2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

External Camera-based Mobile Robot Pose Estimation for Collaborative Perception with Smart Edge Sensors

Simon Bultmann, Raphael Memmesheimer, and Sven Behnke

***Abstract*— We present an approach for estimating a mobile robot's pose w.r.t. the allocentric coordinates of a network of static cameras using multi-view RGB images. The images are processed online, locally on smart edge sensors by deep neural networks to detect the robot and estimate 2D keypoints defined at distinctive positions of the 3D robot model. Robot keypoint detections are synchronized and fused on a central backend, where the robot's pose is estimated via multi-view minimization of reprojection errors. Through the pose estimation from external cameras, the robot's localization can be initialized in an allocentric map from a completely unknown state (*kidnapped robot problem*) and robustly tracked over time. We conduct a series of experiments evaluating the accuracy and robustness of the camera-based pose estimation compared to the robot's internal navigation stack, showing that our camera-based method achieves pose errors below 3 cm and 1° and does not drift over time, as the robot is localized allocentrically. With the robot's pose precisely estimated, its observations can be fused into the allocentric scene model. We show a real-world application, where observations from mobile robot and static smart edge sensors are fused to collaboratively build a 3D semantic map of a ∼240 m² indoor environment.**

I. Introduction

Semantic scene understanding is an important requirement for intelligent robot action, such as collision-free navigation, object manipulation, or human-robot interaction. Scene interpretation from a single sensor view, however, has a limited field of perception. Collaborative perception between mobile robots and distributed static smart edge sensors alleviates this issue and enables to build 3D semantic models of large scenes without being limited by the measurement range or occlusions of a single sensor.

A key prerequisite for fusing observations from different perspectives is knowing the relative sensor poses. While the extrinsic calibration of static sensors can be performed beforehand, a mobile sensor's pose w.r.t. the sensor network must be initialized and tracked to fuse the robot observations into the allocentric model in a consistent manner.

In this work, we propose to estimate a mobile robot's pose w.r.t. the allocentric coordinates of a network of static smart cameras using multi-view RGB images. For this, we build upon our previous work on 3D semantic scene perception using distributed smart edge sensors [1], where images are processed locally on the sensor boards by deep neural networks for semantic image interpretation. For robot pose estimation, we process the images with Convolutional Neural Networks (CNNs) for robot detection and estimation of 2D projections of keypoints defined at distinctive positions of the 3D robot model. Unlike classical robot-to-camera pose estimation systems, our method does not require attaching fiducial markers to the robot. Furthermore, the CNN for keypoint estimation is trained only on synthetic data obtained through randomized scene generation [2]. Robot keypoint detections are streamed to a central backend where they are synchronized and the robot's pose is estimated via multi-view minimization of reprojection errors. Through the pose estimation from external camera views, the robot's localization is initialized in the global map from a completely unknown state, and robustly tracked over time. Furthermore, using multiple sources for localization, i.e. the external camera views together with the robot's internal 2D LiDAR-based navigation, increases robustness in highly cluttered, dynamic real-world environments, where few distinct features, such as walls or columns, are visible in the LiDAR due to occlusions.

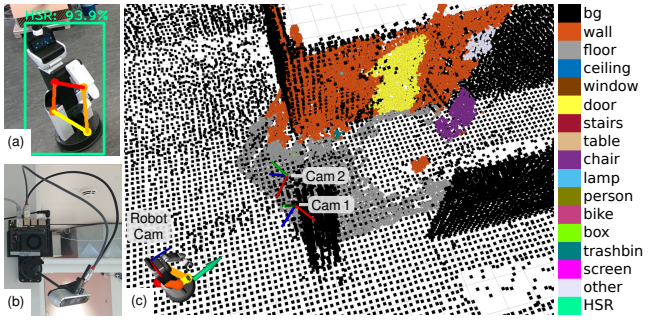

Fig. 1. Robot pose estimation for collaborative perception: (a) Toyota Human Support Robot (HSR) employed in this work, with 2D bounding box and keypoints used for pose estimation detected by static smart edge sensors (b). (c) 3D scene view with robot model and 3D keypoints at the estimated robot pose (green arrow) and robot's semantic observations (colored point cloud) to be fused into the allocentric scene model (black squares). Robot observations fit the allocentric model, showing that the robot's pose is initialized globally coherent through localization by the smart edge sensors.

With the robot's pose precisely estimated, it is integrated as a mobile sensor node into the camera network. We deploy the CNNs for semantic image interpretation from [1] onto the robot's inference accelerator and use its RGB-D camera to obtain semantically annotated point clouds of the robot's view. The robot's observations are then fused into the allocentric scene model to build a 3D semantic map of a large room in collaboration with the static smart edge sensors.

