2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Online Minimization of the Robot Silhouette Viewed From Eye-to-Hand Camera

G. Cortigiani[‡,*], B. Brogi[‡,*], A. Villani[‡], T. Lisini Baldi[‡,§], N. D'Aurizio[‡,§], and D. Prattichizzo[‡,§]

***Abstract*— Redundant robots have the potential to perform internal joints motion without modifying the pose of the end-effector by exploiting the null-space of the Jacobian matrix. Capitalizing on that feature, we developed a control technique for minimizing the robot visual appearance when observed from an eye-to-hand camera. Such algorithm is instrumental in contexts where quickly adjusting the perspective to see objects obstructed by the robot is impractical (e.g., teleoperation in narrow environment). Diminished reality techniques are frequently employed in these cases to mitigate the robot intrusion into the environment, although these techniques may sometimes compromise the perceived realism. The experimental evaluation confirmed the effectiveness of our control algorithm, demonstrating an average reduction of 4.67% of the area covered by the robot within the frame when compared to the case without the optimization action.**

I. Introduction

Virtual and Augmented Reality have found in the Metaverse one of their most promising accomplishments, showcasing a three-dimensional digital world wherein users, embodied in customized avatars, meet each other in a shared environment that augments their physical surroundings [1]. In our previous work [2], we introduced for the first time the concept of *Physical Metaverse*[1], a shared augmented environment where users are able to interact not only with virtual entities but also with tangible objects. This capability is achieved through a novel interface named *Avatarm*, which uses a robotic arm that mimics users interactions with the digital objects by manipulating their real twins. The robot remains hidden from view thanks to an online diminished reality technique which overlays the robot in the frames streamed from an eye-to-hand camera with an image of the background, acquired before placing the robotic arm in the real environment.

Despite this technique successfully improving the users interaction experience in the Physical Metaverse, the overlaying of a static background on the streamed image of the robot is perceived as unnatural by some users, reducing the sense of realism of the surrounding mixed environment. This limitation is particularly relevant when variations in lighting conditions and dynamic changes within the environment occur during the experience.

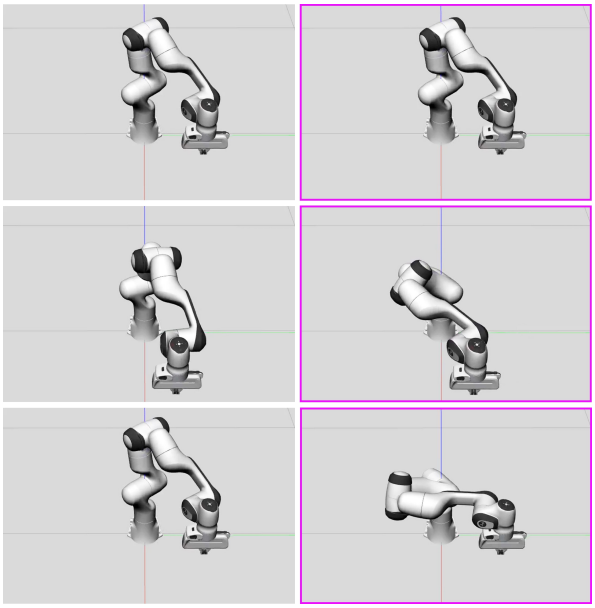


Fig. 1: Three key time instants taken from the execution of a representative end-effector trajectory performed with (right column) and without (left column) the action of the control algorithm minimizing the robot silhouette in the eye-to-hand camera framing. The starting pose (first row) of the robot is the same for both conditions. Throughout the trajectory execution (second and third rows), the algorithm actively reduces the robot visibility while maintaining the end-effector pose along the desired path.

The need for reducing the visual encumbrance of the robot can also be faced in more general critical domains such as teleoperation in unstructured environments [3], or visual servoing applications, in which vision occlusions lead to the control failure [4], [5]. Other scenarios include individuals with disabilities who rely on robots to mitigate their limitations [6]. Indeed, visual perception is one of the most fundamental aspects that affects the quality and accomplishment of teleoperation tasks [7]. Typically, occlusion issues are addressed by employing an eye-in-hand configuration, wherein a camera is mounted on the robot's end-effector. However, this causes an unintuitive remote control, as well as a limited and configuration-dependent field of view [8]. Approaches involving multiple cameras have been proposed to overcome this issue [3], [5], [9], [10], with the disadvantage of requiring more complex setups

The research leading to these results has received funding from the European Union's Horizon Europe programme under grant agreement No. 101070292 of the project "HARIA - Human-Robot Sensorimotor Augmentation - Wearable Sensorimotor Interfaces and Supernumerary Robotic Limbs for Humans with Upper-limb Disabilities". This work was also supported by the Leonardo S.p.A. under Grant LDO/CTI/P/0025793/22.

