Visual control through narrow passages for an omnidirectional wheeled robot

Damiano Morra[1], Enric Cervera[2], Luca Rosario Buonocore[3], Jonathan Cacace[1],
Fabio Ruggiero[1], Vincenzo Lippiello[1], and Mario Di Castro[3]

***Abstract*— Robotic systems are gradually replacing human intervention in dangerous facilities to improve human safety and prevent risky situations. In this domain, our work addresses the problem of autonomous crossing narrow passages in a semi-structured (*i.e.*, partially-known) environment. In particular, we focus on the CERN's Super Proton Synchrotron particle accelerator, where a mobile robot platform is equipped with a lightweight arm to perform measurements, inspection, and maintenance operations. The proposed approach leverages an image-based visual servoing strategy that exploits computer vision to detect and track known geometries defining narrow passage gates. The effectiveness of the proposed approach has been demonstrated in a realistic mock-up.**

I. Introduction

Nowadays, mobile robots are extensively used to assist human workers operating in hazardous environments [1]–[3]. For example, mobile robots have been used for rescue missions in coal mines [4], whose locations are hazardous for humans because non-breathable air and the risk of flooding. Domains in which robots can play a crucial role include explosive devices disposal [5], decontamination of nuclear sites, and handling contaminated waste [6]. At the European Organization for Nuclear Research (CERN), within the world's largest particle accelerator plant, CERNbots [7] are used for various tasks, such as screwing, welding, and inspecting in areas where radiations, oxygen deficiency, and electrical hazards can be a threat [8]. During such operations, robots have to cross small-sized security doors.

In unstructured (*i.e.*, completely unknown) or semi-structured (*i.e.*, partially known) scenarios like those mentioned above, narrow passages (*e.g.*, doors, tunnels, aisles, etc.) represent a big challenge. The narrow passages are identified in [9] using a laser scanner sensor considering neighbor line segments. Other techniques consider that the robot has already detected the gate. In [10], a low-cost method based on infrared and sonar sensors used to detect the angle to which the robot is facing the passage is presented. The robot is correctly oriented using a fuzzy logic controller. A similar approach focused on narrow aisles is presented in [11]. In the solutions mentioned so far, expensive and bulky sensors are often mandatory. A solution based on an RGB-D camera is instead presented in [12]. The depth channel is used as a laser scan, and passages are searched as distance discontinuities of surroundings.

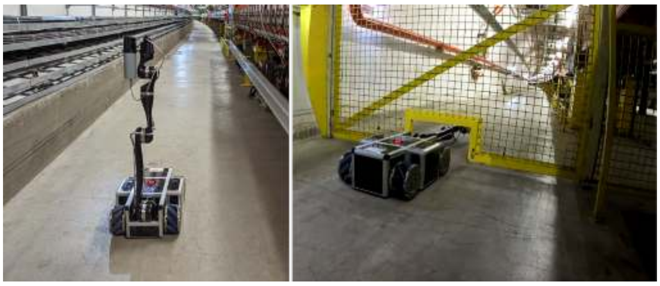

Fig. 1. The MIRA SPS Robot during inspection (left) and the MIRA Robot crossing one of the sector doors (right).

In setups where only monocular camera sensors are available, visual servoing solutions are often adopted. Many solutions rely on the use of artificial landmarks. These are not suitable in unstructured or semi-structured environments where it is mandatory to explore computer vision-like solutions. For instance, computer vision is used in [13] to find rectangular shapes of standard house doors. Other works use optical flow to discern obstacles from the ground plane [14]. For instance, this information is used in [15] to control the robot via a potential field technique, eventually achieving the traversal of a corridor. Optical flow is also exploited in [16] to avoid additional obstacles in the pathway. Lastly, given that a proper data-set of the path is available, machine learning has been employed for semi-structured environments [17].

At the Super Proton Synchrotron (SPS) particle accelerator, inspection and maintenance tasks are carried out by the MIRA SPS Robot, an omnidirectional mobile manipulator [18]. When the particle beam is on, each tunnel sector is kept isolated for security reasons. The robot can move through different sectors thanks to small-sized passages placed at the bottom of the doors (see Fig. 1). The accelerator, whose circumference is 6.9 km, is divided into 19 sectors by the same number of doors. Teleoperation requires, for a skilled operator, an average time of 2 minutes per gate. This means that, for a complete tour of the accelerator, 30 to 60 minutes are used for crossing sector doors. Besides time consumption, this task impacts battery duration and

The research leading to these results has been supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008.

[1]The authors are with the PRISMA Lab, Department of Engineering and Information Technology and CREATE Consortium, University of Naples Federico II, Via Claudio 21, Naples, 80125, Italy {dam.morra}@studenti.unina.it, {jonathan.cacace, fabio.ruggiero, vincenzo.lippiello}@unina.it.