Fig. 1 shows the employed Toyota Human Support Robot (HSR) and static smart edge sensors together with a 3D view of the robot's semantic observations to be fused into the 3D scene model, initialized from a prior without any semantic annotations. As the robot pose is initialized globally coherently through the external keypoint-based pose estimation,

This work was funded by grant BE 2556/16-2 (Research Unit FOR 2535 Anticipating Human Behavior) of the German Research Foundation (DFG)

All authors are with the Autonomous Intelligent Systems group, University of Bonn, Germany; bultmann@ais.uni-bonn.de

its observations consistently fit the allocentric model.
In summary, the main contributions of this paper are:

- A novel method for marker-less mobile robot pose estimation using multi-view keypoint detections to initialize the robot's localization in a global map (*kidnapped robot problem*) and robustly track it over time,
- integration of a mobile robot into a network of static smart edge sensors, fusing the robot's observations from changing viewpoints into the global scene model,
- quantitative evaluation of the pose estimation accuracy and robustness, and
- demonstration of collaborative perception in real-world scenes between mobile robot and sensor network to build a globally consistent 3D semantic map.

II. Related Work

Visual robot detection and localization have been research topics of high interest for a variety of autonomous systems like drones [3], mobile robot platforms [4]–[6], humanoids [7], autonomous cars [8], and underwater robots [9], [10]. External localization of moving objects is also employed for traffic monitoring [11], [12] or action recognition [13].

Detection methods localize robots in images and allow for tracking applications like underwater convoying [9] or monitoring [3]. In contrast, we aim to estimate a robot's pose from multiple views to initialize its localization in a map and to compensate for localization errors in highly cluttered, dynamic environments. Gawel et al. [8] presented a semantic graph-based multi-view approach for robot localization in autonomous driving scenarios. They use semantic estimates from various views like front and aerial views and integrate them in separate target and query graphs. Lu et al. [14] presented an approach for marker-less robot pose estimation via keypoint detection [15]. They aim to define keypoints on the robot maximizing the localization performance in 2D and 3D. Their approach utilizes synthetic training data and transfers well to real sensor data. They estimate the pose of stationary robots from single views, whereas our approach globally tracks a mobile manipulation platform. Lee et al. [16] consider the inverse problem of localizing the camera in relation to an articulated robot. They estimate 2D keypoints of robot joints and finally recover the camera extrinsics w.r.t. the robot manipulator. Pizzaro et al. [4], [5] presented approaches for robot localization from a single view. Shim et al. [17] proposed an approach for multi-view mobile robot localization. They first project the camera images onto a common plane and then localize the robot using its contours after removing shadows. Similarly, Chakravarty and Jarvis [18] utilize a surveillance camera system to localize the robot in the image plane by background subtraction of a color thresholded image. Similar to our work, the surveillance camera view is extended by the robot's view. Obviously, such approaches would fail in highly dynamic and cluttered environments.

The field-of-view of autonomous mobile robots can be extended in comparison to single-view systems using smart edge sensors for global collaborative perception. In prior work, we utilized smart edge sensors to estimate 3D human poses [19] from multiple views and integrate 3D semantic perception [1]. In this work, we focus on the integration of a mobile robot as an additional smart edge sensor node, laying the foundations for many potential applications like external, camera-based control and task planning as well as augmentation of the internal robot view by the integration of human poses or semantic segmentation estimates that are robust against occlusions. Rekleitis et al. [20] presented an approach for collaborative exploration of visual maps using two mobile robots of different capability levels, where one robot was actively collecting visual data for mapping and the second, passive robot visually refined the pose of the moving robot. Dong et al. [21] proposed a multi-robot collaborative approach for dense reconstruction. They associate task views of uncertain or unexplored map areas to the robots based on the traveling salesman problem. In contrast to our approach, they concentrate on an actively moving set of mobile robot platforms and omit semantics. Similarly, Yue et al. [22], propose to combine an Unmanned Aerial Vehicle (UAV) and an Unmanned Ground Vehicle (UGV) for collaborative semantic mapping. Ahmed et al. [23] propose to integrate top-view surveillance cameras for collaborative robotics but assume that the estimations from surveillance cameras are provided to the robot without actively fusing information of the robot's sensors.

With PoseCNN [24], a 6D pose estimation approach jointly estimating semantic labels on a pixel level has been presented. Xiang et al. estimate the translation by regressing the object center in camera coordinates and separately regress the rotation using a newly introduced ShapeMatch-Loss. Wang et al. [25] presented a dense fusion method for 6D pose estimation. They first estimate semantic segmentation and bounding boxes, and then densely fuse RGB and depth images in an embedding. Finally, an iterative pose integrator refines the 6D pose. Keypoint-based approaches for 6D pose estimation adopt a two-stage pipeline, first estimating keypoints for each object instance and, second, retrieving the object poses via PnP [26]. They have been widely used in recent years [16], [27]–[30]. Although most pose estimation approaches are object-centric, in this paper, we propose to use a keypoint-based estimator for the visual pose estimation of a mobile robot.