‡ are with the Department of Information Engineering and Mathematics, University of Siena, Via Roma 56, 53100 Siena, Italy. {cortigiani, brogi, villani, lisini, ndaurizio, prattichizzo}@diism.unisi.it

§ are with the department of Humanoids and Human Centered Mechatronics, Istituto Italiano di Tecnologia, Genova, Italy.

* These authors contributed equally to this work.

[1] http://physicalmetaverse.cloud

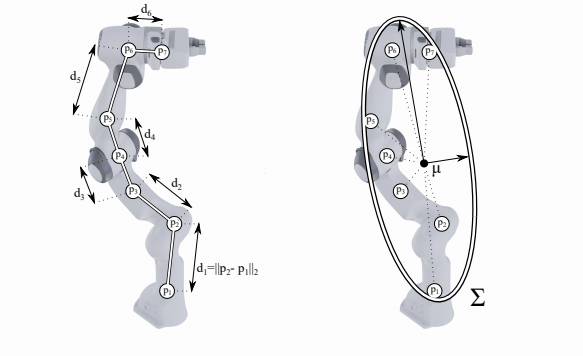


Fig. 2: Visual representation of the two methods for approximating the robot silhouette in the framing. On the left, the link-based method uses the sum of the distances (denoted as d_i) between consecutive joint projections p_i onto the camera frame. On the right, the dispersion-based method computes the determinant of the covariance matrix Σ of the projected joint positions p_i onto the camera frame centered around their mean value μ.

and control strategies. Superimposition techniques [11], [12] may be used to reconstruct the occluded areas, though affecting, as previously mentioned, the perceived realism. Alternatively, improving the visibility can be addressed through the implementation of camera control algorithms to ensure tracking of targeted objects [13]. However, this approach falls short in scenarios in which adjust or modify the pose of the camera is unfeasible.

This paper aims at dynamically reducing the impact of the robot in the user's field of view proposing a control algorithm that minimizes the robot silhouette in a eye-to-hand configuration. In particular, achieving this objective involves leveraging the manipulator redundancy to optimize the robot pose while maintaining the desired pose of the end-effector.

From a mathematical perspective, the null space of the Jacobian matrix can be exploited to accomplish an auxiliary task having no effects at the pose of end-effector [14]. For instance, Liegeois *et al.* [15] used this space to accomplish an additional avoidance action of the robot joint limits. In a similar way, Yokishawa *et al.* maximized the kineto-static and dynamic manipulability measures, as presented in [16] and [17], respectively. More recently, Chen *et al.* in [18] presented a control law based on the null space of the Jacobian matrix integrating self-collision and joint limits avoidance with the maximization of an extended manipulability measure.

In this work, the null space of the Jacobian matrix is exploited to minimize the number of pixels in the camera framing that contains portions of the robot silhouette, and, at the same time, maximize the distance from the joint limits to keep the primary task (i.e., accomplish a desired trajectory) feasible. Fig. 1 visually summarizes a trajectory of the robot performed with and without the action of the proposed control method.

II. Robot Control Algorithm

A prevailing approach to the control of redundant manipulators entails the computation of the desired joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^n$, with n representing the number of joints. Given the desired robot end-effector motion, inverse kinematics can determine an appropriate joint velocity configuration for which the end-effector moves toward the target pose. Since the robot is redundant, there exist multiple solutions that meet the required motion, and thus we can select the one that better satisfies some additional metrics. This idea is based on a hierarchical arrangement of the involved tasks and can be interpreted as a local optimization. The highest priority task is executed employing all the capabilities of the robotic system. The second priority task is then accomplished exploiting the null space of the top priority one. In other words, the secondary task is This concept can be formulated as:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\dot{\boldsymbol{r}}_d + \boldsymbol{K}\boldsymbol{e}) + (\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J})\dot{\boldsymbol{q}}_0$$

where $\dot{\boldsymbol{r}}_d \in \mathbb{R}^6$ is the vector containing the desired end-effector linear and angular velocities, $\boldsymbol{J} \in \mathbb{R}^{6\times n}$ is the Jacobian matrix of the robot, $(\cdot)^{\dagger}$ is the pseudo-inverse operator, $\boldsymbol{e} \in \mathbb{R}^6$ is the error of the end-effector pose, $\boldsymbol{K} \in \mathbb{R}^{6\times 6}$ is a positive definite matrix that ensures the error convergence, $(\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J})$ is the projector on the null space of the Jacobian matrix $\mathscr{N}(\boldsymbol{J})$, and $\dot{\boldsymbol{q}}_0$ is an arbitrary vector that parametrizes the solution sets. The product between the last two terms gives the projected joint velocities that belong to $\mathscr{N}(\boldsymbol{J})$ and do not contribute to the end-effector motion. Therefore, these velocities can be utilized to accomplish a secondary task while guaranteeing the accurate execution of the primary task, i.e., tracking the desired end-effector trajectory.

