{\sum_i P\left(l_i|X_{1:k-1}\right) P\left(l_i|X_k\right)} \,. \tag{3}$$

We again use the implementation of Bayesian fusion in logarithmic form from [3] for better numerical stability. Points labeled as *person* are not included in the semantic map, as dynamic human segments are tracked at a higher update rate via the 3D skeleton representation (cf. Sec. 3.5).

The fused semantic map (Fig. 4 (c)) contains 3D geometry and semantic classes of the areas observed by the smart edge sensors and is completed by the prior information for currently unobserved areas.

To account for moving objects, we adopt a simple ray-tracing approach to update occupancy information of the voxels [21], as illustrated in Fig. 5. Starting from the sensor pose towards the measured voxels, we ray-trace using a 3D

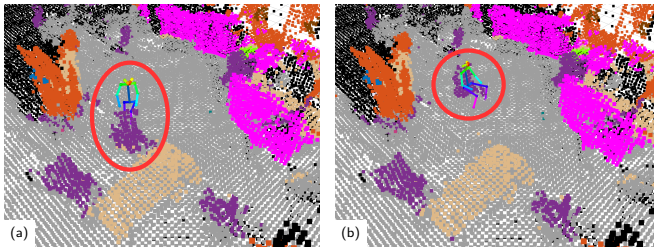

Fig. 5: Map update: 3D Semantic map and 3D person skeleton (a) before and (b) after moving a chair (highlighted with red circle). The semantic map is updated via ray-tracing to account for moving objects.

implementation of Bresenham's algorithm [1]. All voxels between the start and endpoint of the ray are updated as being free space, while the measured voxels are updated as being occupied. The semantic class probability is reset when a voxel state transitions from occupied to free.

3.5 3D Human Pose Estimation with Occlusion Feedback

The 3D joint positions of the detected persons are recovered from a set of 2D keypoint detections from multiple viewpoints via triangulation, and the result is refined using a factor graph skeleton model, as proposed in [2]. Furthermore, a semantic feedback channel from backend to sensors is implemented in our framework that enables the local semantic models of each sensor to incorporate globally-fused 3D pose information.

In this work, we add occlusion information for each joint to the semantic feedback, using the estimated 3D semantic map (cf. Fig. 2). We employ ray-tracing to check each joint for occlusion in the respective local sensor view. For this, we traverse the ray from the respective camera pose to the 3D joint through the 3D map using Bresenham's 3D line-search [1]. When the ray hits a minimum number of $k = 2$ occupied voxels, the joint is marked as occluded in the respective local view.

The benefits of the occlusion information for the local sensor model are illustrated in Fig. 6. Without occlusion feedback, heavy occlusion causes the pose estimation to collapse to the visible side only (Fig. 6 (b)), which cannot be recovered by the feedback on the heatmap level [2]. With occlusion information (Fig. 6 (a)), unreliable, occluded joint detections can be discarded, and the local model is completed by the more reliable semantic feedback. Completely occluded persons can also be added back into the local model (Fig. 6 (d)), making the sensor aware of persons that are going to re-appear in the future. Furthermore, the known occluded joints are excluded from multi-view triangulation in the next forward pass, as no new information can be gained from the respective sensor view. In Sec. 4.2, we show that the added occlusion information improves the overall consistency in terms of reprojection error.

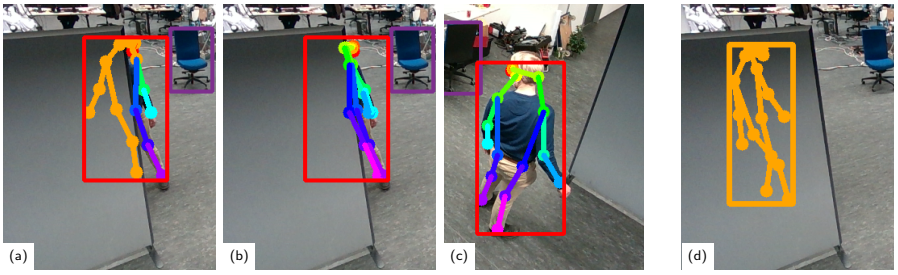

Fig. 6: Occlusion information in semantic feedback: Local 2D pose estimation (a) with and (b) without occlusion information via semantic feedback, (c) reference view without occlusion, (d) fully occluded person. Person detections in red and skeleton keypoints colored by joint index. Occluded joints are marked in orange. Heavy occlusion causes the pose estimation to collapse to the visible side only. With occlusion information, unreliable, occluded joint detections can be discarded, and the local model is completed by the more reliable semantic feedback.

