Door Opening Strategy for Mobile Manipulator with Constrained Configuration

Daniil Pushkarev[1], Konstantin Mironov[1,2](✉), Ilya Basharov[1], Margarita Kichik[1], Sergey Linok[1], Dmitry Yudin[1,2], Muhammad Alhaddad[1], and Aleksandr Panov[2,3]

[1] Moscow Institute of Physics and Technology, Institutskiy Per. 9, 141701 Dolgoprudny, Moscow Region, Russia
[2] AIRI Artificial Intelligence Research Institute, Kutuzovskii Pr, 32/1, 121165 Moscow, Russia
[3] Federal Research Center "Computer Sciences and Control", Vavilova 44/2, 119333 Moscow, Russia

Abstract. We address the task of robotic door opening in office environments. This task is important for providing indoor mobility for collaborative mobile manipulators. In our work, we mainly focus on the use of high-level control opportunities and identification of the door parameters from visual and lidar data. We develop a solution, which includes handle recognition, handle twisting, and opening. The position of the handle is identified from stereo images by a neural-network-based method. We divide the opening procedure into two stages: first, handle twisting and slightly opening, and second, wide opening. The first stage is implemented via high-level task-space control of the robotic arm, while the platform is static. The position of the door axis is identified during the slight opening by fitting lidar data to the kinematic model. At the second stage, both the platform and the arm are active. The trajectory of the platform is defined by the model predictive planner in such a way that it avoids pushing the arm into a singular configuration, while the manipulator is operated via high-level impedance control. In our experiments, a mobile manipulator composed from the wheeled platform and the robotic arm was able to open office doors using the proposed approach.

Keywords: Mobile manipulator · Model predictive control · Interactive manipulation · Object recognition

1 Introduction

The development of mobile manipulators in recent years has faced a number of new robotic tasks and open issues. The ability to open standard doors is useful for the mobility in human-intended environments [1]. It is also theoretically interesting as it requires interaction with a large-size movable mechanical object (comparable to or larger than the size of the robot).

There is a number of works on robotic door opening; however, it still may be considered a relevant scientific task. Door opening may be performed by various types of

A. Ronzhin et al. (Eds.): ICR 2022, LNCS 13719, pp. 130–142, 2022.
https://doi.org/10.1007/978-3-031-23609-9_12

robotic systems e.g. manipulator on a wheeled platform [2–5] or humanoid robot [6, 7]. Control approaches for door opening can be divided into two groups: real-time compliance control methods [2, 8, 9] and planning the trajectory of a robotic system for door opening [3, 4, 6, 7].

The works in the first group consider the following task statement. The endeffector of the robot interacts with the door handle; resistance of the door is identified by the force and torque sensors. The task is to define control inputs based on these sensor data, which helps to move the door in the required direction without damage. This task requires fast communication with sensors and high performance of feedback processing. Communication and control frequency have an order of tens or hundreds Hz. The time of door opening is much longer then for opening by human (e.g., 20 s with 100 Hz communication in [9] and 35 s with 20 Hz in [8]).

The works in the second group aim to determine the trajectory (the reference of robot configurations), which allow the system to open the door. Derivation of control inputs is out of this task statement and can be executed by inner regulators of the robotic system. Obviously, two approaches may be combined, e.g., in [4] compliant controller is used to execute the trajectory, which is defined by the planner. Trajectory planning can be based on a heuristic search or on numerical optimization. The first approach is used in [3] where anytime repairing A* algorithm from [10] is applied to define the executable robot trajectory on a configuration graph. In [4], the sampling-based RRT algorithm was applied for motion planning. [6] claims that the sampling-based search is both overpowered (as the configuration space of door opening is not yet complicated for applying these techniques) and underpowered (as it requires post-processing to make the trajectory smoother) for door opening. Through these statements, the authors substantiate their approach based on the CHOMP [11] trajectory optimization. A bit later, [7] applied for this the TrajOpt [12] method, which is faster than CHOMP. The aforementioned works applied to a robotic system with complicated dynamics and required relatively long time either for trajectory definition or for motion execution and consider path and motion planning as two separated tasks. It seems promising to apply a holistic approach based on the nonlinear model predictive control (MPC), which provides trajectory optimization and a definition of related control inputs as a single operation.

