2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

Skeleton-based Adaptive Visual Servoing for Control of Robotic Manipulators in Configuration Space

Abhinav Gandhi[1], Sreejani Chatterjee [1], and Berk Calli[1]

***Abstract*— This paper presents a novel visual servoing method that controls a robotic manipulator in the configuration space as opposed to the classical vision-based control methods solely focusing on the end effector pose. We first extract the robot's shape from depth images using a skeletonization algorithm and represent it using parametric curves. We then adopt an adaptive visual servoing scheme that estimates the Jacobian online relating the changes of the curve parameters and the joint velocities. The proposed scheme does not only enable controlling a manipulator in the configuration space, but also demonstrates a better transient response while converging to the goal configuration compared to the classical adaptive visual servoing methods. We present simulations and real robot experiments that demonstrate the capabilities of the proposed method and analyze its performance, robustness, and repeatability compared to the classical algorithms.**

I. INTRODUCTION

Vision-based control algorithms have been utilized for robotic manipulators to accomplish various positioning, navigation and manipulation tasks [1]–[4]. Defining the control error using visual information allows to measure and incorporate task-relevant information which is useful for robots operating in unstructured environments; closing the control loop in the task space enables the robot to account for errors in the robot model and calibration [1]. Adaptive control schemes are also applied to vision-based control, allowing to estimate, in an online manner, the relation between corresponding visual features and the robot motion, and controlling the robot even without knowing the robot Jacobian a priori [5]–[7].

Our work focuses on image-based visual servoing (IBVS) for eye-to-hand arrangements [8], in which the robot motion is controlled via an external camera observing the robot. Such algorithms in the literature utilize image features corresponding to the end effector of the robot, and servo the robot so that these features converge to the desired goal locations (obtained from a goal image). Accordingly, these algorithms only focus on the control of robot's end effector pose. In this work, we propose a novel visual servoing algorithm that can control the robot in the configuration space by modeling the robot's shape ('skeleton'), fitting a parametric curve to it, and servoing the robot so that the the curve representing the current robot shape converges to the curve representing the goal configuration (Fig. 1). Specifically, we use the skeletonization algorithm in [9], model the robot skeleton via B-splines [10], and design an adaptive visual servoing algorithm that provides convergence to the goal configuration while estimating the Jacobian between the robot motion and the change in curve features in an online manner. To the best of our knowledge, this algorithm is the first one in literature that allows visual servoing of robotic manipulators in their configuration space.

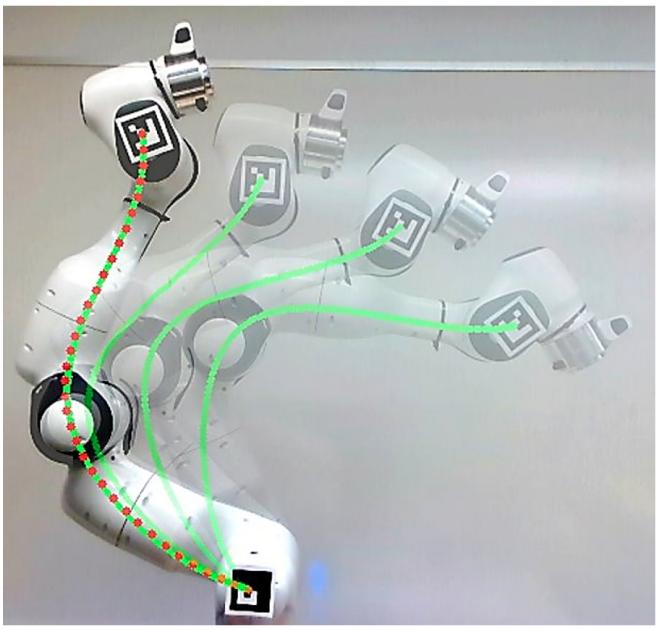

Fig. 1. The proposed skeleton-based adaptive visual servoing method is utilized to make the robot's modeled skeleton (shown with the green curves) converge to a goal configuration (shown with a red dashed curve).

Being able to control the robot in configuration space brings several advantages:

1) Resolving Configuration Ambiguities: We are able to define goal poses and control the robot for specific configurations that correspond to the same end-effector position and/or orientation. For example, such an algorithm can specifically choose to servo the robot to an elbow-up, elbow-down, shoulder-right, or shoulder-left configuration of the same end-effector pose. The classical visual servoing algorithms do not have this ability and would servo to the goal position in a configuration that is closest to the initial configuration of the robot.

2) Better transient response: We experimentally observed that our adaptive visual servoing approach performs better than classical adaptive visual servoing schemes as it employs

*This paper was supported in part by the National Science Foundation under grant IIS-1900953.

[1]Abhinav Gandhi, Sreejani Chatterjee, and Berk Calli are with Department of Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA agandhi2@wpi.edu, schatterjee@wpi.edu

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9981159

more visual information regarding the robot motion.

3) Better robustness: Typically, visual servoing algorithms operate in the kinematics level: essentially, they are velocity reference generators that rely on an inner joint space velocity control loop for executing these references [8]. The more accurate this inner control loop is, the better the visual servoing performance becomes. The accuracy of the inner control loop is especially important for the adaptive visual servoing algorithms to converge to a Jacobian that relates the image feature motions to the robot motion. We experimentally show that our skeleton-based algorithm provides a better transient response in the face of inner control loop deficiencies. While such robustness is important for all visual servoing applications, it would be even more crucial for compliant (low impedance) robots, since accurately controlling them is challenging.