[2]E. Cervera is with the Robotic Intelligence Lab, Department of Computer Science and Engineering, Universitat Jaume-I, Castelló, Spain, ecervera@uji.es.

[3]L.R. Buonocore and M. Di Castro are with European Organization for Nuclear Research, CERN, Route de Meyrin 385, 1211 Geneva, Switzerland, {luca.rosario.buonocore,mario.di.castro}@cern.ch.

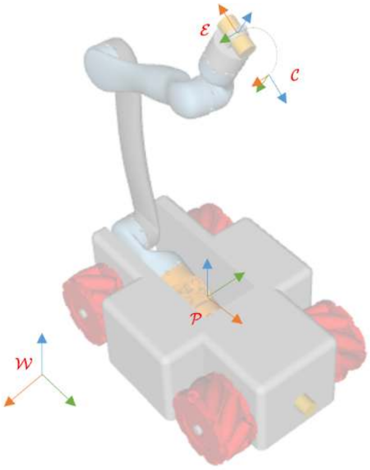

Fig. 2. MIRA SPS robot and employed frames.

can stress the operator. The MIRA SPS Robot was designed based on the gate's size so that there are just 2 cm per side of free space left while crossing. Due to its compact dimensions, it is equipped with monocular RGB cameras only.

The above-described scenario at CERN fits this paper's aim, proposing a visual servoing solution for the autonomous crossing of narrow doors for an omnidirectional platform, set in a semi-structured environment in which site alterations are strictly prohibited. A-priori geometry information about the doors is known and exploited to identify the corners with a computer vision algorithm. Features are tracked to reduce errors and improve robustness. Subsequently, the robot is aligned to the door using an image-based visual servoing (IBVS) approach [19], [20], and it is driven through the other side of the gate via feed-forward control.

II. Image based visual servoing

A. Background

With reference to Fig. 2, let $\mathcal{P} : O_{\mathcal{P}} - (x_{\mathcal{P}}, y_{\mathcal{P}}, z_{\mathcal{P}})$ be the platform frame attached to the base centre. Let $\mathcal{W}$ be the reference fixed world frame, $\mathcal{E}$ be the frame attached to the arm's end-effector, and $\mathcal{C}$ be the camera frame.

Let $\bar{v}_{\mathcal{P}} = \begin{bmatrix} v_{\mathcal{P}}^T & \omega_{\mathcal{P}}^T \end{bmatrix}^T \in \mathbb{R}^6$ be the twist of $\mathcal{P}$ with respect to $\mathcal{W}$, expressed in $\mathcal{P}$,with $v_{\mathcal{P}} \in \mathbb{R}^3$ the instantaneous linear velocity and $\omega_{\mathcal{P}} \in \mathbb{R}^3$ the instantaneous angular velocity. Similarly, let $\bar{v}_{\mathcal{C}} = \begin{bmatrix} v_{\mathcal{C}}^T & \omega_{\mathcal{C}}^T \end{bmatrix}^T \in \mathbb{R}^6$ be the twist of $\mathcal{C}$ with respect to $\mathcal{W}$, expressed in $\mathcal{P}$, with $v_{\mathcal{C}} \in \mathbb{R}^3$ the instantaneous linear velocity and $\omega_{\mathcal{C}} \in \mathbb{R}^3$ the instantaneous angular velocity.

The considered robot is equipped with $n_j = 4$ Mecanum wheels mounted in cross-shape configuration [21] so that movement is omnidirectional. Let $\omega_j \in \mathbb{R}$ be the angular speed of the j-th wheel, with $j = 1, \ldots, n_j$. Let $R_{x,j} > 0$ and $R_{y,j} > 0$ be the distances between the robot center and the center of the j-th wheel along the $x_{\mathcal{P}}$ and $y_{\mathcal{P}}$ axes, respectively, and $R_M > 0$ be the wheel's radius (see Fig. 3).

The differential map between $\bar{v}_{\mathcal{P}}$ and the wheels angular velocities vector, $\omega = \begin{bmatrix} \omega_1 & \cdots & \omega_{n_j} \end{bmatrix}^T \in \mathbb{R}^{n_j}$, is expressed by $\bar{v}_{\mathcal{P}} = J\omega$, where $J \in \mathbb{R}^{6 \times n_j}$ act as a constant mapping matrix, and it is defined as

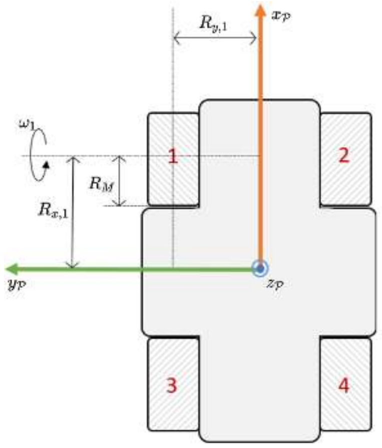

Fig. 3. Top view of the MIRA SPS robot's mobile base.