III. Method

We consider scenarios where N externally mounted cameras $C_i, i = 1, \dots, N$ observe a mobile robot from different viewpoints. The intrinsic and extrinsic calibration of the external cameras are performed beforehand [31], i.e., we assume the transformation ${}^{C_i}_{W}\boldsymbol{T} \in \mathbb{R}^{4\times4}$ from world to camera coordinates and the projection $\Pi_i(\cdot)$ to the image plane of C_i to be known. The cameras observe 2D projections $\boldsymbol{k}_{ij} \in \mathbb{R}^2$ of M keypoints $\boldsymbol{p}_j \in \mathbb{R}^3, j = 1, \dots, M$ defined on distinct positions of the 3D robot model. From these observations, we aim to estimate the robot's pose in world coordinates ${}^{W}_{R}\boldsymbol{T} \in \mathbb{R}^{4\times4}$. As the employed mobile robot moves on the

ground plane, we consider three degrees of freedom (DoF): $\boldsymbol{x} = [x, y]^\top \in \mathbb{R}^2$, and $\theta \in (-\pi, \pi]$, with

$$ {}^W_R\boldsymbol{T}\,(\boldsymbol{x}, \theta) = \begin{bmatrix} \boldsymbol{R}\,(\theta) & \boldsymbol{0} & \boldsymbol{x} \\ \boldsymbol{0}^\top & 1 & 0 \\ \boldsymbol{0}^\top & 0 & 1 \end{bmatrix}, \boldsymbol{R}\,(\theta) \in \mathcal{SO}\,(2)\ . \quad (1) $$

Extending this formulation to more DoFs is straightforward. We estimate the robot pose in a two-stage process, similar to Lee et al. [16] but exploiting multi-view observations: First, robot keypoints are detected on the images, locally on the sensor boards (Sec. III-A), and, second, the robot pose is estimated from a synchronized set of observations via multi-view minimization of reprojection errors (Sec. III-B).

A. Robot Keypoint Detection

We aim to estimate the robot's base pose, independent of the articulation of the robot head or arm, and therefore define the keypoints used for pose estimation on distinct points of the rigid robot base (see Fig. 2 (c)). Keypoints defined manually at distinct points of the 3D model were shown to perform better than automatically defined ones [29].

1) Network Architecture: We employ a top-down approach for robot keypoint detection, i.e., first detecting a bounding box of the robot in the full image (848×480 px) and then estimating keypoints on the crop of the robot. Top-down methods achieve better scale invariance than estimating keypoints on the full input image, as the crop of a detected robot is interpolated to a fixed resolution for keypoint estimation (here 192×256 px). Camera images are processed locally on the smart edge sensors, using a Nvidia Jetson Xavier NX embedded inference accelerator (cf. Fig. 1 (b)). We employ the CNN architectures used in prior work for efficient person keypoint estimation on the embedded hardware [1], [19]: The recent MobileDet architecture [32] is used for robot detection and the network of Xiao et al. [33] with a MobileNet V3 [34] backbone for keypoint estimation.

2) Training Data: To reduce the labeling effort to a minimum, we train the networks predominantly on synthetic data. The CNN for keypoint estimation is trained purely on simulated data (36k samples), while we combine synthetic data and manually annotated real images (12k resp. 3.5k samples) for robot detection. The combination of real and synthetic data helps to boost detector performance in highly cluttered real-world environments and bounding-box labels are less costly to obtain than keypoint annotations. The keypoint CNN generalizes well from only synthetic data.

We employ an extension of the *stilleben*-framework [2], [35] for photorealistic, randomized scene rendering to generate multiple scenes of our robot moving through varying indoor environments, using the 3D robot model. Randomizing scene parameters in addition to image augmentation helps to bridge the reality gap for sim-to-real transfer. Fig. 2 shows examples of the employed training data. Note, that the network also learns to detect occluded keypoints from the surrounding image context.

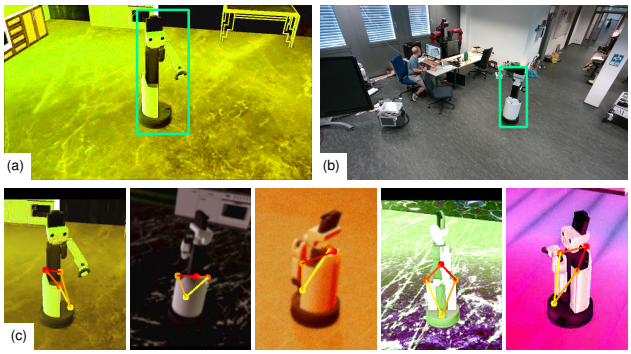

Fig. 2. Examples of training images: (a) synthetic and (b) real images used for the detector; (c) synthetic images used for the keypoint estimation.

B. Robot Pose Estimation

2D robot keypoints are sent over a network to a central backend, where detections from multiple cameras are software-synchronized via their timestamps and associated to corresponding frame-sets. The robot pose ${}^W_R\boldsymbol{T}$ (1) is then recovered by solving a weighted nonlinear least squares problem via minimization of multi-view reprojection errors:

$$ {}^W_R\boldsymbol{T} = \underset{{}^W_R\boldsymbol{T}}{\arg\min} \sum_{i=1}^{N} \sum_{j=1}^{M} w_{ij} \left\| \boldsymbol{k}_{ij} - \Pi_i \left({}^{C_i}_{W}\boldsymbol{T}\, {}^W_R\boldsymbol{T}\, \boldsymbol{p}_j \right) \right\|^2, \quad (2) $$

with weights w_{ij} depending on the confidence of the keypoint detection in the respective camera, similar to [19]. We use the Levenberg-Marquardt algorithm as implemented in the Ceres library [36] for optimization. We discern two different ways to initialize the optimization:

(i) When the robot pose has been initialized in the global map and a current pose estimate is available from its internal navigation stack, we use this to initialize the optimization.

