IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Real-Time Localization for Closed-Loop Control of Assistive Furniture

Lixuan Tang[1], Chuanfang Ning[1], George Adaimi[2], Auke Ijspeert[1], Alexandre Alahi[2], Anastasia Bolotnikova[1]

***Abstract*—For people with limited mobility, navigating in cluttered indoor environment is challenging. In this work, we propose a mobile assistive furniture suite that is designed to ease the life of people with special needs in indoor movement. To enable intelligent coordination of this system, a key component is the localization of each mobile furniture. The challenge is to assess the state of an arbitrary living scenario so that the estimation can be used as a real-time feedback signal for autonomous closed-loop control of mobile furniture. We propose a perception pipeline that addresses these challenges. A machine learning model is designed and trained to jointly achieve multi-object semantic keypoint detection and classification in camera images. The synthetic data generation is employed to augment the training set and boost the model performance. A robust point cloud registration uses the detected semantic keypoints and depth information to estimate poses of the furniture. Tracking is applied to achieve smooth estimation. A high-performance accelerator that optimizes the efficiency of using heterogeneous devices is applied to achieve real-time performance. This visual perception pipeline is used in closed-loop control to steer the mobile furniture from initial to a desired location demonstrated in experiments on real hardware.**

***Index Terms*—Vision-Based Navigation, Localization, Object Detection, Segmentation and Categorization**

I. Introduction

INSTEAD of people adapting to a static environment, the environment could be adaptive to assist its occupants. In this work, we explore the concept of a suite of mobile furniture that, in addition to being functional objects, can be mobile assistive agents: get out of the way, follow a person, autonomously take on a commanded configuration in the room. We develop a hardware solution of such a mobile furniture suite consisting of customized omni-directional robots and adaptive connection mechanism. Besides, to steer the behaviour of the mobile furniture suite, we propose a perception solution to estimate the state of the system in real time.

An inexpensive solution for localization and classification is a key component for intelligent coordination of a suite of mobile furniture in a smart home application. Since a multi-agent system for indoor applications is considered, onboard sensing and processing with high performance are significantly more costly than an external perception system. Thus, we aim to realize these functions with a camera set up at a fixed location in a room. We focus on the following challenges in order to make our assistive application more reliable: **(i) cluttered scenarios** where multiple objects are close to each other, **(ii) partial occlusion** and **(iii) real-time performance**.

We propose a multi-stage pipeline with functions of each component highly coupled. First, inspired by OpenPifPaf [1], a field-based framework which detects 2D keypoint poses without using bounding box, we present its extended formulation for simultaneous multi-object detection and classification. This stage aims to distinguish each 2D pose and its category from cluttered scenarios, meaning that even if multiple objects are close to each other with their bounding box regions overlapping, each 2D keypoint can still be assigned to the right object. Second, 3D coordinates of 2D keypoints are directly retrieved from the depth channel, which are further fed into a robust point cloud registration algorithm to recover 6D poses of all pieces of furniture. This step is able to tolerate missing keypoints or outliers caused by inevitable heavy occlusion. Finally, a Kalman filter based tracker is applied to smoothen estimates of the trajectories among consecutive frames.

A domain specific synthetic dataset is generated in full automation to augment deep neural network (DNN) training. It can tackle heavy (self-)occlusion problems and makes models usable in the real world. To accelerate the execution, the perception pipeline is deployed into our assistive infrastructure via a high-performance engine based on Nvidia TensorRT and C++ thread library, which splits the pipeline into sequential steps, and optimizes the efficiency of using heterogeneous devices (CPUs, GPUs, and I/O devices) through the parallelism.

This indoor localization pipeline is used to extract from the environment at any time the position, orientation, and speed of objects present, which serves as feedback signals for closed-loop control of the mobile assistive furniture. Fig. 1 shows the scheme of this assistive infrastructure with its localization pipeline. Our contributions are summarized as follows:

- Customized omni-directional robots and mechanical connector solution, which renders indoor furniture mobile;
- A perception system for real-time localization and classification in an arbitrary and cluttered environment;
- Evaluation of the capacity of our method to provide feedback signals in closed-loop control, which demonstrates that it is an efficient solution that balances well the real-time performance and the accurate localization.

Manuscript received: 1 March 2023; Revised: 10 April 2023; Accepted: 1 June 2023. This paper was recommended for publication by Associate Editor T.Asfour and Editor P.Vasseur upon evaluation of the reviewers' comments. This work is supported by the EPFL Center for Intelligent Systems (CIS). (Lixuan Tang and Chuanfang Ning are co-first authors.)

[1]Lixuan Tang, Chuanfang Ning, Auke Ijspeert and Anastasia Bolotnikova are with Biorobotics Laboratory, Swiss Federal Institute of Technology in Lausanne - EPFL, Switzerland (email: lixuan.tang@epfl.ch; chuanfang.ning@epfl.ch; auke.ijspeert@epfl.ch; anastasia.bolotnikova@epfl.ch)

[2]George Adaimi and Alexandre Alahi are with Visual Intelligence for Transportation Laboratory, Swiss Federal Institute of Technology in Lausanne - EPFL, Switzerland (email: george.adaimi@epfl.ch; alexandre.alahi@epfl.ch)