The secondary task may involve minimizing (or maximizing) a given scalar objective function $C(\boldsymbol{q}) : \mathbb{R}^n \to \mathbb{R}$. To accomplish this, the gradient of the function can be projected onto the null space by choosing $\dot{\boldsymbol{q}}_0$ (in the case of minimization) as follows:

$$\dot{\boldsymbol{q}}_0 = -\nabla_{\boldsymbol{q}} C(\boldsymbol{q}).$$

This projection can be applied as long as the function C explicitly depends on the joint variables $\boldsymbol{q} \in \mathbb{R}^n$ and is differentiable with respect to $\boldsymbol{q}$.

As the purpose of this work is to minimize the amount of pixels in the framing containing portions of the robot, ideally, we should define the function C to describe this quantity. Unfortunately, directly utilizing this approach proves unfeasible as it is impossible to formulate a differentiable closed-form function that accurately computes the area covered by the robot within the scene for any camera pose. This difficulty arises due to several factors, including robot self-occlusions with respect to the point of view and the irregular shape of its links. Hence, the strategy here presented is to design a differentiable function capable of accurately reflecting the behavior seen in the robot silhouette within

the framing. To achieve this, two distinct methods were devised for comparison purposes, providing insight into their effectiveness in accurately representing the aforementioned area value.

A. Robot Projection Onto the Image Plane

Both methods are based on the projection of the positions of the robot joints onto the image plane of the camera. Let the reference frame of each robot joint i be chosen according to the Denavit-Hartenberg (DH) convention, and let the homogeneous transformation matrix $\boldsymbol{A}_i^0(\boldsymbol{q})$ describe the pose of the i^{th} joint frame with respect to the world reference frame $O_0 - x_0y_0z_0$. This matrix can be computed by exploiting the DH parameters of the robot as:

$$\boldsymbol{A}_i^0(\boldsymbol{q}) = \boldsymbol{A}_1^0(q_1)\boldsymbol{A}_2^1(q_2)\cdots\boldsymbol{A}_i^{i-1}(q_i).$$

The fourth column of the matrix $\boldsymbol{A}_i^0(\boldsymbol{q})$ is the homogeneous vector of the i^{th} joint position, denoted as $\tilde{\boldsymbol{p}}_{\boldsymbol{i}}^0 = [\boldsymbol{p}_i^0; 1] \in \mathbb{R}^4$, where $\boldsymbol{p}_i^0$ represents the vector of spatial coordinates of the i^{th} joint. Afterward, taking into account the pose of the camera in the environment, it is possible to identify the homogeneous transformation matrix $\boldsymbol{T}_0^c$ between the world reference frame and the frame attached to the camera $O_c - x_cy_cz_c$. The homogeneous representation of the i^{th} joint position from the world frame to the camera frame is described as:

$$\tilde{\boldsymbol{p}}_i^c = \boldsymbol{T}_0^c\tilde{\boldsymbol{p}}_i^0.$$

The reference frame on the image plane is considered with axes parallel to the axes x_c and y_c of the camera frame, and the origin is at the intersection of the optical axis with the image plane. Taking into account the frontal perspective transformation [19], it is possible to map the points from the camera frame to the image frame as:

$$\hat{x}_i = \frac{f\tilde{p}_{i,x}^c}{\tilde{p}_{i,z}^c}, \quad \hat{y}_i = \frac{f\tilde{p}_{i,y}^c}{\tilde{p}_{i,z}^c}$$

where f is the focal length of the lens in correspondence of the axis z_c of the camera frame. Further imperfections such as aberrations and geometric distortions were not considered for the purpose of this work, since all the computations were performed in simulation. As a result, the point $\hat{\boldsymbol{p}}_i = [\hat{x}_i\hat{y}_i]^T \in \mathbb{R}^2$ describes the position of the robot joint i on the camera image plane.