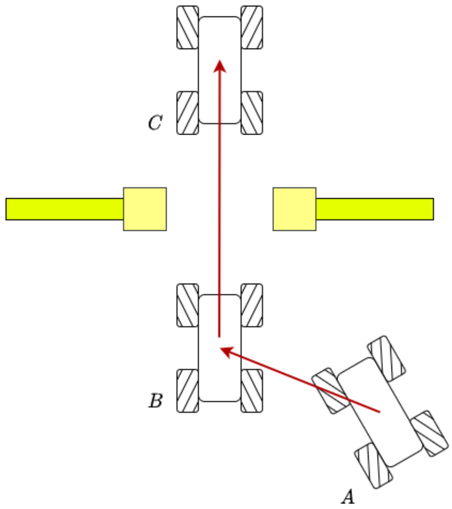

Fig. 4. Schematic of the proposed two-step solution from the top view.

$$J = \frac{R_M}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\dfrac{1}{a+b} & \dfrac{1}{a+b} & -\dfrac{1}{a+b} & \dfrac{1}{a+b} \end{bmatrix}, \tag{1}$$

with $a = R_{x,j}$ and $b = R_{y,j}$, $\forall j \in \{1, \ldots, n_j\}$ [22].

B. Task definition

With reference to Fig. 4, let A be the current point of our mobile base, B be the point in which the robot is perfectly aligned to the narrow passage, and C be the point after the door is crossed. For simplicity, the task has been divided into two sub-steps: the first is the closed-loop alignment to the door (movement from A to B), while the latter is an open-loop forward movement starting from B and leading to C, that has to be performed with the arm lowered as shown in Fig. 1.

The alignment uses only visual information provided by the camera and ensures high accuracy. In this way, there

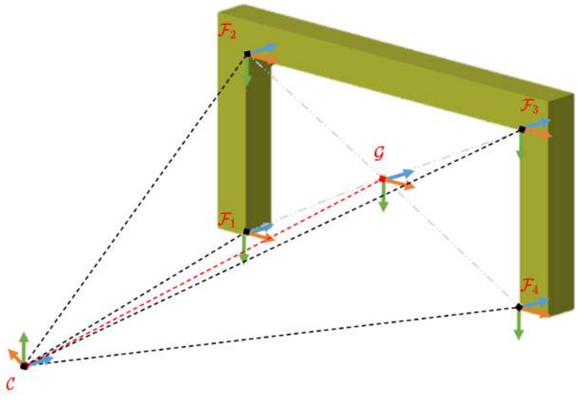

Fig. 5. Schematic of the narrow passage, with the considered corners and the related gate frames.

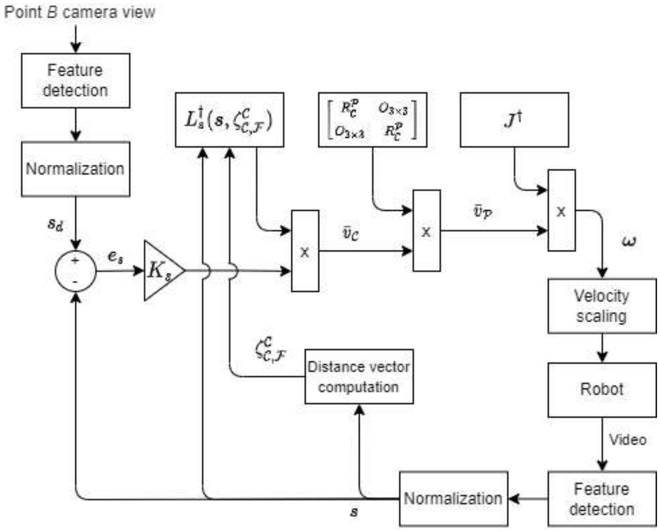
Point B camera view
Feature detection
Normalization
s_d
e_s
+
-
K_s
$L_s^\dagger(s, \zeta_{\mathcal{C},\mathcal{F}}^{\mathcal{C}})$
x
$\bar{v}_{\mathcal{C}}$
$\begin{bmatrix} R_{\mathcal{C}}^{\mathcal{P}} & O_{3\times3} \\ O_{3\times3} & R_{\mathcal{C}}^{\mathcal{P}} \end{bmatrix}$
x
$\bar{v}_{\mathcal{P}}$
$J^\dagger$
x
ω
Velocity scaling
Robot
Video
Feature detection
Normalization
s
$\zeta_{\mathcal{C},\mathcal{F}}^{\mathcal{C}}$
Distance vector computation

Fig. 6. IBVS control scheme

is no collision with the gate during the open-loop forward movement. The aligned pose B has to be chosen close to the gate so that the path length towards C is shortened and, thus, collision risk is reduced. However, it is essential to find a good trade-off between the closeness of the aligned pose and the barrel distortion affecting the algorithm.