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

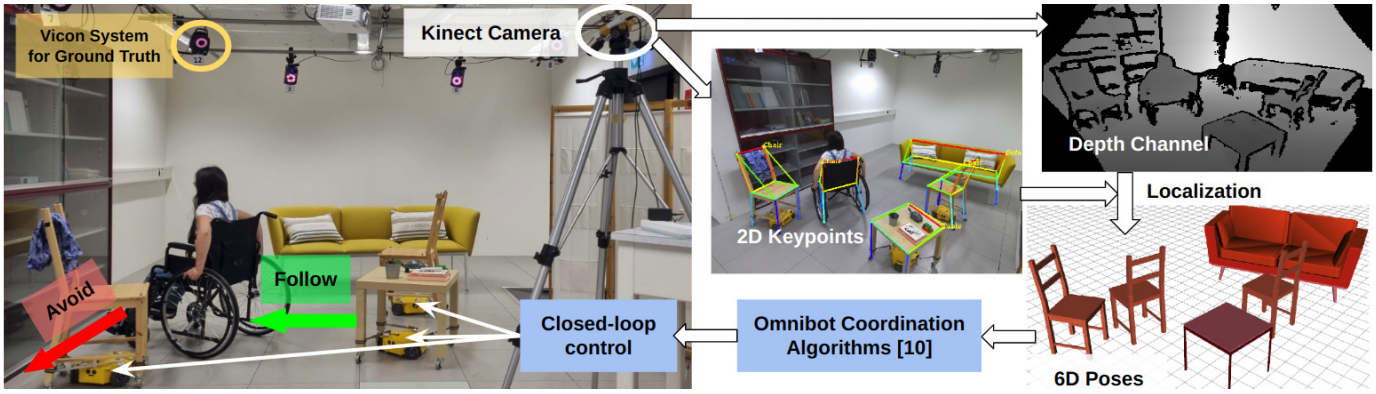

Fig. 1: The mobile furniture is intended to avoid collisions with a person in a wheelchair or to follow a person to help transport objects. To regulate the behaviour of the mobile furniture suite in a closed-loop control, a real-time perception system is proposed to track the state of an arbitrary living scenario that is challenging due to room clutter and object occlusions.

II. Related Work

1) Furniture Arrangement via External Robots: As an early study in 1995, researchers investigated how to organize furniture in a room with a team of robots that can push objects [2]. This work mainly focused on the capacity of several cooperative protocols achieved by the robot team, and complicated physical interaction between the robots and furniture has to be modeled. Besides, the manipulation performed by external agents is also easy to cause direct damage on the furniture. Therefore, we believe that imbuing furniture itself with robot properties is more suitable for the task of indoor furniture arrangement and coordination.

2) Rendering Furniture Mobile: ChairBot [3] is a mobile chair augmented with castors and a robotic vacuum cleaner. It also contains touch sensors, by which people can adjust its motions with small effort. Roombot [4] is a more advanced self-reconfigurable modular robot that can be attached on the furniture to adapt to more complex environments, such as moving steadily on the slope and overcoming obstacles.

3) Furniture Recognition and Localization: To coordinate multiple mobile furniture, a localization system must be introduced. Knight et al. [5] added a screen user interface to control ChairBots and trigger autonomous action. It requires an overhead camera that gives a top-down view, and relies on tracking markers attached on the top of ChairBots. Günther et al. proposed a method [6] that creates a semantic map of an indoor scene based on 3D point clouds. It generates hypotheses for locations of furniture using geometric feature, and verifies them by registering CAD models into the point cloud. Although this work is able to realize furniture localization basically, real-time performance is not achieved since it operates on the whole series of 3D point clouds over time.

With current progress in computer vision, instances described as keypoint skeletons can be extracted from the scenes in a data-driven manner. Recent progress has been made in processing crowded scenes, which is of great importance in applications such as detecting pedestrians for self-driving cars. AlphaPose with SPPE [7] is the latest top-down framework to detect multiple human poses, which first relies on a detector to obtain the bounding box of an object, then estimates joints within the given bounding box. In cluttered scenarios, top-down methods suffer from bounding boxes overlapping when objects stay close to each other. OpenPifPaf [8] [1] deals with this task in a bottom-up way. It directly estimates all the joints and groups them to form separated instances, naturally more suitable for cluttered scenes with inevitable occlusion.

4) Accelerating 2D Human Pose Estimation: At the system deployment stage, achieving both high accuracy and real-time performance is difficult, especially when customizing a certain algorithm for a real-world application. HyperPose [9] implements a high-performance execution engine for 2D human pose estimation task that can dynamically dispatch popular algorithms to heterogeneous devices, thus automatically achieving high utilisation of hardware resources irrespective of deployment environments.

III. Mobile Furniture Localization Pipeline

Our visual localization module[1](Fig. 1) consists of three stages. First, we extract the classified 2D keypoint poses from each frame (Sec. III-A). We augment the DNN training with auto-generated synthetic dataset to get higher accuracy of localization (Sec. III-B). Second, a robust point cloud registration uses the detected semantic keypoints and depth information to recover 6D poses (Sec. III-C). Third, we use Kalman filter to make the estimated trajectories smoother among frames (Sec. III-D). To achieve real-time performance we employ a computational accelerator (Sec. III-E). Our motivation for combining AI and analytical blocks, rather than using end-to-end AI, is in logical decoupling of the pipeline tasks to sub-tasks that can only be addressed robustly by data-driven AI (like semantic keypoint detection), and other sub-tasks that can be done with well understood analytical models, which allows for higher levels of framework explicability.

A. Multi-object 2D Pose Detection and Classification

At this stage, extended OpenPifPaf is applied to extract a set of classified 2D semantic keypoint skeletons for all the furniture detected in a cluttered scene, aiming to assign each