B. Function #1: Link-Based Method

The first function proposed within this work approximates the area of the robot to the cumulative length of the links projected onto the camera plane. More precisely, the distances between the consecutive projected joints are summed, leading to the following cost function:

$$F_l(\boldsymbol{q}) = \sum_{i=1}^{n-1}\sqrt{(\hat{x}_{i+1} - \hat{x}_i)^2 + (\hat{y}_{i+1} - \hat{y}_i)^2}.$$

In a more practical interpretation, this method minimizes the lengths of the link projections, encouraging the links to be oriented along the direction in which the camera is pointing.

C. Function #2: Dispersion-Based Method

The second proposed function computes the dispersion of the projected joint positions. The generalized variance is used as a metric, that is a one-dimensional measure of a multi-dimensional vector variable evaluated as the determinant of its covariance matrix [20].

The function is defined as:

$$F_d(\boldsymbol{q}) = |\boldsymbol{\Sigma}|$$

being $\hat{\boldsymbol{X}}$ and $\hat{\boldsymbol{Y}} \in \mathbb{R}^n$ the vectors containing the coordinates $\hat{x}$ and $\hat{y}$ of the robot joints, $|\cdot|$ the determinant operator, and $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times2}$ the covariance matrix of the two vectors $\hat{\boldsymbol{X}}$ and $\hat{\boldsymbol{Y}}$ having mean $\mu_{\hat{X}}$ and $\mu_{\hat{Y}}$, respectively. The covariance matrix is computed as:

$$\boldsymbol{\Sigma} = \begin{bmatrix}\sigma_{2,0} & \sigma_{1,1}\\ \sigma_{1,1} & \sigma_{0,2}\end{bmatrix}$$

where

$$\sigma_{i,j} = \sum_{\hat{\boldsymbol{X}},\hat{\boldsymbol{Y}}}(\hat{\boldsymbol{X}} - \mu_{\hat{\boldsymbol{X}}})^i(\hat{\boldsymbol{Y}} - \mu_{\hat{\boldsymbol{Y}}})^j.$$

In other words, the objective of this second approach is to minimize the spatial footprint of the manipulator with respect to the camera by prompting the robot to adopt compact poses (i.e., with low dispersion of joints). A graphical representation of the two approximation methods is reported in Fig. 2.

D. Candidate Cost Functions

The functions described in Sect. II-B and Sect. II-C lead to the definition of two corresponding candidate cost functions $C_l(\boldsymbol{q})$ and $C_d(\boldsymbol{q})$ as:

$$C_l(\boldsymbol{q}) = \alpha_l F_l(\boldsymbol{q}) + \beta L(\boldsymbol{q})$$

$$C_d(\boldsymbol{q}) = \alpha_d F_d(\boldsymbol{q}) + \beta L(\boldsymbol{q})$$

where α_l, α_d, and β are scaling factors that appropriately weigh each contribution to the overall cost. The term $L(\boldsymbol{q})$ is added to the cost function to prevent the robot from reaching the joint limits during its movement, and it is defined as:

$$L(\boldsymbol{q}) = \sum_{i=1}^{n}\left(\frac{1}{w_M - w(q_i)} - \frac{1}{w_M}\right).$$

Following the approach described in [19], we can define the normalized distance from the mechanical joint limits as:

$$w(q_i) = \left(\frac{q_i - \bar{q}_i}{q_{iM} - q_{im}}\right)^2 \tag{1}$$

where q_{iM} and q_{im} represent the upper and lower joint limits, respectively, while $\bar{q}_i$ denotes the midpoint of the joint range. The term $w_M = \frac{1}{4}$ is the maximum value that Eq. (1) can assume. Consequently, maximizing the distance of the joints from the limits guarantees that the robot motion is feasible.

The resulting cost of C_l and C_d are meant to be differentiable functions that approximate the characteristics of the projected robot area, while also adjusting potential joint limit violations.

Algorithm 1: Pseudocode for computing $\dot{\boldsymbol{q}}_{t+1}$

Input :
q_t: joint values at time instant t;
$\dot{r}_{d_t}$: desired end-effector velocity at time instant t;
e_t: end-effector position error at time instant t;
T: sample time;
Output :
$\dot{q}_{t+1}$: desired joint velocities at time instant t+T;
Variables :
k: current iteration step of the loop;
q_k: joint values at step k;