C. IBVS

With reference to Fig. 5, let $\mathcal{G}$ be the gate frame placed at the center of the narrow passage, without loss of generality. Besides, let $\mathcal{F}_i$, with $i = 1, \dots, n$, be the frames placed at each feature on the gate. In this case study, the chosen features are the $n = 4$ gate corners (see Fig. 5).

It is worth remarking that the presence of a robotic arm is not mandatory for the devised solution. Nonetheless, using a hand camera helps obtain a better field of view.

The IBVS algorithm controls the robot motion to minimize the difference between the n normalized features detected at the desired configuration $s_d \in \mathbb{R}^n$ and at the current one $s \in \mathbb{R}^n$ (in pixel), directly on the image plane. This technique has the great advantage of not requiring the online robot-gate relative pose computation, except for the camera-feature distance vector measured along the $z-$axis of $\mathcal{C}$. Such distances are stacked in the vector $\zeta_{\mathcal{C},\mathcal{F}}^{\mathcal{C}} = \begin{bmatrix} z_{\mathcal{C},\mathcal{F}_1}^{\mathcal{C}} & \cdots & z_{\mathcal{C},\mathcal{F}_n}^{\mathcal{C}} \end{bmatrix}^T \in \mathbb{R}^n$, with $z_{\mathcal{C},\mathcal{F}_i}^{\mathcal{C}} > 0$, where $i = 1, \dots, n$. The IBVS is also less sensitive to the intrinsic and extrinsic camera calibrations, and the gate is kept inside the field of view along the entire task. Notice that occlusions cannot happen in the addressed scenario. Since the gate is a fixed element of the environment, the time derivative of the desired feature vector, $\dot{s}_d \in \mathbb{R}^n$, is null. For the same reason, it is possible to use the differential equation

$$\dot{s} = L_s(s, \zeta_{\mathcal{C},\mathcal{F}}^{\mathcal{C}})\bar{v}_{\mathcal{C}} \tag{2}$$

to map image-space variables to the operational-space variables. In detail, $\bar{v}_{\mathcal{C}}$ is transformed into the movement of the features, $\dot{s}$, through the $(2n \times 6)$ interaction matrix $L_s(s, \zeta_{\mathcal{C},\mathcal{F}}^{\mathcal{C}}) = \begin{bmatrix} L_{s_1}(s_1, z_{\mathcal{C},\mathcal{F}_1}^{\mathcal{C}}) \\ \cdots \\ L_{s_n}(s_n, z_{\mathcal{C},\mathcal{F}_n}^{\mathcal{C}}) \end{bmatrix}$, whose elements are defined as

$$L_{s_i}(s_i, z_{\mathcal{C},\mathcal{F}_i}^{\mathcal{C}}) = \begin{bmatrix} -\frac{1}{z_{\mathcal{C},\mathcal{F}_i}^{\mathcal{C}}} & 0 & \frac{X_i}{z_{\mathcal{C},\mathcal{F}_i}^{\mathcal{C}}} & X_i Y_i & -(1 + X_i^2) & Y_i \\ 0 & -\frac{1}{z_{\mathcal{C},\mathcal{F}_i}^{\mathcal{C}}} & \frac{Y_i}{z_{\mathcal{C},\mathcal{F}_i}^{\mathcal{C}}} & 1 + Y_i^2 & -X_i Y_i & -X_i \end{bmatrix},$$

with $X_i, Y_i \in \mathbb{R}$ the normalized coordinate of the detected feature in the image plane.

By solving a Perspective-n-Point problem [23], the operational space pose of the gate with respect to the camera is obtained, expressed by the transformation matrix $T_{\mathcal{G}}^{\mathcal{C}} \in SE(3)$. Then, since the geometry of the gate is known (*i.e.*, the pose of $\mathcal{G}$ with respect to the feature frames, $T_{\mathcal{F}_i}^{\mathcal{G}} \in SE(3)$, is known), each $z_{C,\mathcal{F}_i}^{\mathcal{C}}$ can be extracted from the last column of $T_{\mathcal{F}_i}^{\mathcal{C}} = T_{\mathcal{G}}^{\mathcal{C}} T_{\mathcal{F}_i}^{\mathcal{G}}$, with $i = 1, ..., n$ and $T_{\mathcal{F}_i}^{\mathcal{C}} \in SE(3)$ the pose of the feature frame $\mathcal{F}_i$ in $\mathcal{C}$.