[1]https://ponyo.epfl.ch/proj/cis/furniture-localization

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

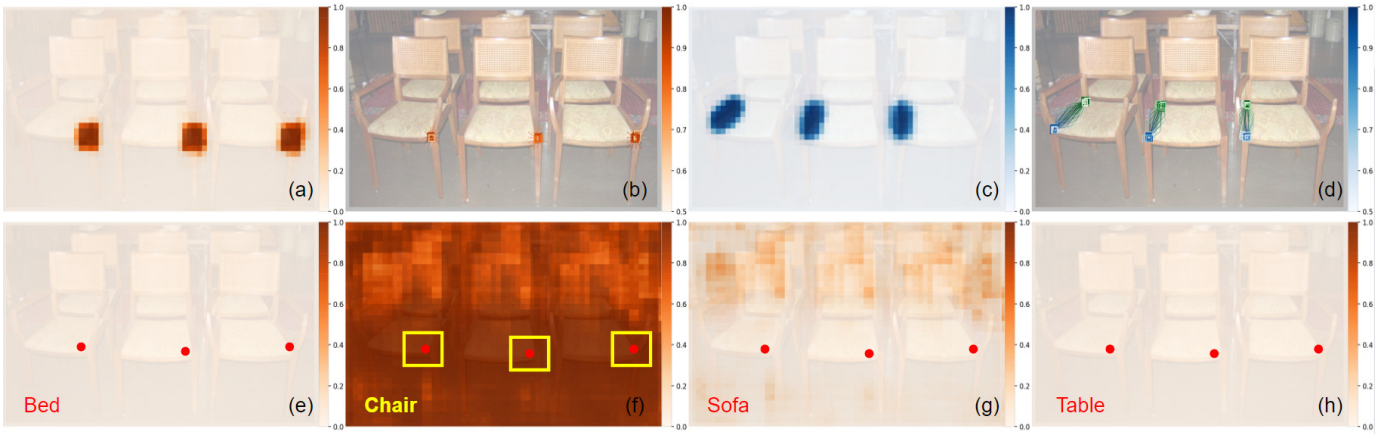

Fig. 2: Composite fields. (a): Confidence of the 6th keypoint. (b): Location regression of the 6th keypoint. (c): Confidence of association between the 5th and 7th keypoints. (d): Location regression of association between the 5th and 7th keypoints. (e)(f)(g)(h): Category channels of the 6th keypoint. Composite fields are visualized if the confidence value is higher than 50%.

keypoint to the right pose even if objects stay very close. The motivation for adding classification lies in the design of coordination algorithms [10], which is not within the scope of this paper. Different types of furniture have different priorities to be triggered to move. For example, tables carrying other things are not convenient to move than blank chairs. The neural network uses ShuffleNetV2K30 as the backbone. It also has a head network to encode keypoint intensities (Composite Intensity Field - CIF), and another for associations (Composite Association Field - CAF). Here we extend its head networks to encode additional category information for classification.

1) Composite Fields Extension: We define CIF at pixel (i, j) describing the furniture semantic keypoint type J as:

$$\mathbf{p}_J^{ij} = \{c, c_{\text{class}}, x, y, b, \sigma\}_J^{ij}, \tag{1}$$

c is the confidence, x and y are regressed real-valued coordinates, b is the uncertainty and σ is the size of the semantic keypoint. We introduce a new component c_{class} to encode N types of furniture using N channels one-hot coding. Similarly, the CAF at pixel (i, j) describing the association between two particular semantic keypoints J_1 and J_2 is defined as:

$$\mathbf{a}_{J_1 \leftrightarrow J_2}^{ij} = \{c, c_{\text{class}}, x_1, y_1, x_2, y_2, b_1, b_2, \sigma_1, \sigma_2\}_{J_1 \leftrightarrow J_2}^{ij}, \tag{2}$$

which contains a confidence, one-hot coding channels for classification, two vectors pointing to connected joints, two spreads for location uncertainty and two semantic keypoint sizes. Fig. 2 gives a visualization of these Composite Fields.

2) Loss Functions Extension: The CIF loss function is:

$$\mathcal{L}_{\text{CIF}} = \sum_{m_c} w(c, \hat{c})\, \text{BCE}(c, \hat{c}) + \sum_{m_v} \frac{1}{\hat{b}} L_2\left(v, \hat{v}\right) + \log \hat{b} \tag{3}$$

$$+ \sum_{m_\sigma} \frac{1}{b_\sigma} \left| 1 - \frac{\hat{s}}{s} \right| + \sum_{m_c} \text{CE}\left(c_{\text{class}}\,, \hat{c}_{\text{class}}\,\right), \tag{4}$$

We introduce cross entropy loss (CE) (4) to train classification c_{class}. Binary cross entropy (BCE) with a Focal loss modification w is used for confidence c. Laplace loss (L_2) attenuated by a predicted spread $\hat{b}$ is used to regress locations v. Laplace loss with a fixed spread b_σ is used to regress keypoint scale s. m_c, m_v and m_σ indicate the masking regions during training. CAF loss follows the same structure. After training, we get an encoder that turns the original image into composite fields.

3) Category Decoding: In order to regress real-valued location of each keypoint, the grid-based composite fields are turned to a high resolution confidence map via a convolution of an unnormalized Gaussian kernel $\mathcal{N}$ with width σ over the regressed targets, weighted by its confidence c:

$$f(v, w) = \sum_{ij} c^{ij} \mathcal{N}\left(v, w \mid x^{ij}, y^{ij}, \sigma^{ij}\right). \tag{5}$$

All real-valued coordinates in this high resolution map with the highest local value serve as seeds of keypoints. Here we present our category decoding algorithm that also follows the box-free and bottom-up manner. Since the category channels of composite fields are the output of the pixel-wise softmax function, shown as Fig. 2(e)(f)(g)(h), each keypoint fetches its category by comparing local value of seeds among all category channels. Then each keypoint votes for its category, with the majority voting of all involved keypoints being the final category of a decoded instance.

B. Training Augmentation Using Synthetic Dataset

Furniture datasets available in literature [11] [12] are not balanced in viewpoints. Labeled objects are often located in the center of images with front viewpoints. It is necessary to gather more samples whose viewpoints are fully captured. We implement automatic generation of synthetic data, which is accurate and productive. To help the synthetic dataset to approximate better to real-world scenes, two aspects can be diversified: **(i) Appearance:** Shown as Fig. 3, indoor layouts are designed in Blender. We diversify the appearance of objects by randomly sampling rendering parameters from corresponding domains, including base colors, material, roughness, texture and light conditions. The texture of each object is sampled from public texture datasets [13] [14]. **(ii) Viewpoint:** The motion of camera is control to vary viewpoints when taking pictures. To track each keypoint, 3D markers are inserted

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Fig. 3: Synthetic images generation. From the left to right, each time the camera takes a picture from a new viewpoint, each object will also sample new parameters of appearance.