(ii) When no prior estimate is available, e.g., to initialize the robot's pose or when the communication link to the robot is unavailable, we use the PnP algorithm [26] to obtain pose candidates from each individual camera view with $M \geq 4$ detected keypoints. The candidate poses are transformed to world coordinates using ${}^{C_i}_{W}\boldsymbol{T}^{-1}$ and projected to the ground plane. The initialization for Eq. (2) is then obtained via interpolation between the candidates, using spherical linear interpolation for the orientation.

In our experiments (Sec. IV), we observe that the pose estimation is very well constrained when the robot is detected in $N \geq 2$ cameras, but less stable when visible in only a single camera and far away from the camera. This is due to the robot body having a small width of 35 cm, providing only a narrow baseline for orientation and depth estimation. To alleviate this issue, we implement a simple yet effective bearing-only heuristic for outlier detection: For pose estimates from a single camera view, we prevent unrealistically high changes in orientation or distance to the camera above a threshold d_θ resp. d_{depth} by updating only the translation orthogonal to the camera's viewing direction.

Lastly, we employ a pose graph [37] to fuse the absolute pose estimations from external cameras, which occur sparsely, at waypoints where the robot is static (cf. Sec. III-C), with the continuous robot-internal odometry. The pose

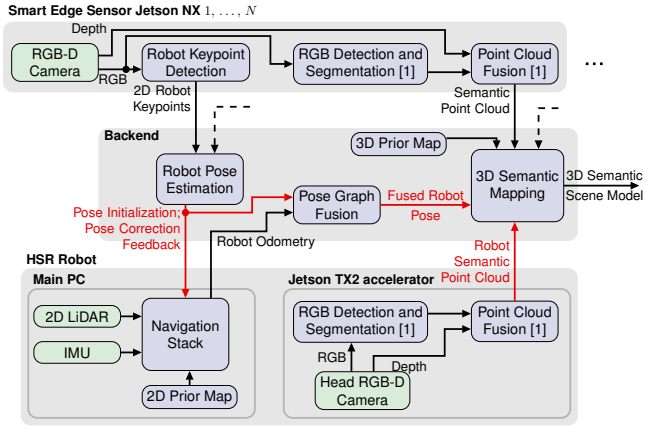

Fig. 3. Overview of the proposed sensor network architecture for collaborative localization and perception. N external smart edge sensors observe mobile HSR robot and scene from static viewpoints. Robot pose is initialized and corrected via external camera pose estimation. Robot observations from changing viewpoints are fused into the allocentric scene model.

estimates from the robot's internal navigation stack are inserted into the pose graph as binary odometry constraints, while the pose estimates from external cameras are used as absolute, unary constraints. The covariance of the camera pose estimates is proportional to the reprojection error residual (2) and inversely to the number of cameras used in the optimization, giving the highest confidence to consistent estimates with low reprojection error in a large number of cameras. The pose graph is optimized each time a new external camera pose estimate occurs.

C. Collaborative Localization and Perception

Fig. 3 gives an overview of the proposed sensor network architecture for collaborative localization and perception. The mobile HSR robot is integrated into a network of static smart edge sensors for 3D semantic scene perception [1]. The robot pose is initialized in the allocentric scene from the external cameras. During operation, external robot pose estimations are sent as pose-correction feedback to the robot sparsely at waypoints where the robot is static, updating the robot's internal particle filter-based localization [38] to keep it globally coherent within the scene model. The interface provided by the robot's navigation stack permits to update the localization from external measurements only when not in movement. To simplify system integration and for better interoperability, we use the robot's navigation stack as provided by the manufacturer and consider it as a closed module. We do not aim to fully control the robot localization via the external cameras for the robot to keep its local autonomy: Its navigation stack integrates information from the camera network when available, but does not depend on it. When, e.g., the communication link is lost, the robot can still navigate autonomously using its integrated sensors.

For collaborative perception, the mobile robot provides changing viewpoints of areas outside the field-of-view of the static smart edge sensors, and its semantic percepts are fused into the allocentric scene model.

1) Semantic Perception onboard the Robot: The HSR's computing system comprises a main PC used for communication, robot navigation, and control and an embedded deep learning inference accelerator (Nvidia Jetson TX2). We employ the latter to deploy CNNs for object detection and semantic segmentation from [1] and obtain semantic point clouds using the Asus Xtion RGB-D camera mounted on the robot's head. The robot's inference accelerator hardware is a previous generation of the compute boards of the static smart edge sensors, with lower, but sufficient resources for semantic point cloud estimation at 1 Hz.

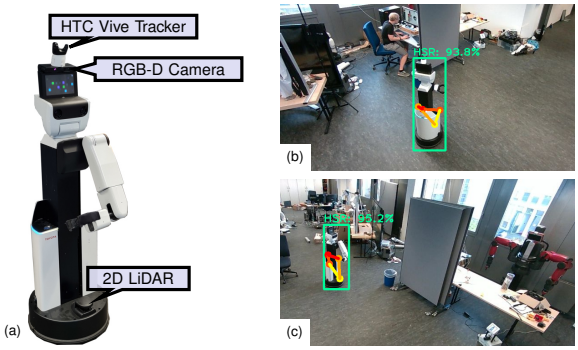

Fig. 4. Robot setup (a) and exemplary robot keypoint detections (b, c).