4) Leveraging redundancy: Since our algorithm can servo the robot to a given goal configuration, it can fully utilize the redundancy of the robot. This would especially be useful for controlling highly redundant robots or soft robotic manipulators such as [11], allowing them to avoid obstacles or navigate in complex environments. Classical visual servoing approaches do not have this ability, since they only consider the end effector pose of the robot.

The Scope and Assumptions of This Paper:

In this work, we focus on 2D implementations of the skeleton-based visual servoing. We provide a thorough analysis for the capabilities, robustness, and repeatability of the proposed algorithm. Specifically, we provide results demonstrating the algorithm's ability to control the robot to different configurations (while also showing its singularities), perform repeated experiments to show its repetability, and present its performance in low impedance case. We also compare these aspects to a classical adaptive visual servoing algorithm. While several 3D skeletonization algorithms exist in the literature [12]–[14] and the methodology has a potential to be expanded to the 3D case, we leave the implementation and analysis of that for the future work.

Our inspiration stems from the success of the skeleton-based human pose tracking algorithms in the literature [15]–[17]. Recent machine learning-based algorithms provide real time pose tracking that is robust to external occlusions (by other objects in the scene) and self-occlusions. In this work, we use an algorithmic skeletonization method [9] and assume that there is no occlusion in the scene. Nevertheless, we believe that adopting the skeleton-based human-pose tracking algorithms to the robots has a great potential to achieve robot configuration control that is robust to occlusions. This aspect is discussed in the last section.

II. Background

Apart from traditional visual servoing approaches [1], [8] that rely on the knowledge of the robot Jacobian, adaptive algorithms such as [18]–[20] estimate the image and robot Jacobian online and require minimum prior knowledge of the system's kinematics. All these methods describe image errors for the end-effector and consequently assert control in the task space of the robotic manipulator.

Using curve features from observed objects is another popular practice for visual servoing. In [21], an image Jacobian is derived for the polar description of an object's contour. Points on the viewed object's boundary are used as features for servoing the robot to its target location. The target location is defined by target image positions for each point on the object boundary. This allows the use of unknown objects in the robot's workspace for the visual servoing task. This approach is found computationally inefficient due to the large number of features used. In [22] shapes are represented by algebraic curves. The curve is decomposed into a sum of product of line factors which are used to derive image features. A point feature interaction matrix is deployed in the control law. Similarly, in [23] the authors use Bezier and Non-Uniform Rational B-Spline (NURBS) curves to model the object's contour and derive image features for an eye-in-hand adaptive IBVS scheme. These methods use the eye-in-hand setup that servos the robot relative to the observed object. To the best of our knowledge our algorithm is the only approach that utilizes robot skeleton and servos the robot in the configuration space using an eye-to-hand arrangement.

Recently, there has been interest in deforming soft bodies into target shapes using visual servoing techniques. Authors in [24] show how shape features can be used to estimate online an adaptive deformation model for a soft body that is deformed with a robotic manipulator. This adaptive model is then applied in the IBVS control law to deform the soft body into a target shape. In [25], B-Splines are used to represent the shape of deformable wires. The deformation matrix is adaptively estimated online to deform the wire into its target shape. These works provide us inspiration to design an approach that utilizes shape representations to control the robot in its configuration space.

III. Skeleton-based Adaptive Visual Servoing

The overall skeleton-based adaptive control scheme is provided by the diagram in Fig. 2. The feature extraction and control system design aspects are explained and discussed in this section.

A. Obtaining Skeletons and Corresponding Features

In this work, we propose to use a representation of robot's shape for visual servoing in configuration space. While there are various options that can be utilized for representing the shape of the robot, e.g. contours [26] and volumetric methods [27], we chose to use a skeleton-based representation, which is more compact compared to these methods, but still gives us the required information about the robot's configuration. In our implementation, we use an RGB-D camera, and first segment the robot's silhouette by distance filtering the depth image. The skeleton is then obtained by applying medial axis transform and thinning methods to this silhouette as in [9]. The extracted skeleton is a single pixel wide curve which closely approximates the robot's shape. As an example,