Variable initialization:
$\dot{q}_0 \leftarrow 0_{7\times 1}$;
$q_k \leftarrow q_t$;
$k \leftarrow 0$;
while *loop time* $< T \wedge$ *joints velocity within limits* **do**
  $\dot{q}_{0_k} \leftarrow -\nabla_q C(q_k)$;
  $q_{k+1} \leftarrow q_k + (I - J(q_t)^\dagger J(q_t))\dot{q}_{0_k})T$;
  if $C(q_{k+1}) < C(q_k)$ **then**
    $\dot{q}_0 \leftarrow \dot{q}_0 + \dot{q}_{0_k}$;
  else
    loop exit;
  end
  $k \leftarrow k + 1$;
end
$\dot{q}_{t+1} \leftarrow J^\dagger(q_t)(\dot{r}_{d_t} + Ke_t) + (I - J^\dagger(q_t)J(q_t))\dot{q}_0$;

For the gradient method to work properly, it is essential to take small steps toward the direction of the optimum. This ensures that the algorithm approaches the minimum without causing the cost function to increase and without surpassing the joint velocity limits. Hence, a small step size is employed in the optimization loop, and the projected joint velocities are continuously updated making sure that they are consistently moved in the descending direction until the command need to be transmitted to the robot, preventing the generation of delay or latency. The detailed description of the robot control algorithm is reported in Alg. 1.

III. Experimental Evaluation

A. Correlation Analysis

As a first step, we experimentally assessed if the cost functions proposed in Sect. II are suitable for approximating the area of the robot projected onto the camera image plane. To this end, we performed a correlation analysis to determine whether there exists a monotonic relation between the output of the candidate functions $F_l(q)$ and $F_d(q)$ and the area of the robot silhouette in the framing (expressed as number of pixels).

A virtual environment, created with Unity Graphic Engine (Unity Technologies Inc., US), was used to collect data for the correlation analysis. The virtual scene comprised a digital Franka Emika Research 3 Robot and a virtual camera. We tested the correlation by collecting data from three different points of view. In what follows, we refer to the three perspectives as Camera 1, Camera 2, and Camera 3. For each camera, the resolution in terms of pixels was fixed to 640×480. In Fig. 3, we show the camera poses with respect to the robot, while the values of the camera positions and orientations, along with the focal length, are reported in Tab. I.

	Camera 1	**Camera 2**	**Camera 3**
Position (x,y,z) [m]	$[2, -0.5, 0.5]$	$[0, -2, 0.5]$	$[2, 0, 2]$
Orientation (r,p,y) [rad]	$\left[-\frac{\pi}{2}, 0, \frac{\pi}{2}\right]$	$\left[-\frac{\pi}{2}, 0, 0\right]$	$\left[-\frac{2\pi}{3}, 0, \frac{\pi}{2}\right]$
Focal Length f [mm]	3.67		

TABLE I: Camera parameters used for the experimental evaluation.

Configuration	ρ	$p-$value
Camera 1	0.743	< 0.001
Camera 2	0.763	< 0.001
Camera 3	0.643	< 0.001

TABLE II: Correlation analysis results for the link-based method. Correlation coefficients (ρ) and corresponding levels of statistical significance (p-value) are reported for each camera configuration.

For each condition, we moved the robot in approximately 1.2×10^7 different poses[2], and, for each pose, we computed the number of pixels projected onto the camera image plane following the methodology described in [2], where an image segmentation algorithm [21] is applied. The latter determines whether a pixel in the image plane contains a portion of the robot based on whether its RGBα values all fall within specific thresholds.

The end-effector was not considered in this assessment, as we assumed that tasks within a $\mathbb{R}^6$ space represent the most comprehensive and broadly applicable scenario. Indeed, in most typical tasks such as pick-and-place or pouring, which demand a precise end-effector pose, it is not possible to modify its orientation to reduce the robot visible area. Acquisitions of the three virtual cameras were separately analyzed. For each camera perspective, a preliminary analysis showed the aforementioned relationship to be monotonic for the link-based method $F_l(\boldsymbol{q})$ and not monotonic for the dispersion-based method $F_d(\boldsymbol{q})$, as assessed by visual inspection of the relative scatterplots (see Fig. 4). Therefore, the cost function related to the dispersion-based method was not considered for the subsequent analyses.

For each virtual camera, a Spearman's rank-order correlation test was run to assess the relationship between number of pixels of the robot on the camera image plane and the output of the objective function $F_l(\boldsymbol{q})$. Resulting correlation factors (ρ) and significance levels (p-value) are reported in Tab. II, showing a strong correlation for Camera 1 and Camera 2, and a moderate correlation for Camera 3, in accordance with [22].

[2]This value was obtained moving each of the 7 joints, within their specified limits, with a step of 0.4 radians.