2) Semantic Map Fusion: The robot's semantic observations from changing viewpoints are integrated into the allocentric 3D semantic map using the probabilistic fusion proposed in [1]. Because we consider the mobile robot's pose less reliable than the a-priori calibrated static sensor poses, we perform an ICP alignment step before integrating the robot's data into the map. This compensates for the drift accumulated between pose corrections and tolerances in the robot's forward kinematics. As a good initialization is critical for ICP, precise robot localization is necessary for initializing the alignment.

IV. Evaluation

A. Experiment Setup

During the experiments, the HSR robot operates in a challenging, cluttered indoor environment of $\sim$240 m^2 size. Four external smart edge sensors are mounted at $\sim$2.5 m height in the center of the room to initialize and correct the robot localization. As a reference for pose estimation, we employ the affordable and easy-to-use HTC Vive Pro tracking system, which was shown to yield position accuracies within a few millimeters [39]. For this, we place an HTC tracker on the robot's head, as shown in Fig. 4. For evaluation of the pose estimation accuracy, we define seven waypoints in the area observed by the external cameras and connect them in different ways to three different trajectories (cf. Fig. 5). As the tracking system and the camera network don't operate in the same reference frame, trajectories are rigidly aligned via Procrustes analysis [40] before evaluation.

B. Pose Estimation Accuracy

During a first set of experiments, we record a dataset of three iterations of each trajectory in both forward and reverse order, resulting in 18 sample trajectories. At the beginning of each trajectory, the robot's pose is initialized from observations of all four cameras and we verify that

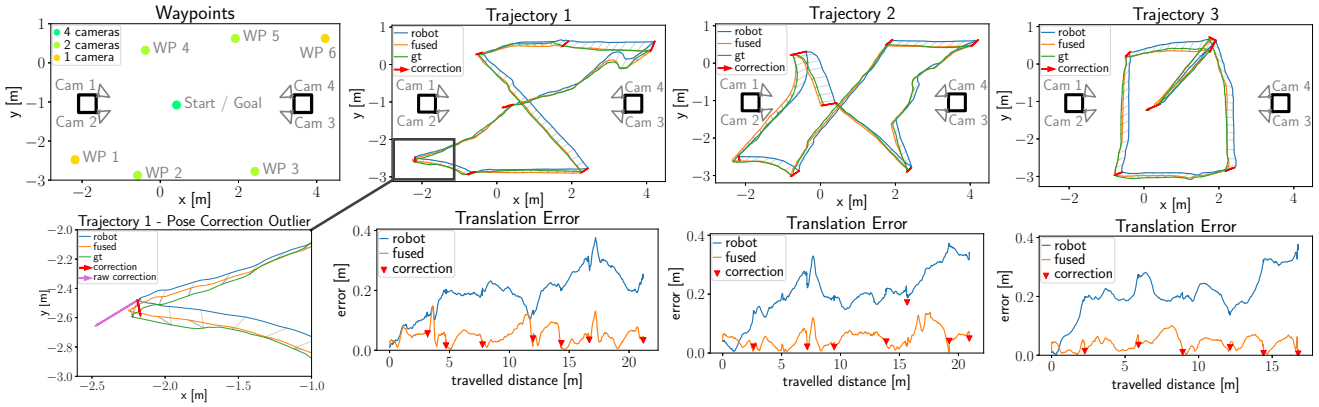

Fig. 5. Evaluation of robot pose estimation: top-left: Room-setup with camera positions and waypoints colored by number of cameras from which waypoints are visible; center and right: plots of one resp. iteration of the three trajectories used for evaluation of the translation error over the traveled distance; bottom-left: example for outlier detection at WP 1 observed from only a single camera. The pose correction from the external cameras is not sent as feedback to the robot during these experiments to measure the deviation over the full trajectory length. See Sec. IV-B for further explanations.

TABLE I

TRANSLATION AND ORIENTATION ERROR (MEAN ± STD) AT WAYPOINTS, BY NO. OF CAMERAS WITH ROBOT DETS. AND POSE ESTIMATION SOURCE.