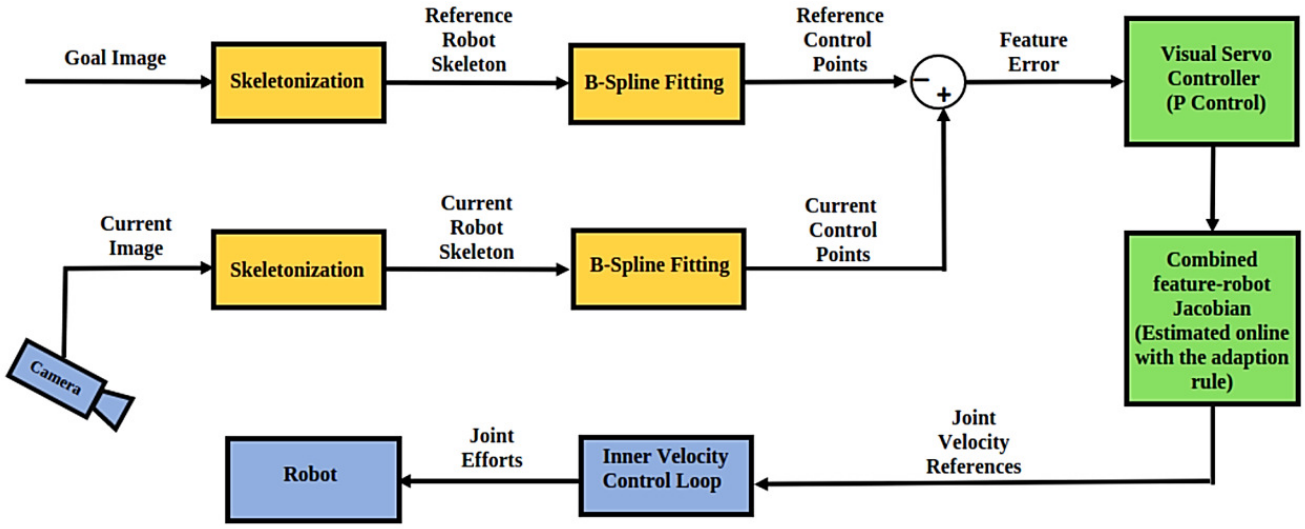

Fig. 2. Block diagram of the Skeleton-based Adaptive Visual Servoing algorithm. Orange boxes are related to obtaining skeletons and the related features. Green blocks comprise the adaptive vision-based control rule. Blue blocks are representing the camera, the robot and the robot's controller.

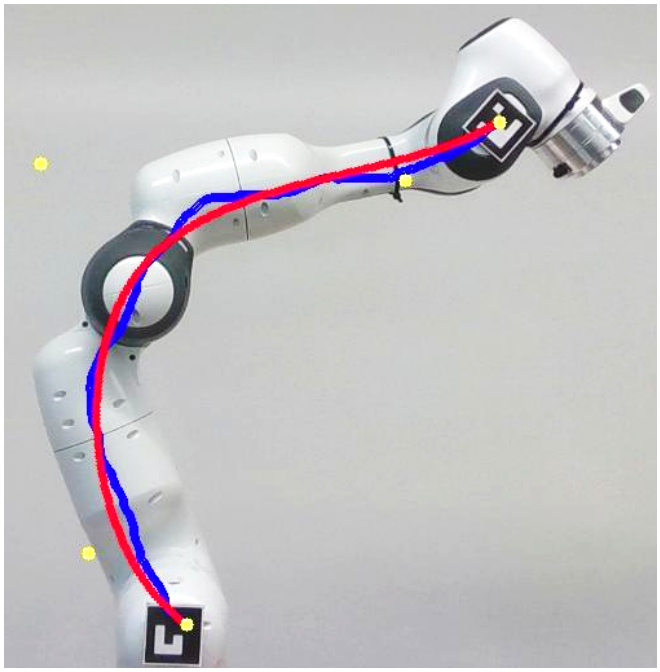

Fig. 3. Skeleton (blue curve) and approximated B-spline (red curve) for the robot configuration shown in the image. The yellow points denote control points of the curve.

Fig. 3 shows the robot's skeleton with a blue curve for the given configuration after the skeletonization algorithm is applied (the thickness of the skeleton is exaggerated for visibility in the figure).

We then generate a mathematical model of this curve. Among the implicit, explicit, and parametric methods for curve modelling, the parametric methods are the most suitable type for our case: Implicit and explicit representations (in the form of $y - f(x) = 0$ and $y = f(x)$ respectively in the 2D case) require to have a single x value for each y value of the skeleton pixels, which significantly limits the robot configurations that can be represented. The parametric methods model the $x(t)$ and $y(t)$ coordinates of the skeleton separately with a common t parameter, and therefore, do not limit the possible configurations in the robot's workspace. Among the parametric curve modelling options, we use Normalized B-splines [10] due to the following reasons. B-splines constitute multiple polynomial segments that are linearly combined with smoothness constraints. This piecewise nature of B-splines helps our method represent complex shapes (configurations) with a collection of relatively low degree polynomials. The individual polynomial pieces are determined by a control polygon formed by control points. As such, the control points uniquely represent the shape of the B-spline that is used to approximate the robot skeleton [10]. We use these control points as features in our visual servoing scheme.

It is also important to note that B-splines provide a spatial curve model, which allows to represent robot configurations in 3D space. Furthermore, they are invariant to translation, rotation, scaling and perspective projection, and therefore there is a one-to-one correspondence between the features (control points) in 3-D space and the image plane [23], [28]. These properties enable using B-splines and their control points to servo the robot in 3-D space using image-based visual servoing. While all the implementations in this paper are 2D planar visual servoing, we believe that utilizing B-splines will allow us to extend our implementation to 3D configurations in the future.

Fig. 3 shows the B-spline curve in red, which is fitted to the skeleton of the robot in blue. In this figure, we model the skeleton using a B-spline with two segments, each of which are parameterized by third degree polynomials. This B-spline is represented by five control points which are shown in yellow color.

B. Adaptive Visual Servoing

As mentioned in Section III-A, we use control points of the B-spline (that is fitted to the skeleton of the robot) as features for visual servoing. Our feature error vector $\boldsymbol{e}$ is defined in the image space as the difference between the goal feature vector $\boldsymbol{c^*}$, which contains image co-ordinates of the control points c_i^* for the robots desired configuration, and the current feature vector $\boldsymbol{c}$, which contains the image co-ordinates of the control points c_i in the robots current configuration. Here, $\boldsymbol{c}$ and $\boldsymbol{c^*} \in R^{2p}$ where p is the number of control points used to compute $\boldsymbol{e}$ as shown in (1). Each control point c_i and c_i^* (in $\boldsymbol{c}$ and $\boldsymbol{c^*}$ respectively) $\in R^2$.