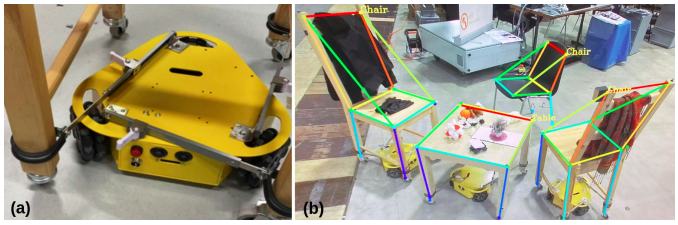

Fig. 4: Omni-directional robot and attachment configuration.

into the furniture. According to the camera projection model, 2D labels of each image can be computed by projecting 3D coordinates of markers onto the 2D image plane.

C. 6D Pose Estimation

2D keypoint skeletons serve as low-dimensional clues for detected objects. The depth channel is used to add 3^{rd} dimension to each keypoint. A benchmark keypoint set is prepared for each furniture category. The 6D pose of the furniture, which is the transformation from the benchmark keypoint set to the detected keypoint set, is calculated by the robust 3D point cloud registration. In this work, we use TEASER++ [15], a recent robust and certifiable registration algorithm, to recover the 6D pose of each detected object. In many cases the depth of some 2D keypoints is missing or wrong due to occlusions in the front. This method is applied to not only tolerate outliers or missing points, but also adapt benchmark keypoint sets to objects with different geometric parameters.

D. Tracking Management

Inspired by [16], this stage aims to smoothen trajectories among consecutive frames via tracking. Each valid 6D pose up to current frame $t-1$ is assigned with a tracker based on Kalman filter to maintain its state, including the position, orientation and their corresponding first order derivative, denoted as $S^i_{t-1} = (x, y, \theta, v_x, v_y, w_\theta)$. For the time update step of Kalman filter, we adopt the constant velocity model to predict the possible set of states $S_{\text{pre}} = \{S^i_{\text{pre}} | i = 1, 2...n_{t-1}\}$ in the new frame t consisting of n_{t-1} instances:

$$x_{\text{pre}} = x + v_x \Delta t,\ y_{\text{pre}} = y + v_y \Delta t,\ \theta_{pre} = \theta + w_\theta \Delta t \quad (6)$$

where Δt means the time interval between frames t and $t-1$. When dealing with the new frame t, the newest set of poses $D_t = \{D^i_t = (x, y, \theta) | i = 1, 2...n_t\}$ is recovered. To associate it with predicted poses S_{pre}, we treat it as a bipartite graph matching problem, which is solved by the greedy algorithm, meaning that a newly recovered pose D^i_t will match a predicted pose S^j_{pre} if their Euclidean distance is the shortest among the rest. In addition, we reject a pair of matching if their Euclidean distance is higher than a threshold, which can be approximated according to the maximal speed of robots. Finally, for each accepted pair of matching $(D^i_t, S^j_{\text{pre}})$, we treat D^i_t as a new measurement for S^j_{pre}, and fuse them to get its current state S^j_t via the measurement update step of Kalman filter. If a recovered pose D^i_t with high confidence is not matched, a new tracker is assigned to add it into memory. If a prediction S^j_{pre} is not matched for several frames, its tracker is deleted. This mechanism also helps to prevent 6D poses from disappearing if they are just temporarily occluded.

E. Localization Pipeline Accelerator

This pipeline is optimized from two aspects. First, the DNN model is converted into ONNX format, which is further optimized at execution by GPU inference operator based on Nvidia TensorRT. Furthermore, we implement a high-performance execution engine to accelerate the whole pipeline from input images to the 6D poses of localized objects. Inspired by the dataflow in HyperPose [9] that extracts 2D human pose from a single image, we split our localization pipeline (apart from tracking management) into four sequential steps: The input image is (i) pre-processed into expected layout; (ii) encoded into Composite Field by DNN via GPU inference operator; (iii) decoded into 2D skeletons; (iv) 6D poses recovery. The computation operator at each stage shares a concurrent FIFO channel, scheduled by the C++ standard thread library. This accelerator maximise the efficiency of utilizing CPUs, GPUs, and I/O devices through pipeline parallelism.

IV. Assistive Infrastructure Prototype

Shown as Fig. 1, this prototype consists of a static Kinect camera, a host PC, and a suite of mobile assistive furniture driven by omni-directional robots (Omnibot). The communication among the host PC and robots is realized by the Robot Operating System. Ground truth in real-world experiments is provided by Vicon motion capture system. We develop the pre-built Nexus omni-directional drive platform to render furniture mobile according to patients' needs, which is easy to replicate and configure for future extensions in assistive environment. We adopt a Tripous attachment configuration (Fig. 4) that connects the Omnibot to furniture in a secure and interchangeable way. Tripous reaches out its arms to grab legs of the furniture and drive them as the main body of robot moves. Three arm connections ensure that the furniture is fully constrained and will not slip when driven by the robot.