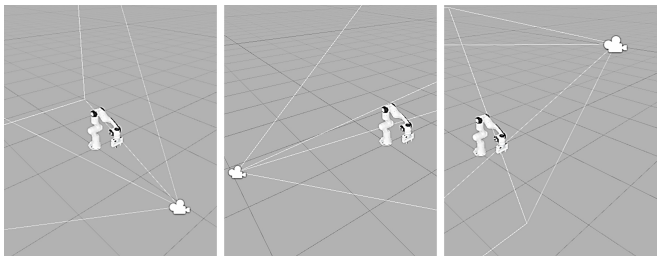
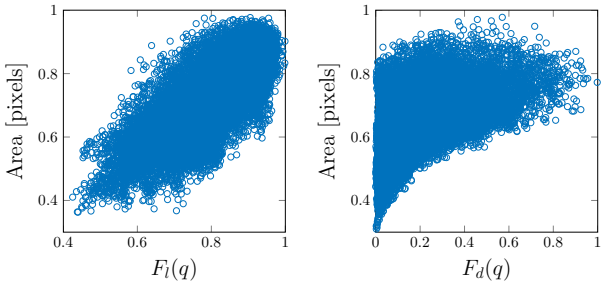


Fig. 3: The three perspectives used in the experimental evaluation. From left to right, the configurations of Camera 1, Camera 2, and Camera 3 with respect to the robot.



(a) (b)

Fig. 4: Two representative scatterplots for number of pixels of the robot in the image plane captured by Camera 2. In (a), real area values versus values estimated with the link-based method. In (b), real area values versus values estimated with the dispersion-based method. Data are normalized for a comparison purpose.

B. Minimization Analysis

Given the results of the correlation analysis, we decided to proceed the experimental evaluation considering only the link-based method.

To test the effectiveness of the control algorithm that minimizes the robot silhouette in the framing by exploiting the cost function based on the link-based method, we simulated four different trajectories using Gazebo and ROS. More precisely, the four trajectories (depicted in Fig. 5) are:

- **T1:** Planar Lissajous Path in the x-y plane with a duration of 20 seconds.
- **T2:** Planar Lissajous Path in the x-y plane with a duration of 30 seconds.
- **T3:** Pouring Trajectory with a duration of 23 seconds.
- **T4:** Pick and Place Trajectory with a duration of 14 seconds.

The first two trajectories delineate paths challenging to be followed due to the differences in curvature and velocity at every point [23], while the latter two trajectories were selected to exemplify tasks of a teleoperation scenario. While T1 and T2 were artificially generated starting from their mathematical equation, T3 and T4 were recorded teleoperating a robot for performing a pouring and a pick-and-place task, respectively. Each trajectory was executed in Gazebo recording the scene from the three different points of view utilized in Sect. III-A, i.e., Camera 1, Camera 2, and Camera 3.

The four trajectories were performed both with and without the minimization action. The robot control constant parameters were fine-tuned through empirical methods, resulting in the following values: $\beta = 0.003, \alpha_s = 0.1, \boldsymbol{K} = \mathrm{diag}\{10, \cdots, 10\}$. The robot control loop was run at $1\,\mathrm{kHz}$. The difference in the number of pixels containing robot portions in both cases was used to quantify the actual reduction of robot appearance in each camera framing. In Tab. III, area reductions are reported for each trajectory and for each point of view, while the corresponding end-effector position and orientation errors are in Tab. IV. In Fig. 6, a plot showing the area with and without the action of the minimization algorithm is depicted.

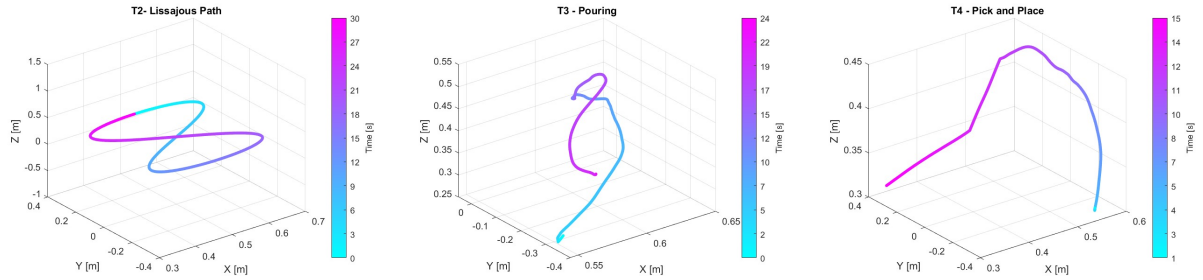
Results and Discussion: Outcomes of the experiments reveal an average reduction of 4.67% in the robot area visible from the camera, with significant variability contingent upon the camera configuration. Specifically, we observe an average reduction of 4.21%, 1.00%, and 8.8% considering the Camera 1, Camera 2, and Camera 3 points of view, respectively. Concerning the first two trajectories T1 and T2, it is worth noting how, on average, T2 presents a more substantial reduction, even though T1 and T2 describe the same path. This is attributed to the longer duration of T2, which gives the robot ample time for reconfiguration into a more advantageous pose.