We present a novel approach for wide door opening using a mobile manipulator. Trajectory planning is made for a wheeled platform while the control of the arm is defined to adjust the trajectory of the end-effector to the door opening trajectory under impedance of the door handle. This statement is a bit similar to [3], but contrary to this work, we, first, develop our own planner based on trajectory optimization and, second, apply high-level force control of the arm. The use of high-level force control simplifies the opening task; however, it is challenging as the arm must avoid singular configurations. We insert singularity avoidance into objectives and constraints for the optimization task. The wide opening of the door is predated by the door positioning and unlocking. Unlocking is implemented via simple high-level control rules, while the positioning of the door is based on machine vision. We have developed a complex neural solution for handle positioning from stereo camera images and estimate the door kinematic model based on lidar data. We have implemented and tested our approach on a mobile manipulator, which is shown in Fig. 1. It consists of the differential drive Husky platform [13] and

UR5 [14] the robotic arm with two-fingers gripper and stereo camera mounted on its end-effector. Compared with purely indoor mobile manipulators (e.g. TIAGo [15]) Husky has a relatively big footprint, which makes collision avoidance challenging and reduces effective workspace of the arm. The rest of the paper is organised as follows. In Sect. 2, we present our common view on the opening pipeline. The following three sections focused on the three main subtasks of the solution: recognition of the door handle (3), handle twisting and slightly opening with simultaneous identification of door kinematics (4), and wide opening (5). In Sect. 6, we present the experimental results, while Sect. 7 includes the concluding remarks.

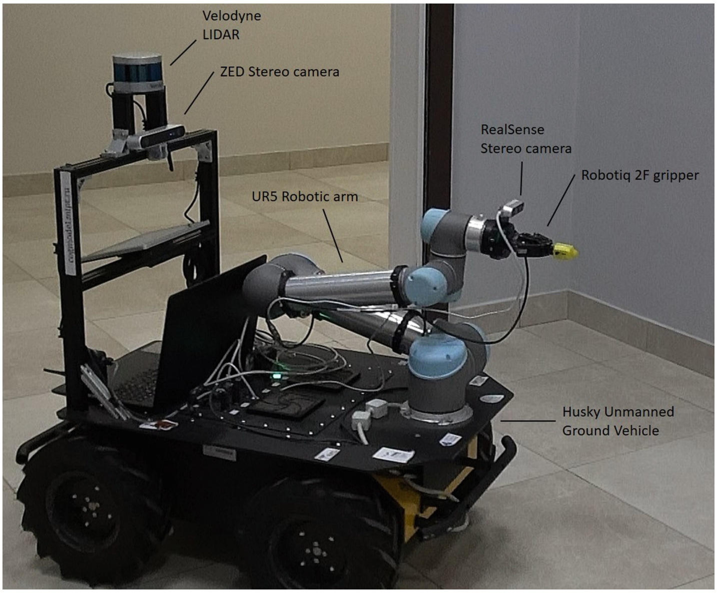

Fig. 1. Experimental mobile manipulator composed of a four-wheels platform and a robotic arm.

2 Common Architecture of the Solution

Following the STRL architecture for cognitive agents [16–18], we divide the components of the solution into strategic, tactical, and reactive layers. A strategic layer corresponds to high-level cognitive tasks, the tactical one corresponds to local perception and planning for executing high-level action, while the reactive one includes low-level data processing and control. We consider door opening as a single task, which is stated on the strategic layer and executed on the tactic and reactive layers. Common architecture is shown in Fig. 2. We divide the opening procedure into three stages:

1. Positioning of the door and wheelbase before opening. The spatial position of the door handle is identified from stereo images (Sect. 3; the handle recognition component is shown in Fig. 2).