$$\boldsymbol{e} = \boldsymbol{c} - \boldsymbol{c^*} \tag{1}$$

This image error is utilized in a vision-based control loop in [8] as follows:

$$\boldsymbol{v_c} = -\lambda J_{im}^{+} \boldsymbol{e}, \tag{2}$$

where $\boldsymbol{v_c}$ is the desired velocity for the control points in Cartesian space, which would make the current control points converge to their goal locations in the acquired image. J_{im}^{+} is the pseudo inverse of the image Jacobian. This desired Cartesian space velocity is converted to desired joint velocities, $\dot{\boldsymbol{q}}_{\boldsymbol{r}} \in R^m$ (m is the number of controllable joints in the robot), using the robot Jacobian, J_r, as follows:

$$\dot{\boldsymbol{q}}_{\boldsymbol{r}} = J_r^{+} \boldsymbol{v_c} \tag{3}$$

Combining eq. (2) and eq. (3) results in

$$\dot{\boldsymbol{q}}_{\boldsymbol{r}} = -\lambda J_r^{+} J_{im}^{+} \boldsymbol{e} \tag{4}$$

In our adaptive visual servoing scheme, we estimate the combined image-robot Jacobian, $J_c^{+} = J_r^{+} J_{im}^{+}$. This estimation is performed by using the last n collected samples of the input joint velocities to the robot ($\dot{Q} \in R^{n \times m}$) and observed feature velocities from the acquired images ($\dot{C} \in R^{n \times 2p}$) as follows:

$$\dot{Q} = [\dot{\boldsymbol{q}}[k-n+1],\ \dot{\boldsymbol{q}}[k-n+2],\ \ldots,\ \dot{\boldsymbol{q}}[k]]^T \tag{5}$$

$$\dot{C} = [\dot{\boldsymbol{c}}[k-n+1],\ \dot{\boldsymbol{c}}[k-n+2],\ \ldots,\ \dot{\boldsymbol{c}}[k]]^T \tag{6}$$

Here, k is the iteration number of the control loop, and n is also called the "window size" of the adaptation. Let's call the estimated combined Jacobian $\hat{J}_c$. The adaptive updates are performed for each row of $\hat{J}_c$ as shown in [24]:

$$\Delta \hat{J}_{c_i}[k] = \gamma \dot{Q}^T (\dot{Q} J_{c_i}[k] - \dot{C}) \tag{7}$$

$$\hat{J}_{c_i}[k+1] = \Delta \hat{J}_{c_i}[k] + \hat{J}_{c_i}[k] \tag{8}$$

In these equations, $\hat{J}_{c_i}[k]$ is the i^{th} row of the combined Jacobian at instance k, and γ is the gain of the adaptive process. Essentially this update rule calculates the difference between the expected and measured feature displacements, and projects it as a corrective term after scaling it with γ.

This adaptive scheme requires to be initialized by collecting the first n samples of $\dot{\boldsymbol{c}}$ and $\dot{\boldsymbol{q}}$. This is achieved by providing an initial trajectory to the robot and recording these values. The window size is an important design parameter for an adaptive scheme: A large window size makes the system robust to sharp changes in velocity, however it increases the time required for the Jacobian to converge. A small window size may make the system more vulnerable to measurement noise and may cause sudden changes in robot velocity.

IV. Simulations and Experiments

In this section, we analyze the capabilities of the skeleton-based adaptive visual servoing, and compare its performance to the classical adaptive visual servoing scheme in [19].

A. Configuration Space Control:

A capability of the skeleton-based scheme is to control the robot in configuration space. We demonstrate this capability via simulations with a two-link planar robot in Gazebo simulator. Fig. 4 presents two examples of servoing the robot to two different configurations (elbow-left and elbow-right) that correspond to the same end-effector positions. It is important to note that this is beyond the capabilities of classical image-based visual servoing schemes, since they only utilize the features on the end effector. These algorithms would converge to a given end effector pose with the closest configuration to the initial one. Thanks to the skeleton representation that provides features to the control algorithm all along the robot's shape, we can define goals and achieve convergence for different configurations.

We also observed that the algorithm sometimes gets stuck in local minima as presented in Fig. 5. These local minima are observed to be at the axial symmetry of the desired configuration. We believe that this phenomenon might be a limitation of our curve modeling technique, and will explore various different modeling methods to avoid the local minima in the configuration space. Nevertheless, we believe these minima can be overcome by providing intermediate goal configurations between the initial and final configurations.

We also observed that both successful convergence and local minima cases are repeatable; the system followed very similar trajectories for each initial-goal configuration pair for five consecutive runs.

B. Controlling a Redundant Manipulator

Proof of concept experiments for the capability of the skeleton-based algorithm to leverage the redundancy of the robot is given in Fig. 6. In this example, we run simulations with a 3 DOF planar robot. The skeleton-based algorithm was able to make the system converge to the desired configurations utilizing all 3 joint motions.

C. Performance Comparison:

We run experiments with a real robot in order to compare the performance of the skeleton-based adaptive approach to the classical adaptive visual servoing algorithms. We utilized a Franka Emika Panda robot along with an Intel RealSense

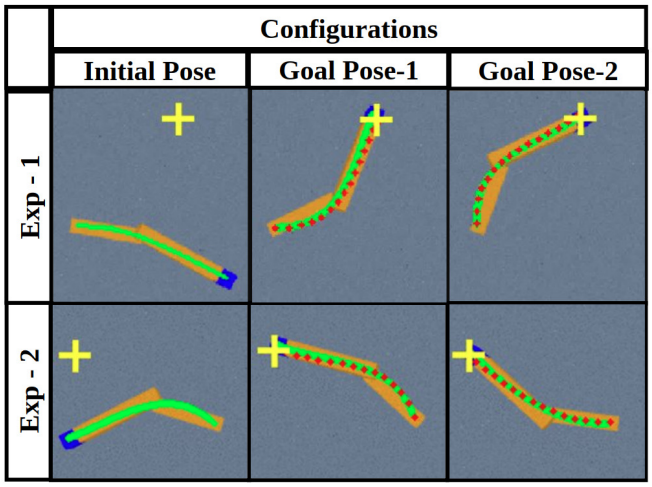

Fig. 4. Configuration space control for the same goal position (as shown by the yellow "+" markers) and initial configuration. Results shown for two experiments. Left panels: Initial configuration of the robot. Middle panels: Robot at the target with elbow right configuration. Right panels: Robot at the target with elbow left configuration. Red dotted curves are the provided targets, and green curves are the modeled skeleton of the robot.