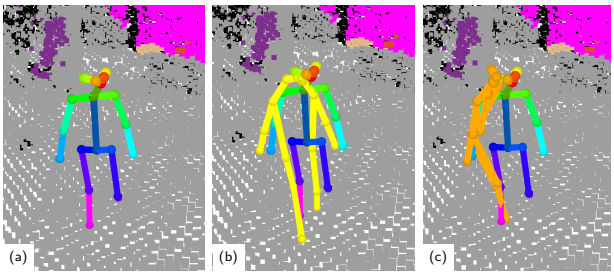

Fig. 7: Comparison of multi-view triangulation and local depth for estimating person keypoints in 3D: (a) multi-view triangulated 3D skeleton, (b) local depth from front view, and (c) local depth from side view. The local depth estimate results in a good approximation of the 3D skeleton for a front view, but is inaccurate in case of self-occlusion, e.g. from a side view. Multi-view triangulation is more robust but requires synchronization with other sensors.

We further investigate the reliability of the local depth estimate of skeleton joints from the Jetson NX smart edge sensors. The local depth enables estimating 3D joint positions from a single camera only, without dependence on other sensors. However, the RGB-D depth suffers from significant noise at larger distances and the depth measurement for a joint often is obstructed by occlusions or self-occlusions, as illustrated in Fig. 7. The local depth estimate results in a good approximation of the 3D skeleton for a front view, but is inaccurate in case of self-occlusion, e.g. from a side view. Multi-view triangulation is more robust to these issues but requires synchronization with other sensors.

The local depth estimate, however, can still be used as an indication to constrain the data association between cameras for multi-view triangulation. Person detections from different camera views are associated based on the epipolar

distance of their joints using the efficient iterative greedy matching proposed by Tanke et al. [24]. Keypoint detections from one image are projected as epipolar lines into the other cameras, where the distance from corresponding joint detections to the epipolar line is used as data-association cost. When a depth estimate is available, including an uncertainty interval computed from the keypoint confidence and the distribution of local depth readings, the matching can be restricted to a line segment. This helps to resolve ambiguous situations, where keypoints from multiple persons have a low distance to the epipolar line but are located at different positions along the line. Keypoints located on the line segment close to the projected depth estimate will receive lower data association cost while correspondences outside the projected depth interval will be discarded.

4 Evaluation

We evaluate the proposed system in challenging, cluttered real-world indoor scenes with multiple persons.

4.1 Implementation Details

Our sensor network consists of 20 smart edge sensors, thereof 4 based on the Jetson NX board, as introduced in this paper, and 16 based on the Google Edge TPU [2]. The boards are connected to mains power supply and the power consumption of an NX board is 20 – 25 W during inference, thereof ∼5 W for powering the RGB-D camera, compared to 7 W for the Edge TPU board. The sensors cover an area of roughly 12×22 m. The cameras face downward towards the center and run at 30 Hz. We conduct experiments with the proposed, extended sensor network, with 8 persons moving in the covered area, which are evaluated in the following using a sequence of 106 s containing ∼3,000 frames per camera.

4.2 Quantitative Results

To analyze the consistency between local and globally-fused human pose estimation, we evaluate the error between 2D poses detected in the individual sensor

Table 1: Evaluation in real-world multi-person scenes with 20 cameras and 8 persons: Reprojection error (px) per joint class between detected 2D poses and fused 3D poses.