2. Handle twisting and slightly opening via the manipulator (Sect. 4). The manipulator unlocks the handle and slightly opens the door (the handle twisting and slightly opening component is shown in Fig. 2). In the meantime, the kinematic model of the door is identified based on LIDAR data (the door state recognition component is shown in Fig. 2).
3. Door opening via the whole-body motion. The wheelbase and the manipulator are acting in order to open the door wide (Sect. 5).

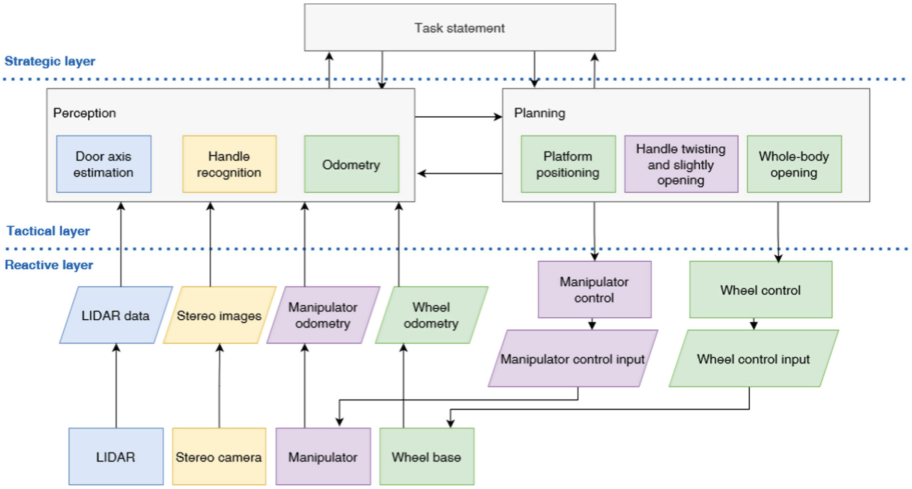

Fig. 2. Architecture of the solution components.

3 Recognition and Positioning of the Door Handle

This section describes in detail the method for determining the door handle position, as well as the collection and annotation of the necessary data for 2D detection and keypoint regression tasks. The dataset includes the markup of:

1. door handle localization data via 2D bounding boxes b_i,
2. each b_i contains inside six 2D keypoints $k_{ij}, j \in \{1, \ldots, 6\}$.

So, the dataset helps us solve two tasks simultaneously: 2D object detection and keypoint regression. To determine the position in 3D space, the door handle can be represented as a simplified skeleton model consisting of six keypoints.

Using the symmetry of an object, 1–2-3 points are assigned as «start» and 4–5-6 as the “end” of a handle. Points 2–3 and 4–5 are responsible for bending along the main axis 3–4 (Fig. 3). Photos of door handles are taken from different angles and scales. The dataset does not include information about keypoint visibility.

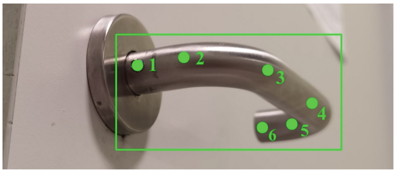

Fig. 3. Keypoints location overview on the created dataset.

Table 1. Distribution of the Train Test Split.

	Images	Boxes	Keypoints
Train	212	258	1548
Test	52	67	402
Total	264	325	1950

As a result of manual marking up, 264 images were collected, which contain 325 2D bounding boxes in total. The dataset was divided according to Table 1.

The model for obtaining 3D door handle coordinates consists of three main blocks: detection, regression of 3D coordinates, and projection modules (see Fig. 4). We will describe each of them in more detail (Table 2).

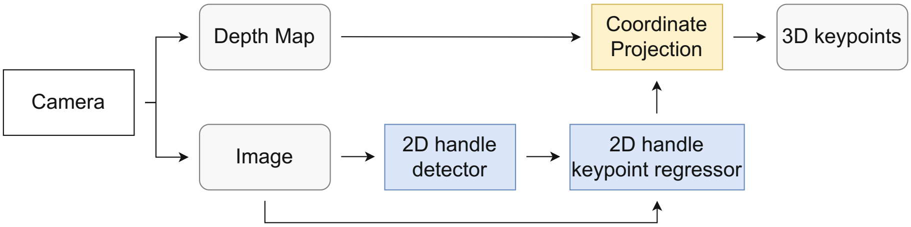

Fig. 4. Data pipeline to get 3D door handle keypoints.