Pose Estimation	1 Camera		2 Cameras		4 Cameras		**Average**	
Robot	20.6 ± 7.6 cm	1.17 ± 1.19°	17.1 ± 6.9 cm	1.03 ± 1.23°	25.0 ± 6.9 cm	1.30 ± 1.96°	19.1 ± 7.6 cm	1.11 ± 1.38°
Cameras (raw)	13.8 ± 10.1 cm	3.39 ± 2.87°	2.59 ± 1.49 cm	0.97 ± 0.79°	2.88 ± 1.42 cm	1.02 ± 0.76°	4.65 ± 6.22 cm	1.44 ± 1.72°
Cameras (1 frame)	8.23 ± 5.23 cm	1.40 ± 1.31°	2.59 ± 1.49 cm	0.97 ± 0.79°	2.88 ± 1.42 cm	1.02 ± 0.76°	3.65 ± 3.36 cm	1.06 ± 0.92°
Cameras (5 frames)	7.77 ± 5.47 cm	1.18 ± 1.27°	**2.58 ± 1.48 cm**	0.86 ± 0.68°	2.82 ± 1.43 cm	**0.97 ± 0.74°**	3.54 ± 3.31 cm	0.94 ± 0.84°
Fused	**4.25 ± 1.57 cm**	**1.12 ± 1.08°**	2.64 ± 1.48 cm	**0.79 ± 0.65°**	**2.73 ± 1.41 cm**	**0.97 ± 0.76°**	**2.93 ± 1.60 cm**	**0.88 ± 0.77°**

robot observations are consistent with the global model (cf. Fig. 1 (c)) to confirm a good initialization. The 2D grid map used by the robot for LiDAR navigation is initialized with the prior model of the empty room. The pose correction from external cameras is not sent to the robot during these experiments to measure the deviation over the full trajectory.

Fig. 5 shows an exemplary iteration of each trajectory, comparing the path estimated by the robot's internal navigation stack and the fused trajectory estimate, obtained using pose corrections from the external cameras and pose graph optimization, with the ground-truth obtained from the HTC reference tracking system. Furthermore, the evolution of translation error is shown over the traveled distance. While the robot's internal localization quickly accumulates errors of 20 – 30 cm, the error of the external camera pose estimation stays below 5 cm, when the robot is observed from at least two cameras. The error can be higher when observing the robot in only a single camera, e.g. WP 1 (1st WP of Traj 1, resp. 5th WP of Traj. 2), but always improves upon the robot's internal estimate. The single-camera observations can further be improved by fusing with robot odometry via the pose graph. In Fig. 5 (bottom-left), we illustrate the outlier detection for pose estimation from a single camera: The raw pose estimate shows an unrealistically high change in distance to the observing camera; therefore, the pose correction is restricted to the lateral direction.

TABLE II

ROOT MEAN SQUARE TRANSLATION ERROR IN CM.

Pose Estimation	Traj. 1	Traj. 2	Traj. 3	**Avg.**
Robot	20.5	19.4	15.8	18.5
Fused	**5.76**	**4.64**	**3.03**	**4.48**

Tab. I reports a quantitative evaluation of the translation and orientation errors at the waypoints, ordered by the number of cameras from which the robot is observed. The pose error is lowest when the robot is observed from two or four cameras and amounts to 2.93 cm and 0.88° averaged over all waypoints, significantly improving over the robot internal localization with an average error of 19.1 cm and 1.11°. The outlier detection for single-camera pose estimation significantly improves the accuracy in the resp. waypoints by 5.6 cm resp. 2.0° w.r.t. the raw estimate and prevents a worsening of the orientation estimate w.r.t. the robot's internal localization. Averaging the pose estimates of multiple frame-sets and fusion with the robot odometry via the pose graph give further improvements.

Tab. II shows the root mean square (RMS) of the translation error for the trajectories. The shorter Traj. 3, without WP 1 and WP 6 visible in only one camera, has the lowest trajectory error. The fused trajectory estimate using external camera pose estimation and robot odometry gives a significant improvement from 18.5 cm to 4.48 cm, averaged over the dataset.

In a second set of experiments, we compare the fused trajectory calculated on the central backend with the robot's internal estimate when integrating the pose correction feedback at static waypoints. For this, we record two iterations of a longer trajectory (∼40 m, 3-times Traj. 3), with and without applying the feedback on the robot. Fig. 6 shows the translation error over the traveled distance. The pose correction feedback significantly improves the robot's localization, reducing the error to the order of magnitude of the fused path calculated on the backend. Comparing the RMS translation error of the complete trajectories, the robot estimate without

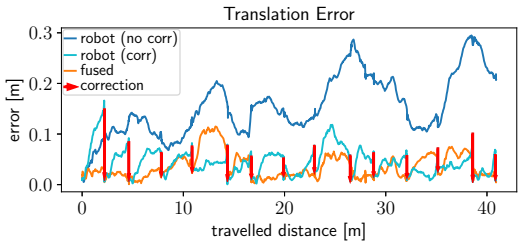

Fig. 6. Translation error with and without applying pose correction feedback to the robot's localization for the second experiment.

correction amounts to 17.5 cm and is improved significantly to 5.51 cm by the pose correction feedback. The pose graph fusion further improves the RMSE to 3.34 cm. To integrate the pose graph fusion on the robot, a closer coupling with the backend or a re-implementation of the robot navigation stack would be required.

C. Collaborative Semantic Mapping

In further experiments, we demonstrate the integration of the robot as a mobile sensor node into the smart edge sensor network. Fig. 7 shows the consistency of the robot observations with the allocentric scene model. Without pose correction feedback, error accumulates in the localization and the observations have low consistency with the model. After pose correction through the feedback from the smart edge sensors, the observations fit the model well and can consistently be fused into the semantic map.