During the task, the arm is still, so that the hand camera is fixed with respect to the platform. This means that $\bar{v}_{\mathcal{C}}^{\mathcal{P}} \equiv \bar{v}_{\mathcal{P}}$, with $\bar{v}_{\mathcal{C}}^{\mathcal{P}}$ the twist of $\mathcal{C}$ with respect to $\mathcal{W}$, expressed in $\mathcal{P}$. The IBVS control law can be designed by pseudo-inverting (2), leading to $\bar{v}_{\mathcal{C}} = L_s^\dagger(s, \zeta_{\mathcal{C},\mathcal{F}}^{\mathcal{C}})K_s(\boldsymbol{s}_d - \boldsymbol{s})$, with $K_s \in \mathbb{R}^{n \times n}$ a positive definite matrix and $\dagger$ the pseudo-inversion operator. Then, to obtain the required velocities for each wheel, the mapping relationship $\omega = J^\dagger \bar{v}_{\mathcal{P}}$ must be applied, with $\bar{v}_{\mathcal{P}} \equiv \bar{v}_{\mathcal{C}}^{\mathcal{P}} = \begin{bmatrix} R_{\mathcal{C}}^{\mathcal{P}} & O_{3\times3} \\ O_{3\times3} & R_{\mathcal{C}}^{\mathcal{P}} \end{bmatrix} \bar{v}_{\mathcal{C}}$, where $O_\times$ is the zero matrix of proper dimensions and $R_{\Sigma_1}^{\Sigma_2} \in SO(3)$ is the rotational matrix from a generic frame Σ_1 to Σ_2. As long as the gain matrix $K_s \in \mathbb{R}^{n \times n}$ is positive definite, the evolution of the error $e_s = s_d - s$ is stable, *i.e.*, $\dot{\boldsymbol{e}}_s + K_s \boldsymbol{e}_s = 0$. Asymptotic stability is not achievable due to mapping matrix J introduced in (1) not being full rank [20].

An overview of the algorithm is shown in Fig. 6. The wheels' speed is increases when the robot is very distant from the aligned pose. It is then essential to consider the motor

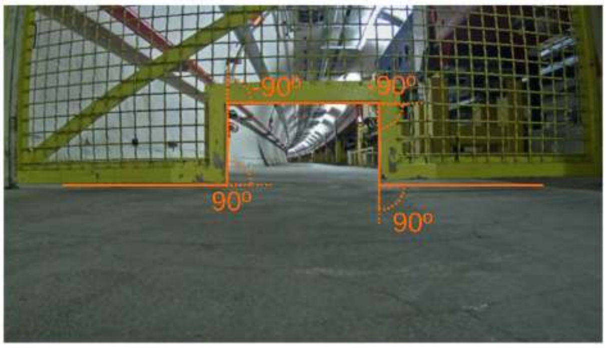

Fig. 7. Example of contour template for the metallic door.

nominal speed: if one or more motors saturate, the resulting platform movement is unpredictable. Moreover, high-speed velocities cause a slippage effect. On the other hand, when the error e_s has a low value, the alignment gets slower. This means that the algorithm slows down once the robot is close to the goal pose. Therefore, a scaling solution has been adopted: ω is halved whenever $\omega_j > \omega_{lim}$, with $j = 1, ..., n_j$ and $\omega_{lim} > 0$. Therefore, K_s speeds up the algorithm when the robot is close to the door, and ω_{lim} prevents critical high speed whenever the robot is far from it.

Significant factors impacting precision are poor camera calibration, frames misalignment, and slippage. The operator supervising the task can switch back to manual control in faulty situations.

III. Computer vision

Our visual processing aims to detect and track the doors in the tunnel and use them for manoeuvring the robot. In the facility, constant illumination is provided by spotlights. The vision system should be robust with respect to changes in lighting conditions given by the movement of the on-robot camera. We present a novel combination of color object detection and edge tracker that robustly extracts visual features for controlling the robot platform.

The characteristics of the doors determine the visual processing algorithm. Doors are metallic, possibly reflective, texture-less, and solid-colored. Poly-lines can approximate their contour. A contour template is defined by a set of relative angles between consecutive contour segments, whose dimension is neglected. Thus, the template is invariant to the door scale. The angles between segments are relative: the proposed approach is invariant to the rotation around the camera axis. An example is shown in Fig. 7. The list of angles for the template of the yellow metallic door is $(90°, -90°, -90°, 90°)$.

The vision processing consists of color thresholding, removing noise with morphological operations (opening and closing), and contour detection. The best match between the contour template and each subset of the most significant detected contour is returned. The contours detection is noisy due to the cluttered background and lighting conditions. Nevertheless, the detected points are helpful for the initialization of a line tracker. We chose a moving edge tracker [24] that can track line segments in monochrome images. Thus, it is commonly used with the image intensity channel. Since we are tracking highly saturated color objects, the edges in the monochrome images of the saturation channel are more distinctive to the grey floor and walls.