Table 2. Methods evaluation.

Steps/Tasks	Detection	Regression	
Method	YOLOX-s [19]	HRNet [21]	HRNet-Lite [20]
Metric[a]	mAP: 0.82	PCK: 0.38	PCK: 0.31
FPS[b]	54	32	51

[a] Door handle test set
[b]NVIDIA TeslaV100

YoloX [19] was taken as a detection module. This high-performance anchorfree method tiny version works 23 FPS on Jetson Xavier NX with TensorRT optimization. Additionally, it has a good speed-accuracy trade-off. It has been trained with original instructions from authors with 50 epochs on TeslaV100 GPU. HRNetLite [20] was taken as a keypoint regressor module. This is an updated light-weight version of HRNet [21]. Thanks to the multi-scale architecture, the model enables one to cope well with the problems of rotating and obscuring keypoints. Using the MMPose framework [22], it has been trained in the 256 × 192 image scale with 20 epochs. To compensate the lack of data and make the keypoint regressor more robust, different augmentation techniques were applied.

1. Detection module. To make our method resistant to color changes, such as shadows, light environment, etc., we use RandomHSV. Also, since the manipulator needs to observe the handle from different sides, we apply image transformations as RandomFlip and RandomAffine. The latter includes RandomShear, RandomTranslation, RandomRotation. To make inference more robust, we apply MixUp and Mosaic augmentations.
2. Regression module. To accurately determine the skeleton of the handle, the following transformations were selected: RandomFlip and RandomAffine.

   Using a depth map and an intrinsic matrix, we can obtain 3D coordinates:

$$\begin{bmatrix} u \\ v \\ 1 \\ 1/z \end{bmatrix} = \frac{1}{z} \underbrace{\begin{bmatrix} K & 0 \\ 0 & 1 \end{bmatrix}}_{4\times4} \underbrace{\begin{bmatrix} R & t \\ 0 & 0 \end{bmatrix}}_{4\times4} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

where rotation matrix **R**, translation vector **t**, and the intrinsic matrix **K** make up the camera projection matrix. *u, v* is a camera coordinate system, and *z*, *y*, *z* is a world coordinate system.

4 A Strategy for the Slight Opening

Geometric appearance for the slightly opening procedure is shown at Fig. 5. At the beginning the only known information is the position of the handle center H in the coordinate system related to the robot base (it is got from the recognition component). This information is enough to perform three consecutive actions: first, capture the handle by the gripper at point H, second, twist the handle via compliant rotation of the gripper, and, third, slightly open the door by pulling the handle. Wide opening of the door require knowledge about the position of the door axis O and rotation radius r. This task is equivalent to estimating the transition between base-related and door-related coordinate systems. Estimation of the door axis is made from LIDAR data while slightly opening.

Some modern robotic arms, including UR5, are able to be controlled by forces in Cartesian task space (if joint configuration is not singular). Our common philosophy for operating a mobile manipulator is to use this ability as much as we can. This mode called “force mode” could be used for soft operations avoiding destruction of the objects and

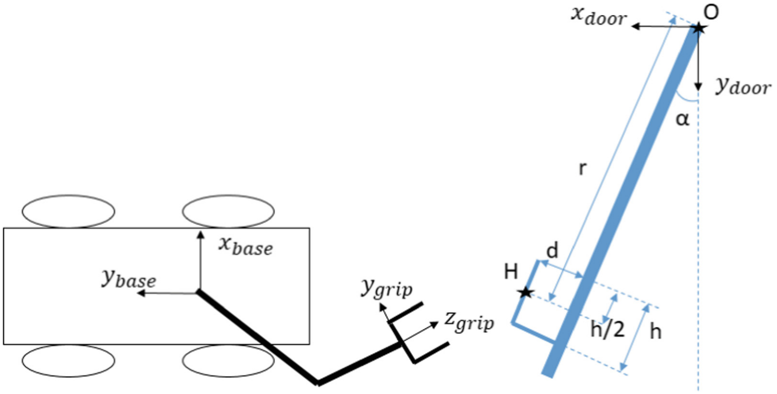

Fig. 5. Relative positions of the robot and the door.