A. Electronics Extension

Omnibot is equipped with Arduino Mega extended by a custom Printed Circuit Board. To enable teleoperation with higher level language in ROS structure, the basic functionality of sensors and actuators has been implemented on-board. The embedded program includes interfaces for: **(i) Bluetooth:** Omnibot is teleoperated via the serial communication over

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

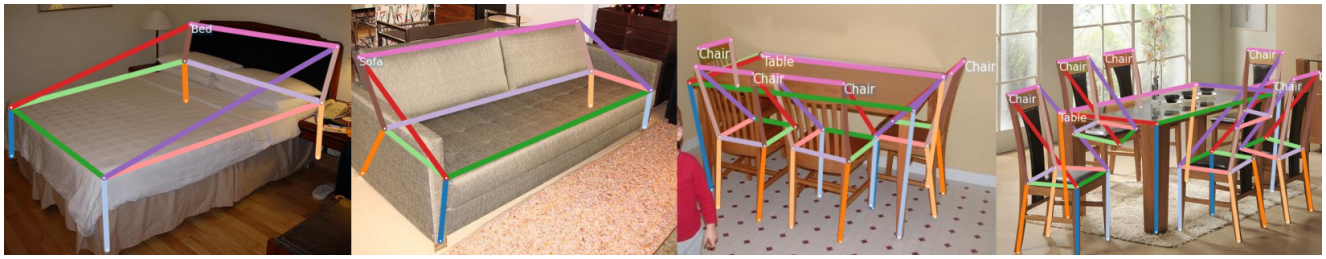

Fig. 5: Predicted results on the Keypoint-5 (left two images) and Pascal3D+ (right two images) datasets.

Method	Class	PCP	AE	$AP^{0.5:0.95}$	$AP^{0.5}$	$AP^{0.75}$	$AR^{0.5:0.95}$	$AR^{0.5}$	$AR^{0.75}$	Precision	Recall
3DINN [11]	Bed	77.4	1.16	-	-	-	-	-	-	-	-
	Chair	87.7	0.92	-	-	-	-	-	-	-	-
	Sofa	77.4	1.14	-	-	-	-	-	-	-	-
	Swivel Chair	78.5	1.19	-	-	-	-	-	-	-	-
	Weighted Average	80.8	1.08	-	-	-	-	-	-	-	-
Original OpenPifPaf (no classification)	Bed	76.5	1.15	77.9	96.6	87.3	82.2	97.3	89.9	-	-
	Chair	75.3	1.15	86.8	92.8	88.9	88.8	93.1	90.8	-	-
	Sofa	78.3	1.09	81.9	95.0	89.6	84.5	95.3	90.5	-	-
	Swivel Chair	65.7	1.18	71.1	82.1	74.1	74.4	83.5	77.2	-	-
	Weighted Average	74.7	1.14	80.6	92.3	86.0	83.5	92.9	88.0	-	-
Extended OpenPifPaf (with classification)	Bed	82.4	1.02	81.3	97.4	91.9	86.1	99.0	94.9	97.6	95.9
	Chair	77.4	1.09	88.9	95.7	90.5	91.1	96.8	92.9	98.0	91.7
	Sofa	83.1	0.93	85.6	96.0	92.7	88.0	96.3	93.8	99.7	95.0
	Swivel Chair	77.3	0.97	79.6	88.1	82.5	82.6	89.4	85.4	97.4	89.8
	Weighted Average	80.3	1.01	84.7	94.6	90.2	87.6	95.7	92.2	98.3	93.2
Extended OpenPifPaf (with classification) + Synthetic Dataset Augmentation	Bed	82.6	1.00	82.5	97.5	93.5	87.3	99.0	95.6	99.0	96.3
	Chair	80.7	1.05	91.9	96.8	93.9	93.8	97.9	95.4	97.8	92.6
	Sofa	85.7	0.92	86.8	96.9	94.6	89.5	97.8	95.5	99.0	95.8
	Swivel Chair	81.0	0.95	84.2	94.2	87.3	87.6	95.3	90.9	97.9	93.3
	Weighted Average	**82.7**	**0.98**	**86.9**	**96.6**	**92.6**	**90.0**	**97.6**	**94.7**	**98.4**	**94.4**

TABLE I: Evaluation on the Keypoint-5 dataset using a V100 GPU

Bluetooth JY-MCU at 115200 baud rate. **(ii) Motors:** Omnibot is driven by a PID speed control loop with 3 DC Coreless motors and encoder feedbacks; **(iii) LED:** 48 ARGB-LEDs pasted at the bottom of Omnibot indicate the status and direction of motion; **(iv) Sonars:** The pre-built platform comes with 3 sonars for distance sensing. The Mega board communicates with sonars using message templates via RS-485 interface.

B. Interactive User Interface

People with limited mobility may suffer from other issues, such as above-elbow amputee or extremities paralysis by the spine cord injury, which impacts on their interaction abilities as well [17]. Omnibot is equipped with four types of interactive control to help people with different needs, including: **(i) Automatic Control:** Program interface is designed for carer of people for configuration purposes; **(ii) Voice Control:** It relies on DeepSpeech [18], an speech-to-text engine based on recurrent neural network which ingests speech spectrograms and generates text transcriptions; **(iii) Gesture Control:** It relies on MediaPipe [19] framework, which translates the detected Hand Landmark Model into robot commands; **(iv) Tablet Control:** Omnibot can be controlled with a tablet application on Android in the interactive mode.

V. Experimental Validation

Evaluation is performed at each stage of the pipeline. First, we compare the performance of our method with baselines in terms of extracting 2D keypoints on two public datasets and our proposed synthetic dataset, with the latter focusing on cluttered scenarios. Classification is also evaluated at this stage. Second, this pipeline is integrated into our indoor assistive infrastructure, and we evaluate its ability to track 6D poses. Third, we evaluate the accuracy of closed-loop control of the assistive furniture, which relies on visual feedback signals. In addition, the execution speed is also evaluated.