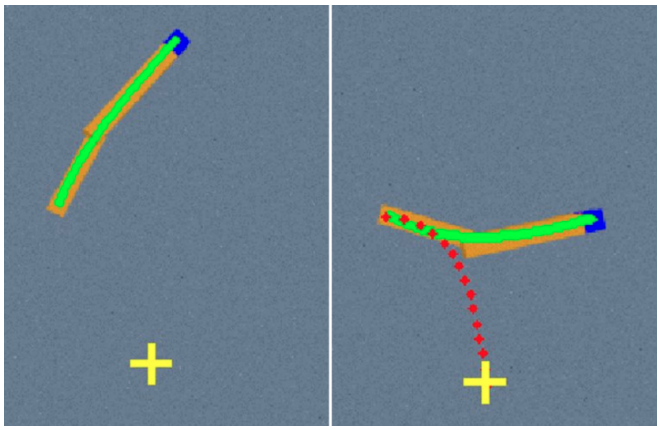

Fig. 5. Singularity in the curve space.

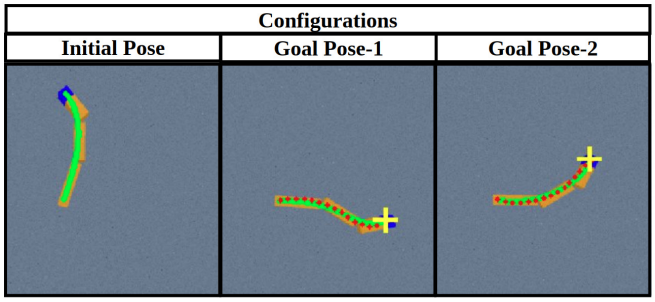

Fig. 6. Skeleton-based visual servoing is also suitable for leveraging the redundancy of the robot. Experiments with a 3 DOF planar robot.

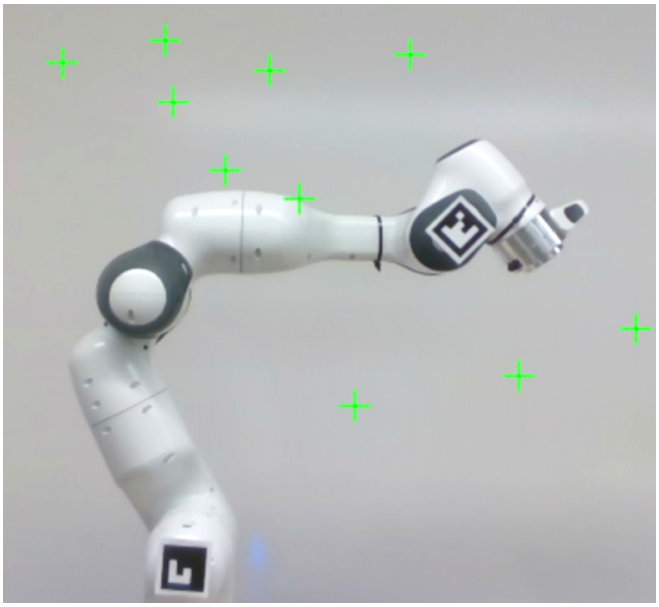

Fig. 7. Experimental setup and distribution of target poses for the experiments.

d435i depth sensor to acquire visual feedback. These experiments are constrained to the planar case and we use two links of the robot. We use Aruco markers for simplifying the image processing as follows: In the implementation of the classical adaptive visual servoing, we track the pixel position of the marker center and utilize it in the vision-based control loop. For our skeleton-based algorithm, we use the markers for constraining the two ends of the extracted skeleton.

We performed 20 experiments with each of the algorithms for two different initial positions and the 10 targets that corresponds to the marked end-effector positions in Fig. 7 (For the skeleton-based algorithm, reference curves that correspond to these target locations are provided).

Prior to conducting the experiments, both algorithms are tuned with a target location that has an end-effector image error of 100 pixels in each axis of the image plane. The estimation window size (n in eqs. (5) and (6)) is set to 50, and a control loop frequency of 10 Hz is used. Both algorithms are provided with the same velocity profile for collecting the first 50 samples. The controller gains, λ, and adaptation gain γ are extensively tuned for the best performance for each algorithm while ensuring that the overshoot in end-effector position errors are less than 5% and the steady state errors are within $\pm 2\%$ of the target position.