dropping the control box in the safety block. It was used for gripping and twisting the handle. For the handle twist stage, forces were determined in the gripper frame: down pressure and small rotation force around z_{grip}. After that, the manipulator could be able to slightly open the door by pulling the handle toward the negative z_{grip}. At the final stage, the manipulator returns the handle to the original orientation, for which forces were given as reversed forces at the twist stage and iteratively checking the alignment of the yz_{grip} plane and xy_{base} plane.

The position of the door plane is obtained by solving the plane-popout task on the LIDAR pointcloud. A subset of the points near the handle are taken from the pointcloud. RANSAC is applied to fit the plane to these points. As a result, we get the parameters of the plane equation in base-related 3D coordinates. As we consider the plane to be vertical, this equation is reduced to the straight line equation in base-related planar coordinates. During the slight opening, several such estimations are made resulting in a set of line equations. By assuming that all these lines intersect in O, we can define a set of linear equations with unknown coordinates of O. Each equation looks as follows:

$$\cos \alpha_i x_{O(base)} + \sin \alpha_i y_{O(base)} = p_i \tag{2}$$

Here α_i and p_i are the door plane parameters for the single lidar measurement; specifically, α_i is the door opening angle with respect to robot base. If we have more than two such equations, we can estimate O by the least squares method.

5 Planning for the Whole-Body Opening Motion

The geometric appearance for the wide opening task is shown in Fig. 6. If we want to exploit the Cartesian force control of the arm, we should define the platform trajectory in such a way that it avoids the pushing arm into a singular configuration. The non-singular configuration requires that the position of the end-effector be within the certain limits. The end-effector must be within the black-dashed circle in Fig. 6. The center of the circle is at distance b forward to the robot base. If the base joint of the arm is rotated with a certain angle, then the center of the non-singular area is also rotated with this angle. The radius of non-singular area Δr_{max} is equal to approximately 30 cm for UR5 robotic arm. Another aspect of singularity avoidance is that the difference between the azimuth of the end-effector and the azimuth of the base joint must not exceed certain value φ (for UR5 $\varphi \approx 45°$). The singularity avoidance work as force that push end-effector from danger zone, strength determined from experiments. The opening trajectory of the wheeled platform should be defined in such a way that the point, which is b forwards to the robot base, is within the corridor between the two red-dash arcs. In this case, the end-effector pulling the door handle is within a non-singular area. Also, the angle between b-line and the door normal must not exceed φ. This angular constraint is harder to satisfy, but its violation is less critical. If the angular constraint is violated, but other constraints are satisfied, the arm may be switched to a fully compliant mode, which has no singularity restrictions. As the platform is moving through the planned path within the corridor, a compliant mode will allow the arm to open the door.

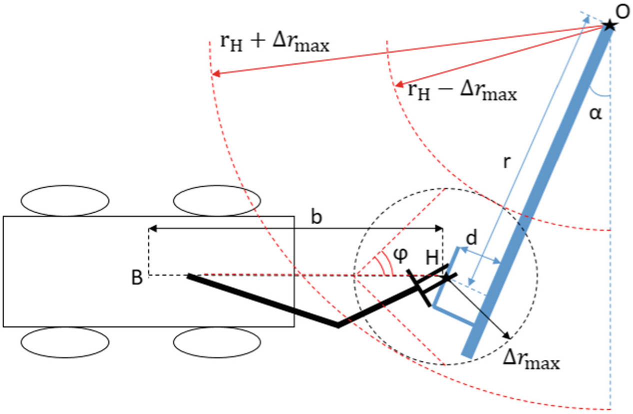

Fig. 6. Geometric singularity-avoidance constraints for wide-opening.