	Camera 1	**Camera 2**	**Camera 3**	**Trajectory Mean**
T1	-5.73%	-0.31%	-3.59%	-3.21%
T2	-6.38%	-0.26%	-5.99%	-4.21%
T3	-3.48%	0.16%	-10.98%	-4.77%
T4	-1.25%	-3.58%	-14.66%	-6.50%
Camera Mean	-4.21%	-1.00%	-8.80%	**-4.67%**

TABLE III: Minimization analysis results. Area reductions obtained with the robot control algorithm based on the link-based method are given for each trajectory and each point of view as percentage of the area covered by the robot when no optimization was applied.

The influence of the camera configuration on the minimization performance can be attributed to a combination of factors, including the robot task space location and the extrinsic camera parameters. For instance, Camera 2 was positioned laterally and orthogonally to the task space of the robot. Hence, its pose made the achievement of effective minimization along the trajectory challenging. Furthermore, in such configuration, the link section dimensions could also be less negligible.

In the specific case T3 with Camera 2, we observed a negligible average increase in the exposed robot area (+0.16%) compared to the baseline. This lack of improvement can be attributed to the nature of the algorithm as an online optimization method, which tries to optimize the robot pose in a real-time manner without knowledge of future desired



(a) (b) (c)

Fig. 5: In (a), (b), and (c), the desired end-effector trajectories T2, T3, and T4 used for the minimization analysis. It is worth noticing that T1 and T2 have the same path, but with different time duration.

	Position	**Orientation**		
	$[m \times 10^{-3}]$	$[rad \times 10^{-3}]$		
	$\|\cdot\|_2$	roll	pitch	yaw
Baseline	0.74	0.7	1.8	0.6
Camera 1	0.87	1.47	1.58	1.03
Camera 2	0.96	1.21	1.59	1.26
Camera 3	0.62	0.82	1.48	0.72

TABLE IV: Average end-effector position and orientation errors among the trajectories for each camera compared to the baseline without the minimization algorithm. Values are calculated as the root mean square error (RMSE) of the Euclidean distance for position, and as RMSE of roll, pitch, and yaw components for orientation.

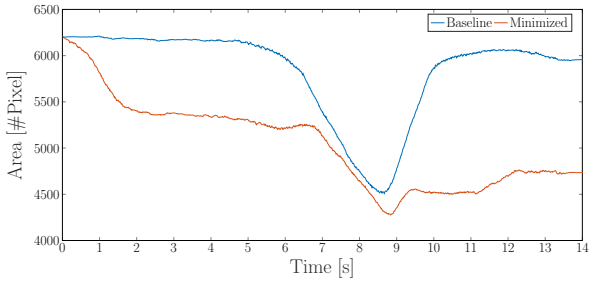


Fig. 6: Representative trial from the minimization analysis. The red and blue lines report the number of pixels containing portions of the robot obtained with and without the action of the robot control algorithm based on the link-based method, respectively. The robot is moving along T4 and the scene is recorded from Camera 3.

positions. Consequently, it initiated a reconfiguration that minimized the area during the first time intervals. However, this subsequently resulted in the robot assuming a less advantageous posture, followed by a conservative strategy with no further improvements. Additionally, the method discourages the robot from operating too close to its joint limits. Therefore, in scenarios where the desired trajectory approaches these limits, the algorithm encourages the robot to exploit its null space to move away from the limits rather than prioritizing the reduction of the robot area.

IV. Conclusions and Future Work

We presented a novel methodology aimed at minimizing the visual footprint of a robotic arm as captured by a camera. This approach holds significant implications, not only for enhancing realism within the Physical Metaverse, but also in critical domains such as teleoperation and telemanipulation. Indeed, users often face occlusion problems when manipulating robots, due to the spatial arrangement of the environment or the presence of the robot body obstructing the user field of view. This issue is especially pertinent for individuals with disabilities who rely on robots to mitigate their limitations, as they may face difficulties in adjusting their own position or that of the robot.