A. Setups and Metrics

1) 2D Keypoint Pose Detection and Classification: **(i) Object Keypoint Similarity Score (OKS):** Each keypoint is assigned with a bounding box according to the area of the furniture object. Then the overlaps between ground truth and predicted bounding boxes are computed to get the standard detection metrics average precision (AP) and average recall (AR); **(ii) Percentage of Correct Parts (PCP):** Defined as the percentage of keypoints localized within 1.5 times of the standard deviation of annotations; **(iii) Average Error (AE):** Defined as the mean distance between a predicted keypoint and the ground truth location, bounded by 5; **(iv) Percentage of Correct Keypoints (PCK):** A keypoint is correct if its L2 pixel distance from the ground truth location is less than $0.1 \times \max(h, w)$, where h and w are the object's bounding box dimensions. Note that, OKS is applied to experiments on all the related furniture datasets. And we use the rest metrics to compare with baseline methods on their corresponding datasets respectively. **(v) Precision and Recall for Multi-class Classification:** We conduct experiments on Keypoint-5

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Method	Class	PCK	$AP^{0.5:0.95}$	$AP^{0.5}$	$AP^{0.75}$	$AR^{0.5:0.95}$	$AR^{0.5}$	$AR^{0.75}$
StarMap [20]	(Swivel) Chair	**91.7**	-	-	-	-	-	-
	Sofa	89.7	-	-	-	-	-	-
	Table	90.0	-	-	-	-	-	-
Original OpenPifPaf (no classification)	(Swivel) Chair	91.4	80.5	92.7	84.5	82.9	93.9	86.9
	Sofa	88.7	72.3	91.7	81.7	74.9	92.6	83.6
	Table	89.8	69.5	85.2	76.0	79.4	93.5	85.1
Extended OpenPifPaf (with classification)	(Swivel) Chair	92.3	83.6	95.4	86.8	85.7	96.0	88.7
	Sofa	91.2	75.9	93.7	84.1	78.8	94.7	85.2
	Table	88.3	67.7	84.4	75.8	78.4	92.9	85.1
Extended OpenPifPaf (with classification) + Synthetic Augmentation	(Swivel) Chair	**95.2**	87.3	97.7	91.0	89.4	98.5	92.0
	Sofa	**93.5**	79.4	93.6	87.6	82.4	94.7	88.9
	Table	**90.0**	71.1	87.0	76.0	81.1	94.0	85.1

TABLE II: Evaluation on the Pascal3D+ dataset using a V100 GPU

Metrics	AlphaPose with SPPE [7]	Extended OpenPifPaf
$AP^{0.5:0.95}$	40.0	**68.6**
$AP^{0.5}$	45.5	**75.0**
$AP^{0.75}$	40.4	**70.6**
$AR^{0.5:0.95}$	48.1	**69.9**
$AR^{0.5}$	53.5	**75.6**
$AR^{0.75}$	48.5	**71.5**

TABLE III: Evaluation on the synthetic furniture dataset using a V100 GPU. In [7], SPPE, used for crowded scenes, has been integrated into AlphaPose. Keypoint skeletons of both methods are changed from the original human pose to the furniture pose

dataset because it can provide only one category of object in each image, which is suitable for classification task.

2) 6D Pose Localization and Tracking: We arrange a living room scene using the table, sofa and chairs. We use the tablet user interface to control motion of a mobile chair for 4 minutes, during which occlusions occur along its way. The ground truth pose is given by Vicon motion capture system. We compare at any time the position and orientation of the localized chair with its ground truth. This experiment is repeated for 10 times within the same setups, among which the **Average Absolute Trajectory Error** and **Maximum Absolute Trajectory Error** are calculated. We assume the robot is never turned over in normal state, so only the X and Y axes are evaluated for position, and yaw for the orientation.

3) Closed-loop Control: In this setup, the automatic control mode of robots is turned on, with goals of both mobile chairs manually specified, and published by two separate ROS topics. Each chair subscribing to its topic is driven to the goal in full automation, where feedback signal is provided by our perception pipeline. As for control approach, each robot moves at a fixed translational velocity to the target position until translation error falls within $0.03\,\mathrm{m}$, then moves at a fixed angular velocity to the target direction until rotational error falls within 0.05 radian. The velocity is corrected at 15Hz based on the error between visual feedback and the target goal. We record errors between goals and current states of two mobile chairs over time. This experiment is repeated for 10 times with the goals of two chairs kept the same. Although the staring points of two mobile chairs and the arrangement of the rest furniture vary among different runs, we make sure no collision occurs. The overall **rate of successful reaching** to the goals and **Steady-State Errors** are reported.

4) Execution Speed: We record the time consumed at each stage III-E of the localization pipeline, and average the overall frame per second (FPS) among 10 runs of execution.

B. Results

1) Evaluation on Keypoint-5 and Pascal3D+ Dataset: Table I and Table II show results on Keypoint-5 and Pascal3D+ datasets. First, a multi-task manner that equips original OpenPifPaf with classification improves the accuracy on its original task (keypoint detection), which helps it to achieve comparable performance to baselines. Second, by introducing synthetic data into training, further improvement on both detection and classification can be noticed. These trends can be revealed by all the metrics in this experiment. Fig. 5 shows predictions on the two datasets. Synthetic dataset augmentation can supplement more viewpoints of furniture and cluttered scenarios, which are out of the distribution of Keypoint-5 and Pascal3D+ datasets. Here we mention that models trained only on these two datasets are insufficient in real use. This experiment can also quantify the benefit of synthetic dataset augmentation, which makes models usable in real world.

2) Evaluation on Synthetic Furniture Dataset: Table III shows the quantitative result on the synthetic furniture dataset. When facing cluttered indoor environment, extended OpenPifPaf reports a higher OKS score, compared with AlphaPose which relies on bounding boxes. Fig. 6 presents the qualitative result, showing that OpenPifPaf is better at extracting skeletons even under a certain degree of occlusion.