The motion planning for a wheeled platform is achieved by turning the planning task into a model-predictive control statement, which represents the optimization problem. A numerical solution is achieved with the "Acados" toolkit [21]. Let $s = \left[x, y, v, \theta\right]^T$ and $u = [a, w]^T$ be the state and the input vectors of the system. Here x, y and θ are the coordinates and orientation of the geometric center of the robot in the inertial frame of the door, v and a are the linear velocity and acceleration of the robot, and ω is the angular velocity of the robot. The kinematic model of the differential-drive platform is the following:

$$\begin{aligned} \{\dot{x}\} &= v cos\theta, \\ \{\dot{y}\} &= v sin\theta, \\ \{\dot{v}\} &= a, \\ \left\{\dot{\theta}\right\} &= \omega. \end{aligned} \tag{3}$$

The optimization problem for the model (3) has a cost function, which should be minimized in the presence of constraints on s and u. For the planning of the motion of the mobile robot, it is suggested that the position of the base link of the manipulator should be as near as possible to the arc-path of the door handle; therefore, it is necessary to introduce the coordinates of the base link in the inertial door frame:

$$\begin{aligned} x_b &= -b\cos(\theta) + x, \\ y_b &= -b\sin(\theta) + y. \end{aligned} \tag{4}$$

The position of the base link is given as:

$$z = \sqrt{x_b^2 + y_b^2} \tag{5}$$

The cost function of the studied problem consists of three terms: square deviations of the parameters of the state vector from the desired values, square values of the input vector from zeros, and the position of the base link from the arc-path of the door handle. The general cost function of the problem is written as:

$$J = \int_0^T (s - s_d)^T W_1 (s - s_d) + +(u - u_d)^T W_2 (u - u_d) + (z - z_d)^T W_3 (z - z_d) d\tau, \tag{6}$$

where $W_1 \in \mathbb{R}^{n \times n}$, $W_2 \in \mathbb{R}^{m \times m}$ and $W_3 \in \mathbb{R}$ are the weight matrices of the optimization problem, s_d, u_d are reference values (the initial guess) of the state and input vector respectively, and z_d equals the radius of the arc-path of the door handle. In another formulation of the optimization problem, the desired position of the base link of the manipulator can be added by a term-constrained minimum and maximum distance from the arc-path. In this case, the cost function (6) contains only the first two terms. The initial guess for the optimization problem can be constructed as a straight line from the position of the robot in front of the closed door to the end position behind the opened door with angle $\pi/2$, so the end position of the initial guess is fixed with respect to the door frame. The change in scenarios of the initial guess depends on the initial position and orientation of the robot only. Figure 7 shows the initial and optimized path of the center point of the robot, in case that the initial posture of the robot is (-0.8, 0.8, -1.57) and the end posture is (0.8, 0.9, $-\pi$) where the handle is located at position (0.8, 0) for the closed door and at (0, 0.8) for the opened door.

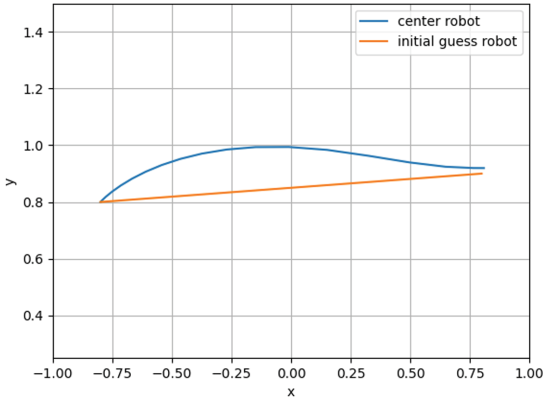

Fig. 7. Initial guess and optimized trajectory.

6 Experiments

The experimental evaluation was made in two stages. At the first stage, we validated the accuracy of identifying door and handle. At the second stage, the common pipeline was tested on a real mobile manipulator.