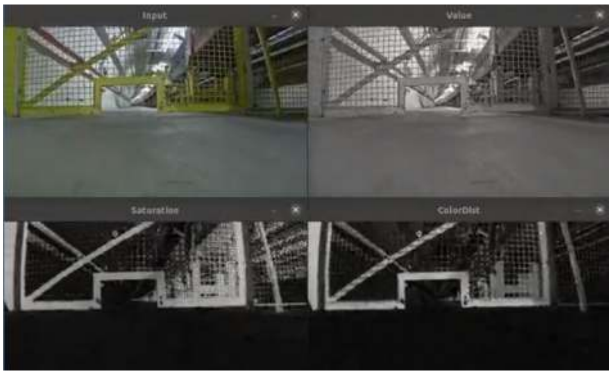

Fig. 8. Original input image (top-left) and monochrome channels for tracking: top-right, intensity (value) channel; bottom-left, saturation channel; bottom-right, proposed *color distance* channel.

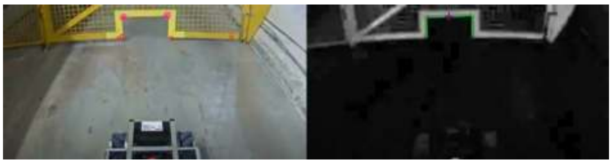

Fig. 9. Contour points (left image) used for the initialization of the segment tracker (right image).

Figure 8 depicts a sample frame of the yellow door: in the monochrome intensity (value) image, the door's brightness and the floor are very similar; thus, the edges are difficult to track. However, the saturation channel produces evident edges since the floor and walls appear nearly black. The saturation channel's drawback is that its computation is based on the value channel, which can be computed in several ways and produce different results to the perceived colors.

We have introduced a new color distance measurement that improves the image to be tracked. This color distance is computed as the norm of the vector product of the RGB value of each pixel with the vector along the diagonal of the RGB cube $(0, 0, 0) \rightarrow (1, 1, 1)$. As shown in Fig. 8 (bottom-right), this measurement can highlight the colored objects with less noise and more proportional intensity than the saturation channel. The tracker is automatically initialized with the points computed by the contour detector (Fig. 9). As long as the result of the tracker is valid, it is used for the servoing task. If the tracker fails, it will be reinitialized again with the points of the detector.

The combination between contour detection and edge tracking produces a solid result for the image points used in the servoing task. If the tracker fails and the detector cannot be re-initialized, the system falls back to manual control.

IV. Experiments

As stated before, the employed robot is the MIRA SPS, a mobile manipulator composed of an aluminum wheeled platform, $n_j = 4$ Mecanum wheels, and a Kinova JACO

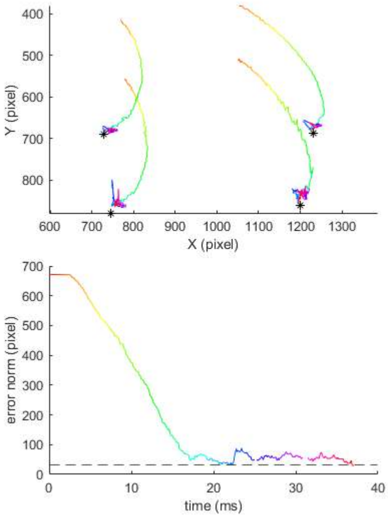

Fig. 10. First experimental result of the visual servoing test, achieved with $k = 1$ and $\omega_{lim} = 0.05$. Top graph shows image plane features evolution, while bottom graph shows error norm evolution.

6S arm. The employed motors are Maxon EC-4 200W 14-1, driven by Elmo Ethercat PELM027 drivers. Robot maximum linear forward velocity is 2 m/s, while slip-speed was found at 0.4 m/s on SPS tunnel's floor. Referring to (1), the parameters are $a = 18.3$ cm, $b = 13.95$ cm, and $R_M = 7.5$ cm. The whole robot dimensions are $52 \times 35 \times 19$ cm and it weights 40 kg.

Experiments were conducted in the laboratory using a mockup of the door. A video of the experiments can be found at the following link[1]. The camera stream was set at 12.5 frames per second, with a resolution of 1920×1080 pixels. The onboard computer is a Lenovo M90n-1, with an integrated graphical unit.

The gain matrix K_s has been experimentally tuned to $K_s = \text{diag}(k, k, ..., k)$, with $k > 0$ varying in the addressed case studies. Other choices, such as $K_s = \text{diag}(k, k, ..., 2k)$, are useful to boost the rotational movement over others. This is particularly effective when $\zeta^{\mathcal{C}}_{\mathcal{C},\mathcal{F}}$ detection is not accurate.

The following experiments show the behavior with different values of k and ω_{lim}, given the same starting pose A and the same aligned pose B. The top pictures in Fig. 10-13 show the detected corner points movement across the image plane while moving to reach the goals (the black stars). Instead, the bottom pictures show the not normalized error norm. The algorithm stops when the error norm, $|e_s|$, is below the threshold of 0.03: this was found to be the best trade-off value between alignment time and precision. Values higher than 0.04 do not guarantee gate crossing. Blank spots in graphs are due to outliers' cancellation.