We show the collaborative semantic mapping in a lab-scale experiment in Fig. 8. The semantic map is initialized from a prior model of the empty room, without any semantic information. The static smart edge sensors observe the room only partly, due to occlusions and limited measurement range, providing semantic information for ∼50 % of the voxels. The mobile robot provides changing sensor perspectives and can actively perceive the areas not observed by the static sensors. The waypoints for completing the map were defined manually in the experiment but could be set automatically using approaches for exploration and viewpoint optimization [21]. Through collaboration, mobile robot and static sensors build a complete semantic map of the ∼240 m^2 environment with semantic information for over 90 % of the voxels.

D. Localization Robustness

To evaluate the robustness of our collaborative localization approach, we repeat the lab-scale experiment ten times, five times with applying the pose correction feedback and five times without, and report the success rate in Tab. III. Using only the internal LiDAR-based localization, the robot cannot reach all waypoints in three of five trials, an emergency stop

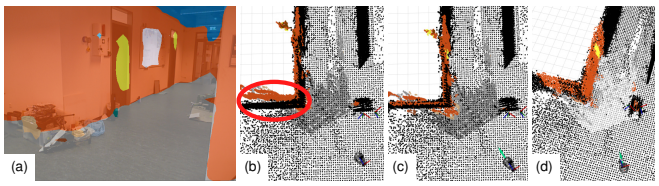

Fig. 7. Consistency of robot observations with global map: (a) local robot view; (b) 3D view with accumulated drift after Traj. 3 and (c) after pose correction (green arrow); (d) robot observations fused into map.

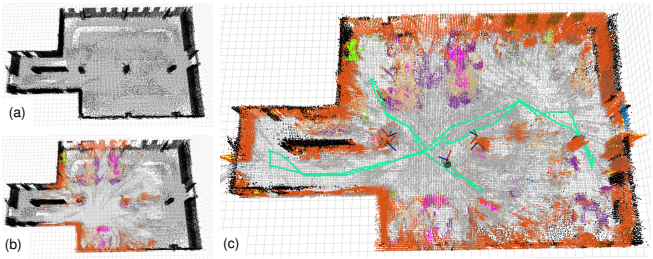

Fig. 8. Completion of semantic map by fusing with robot observations: (a) initial 3D map, (b) incomplete semantic map with observations from static smart edge sensors, (c) semantic map completed with robot observations and exploration path.

TABLE III

ROBUSTNESS DURING 5+5 LAB-SCALE EXPERIMENTS.

	WP not reached	Emergency Stop	Success Rate
w/o correction fb	3 / 5	1 / 5	1 / 5
w/ correction fb	**0 / 5**	**0 / 5**	**5 / 5**

due to collision with an obstacle occurred in one trial, and, hence, the success rate reaches only 1 / 5. Failures occur after 38 m of traveled distance, on average.

With the proposed localization feedback, the robot completes the ∼60 m long trajectory successfully in all trials. The external camera-based pose estimation compensates for the difficulties of the robot-internal navigation to localize in the highly cluttered, dynamic environment where only few distinctive features, such as walls or columns, are visible in the LiDAR, significantly increasing the system's robustness.

V. CONCLUSIONS

We presented a novel method for marker-less mobile robot pose estimation using multi-view keypoint detections from a network of external smart cameras. We use this to initialize and continuously update a mobile robot's localization in the allocentric scene model of the smart edge sensor network and build a system for collaborative perception between mobile robot and static smart sensors. The typical position error of our proposed method for mobile robot pose estimation is below 2.8 cm when detected in at least two cameras, and below 4.3 cm when detected in a single camera, while the robot localization typically deviates more than 19 cm after only 5 m of traveled distance.

Precisely initializing and tracking the robot's localization w.r.t the camera network allows to fuse its semantic observations in a globally consistent way into the allocentric scene model. The robot as a mobile sensor node provides changing viewpoints and can explore areas not covered by the static sensors due to occlusions and limited measurement range. We demonstrate a real-world application where a mobile robot and distributed smart edge sensors collaboratively build a 3D semantic map of a large room.

In future work, we plan to use our system for collaborative perception to enable anticipative human-aware navigation and human-robot interaction in a shared workspace.

ACKNOWLEDGEMENT

The authors would like to thank Jonas Bode and Julian Hau for their help with preparing the training data.

REFERENCES

[1] S. Bultmann and S. Behnke, "3D semantic scene perception using distributed smart edge sensors," in *Int. Conf. on Intelligent Autonomous Systems (IAS)*, 2022.

[2] M. Schwarz and S. Behnke, "Stillleben: Realistic scene synthesis for deep learning in robotics," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 10 502–10 508.

[3] M. W. Ashraf, W. Sultani, and M. Shah, "Dogfight: Detecting drones from drones videos," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7067–7076.

[4] D. Pizarro, M. Mazo, E. Santiso, and H. Hashimoto, "Mobile robot geometry initialization from single camera," in *Field and Service Robotics (FSR)*, 2008, pp. 93–102.

[5] D. Pizarro, M. Mazo, E. Santiso, M. Marron, D. Jimenez, S. Cobreces, and C. Losada, "Localization of mobile robots using odometry and an external vision sensor," *Sensors*, vol. 10, no. 4, pp. 3655–3680, 2010.