3) Evaluation of 6D Pose Localization and Tracking: Fig. 7 visualizes one run of localizing a mobile chair with its ground truth. Our prediction matches the ground truth with high accuracy. Fig. 8 shows three samples of the qualitative result where the tracked chair is partially occluded. In Fig. 8(a)(b), the bounding box region of the chair overlaps with those of others. Our bottom-up framework can distinguish them in this cluttered scenario. In spite of missing keypoints (the bottom of the chair) and some wrong depth value (the occluded seat of chair), the robust registration can tolerate outliers and recover right 6D poses. In Fig. 8(c), although the 2D keypoint skeleton is lost, Kalman filter keeps the tracker in memory until it is detected again. The Average Absolute Trajectory Errors on

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

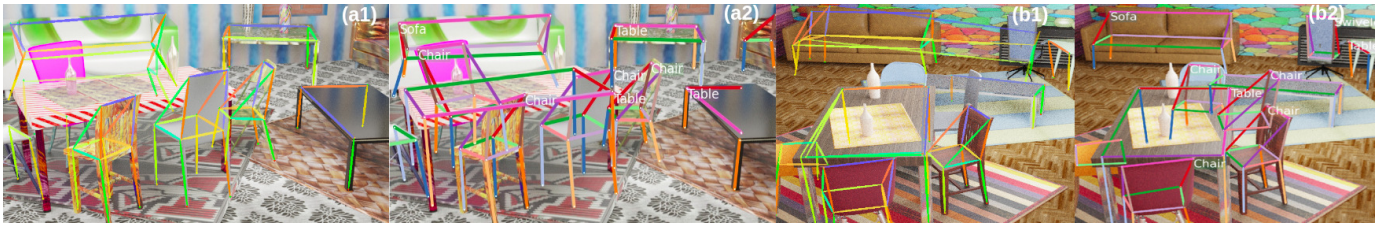

Fig. 6: Inference on synthetic images. (a1)(b1) are predicted by AlphaPose, and (a2)(b2) ares inferred by extended OpenPifPaf.

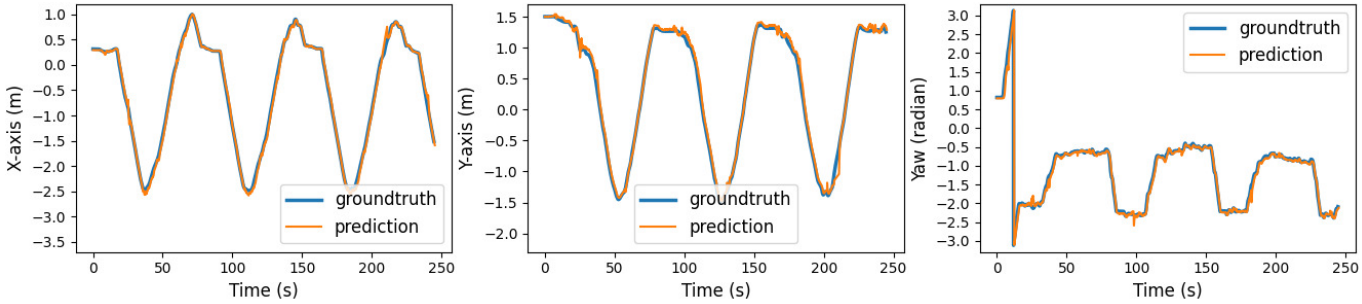

Fig. 7: The pose tracked by our method (orange), and the ground truth from Vicon (blue). Yaw is between $-\pi$ and π.

Fig. 8: Examples of predicted 2D keypoint skeletons and tracked 6D poses in presence of occlusions in a cluttered living room.

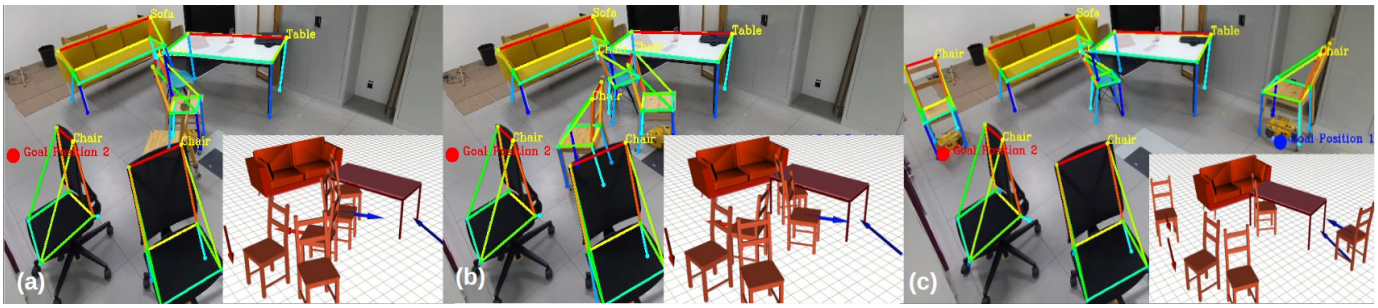

Fig. 9: Closed-loop control of two mobile chairs using visual feedback signal to reach the goals. See associated video for full demonstration.

X, Y axis and Yaw orientation among 10 runs are $0.042\,\mathrm{m}$, $0.054\,\mathrm{m}$ and 3.55° respectively. The corresponding Maximum Absolute Trajectory Errors are $0.221\,\mathrm{m}$, $0.259\,\mathrm{m}$ and 22.27° respectively, which show its behavior in the worst cases caused by heavy occlusions in the single camera setup.