Feedback	Cams	Pers	Head	Hips	Knees	Ankls	Shlds	Elbs	Wrists	**Avg**
w/o fb	20	8	5.09	5.98	5.75	6.87	4.67	5.53	6.95	5.69
fb [2]	20	8	4.76	5.51	4.98	5.94	4.34	4.88	5.66	5.08
fb + occl.	20	8	4.36	4.68	4.37	5.44	3.97	4.38	**5.04**	4.56
fb + occl. + local depth	20	8	**4.30**	**4.63**	**4.32**	**5.42**	**3.91**	**4.33**	**5.04**	**4.51**

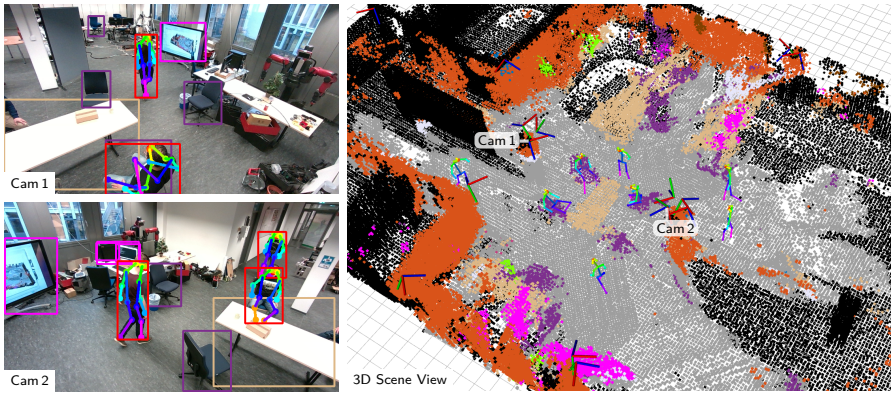

Fig. 8: Experiments in real-world multi-person scenes: Local detections of two reference sensor views and 3D semantic scene view with 3D poses of eight persons estimated in real-time. The geometry of the room is accurately represented by the map, fusing prior map (black) and current observations of smart edge sensors with their semantic classes (e.g., tables, chairs, computers). Interactions between persons and the scene, e.g., persons sitting on chairs (violet), are explained in a physically plausible way by the scene model.

views and reprojected fused 3D poses in Tab. 1. The reprojection error decreases for all joint classes when using the semantic feedback proposed in [2] over a purely feed-forward pipeline. Adding the occlusion information, as proposed in this work, further decreases the reprojection error, as unreliable occluded keypoint detections can be discarded and excluded from multi-view triangulation. Constraining the data association using the local depth estimates of the RGB-D cameras gives a further small improvement. The proposed pipeline leads to the lowest reprojection error for all joint classes, amounting to 4.51 px on average, indicating that the consistency between local and globally-fused pose estimation increases through the semantic feedback with occlusion information and by using local depth estimates in the data association step for multi-view triangulation.

4.3 Qualitative Results

An exemplary scene of the real-world multi-person experiments is shown in Fig. 8. Local detections and pose estimation in two reference camera views are depicted together with the 3D semantic scene view. 3D poses of eight persons are estimated online, in real-time during the experiment. The semantic map represents the 3D geometry of the scene, fusing prior map and current sensor observations, including semantic class probabilities. Interactions between persons and objects in the scene, e.g., persons sitting on chairs, are explained in a physically plausible manner by the scene model.

Table 2: Average inference time and validation score (given as mAP for detectors and pose estimation and mIOU for segmentation) of CNN models (batch size 1) on different embedded hardware and for different numerical precision.

Model	Input Res.	Edge TPU (int8)		Jetson NX (fp16)		Jetson NX (int8)	
		time	val. score	time	val. score	time	val. score
RGB det.	848×480	-	-	24.1 ms	36.2 %	**11.8 ms**	36.0 %
RGB det.	640×480	21.5 ms	**36.7 %**	-	-	-	-
Pose est.	192×256	4.5 ms	68.4 %	4.0 ms	**69.3 %**	**3.5 ms**	68.6 %
Thermal det.	160×120	-	-	13.9 ms	**25.4 %**	**6.0 ms**	24.9 %
RGB segm.	849×481	-	-	27.0 ms	**50.0 %**	**20.0 ms**	48.8 %

Table 3: Average processing time for pose estimation (inference + post-processing) for increasing number of person detections per image. Batch processing can be used on Jetson NX while crops are processed one by one on the Edge TPU.