The mean and standard deviation for rise time, overshoot and settling time of 20 experiments for each algorithm is given in Table I. This data shows that our algorithm provides faster convergence with less overshoot. We believe that the higher performance is due to utilizing more information regarding the motion of the robot via the shape representation during the online adaptation. Fig. 8 provides the end-effector trajectory for the skeleton-based algorithm in yellow. The red dotted curve represents the target configuration and the green curve represents the final configuration at the end of the experiment. For this experiment, the end-effector error and normalized feature error is provided in fig. 10, the joint velocities are shown in fig. 9, and the model-error for estimation of the combined Jacobian, J_c, is given in Fig. 11

(our sampling and control frequency is 10 Hz, and therefore 10 iterations = 1 second). The first 50 iterations of the plot represent the model error during the initial estimation period, while the rest of the iterations represent the model-error during the control phase. We stop updating our estimation of the interaction matrix once the robot is close to the target configuration. The algorithm provides smooth decay of the error with close-to-optimum trajectory even when the Jacobian error does not converge to zero.

TABLE I
PERFORMANCE COMPARISON SUMMARY

	Baseline IBVS	**Proposed algorithm**
Rise time (s)	9.025 ± 3.316	6.870 ± 3.164
Overshoot (%)	4.83 ± 0.095	2.14 ± 0.044
Settling time (s)	15.585 ± 4.570	$11.232 \pm 4.029^{*}$

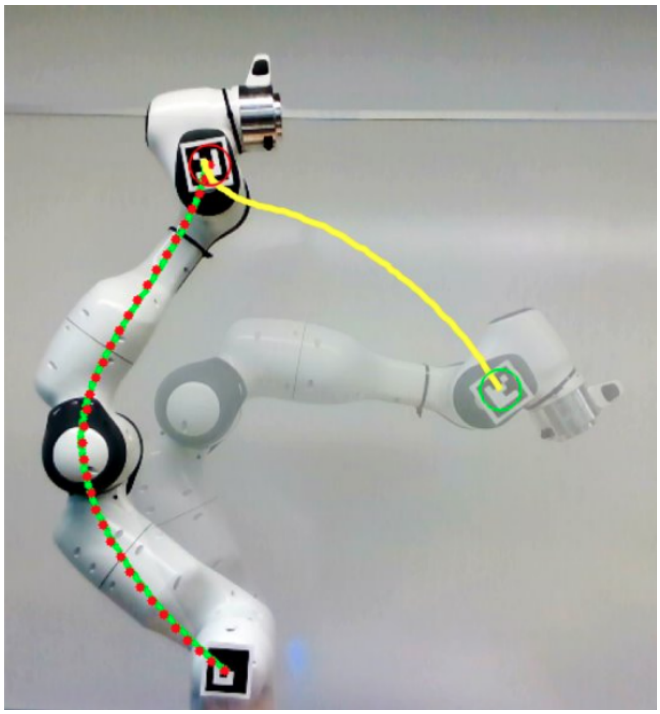

Fig. 8. End effector trajectory (as denoted by the yellow curve) plot for our proposed algorithm. The green curve shows the modeled skeleton for the current image and the red dotted curve shows the desired modeled skeleton curve.

D. Repeatability:

A repeatability test is conducted for the skeleton-based adaptive algorithm and the classical adaptive algorithm by running 5 trials for the same initial and reference positions. The results are summarized in Table II. In general, the transient response results look quite consistent for both the classical adaptive algorithm and our proposed algorithm. We notice one trial with a larger overshoot for our novel algorithm and hence the much larger mean overshoot percentage.

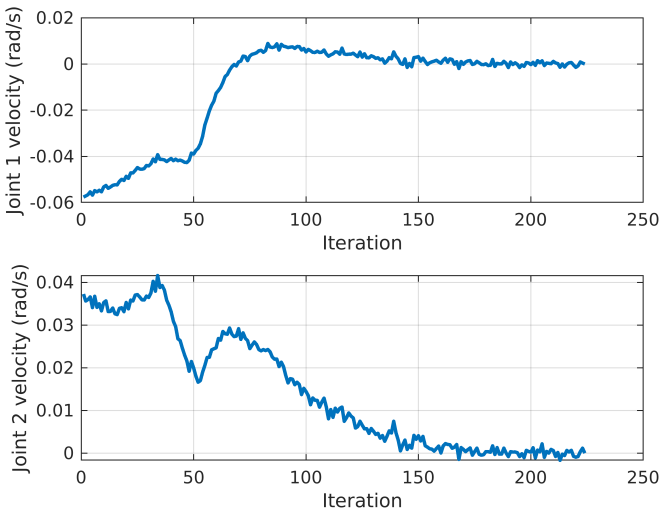

Fig. 9. Applied joint velocities.

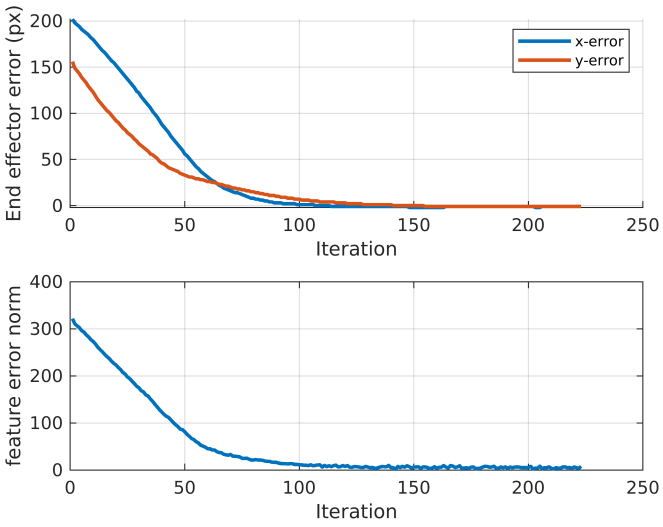

Fig. 10. End-effector error (above) and feature error norm (below).