The first experiment is carried out by choosing $k = 1$ and $\omega_{lim} = 0.05$. With reference to Fig. 10, it clearly shows that error evolution is stable. It can be split into a settling phase and a fine-alignment phase. The whole alignment takes 36 s because of the low value of ω_{lim}. Since k is low as well, the fine-alignment (small corrections that start at 17 s) takes half of the total time. Detected points move smoothly on the image plane, reflecting the robot's untroubled movement.

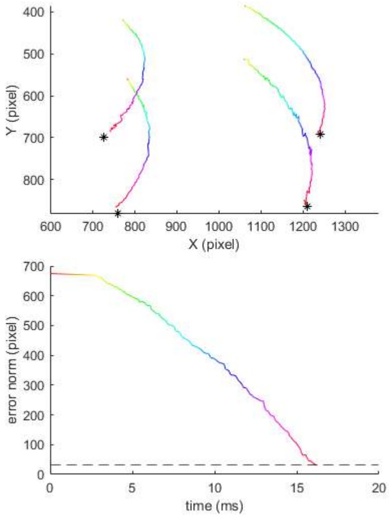

Fig. 11. Second experimental result of the visual servoing test, achieved with $k = 5$ and $\omega_{lim} = 0.05$. Top graph shows image plane features evolution, while bottom graph shows error norm evolution.

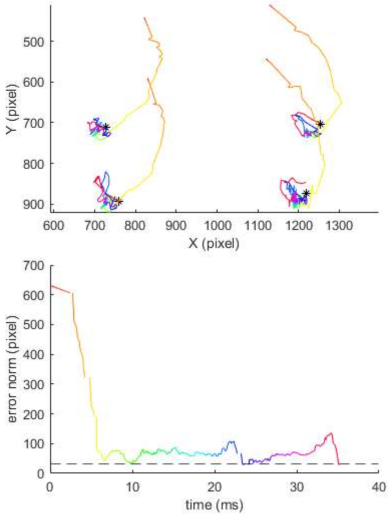

Fig. 12. Third experimental result of the visual servoing test, achieved with $k = 1$ and $\omega_{lim} = 0.2$. Top graph shows image plane features evolution, while bottom graph shows error norm evolution.

The second experiment sees the value of k increased by 5 times, that is $k = 5$ and $\omega_{lim} = 0.05$. Plots are depicted in Fig. 11, showing how fine-alignment overtime can be canceled. The total time is 16 s, and the movement on the image plane is smooth.

In the third experiment, compared to the first one, only ω_{lim} increased: $k = 1$ and $\omega_{lim} = 0.2$. The settling alignment duration is reduced from 17 s to 5 s (see Fig. 12). However, the fine-alignment duration is larger than the first case: this also reflects a harsher motion in the image plane.

In the fourth experiment, a trade-off between k and ω_{lim} was considered, with $k = 2$ and $\omega_{lim} = 0.1$. Settling time is

[1] https://youtu.be/N0FE8g50Qik

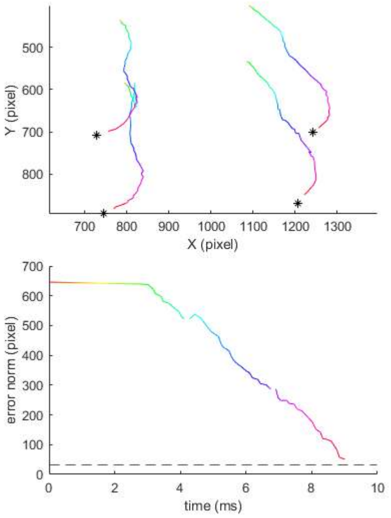

Fig. 13. Fourth experimental result of the visual servoing test, achieved with $k = 2$ and $\omega_{lim} = 0.1$. Top graph shows image plane features evolution, while bottom graph shows error norm evolution.

9 s, while no fine-alignment is needed. Points evolution on the image plane is not too harsh (see Fig. 13).

In each experiment, tracked points in the image plane do not cause large motions that can cause features to exit the field of view. Noise is instead caused by movement vibrations, platform drifting (due to non-perfect flatness of the ground), and detection displacements.

In the first and third experiments, Fig. 10 and Fig. 12, respectively, the errors evolution reaches local minima before starting the fine-alignment phase. Increasing the threshold stop value reduces the servoing time drastically. Nevertheless, the abrupt speed changes of the settling phase lead to platform slippage, invalidating the alignment.

V. Conclusion

We addressed the problem of crossing narrow doors in a semi-structured environment. We employed a visual servoing approach where the point features are extracted from a robust detector and tracker of the edge lines of a colored contour. The points are obtained from the intersections of the lines. A novel color segmentation based on the distance of the pixel colors to the diagonal of the RGB cube is proposed as an alternative to the intensity or saturation channels of the image. The devised solution was tested in the CERN's SPS: a mobile robot platform is equipped with a lightweight arm to perform inspection and maintenance tasks remotely driven by an operator. We showed the ability to drastically reduce crossing time with smooth movements: 30 s (alignment, arm lowering, forward movement, arm raising), versus 2 minutes of the teleoperation case.