Sensor Type	Precision	1	2	3	4	5	6
Edge TPU	int8	14.4 ms	23.6 ms	33.0 ms	43.9 ms	54.3 ms	65.9 ms
Jetson NX	int8	12.1 ms	**15.0 ms**	**18.0 ms**	27.4 ms	**31.0 ms**	**38.8 ms**
Jetson NX	fp16	**11.8 ms**	17.4 ms	21.9 ms	**26.4 ms**	31.7 ms	41.5 ms

4.4 Run-time Analysis

We analyze the run-time of CNN inference and the validation score on the respective training dataset (cf. Sec. 3.2) on different embedded hardware accelerators and for different numerical precision for the employed models in Tab. 2. Thermal detector and RGB segmentation are only executed on the Jetson NX, as the Edge TPU does not have enough computational power to run all models in parallel. The run-times on Jetson NX in 16-bit floating-point mode (fp16) are comparable to 8-bit quantized (int8) inference on the Edge TPU. The inference times roughly halve when using int8 precision on Jetson NX for the detectors and also decrease for the segmentation. For pose estimation, here stated for a single crop and batch size 1, the difference is less significant. The inference time is only about 4 ms, and the precision is less relevant compared to other overhead from the inference framework in this case. The validation score is given as bounding-box or keypoint mean average precision (mAP) for the *person* class as defined for the COCO dataset [12] for the detectors or pose estimation, respectively, and as mean intersection over union (mIOU) for the semantic segmentation. It decreases between 0.2 and 1.2 % when using int8 precision instead of fp16. The slightly better performance of the RGB detector on the Edge TPU can be explained as it was trained for the *person* class only, while the detector used on Jetson NX was trained for *person* and 12 indoor object classes.

Table 3 shows the scaling of processing time for pose estimation, including CNN inference and post-processing, with an increasing number of person detections per image. During our experiments with 8 persons in the scene, a maximum

of 6 persons were visible at a time in one camera. On the Edge TPU sensors [2], crops are processed one by one, as only a batch size of one is supported, and the runtime scales linearly. Up to three persons can be tracked at the full camera frame rate of 30 Hz. On the Jetson NX platform, batch-processing is possible, and therefore the run-times scale sub-linearly. Up to five persons can be tracked at the full camera frame rate. For pose inference, there is only a small difference in run-time between fp16 and int8 mode on Jetson NX.

We run CNN inference in fp16 mode on the Jetson NX smart edge sensors during our online experiments to benefit from the higher numerical precision, as 8-bit quantization gives only little gains in run-time for the pose estimation with strong real-time constraints, and the fp16 inference time is sufficient for the other models that run with lower priority at a 1 Hz update rate.

5 Conclusion

In this work, we presented a network of distributed smart edge sensors for 3D semantic scene perception, including static or slowly moving geometry as well as dynamic human motions. RGB-D and thermal camera images are processed locally on the sensor boards with vision CNNs for person and object detection, semantic segmentation, and human pose estimation. 2D human keypoint estimations, augmented with the RGB-D depth estimate, as well as semantically annotated point clouds are streamed from the sensors to a central backend, where multiple viewpoints are fused into an allocentric 3D semantic scene model. The individual sensors incorporate global context information into their local models via a semantic feedback channel. For this, the globally-fused 3D human poses are projected into the sensor views, where they are fused with the local detections. The estimated 3D geometry enables to add occlusion information for each joint to the semantic feedback, such that unreliable, occluded joint detections can be discarded on the sensors, and the local models can be complemented by the more reliable feedback joint positions. We built a sensor network of 20 smart edge sensors, thereof 4 based on the novel Jetson NX board, covering an area of about 12×22 m, and evaluated the proposed system in challenging, cluttered real-world scenes with up to 8 persons. Dynamic human motions are estimated in real time and the semantically annotated 3D geometry provides a complete scene view that also explains interactions between persons and objects in the scene.

Future work includes using the 3D semantic scene model and human poses estimated by the smart edge sensors to enable anticipative robot behavior and safe human-robot interaction in a shared workspace. Mobile sensor nodes could further be added to the sensor network for active exploration of areas not covered by the permanently installed sensors.