To obtain groundtruth data about the handle position and the door kinematics, we applied the ARUCO markers recognition module from OpenCV [24]. This module returns pixel coordinates of the marker's angles, which are then converted into 3D according to (1). Their 3D positions are used to define the door normal and transform between ARUCO-related, camera-related and robot-related coordinate systems. Distance between the marker and the center point of the door handle is measured manually. Thanks to augmentations the 2D keypoints, the regressor works in the most difficult situations. Using a depth map, the method can transform coordinates in 3D (Fig. 8). TensorRT implementation increases the processing speed, which enables one to work in real time. The comparison of 17 cases of positioning with the Aruco data showed that the standard deviation in positioning is equal to 9 mm. To validate the method of the door axis estimation, a set of experiments were conducted. We manually measured the angle of the door in several positions and recorded lidar scans to apply our algorithm. The results of the example procedure of axis estimation are given Table 3. As a result, we calculate the mean door angle error equal 1.2° and max door angle error equal 2.9°. The LS estimation of the door axis based on lidar measurements provides accuracy improving up to 1–2 cm.

At the second stage of the experiments, the real robot was able to open the office door with an 0.8-m radius. The unlocking and slight opening stage took around 15 s, the wide opening for 90° – around 8 s. See the supplementary video [25] for the details.

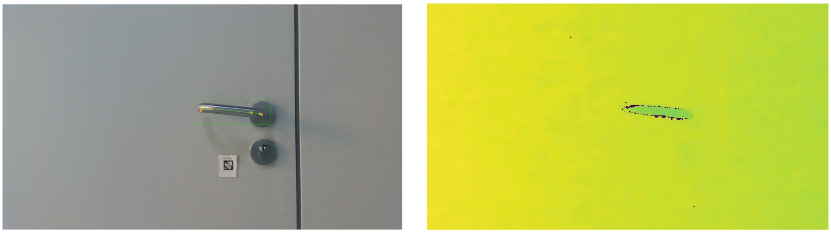

Fig. 8. Network output and depth map.

Table 3. Door angle errors, °.

Lidar angle, °	0.4	2.3	5.5	8	11.3	14.8	20	22.8
GT angle, °	0	2.9	5.8	8.7	11.5	14.5	17.5	20.5
Angle error, °	0.4	0.6	0.3	0.7	0.2	0.3	2.5	2.3
Axis error, cm	–	42	10	10	10	9	9	9
Lidar angle, °	25.6	29.7	32.1	35	38	41.2	44.5	47.5
GT angle, °	23.5	26.8	30	33.4	36.9	40.5	44.5	48.6
Angle error, °	2.1	2.9	2.1	1.6	1.1	0.7	0	0.9
Axis error, cm	8	7	6	4	3	3	1	1

7 Conclusion

In this work, we have developed a solution for robotic door opening based on LIDAR and stereo camera measurements. The opening motion of a mobile manipulator is determined by a combination of compliant control for the robotic arm in the task space and model predictive trajectory planning for the wheeled platform. Numerical experiments showed that the proposed methods for positioning the door handle and the axis satisfy accuracy requirements of a real robotic system. Our approach was implemented on a real mobile manipulator and enabled it to open standard doors in an office environment. These results may help to provide the mobility of mobile manipulators in office environments.

Acknowledgements. This work was supported by the Ministry of Science and Higher Education of the Russian Federation under Project 075–15-2020–799.

References

1. Mironov, K., Mambetov, R., Panov, A., Pushkarev, D.: Model predictive control with torque constraints for velocity-driven robotic manipulator. In: 2021 20th International Conference on Advanced Robotics (ICAR), pp. 107–112. IEEE (2021). https://doi.org/10.1109/ICAR53236.2021.9659428