4) Evaluation of Closed-Loop Control: Fig. 10 shows how errors decrease in one run of closed-loop control. Fig. 9 presents three moments within this run. Two chairs were initialized at the left and right ends of the field of view, with goals set on their opposite ends. In Fig. 9(a), heavy occlusion occurs as four chairs form a straight line in the camera view. Although extended OpenPifPaf lost detection of one mobile chair and one static chair, Kalman filter maintained 6D poses in memory via state prediction, thus prevents them from disappearing. Error of state prediction accumulates as the time of losing detection increases. Then two mobile chairs end overlapping, with missing skeletons detected again (Fig. 9(b)). This new detection helps trackers to correct the state, shown as the pink region in Fig. 10(a). Finally, two chairs reach their goals successfully (Fig. 9(c)). The overall rate of successful

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Execution Platform	Resized Input	Pre-process	Inference (GPU)	Decode	Pose Recovery (ms)	FPS
Pytorch	640 x 480	15.8	240.6	4.32	0.63	3.82
	432 x 368	14.7	149.6	2.35	0.63	5.98
Accelerator (C++/TensorRT)	640 x 480	1.31	39.5	7.20	0.31	20.7
	432 x 368	1.31	19.6	5.21	0.31	37.4

TABLE IV: Execution speed at each step of the pipeline on different platforms. The input resolution is 1920 x 1080

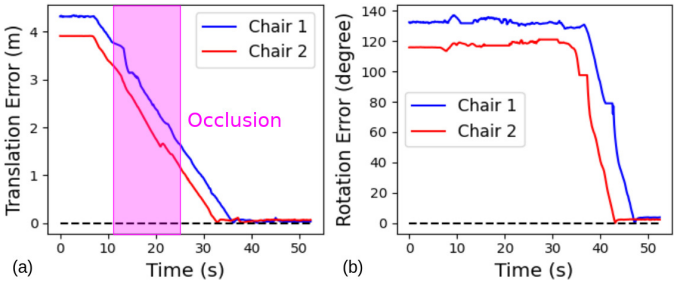

Fig. 10: The tracking error evolution during closed-loop control.

reaching to goals is 90% among 10 runs. One trial failed because of far distance from the camera, making the detection of 2D skeletons unstable. The overall Steady-State Errors on transition and rotation are $0.033\,\mathrm{m}$ and $3.12°$ respectively.

5) Evaluation of Execution Speed: Shown as Table IV, all the data is averaged among 10 times of execution. It is noticeable that GPU inference takes up majority of the total time budget, which is significantly reduced by the inference operator based on TensorRT. The time spent on almost all steps is reduced via the pipeline parallelism, except for decoding stage. The reason is that the decoder of original system has already been optimized by C++ extension of Pytorch. The resizing scale also has a great impact on the speed. Here we report that resizing the original input into 640 x 480 or 432 x 368 can achieve good balance between the accuracy of localization and real-time performance.

VI. Conclusion And Future Work

This work proposes a concept of an intelligent assistive infrastructure, aiming to ease the life of people with limited mobility. We establish a localization pipeline, and demonstrate that it is efficient for the closed-loop control of assistive furniture, which balances well the real-time performance and high accuracy in cluttered scenarios with partial occlusions. Here we also mention that although the single-camera configuration is able to tolerate full occlusions for short duration (around 3 seconds) where one object is completely invisible for a camera, we believe that increasing the number of sensors, and exploring redundancy configuration for multi-view solutions in the future work are more practical to address full occlusions. Coordination algorithms will also be integrated to avoid collision and achieve assistive actions in higher levels.

References

[1] S. Kreiss, L. Bertoni, and A. Alahi, "OpenPifPaf: Composite fields for semantic keypoint detection and spatio-temporal association," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13498–13511, 2021.

[2] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 235–242, 1995.

[3] H. Knight, T. Lee, B. Hallawell, and W. Ju, "I get it already! the influence of chairbot motion gestures on bystander response," in *IEEE International Symposium on Robot and Human Interactive Communication*, pp. 443–448, 2017.

[4] S. Hauser, M. Mutlu, P.-A. Léziart, H. Khodr, A. Bernardino, and A. Ijspeert, "Roombots extended: Challenges in the next generation of self-reconfigurable modular robots and their application in adaptive and assistive furniture," *Robotics and Autonomous Systems*, vol. 127, p. 103467, 2020.

[5] A. Fallatah, B. Stoddard, M. Burnett, and H. Knight, "Towards user-centric robot furniture arrangement," in *IEEE International Conference on Robot Human Interactive Communication*, pp. 1066–1073, 2021.

[6] M. Günther, T. Wiemann, S. Albrecht, and J. Hertzberg, "Model-based furniture recognition for building semantic object maps," *Artificial Intelligence*, vol. 247, pp. 336–351, 2017.

[7] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, "CrowdPose: Efficient crowded scenes pose estimation and a new benchmark," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10863–10872, 2019.

[8] S. Kreiss, L. Bertoni, and A. Alahi, "PifPaf: Composite fields for human pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11977–11986, 2019.

[9] Y. Guo, J. Liu, G. Li, L. Mai, and H. Dong, "Fast and flexible human pose estimation with HyperPose," in *ACM International Conference on Multimedia*, 2021.

[10] F. M. Conzelmann, L. Huber, D. Paez-Granados, A. Bolotnikova, A. Ijspeert, and A. Billard, "A dynamical system approach to decentralized collision-free autonomous coordination of a mobile assistive furniture swarm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7259–7265, 2022.

[11] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman, "3D interpreter networks for viewer-centered wireframe modeling," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 1009–1026, 2018.

[12] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3D object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 75–82, 2014.

[13] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi, "Describing textures in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[14] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material recognition in the wild with the materials in context database," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3479–3487, 2015.

[15] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.

[16] X. Weng, J. Wang, D. Held, and K. Kitani, "3D Multi-Object Tracking: A Baseline and New Evaluation Metrics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[17] I. Moon, M. Lee, J. Ryu, and M. Mun, "Intelligent robotic wheelchair with emg-, gesture-, and voice-based interfaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, 2003.

[18] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, pp. 173–182, 2016.

[19] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, *et al.*, "MediaPipe: A Framework for Building Perception Pipelines ," *arXiv preprint arXiv:1906.08172*, 2019.

[20] X. Zhou, A. Karpur, L. Luo, and Q. Huang, "StarMap for category-agnostic keypoint and viewpoint estimation," in *European Conference on Computer Vision*, pp. 318–334, 2018.