E. Robustness:

As summarized in the introduction section, visual servoing algorithms typically operate in the kinematic level, and they rely on accurate inner loop velocity controllers. The accuracy of this inner loop controller is especially crucial for the adaptive visual servoing algorithms to achieve convergence. However, there can be many cases, where an accurate inner loop controller is not available, e.g. low-impedance robots, soft robots. To simulate these conditions, we decrease the joint stiffness of the Franka Emika manipulator by a factor of 20. This significantly reduces the tracking accuracy of the low-level controllers and adds undesired dynamics for the visual servo controllers. We perform a total of eight experiments with two target positions for each algorithm. For each target, we run two trials. First, with the robot's stiffness set to maximum. This serves as a benchmark experiment and the second, with the reduced robot stiffness. Figure 12 shows the experiment trajectories of the end effector. The images

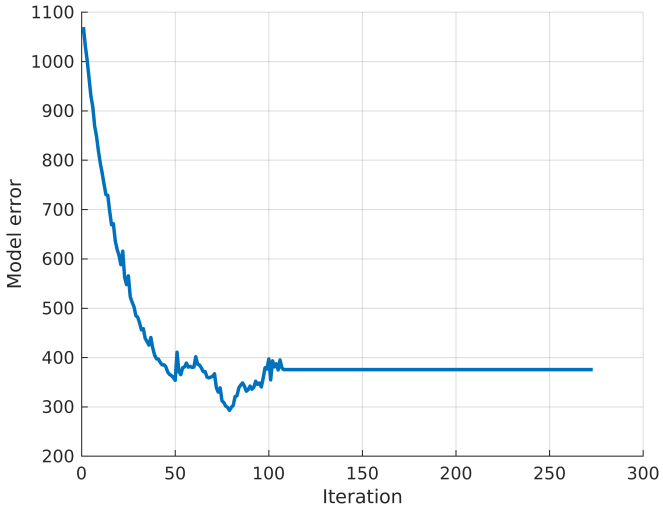

Fig. 11. Model-error for the estimation of interaction matrix

TABLE II
SUMMARY OF REPEATABILITY TESTS

	Baseline IBVS	Proposed algorithm
Rise time (s)	7.26 ± 0.17	3.620 ± 0.08
Overshoot (%)	0.33 ± 0.00	1.37 ± 0.01
Settling time (s)	14.74 ± 0.26	7.90 ± 0.28

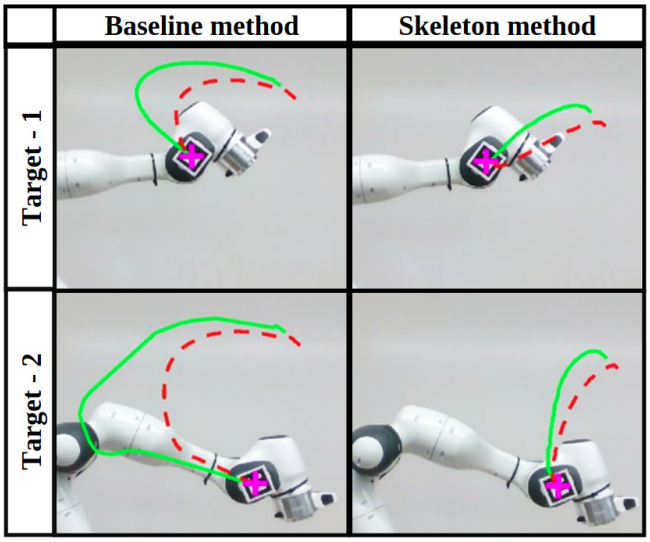

Fig. 12. End-effector trajectories for two targets. From the left: target 1 - a)baseline algorithm, b)proposed algorithm; target 2 - c)baseline algorithm, d)proposed algorithm. Benchmark trajectory shown in red and trajectory for reduced joint stiffness shown in green.

are cropped to focus on the region of interest. We can see that trajectories with reduced stiffness (shown in green) for our proposed algorithm are significantly closer to their respective benchmarks (shown in red) than for the baseline algorithm. This robustness to the system dynamics can be attributed to the shape features used to estimate the image Jacobian for our novel algorithm.

V. CONCLUSION & FUTURE WORK

In this paper, we show how skeletonization and curve fitting techniques can be leveraged to obtain a representation of the robot's shape, which then can be used for vision-based control purposes. We proposed an adaptive scheme that estimates the relation between the changes in the robot's curve representation and the input joint velocities in an online manner. We provide simulation and experimental results that demonstrate the capabilities of our algorithm and compare it's performance to the classical adaptive visual servoing approach. Our algorithm shows promising results with faster convergence, minimal overshoot and low steady state errors. Additionally, it provides the benefit of configuration space control which is not possible with classical visual servoing algorithms.

We believe that this work unlocks new possibilities for vision-based control of robotic manipulators and poses important new research questions. We are particularly interested in adopting newer and faster skeletonization methods such as skeleton-based human pose tracking techniques [15]–[17] for robot configuration tracking to achieve accurate shape extraction while being robust to occlusions. The skeletonization method used in our approach is the bottleneck for our algorithm currently and using faster methods that are robust to noise and occlusions in the acquired depth images would significantly improve controller performance. The extension of our approach to the configuration control in 3D space is of our interest. We expect our current representation to scale well in terms of spatial modelling capabilities. However, with redundant robots in 3D space we expect other issues such as self-occlusion. We would like to consider the following options for circumventing these problems: **a)** using multiple views of the robot to reconstruct its skeleton, **b)** using an approximate physical model of the robot to augment the information provided by vision sensor. Avoiding local minima of the curve representation would also boost the performance of the algorithm. For this purpose, we are planning to try other parametric curve modeling techniques. Currently, our algorithm is unable to validate a given target configuration and we assume that the target configurations provided are within the robot's workspace and valid. The ability to generate valid targets and intermediate configurations between the initial and final configurations is also expected to be an effective solution for avoiding local minima and unreachable target configurations. Another interesting direction would be to utilize the skeleton-based adaptive visual servoing in soft robot control. Since we use normalized B-splines, we do not expect degradation in control performance due to changes in length of the robot's skeleton as in soft robots. Skeletons are expected to provide very effective representations of such robots, and leverage their redundancy for avoiding obstacles, compensating for unwanted deformations in the robot shape, and navigating complex environments.