Here, we adopted an online control solution based on the local optimization of an objective function in the null space of the Jacobian matrix. The cost function estimates the number of pixels of the robot projection on the camera image plane, approximating the area of the robot to the sum of the lengths of the segmented and projected links.

In contrast to approaches based on multiple or controllable cameras, the proposed method only relies on the kinematic control of the robotic arm, thus it can be easily adapted to various setups and scenarios. For instance, other cameras or different robotic manipulators can be used simply by updating the corresponding parameters. Additionally, the cost function explicitly depends on the camera pose, which allows for minimization of the robot area visible from the point of view even when the latter is not fixed and may change during the execution of the trajectory.

The experimental evaluation revealed the effectiveness of our control algorithm and some limitations that will be addressed in future work. In forthcoming studies, we intend to explore an objective function that incorporates the three-dimensional characteristics of the links and considers the boundaries of the image plane, possibly conducting an experimental validation on a real robot. Additionally, we will focus on selecting regions or objects within the image that should experience minimal occlusion. This approach will achieve a more precise and effective minimization, strategically positioning the robot to partially exit the camera field of view.

REFERENCES

[1] H. Ning, H. Wang, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand, "A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges," *IEEE Internet of Things Journal*, 2023.

[2] A. Villani, G. Cortigiani, B. Brogi, N. D'Aurizio, T. Lisini Baldi, and D. Prattichizzo, "Avatarm: an avatar with manipulation capabilities for the physical metaverse," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 626–11 632.

[3] M. Kamezaki, J. Yang, H. Iwata, and S. Sugano, "Visibility enhancement using autonomous multicamera controls with situational role assignment for teleoperated work machines," *Journal of Field Robotics*, vol. 33, no. 6, pp. 802–824, 2016.

[4] F. Chaumette, S. Hutchinson, and P. Corke, "Visual servoing," *Springer handbook of robotics*, pp. 841–866, 2016.

[5] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, "Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 796–803, 2018.

[6] C.-S. Chung, H. Wang, and R. A. Cooper, "Functional assessment and performance evaluation for assistive robotic manipulators: Literature review," *The journal of spinal cord medicine*, vol. 36, no. 4, pp. 273–289, 2013.

[7] L. Almeida, P. Menezes, and J. Dias, "Interface transparency issues in teleoperation," *Applied Sciences*, vol. 10, no. 18, p. 6232, 2020.

[8] B. Vagvolgyi, W. Niu, Z. Chen, P. Wilkening, and P. Kazanzides, "Augmented virtuality for model-based teleoperation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3826–3833.

[9] D. Rakita, B. Mutlu, and M. Gleicher, "An autonomous dynamic camera method for effective remote teleoperation," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 325–333.

[10] G. Flandin, F. Chaumette, and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 3. IEEE, 2000, pp. 2741–2746.

[11] F. Cosco, C. Garre, F. Bruno, M. Muzzupappa, and M. A. Otaduy, "Visuo-haptic mixed reality with unobstructed tool-hand integration," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 1, pp. 159–172, 2012.

[12] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2442–2447.

[13] R. Moccia and F. Ficuciello, "Autonomous endoscope control algorithm with visibility and joint limits avoidance constraints for da vinci research kit robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 776–781.

[14] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of intelligent and robotic systems*, vol. 3, pp. 201–212, 1990.

[15] A. Liegeois *et al.*, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE transactions on systems, man, and cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

[16] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[17] ——, "Dynamic manipulability of robot manipulators," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 1033–1038.

[18] F. Chen, M. Selvaggio, and D. G. Caldwell, "Dexterous grasping by manipulability selection for mobile manipulator with visual guidance," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1202–1210, 2018.

[19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2010.

[20] A. Sengupta, "Generalized variance," *Encyclopedia of statistical sciences*, vol. 3, no. 1, 2004.

[21] N. Kulkarni, "Color thresholding method for image segmentation of natural images," *International Journal of Image, Graphics and Signal Processing*, vol. 4, no. 1, p. 28, 2012.

[22] P. Schober, C. Boer, and L. A. Schwarte, "Correlation coefficients: appropriate use and interpretation," *Anesthesia & analgesia*, vol. 126, no. 5, pp. 1763–1768, 2018.

[23] K. Hoang-Dinh, M. Leibold, and D. Wollherr, "A fast and close-to-optimal receding horizon control for trajectory generation in dynamic environments," *Robotics*, vol. 11, no. 4, p. 72, 2022.