[6] D. Meger, D. Marinakis, I. Rekleitis, and G. Dudek, "Inferring a probability distribution function for the pose of a sensor network using a mobile robot," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009, pp. 756–762.

[7] A. Amini, H. Farazi, and S. Behnke, "Real-time pose estimation from images for multiple humanoid robots," in *RoboCup Int. Symposium*, 2022, pp. 91–102.

[8] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena, "X-view: Graph-based semantic multi-view localization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 3, pp. 1687–1694, 2018.

[9] F. Shkurti, W.-D. Chang, P. Henderson, M. J. Islam, J. C. G. Higuera, J. Li, T. Manderson, A. Xu, G. Dudek, and J. Sattar, "Underwater multi-robot convoying using visual tracking by detection," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017, pp. 4189–4196.

[10] B. Joshi, M. Modasshir, T. Manderson, H. Damron, M. Xanthidis, A. Q. Li, I. Rekleitis, and G. Dudek, "DeepURL: Deep pose estimation framework for underwater relative localization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 1777–1784.

[11] L. Strand, J. Honer, and A. Knoll, "Systematic error source analysis of a real-world multi-camera traffic surveillance system," in *25th International Conference on Information Fusion (FUSION)*, 2022.

[12] S. Severi, J. Härri, M. Ulmschneider, B. Denis, M. Bartels, *et al.*, "Beyond GNSS: Highly accurate localization for cooperative-intelligent transport systems," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2018.

[13] W. Wang, Y. Yan, L. Zhang, R. Hong, and N. Sebe, "Collaborative sparse coding for multiview action recognition," *IEEE MultiMedia*, vol. 23, no. 4, pp. 80–87, 2016.

[14] J. Lu, F. Richter, and M. C. Yip, "Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 4622–4629, 2022.

[15] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning," *Nature Neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018.

[16] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-robot pose estimation from a single image," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 9426–9432.

[17] J.-H. Shim and Y.-I. Cho, "A mobile robot localization using external surveillance cameras at indoor," *Procedia Computer Science*, vol. 56, pp. 502–507, 2015.

[18] P. Chakravarty and R. Jarvis, "External cameras and a mobile robot: A collaborative surveillance system," in *Australasian Conf. on Robotics and Automation (ACRA)*, 2009.

[19] S. Bultmann and S. Behnke, "Real-time multi-view 3D human pose estimation using semantic feedback to smart edge sensors," in *Robotics: Science and Systems (RSS)*, 2021.

[20] I. Rekleitis, R. Sim, G. Dudek, and E. Milios, "Collaborative exploration for the construction of visual maps," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 3, 2001, pp. 1269–1274.

[21] S. Dong, K. Xu, Q. Zhou, A. Tagliasacchi, S. Xin, M. Nießner, and B. Chen, "Multi-robot collaborative dense scene reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–16, 2019.

[22] Y. Yue, C. Zhao, Z. Wu, C. Yang, Y. Wang, and D. Wang, "Collaborative semantic understanding and mapping framework for autonomous systems," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 978–989, 2020.

[23] I. Ahmed, S. Din, G. Jeon, F. Piccialli, and G. Fortino, "Towards collaborative robotics in top view surveillance: A framework for multiple object tracking by detection using deep learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 7, pp. 1253–1270, 2021.

[24] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Robotics: Science and Systems XIV*, 2018.

[25] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3343–3352.

[26] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *International Journal of Computer Vision*, vol. 81, no. 2, p. 155, 2008.

[27] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2017.

[28] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4561–4570.

[29] M. Zappel, S. Bultmann, and S. Behnke, "6D object pose estimation using keypoints and part affinity fields," in *RoboCup Int. Symposium*, 2021, pp. 78–90.

[30] A. Amini, A. S. Periyasamy, and S. Behnke, "YOLOPose: Transformer-based multi-object 6D pose estimation using keypoint regression," in *Int. Conf. on Intelligent Autonomous Systems (IAS)*, 2022.

[31] B. Pätzold, S. Bultmann, and S. Behnke, "Online marker-free extrinsic camera calibration using person keypoint detections," in *44th DAGM German Conf. on Pattern Recognition (GCPR)*, 2022, pp. 300–316.

[32] Y. Xiong, H. Liu, S. Gupta, B. Akin, G. Bender, Y. Wang, P.-J. Kindermans, M. Tan, V. Singh, and B. Chen, "MobileDets: Searching for object detection architectures for mobile accelerators," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3825–3834.

[33] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Europ. Conf. on Computer Vision (ECCV)*, 2018, pp. 466–481.

[34] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 1314–1324.

[35] A. Boltres, A. Villar-Corrales, J. Nogga, and P. Schütt, "Sl-cutscenes," 2022. [Online]. Available: https://github.com/AIS-Bonn/sl-scenes

[36] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 3 2022. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[37] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends in Robotics (FNT)*, vol. 6, no. 1-2, pp. 1–139, 2017.

[38] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.

[39] P. Bauer, W. Lienhart, and S. Jost, "Accuracy investigation of the pose determination of a VR system," *Sensors*, vol. 21, no. 5, 2021.

[40] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580.