2. Chung, W., Rhee, C., Shim, Y., Lee, H., Park, S.: Door-opening control of a service robot using the multifingered robot hand. IEEE Trans. Indus. Electron. **56**(10), 3975–3984 (2009). https://doi.org/10.1109/TIE.2009.2025296
3. Chitta, S., Cohen, B., Likhachev, M.: Planning for autonomous door opening with a mobile manipulator. In: 2010 IEEE International Conference on Robotics and Automation, pp. 1799–1806. IEEE (2010). https://doi.org/10.1109/ROBOT.2010.5509475
4. Arduengo, M., Torras, C., Sentis, L.: Robust and adaptive door operation with a mobile robot. Intel. Serv. Robot. **14**(3), 409–425 (2021). https://doi.org/10.1007/s11370-021-00366-7
5. Zuo, W., Venkatraman, R., Song, G., Chen, Z.: A novel design of mobile robotic system for opening and transitioning through a watertight ship door. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1378–1383. IEEE (2012). https://doi.org/10.1109/IROS51168.2021.9635942
6. Zucker, M., Jun, Y., Killen, B., Kim, T.G., Oh, P.: Continuous trajectory optimization for autonomous humanoid door opening. In: 2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA), pp. 1–5. IEEE (2013). https://doi.org/10.1109/TePRA.2013.6556358
7. Banerjee, N., Long, X., Du, R., Polido, F., Feng, S., Atkeson, C.G., Gennert, M., Padir, T.: Human-supervised control of the ATLAS humanoid robot for traversing doors. In: 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pp. 722–729. IEEE (2015). https://doi.org/10.1109/HUMANOIDS.2015.7363442
8. Ding, L., Xia, K., Gao, H., Liu, G., Deng, Z.: Robust adaptive control of door opening by a mobile rescue manipulator based on unknown-force-related constraints estimation. Robotica **36**(1), 119–140 (2018). https://doi.org/10.1017/S0263574717000200
9. Stuede, M., Nuelle, K., Tappe, S., Ortmaier, T.: Door opening and traversal with an industrial cartesian impedance controlled mobile robot. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 966–972. IEEE (2019). 10.1109/ ICRA.2019.8793866
10. Likhachev, M., Gordon, G.J., Thrun, S.: ARA*: Anytime A* with provable bounds on sub-optimality. Adv. Neural Inf. Process. Syst. **16** (2003)
11. Ratliff, N., Zucker, M., Bagnell, J.A., Srinivasa, S.: CHOMP: Gradient optimization techniques for efficient motion planning. In: 2009 IEEE International Conference on Robotics and Automation, pp. 489–494. IEEE (2009). https://doi.org/10.1109/ROBOT.2009.5152817
12. Schulman, J., et al.: Motion planning with sequential convex optimization and convex collision checking. Int. J. Robot. Res. **33**(9), 1251–1270 (2014)
13. Husky, U.G.V: Outdoor field research robot by Clearpath. https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/
14. The UR5: A flexible collaborative robot arm, https://www.universal-robots.com/products/ur5-robot/
15. PAL Robotics, http://pal-robotics.com/robots/tiago/
16. Multilayer cognitive architecture for UAV control: Emel'yanov, S., Makarov, D., Panov, A.I., Yakovlev, K. Cogn. Syst. Res. **39**, 58–72 (2016). https://doi.org/10.1016/j.cogsys.2015.12.008
17. Panov, A.I.: Goal setting and behavior planning for cognitive agents. Sci. Tech. Inf. Process. **46**(6), 404–415 (2019). https://doi.org/10.3103/S0147688219060066
18. Panov, A.I.: Simultaneous learning and planning in a hierarchical control system for a cognitive agent. Autom. Remote Control **83**(6), 869–883 (2022). https://doi.org/10.1134/S0005117922060054
19. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv:2107.08430 (2021)
20. Yu, C., Xiao, B., Gao, C., Yuan, L., Zhang, L., Sang, N., Wang, J.: Lite-hrnet: A lightweight high-resolution network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10440–10450 (2021)

21. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5693–5703. (2019)
22. MMCV Contributors. MMCV: OpenMMLab Computer Vision Foundation. https://github.com/open-mmlab/mmcv
23. Verschueren, R., et al.: ACADOS—A modular open-source framework for fast embedded optimal control. Math. Prog. Comput. **14**(1), 147–183 (2022)
24. ArUco Marker Detection, https://docs.opencv.org/3.4/d9/d6d/tutorial_table_of_content_aruco.html
25. Autonomous door opening by mobile manipulator, https://youtu.be/tuRkFwhmRws