REFERENCES

[1] F. Chaumette, S. Hutchinson, and P. Corke, "Visual servoing," in *Springer Handbook of Robotics*. Springer, 2016, pp. 841–866.

[2] Y. Zhao, L. Gong, Y. Huang, and C. Liu, "A review of key techniques of vision-based control for harvesting robot," *Computers and Electronics in Agriculture*, vol. 127, pp. 311–323, 2016.

[3] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, "A survey on vision-based uav navigation," *Geo-spatial information science*, vol. 21, no. 1, pp. 21–32, 2018.

[4] M. Hussein, "A review on vision-based control of flexible manipulators," *Advanced Robotics*, vol. 29, no. 24, pp. 1575–1585, 2015.

[5] A. Astolfi, L. Hsu, M. S. Netto, and R. Ortega, "Two solutions to the adaptive visual servoing problem," *IEEE transactions on robotics and automation*, vol. 18, no. 3, pp. 387–392, 2002.

[6] H. Wang, B. Yang, J. Wang, X. Liang, W. Chen, and Y.-H. Liu, "Adaptive visual servoing of contour features," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 811–822, 2018.

[7] J. Pomares, F. Chaumette, and F. Torres, "Adaptive visual servoing by simultaneous camera calibration," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 2811–2816.

[8] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[9] T. C. Lee, R. L. Kashyap, and C. N. Chu, "Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms," *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 6, pp. 462–478, 11 1994.

[10] C. de Boor, "On Calculating with B-Splines," *Journal of Approximation Theory*, vol. 6, no. 1, pp. 50–62, 1972.

[11] J. Santoso and C. D. Onal, "An origami continuum robot capable of precise motion through torsionally stiff body and smooth inverse kinematics," *Soft Robotics*, vol. 8, no. 4, pp. 371–386, 2021.

[12] S. Tran and L. Shih, "Efficient 3d binary image skeletonization," in *2005 IEEE Computational Systems Bioinformatics Conference-Workshops (CSBW'05)*. IEEE, 2005, pp. 364–372.

[13] Ó. G. Hernández, V. Morell, J. L. Ramon, and C. A. Jara, "Human pose detection for robotic-assisted and rehabilitation environments," *Applied Sciences*, vol. 11, no. 9, p. 4183, 2021.

[14] Y. Yoshiyasu, R. Sagawa, K. Ayusawa, and A. Murai, "Skeleton transformer networks: 3d human pose and skinned mesh from single rgb image," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 485–500.

[15] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

[16] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, "Semantic graph convolutional networks for 3d human pose regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3425–3435.

[17] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4929–4937.

[18] A. Shademan, A. M. Farahmand, and M. Jägersand, "Robust Jacobian estimation for uncalibrated visual servoing," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5564–5569, 2010.

[19] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true Jacobian," in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, 2002, pp. 186–193.

[20] T. Y. Chang, W. C. Chang, M. Y. Cheng, and S. S. Yang, "Dynamic visual servoing with Kalman filter-based depth and velocity estimator," *International Journal of Advanced Robotic Systems*, vol. 18, no. 3, may 2021. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/17298814211016674

[21] C. Collewet and F. Chaumette, "A contour approach for image-based control on objects with complex shapes," in *IEEE International Conference on Intelligent Robots and Systems*, 2000, pp. 751–756.

[22] A. Y. Yazicioglu, B. Calli, and M. Unel, "Image based visual servoing using algebraic curves applied to shape alignment," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2009, pp. 5444–5449. [Online]. Available: http://ieeexplore.ieee.org/document/5354310/

[23] H. Wang, B. Yang, J. Wang, X. Liang, W. Chen, and Y. H. Liu, "Adaptive Visual Servoing of Contour Features," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 811–822, 2018.

[24] D. Navarro-Alarcon and Y. H. Liu, "Fourier-Based Shape Servoing: A New Feedback Method to Actively Deform Soft Objects into Desired 2-D Image Contours," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 272–279, 2018.

[25] R. Lagneau, A. Krupa, and M. Marchal, "Automatic Shape Control of Deformable Wires Based on Model-Free Visual Servoing," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5252–5259, 2020.

[26] A. Pourramezan Fard, H. Abdollahi, and M. Mahoor, "Deep active shape model for face alignment and pose estimation," *arXiv e-prints*, pp. arXiv–2103, 2021.

[27] X. Luo, B. Berendsen, R. T. Tan, and R. C. Veltkamp, "Human pose estimation for multiple persons based on volume reconstruction," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 3591–3594.

[28] L. Piegl and W. Tiller, "The NURBS book," *Choice Reviews Online*, vol. 35, no. 02, pp. 35–0952–35–0952, 1997. [Online]. Available: http://link.springer.com/10.1007/978-3-642-97385-7