References

[1] J. Cacace, A. Finzi, V. Lippiello, M. Furci, N. Mimmo, and L. Marconi, "A control architecture for multiple drones operated via multimodal interaction in search amp; rescue mission," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2016.

[2] "A semi-autonomous mobile robot for bridge inspection," *Automation in Construction*, vol. 91, pp. 111–119, 2018.

[3] L. F. Manta, D. Popescu, O. Unguritu, H. Roibu, M. Marian, and M. Abagiu, "Software architecture for a mobile robot designed for rescue missions support in hazardous environments," in *2020 21th International Carpathian Control Conference (ICCC)*.

[4] A. H. Reddy, B. Kalyan, and C. S. Murthy, "Mine rescue robot system – A review," *Procedia Earth and Planetary Science*, vol. 11, pp. 457–462, 2015.

[5] S. Kang, C. Cho, C. Park, J. Lee, D. Ryu, and M. Kim, *ROBHAZ-DT2: Passive Double-Tracked Mobile Manipulator for Explosive Ordnance Disposal*. Springer Berlin Heidelberg, 2006.

[6] J. Petereit, J. Beyerer, T. Asfour, S. Gentes, B. Hein, U. D. Hanebeck, F. Kirchner, R. Dillmann, H. H. Götting, M. Weiser, M. Gustmann, and T. Egloffstein, "ROBDEKON: Robotic systems for decontamination in hazardous environments," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2019.

[7] M. Di Castro, "A Novel Robotic Framework for Safe Inspection and Telemanipulation in Hazardous and Unstructured Environments," 2019. [Online]. Available: http://cds.cern.ch/record/2708835

[8] K. Kershaw, B. Feral, J.-L. Grenard, T. Feniet, S. D. Man, C. Hazelaar-Bal, C. Bertone, and R. Ingo, "Remote inspection, measurement and handling for maintenance and operation at cern," *International Journal of Advanced Robotic Systems*, vol. 10, no. 11, p. 382, 2013.

[9] Z. Xiang, V. Santos, and J. Liu, "Locating and crossing doors and narrow passages for a mobile robot," in *1st International Conference on Informatics in Control, Automation and Robotics*, 2004.

[10] Y. Matsumoto, C. Premachandra, B. H. Sudantha, and S. Sumathipala, "A study on autonomous entering into narrow path using a mobile robot," in *2018 3rd International Conference on Information Technology Research*, 2018.

[11] J. Borenstein, D. Wehe, L. Feng, and Y. Koren, "Mobile robot navigation in narrow aisles with ultrasonic sensors," in *ANS 6th Topical Meeting on Robotics and Remote Systems*, 12 1995.

[12] G. Maciel, M. Faria Pinto, I. Junior, and A. Marcato, "Methodology for autonomous crossing narrow passages applied on assistive mobile robots," *Journal of Control, Automation and Electrical Systems*, vol. 30, pp. 943–953, 2019.

[13] V. Karakkat Narayanan, F. Pasteau, M. Babel, and F. Chaumette, "Visual servoing for autonomous doorway passing in a wheelchair using a single doorpost," in *IEEE/RSJ IROS Workshop on Assistance and Service Robotics in a Human Environment*, 2014.

[14] H. Chao, Y. Gu, and M. Napolitano, "A survey of optical flow techniques for robotics navigation applications," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1, pp. 361–372, 2014.

[15] N. Ohnishi and A. Imiya, "Appearance-based navigation and homing for autonomous mobile robot," vol. 31, no. 6–7, pp. 511—-532, 2013.

[16] M. Ferro, A. Paolillo, A. Cherubini, and M. Vendittelli, "Vision-based navigation of omnidirectional mobile robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2691–2698, 2019.

[17] K. Ibrahim Khalilullah, S. Ota, T. Yasuda, and M. Jindai, "Development of robot navigation method based on single camera vision using deep learning," in *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan*, 2017.

[18] C. Prados Sesmero, L. R. Buonocore, and M. Di Castro, "Omnidirectional robotic platform for surveillance of particle accelerator environments with limited space areas," *Applied Sciences*, vol. 11, no. 14, 2021.

[19] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[20] ——, "Visual servo control, Part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[21] Y. Li, S. Dai, L. Zhao, X. Yan, and Y. Shi, "Topological design methods for mecanum wheel configurations of an omnidirectional mobile robot," *Symmetry*, vol. 11, no. 10, 2019.

[22] H. Taheri, B. Qiao, and N. Ghaeminezhad, "Kinematic model of a four mecanum wheeled mobile robot," *International journal of computer applications*, vol. 113, no. 3, pp. 6–9, 2015.

[23] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer-Verlag London Limited, 2008.

[24] A. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 615–628, 2006.