Acknowledgments

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

References

1. Amanatides, J., Woo, A.: A fast voxel traversal algorithm for ray tracing. In: 8th Europ. Computer Graphics Conference and Exhibition (EuroGraphics) (1987)
2. Bultmann, S., Behnke, S.: Real-time multi-view 3D human pose estimation using semantic feedback to smart edge sensors. In: Robotics: Science and Systems (RSS) (2021)
3. Bultmann, S., Quenzel, J., Behnke, S.: Real-time multi-modal semantic fusion on unmanned aerial vehicles. In: Europ. Conf. on Mobile Robots (ECMR) (2021)
4. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 43(1), 172–186 (2021)
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Europ. Conf. on Computer Vision (ECCV). pp. 833–841 (2018)
6. Chen, L., Ai, H., Chen, R., Zhuang, Z., Liu, S.: Cross-view tracking for multi-human 3D pose estimation at over 100 FPS. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 3276–3285 (2020)
7. Clapés, A., Jacques Junior, J.C.S., Morral, C., Escalera, S.: ChaLearn LAP 2020 challenge on identity-preserved human detection: Dataset and results. In: IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG). pp. 801–808 (2020)
8. Dengler, N., Zaenker, T., Verdoja, F., Bennewitz, M.: Online object-oriented semantic mapping and map updating. In: Europ. Conf. on Mobile Robots (ECMR) (2021)
9. Dong, J., Jiang, W., Huang, Q., Bao, H., Zhou, X.: Fast and robust multi-person 3D pose estimation from multiple views. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) pp. 7784–7793 (2019)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
11. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V., Adam, H.: Searching for MobileNetV3. In: IEEE Int. Conf. on Computer Vision (ICCV). pp. 1314–1324 (2019)
12. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Europ. Conf. on Computer Vision (ECCV). pp. 740–755 (2014)
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: Europ. Conf. on Computer Vision (ECCV). pp. 21–37 (2016)
14. McCormac, J., Handa, A., Davison, A., Leutenegger, S.: SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In: IEEE Int. Conf. on Robotics and Automation (ICRA). pp. 4628–4635 (2017)
15. Qiu, H., Wang, C., Wang, J., Wang, N., Zeng, W.: Cross view fusion for 3D human pose estimation. In: IEEE Int. Conf. on Computer Vision (ICCV). pp. 4341–4350 (2019)
16. Quenzel, J., Behnke, S.: Real-time multi-adaptive-resolution-surfel 6D LiDAR odometry using continuous-time trajectory optimization. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). pp. 5499–5506 (2021)
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 779–788 (2016)

18. Remelli, E., Han, S., Honari, S., Fua, P., Wang, R.: Lightweight multi-view 3D pose estimation through camera-disentangled representation. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 6039–6048 (2020)
19. Rusu, R.B., Cousins, S.: 3D is here: Point cloud library (PCL). In: IEEE Int. Conf. on Robotics and Automation (ICRA) (2011)
20. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted residuals and linear bottlenecks. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 4510–4520 (2018)
21. Schleich, D., Beul, M., Quenzel, J., Behnke, S.: Autonomous flight in unknown GNSS-denied environments for disaster examination. In: Int. Conf. on Unmanned Aircraft Systems (ICUAS). pp. 950–957 (2021)
22. Stückler, J., Biresev, N., Behnke, S.: Semantic mapping using object-class segmentation of RGB-D images. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). pp. 3005–3010 (2012)
23. Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural metworks. In: Int. Conf. on Machine Learning (ICML). pp. 6105–6114 (2019)
24. Tanke, J., Gall, J.: Iterative greedy matching for 3D human pose tracking from multiple views. In: German Conf. on Pattern Recognition (GCPR). pp. 537–550 (2019)
25. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Europ. Conf. on Computer Vision (ECCV). pp. 466–481 (2018)
26. Xiong, Y., Liu, H., Gupta, S., Akin, B., Bender, G., Wang, Y., Kindermans, P.J., Tan, M., Singh, V., Chen, B.: MobileDets: Searching for object detection architectures for mobile accelerators. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 3825–3834 (2021)
27. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ADE20K dataset. International Journal of Computer Vision 127(3), 302–321 (2019)