DOI: 10.3217/978-3-85125-663-5-17

Object Grasping in Non-metric Space Using Decoupled Direct Visual Servoing

Bernhard Neuberger[1], Geraldo Silveira[2], Marko Postolov[3] and Markus Vincze[4]

Abstract— **In this paper we present a robotic system for grasping novel objects. Using a low-cost camera mounted on the end-effector, our system utilizes visual servoing control to command the gripper to a grasp position that is prescribed during a teach-in phase when the object is presented to the system. By using decoupled direct visual servoing, an intensity-based approach, object grasping is done without any 3D input and requires no metric information about the object. Although the robot moves in the 3D Euclidean space and is controlled in the joint space, the command signal is derived completely from pixel information from the input image in the 2D projective space. Furthermore, the control strategy is extended for trajectory following in the control error space to generate smoother and more stable trajectories. This enables more direct and accurate positioning of the end-effector. A set of experiments is performed with a 7 DoF KUKA LWR IV robotic arm and shows the capability of precisely grasping objects from cluttered scenes. The system also shows robustness to object movement during the grasping process as well as robustness against errors in the camera calibration.**

I. INTRODUCTION

Robotic arms are widespread in industrial environments for production tasks and are capable of precise sub millimeter positioning. In order to exploit such high precision the system is required to perceive the environment with similar accuracy.

One common task in robotics is grasping, which works well when considering controlled conditions but becomes increasingly more challenging when its necessary to adapted to an changing environment.

The focus in this work is on the task of grasping objects with a robotic arm. The robotic system is equipped with a low cost camera that provides the perception. The described approach is capable of grasping objects that are newly presented to the robot and only require a teach-in phase to generate a single reference image.

State-of-the-art methods [5], [13], [9] have shown that visual servoing control can stabilize a robotic system around an equilibrium space. Here the work from Silveira et. al. [15] is extended such that the visual servoing approach is used for the task of grasping. The considered objects have a planar and textured surface that will be visible to the robot during the task of grasping.

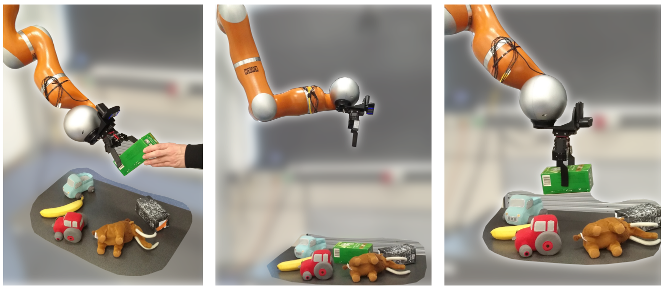


Fig. 1: Teaching the robot to grasp a newly presented object.

In order to use visual servoing for grasping the task was broken down to an end-effector positioning. Therefore first experiments were conducted were it is shown that the system is able to position itself in regards to a reference image. This approach is extended in such a way that the robot is guided along a desired trajectory to the final pose. For this guided movement a desired control error is predetermined so that the initial control error is gradually reduced. This enables a direct smooth movement and is used to grasp objects in a cluttered environment without colliding with the surroundings of the target object. For first results on grasping the clutter was placed such that it was guaranteed that the target object was good visible and the desired grasp was reachable.

In the teach-in phase a new object is learned by showing it to the robot. Therefore it is needed to present the object to the camera of the system as it would be seen when grasped by the gripper. We do not require any form of object model or depth information and purely rely on the input from the mounted camera. Figure 1 shows the teach-in phase, the system state when starting the grasping process and the state of the robot after grasping the newly learned object.

Additionally, a number of experiments was conducted which test the system performance in regard to errors in the camera parameters. For this case wrong camera intrinsic parameters, particularly the focal lengths f_x and f_y in the algorithm were disturbed when a positioning task was performed.

The presented system builds on the work from [15] and contributes with the following additional features:

- visual servoing is used as a tool for grasping newly presented objects

*This work was not supported by any organization

[1]Bernhard Neuberger is with Faculty of Electrical Engineering, ACIN, V4R, Technische Universität Wien, 1040 Wien, Austria neuberger@acin.tuwien.ac.at

[2]Geraldo Silveira is with DRVC Division, CTI Renato Archer, CEP 13069-901, Campinas/SP, Brazil Geraldo.Silveira@cti.gov.br

[3]Marko Postolov is with Chair of Automatic Control Engineering, Technische Universität München, 80333, München, Germany marko.postolov@tum.de

[4]Markus Vincze is with Faculty of Electrical Engineering, ACIN, V4R, Technische Universität Wien, 1040 Wien, Austria vincze@acin.tuwien.ac.at

- an easy to use teach-in phase is used that requires a human supervisor to take a single reference image of the object in the desired grasp pose
- a desired error trajectory is presented for smoother movement that enables grasping in clutter
- we show that the approach has high robustness against errors in the camera intrinsic parameters
- object grasping remains successful even when the target is repositioned during the process

II. RELATED WORK

Saxena et. al. [11] present a robotic system that is able to grasp newly seen objects. Their approach calculates corresponding points for grasping the object and then calculates a 3D grasp point from a set of sparse points on the object. In contrary to this work we are not able to grasp completely unseen objects but we need to teach the object to the robot which doesn't require any 3D object information.

Fischinger et. al. [6] present a grasping approach that uses depth data to extract features for grasping objects in piles and cluttered scenes. They present a method that is able to grasp newly presented objects without any object knowledge.

In [7] Levine et. al. show a deep learning approach that learns to move the robot in the task space such that it results in a high probability for a successful grasp. They show that continuous servoing corrects the mistakes from the network and improve the grasp quality. Their method requires a large number of training data compared to a single reference image in our method.

Another deep learning based method for grasping is presented by Mahler et. al. in [8]. They show that their network is able to predict grasp points with a high success rate when trained on a large synthetic dataset. They also present a grasp planner that is needed to position a robot within workspace constraints. In contrary our method directly positions in regards to the target and does not use grasp points at all.

Chaumette and Hutchinson present in [3] and [4] an overview of various state-of-the-art visual servoing approaches. They present control strategies for image-based visual servo control (IBVS) and position-based visual servo control (PBVS). The difference between IBVS and PBVS are examined and stability of the strategies are investigated. PBVS methods such as [17] and [16] have the advantage of having full control over the trajectories in the Cartesian space but have the disadvantage of being sensitive to camera parameters. Our approach is similar to IBVS methods as presented in [2] but compared to them we don't use image features. Instead pixel intensities are used directly as presented in [14] and use it for the specific robotic task of grasping.

In [9] Mariottini et. al. show how to use IBVS to control a nonholonomic mobile robot to a desired pose. Similar to our approach they don't use any metric information and control the robot in the epipolar geometry. Contrary to our method they use extracted features from the images for the IBVS. In our system the visual tracker from [10] is used.

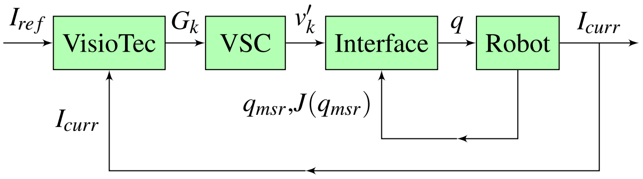


Fig. 2: Conceptional Overview of the Visual Control

Nogueira presents in [10] an intensity-based homography estimation which is implemented in the robot operating system(ROS). In this paper we will refere to this software as the "VisioTec" that is publicly available[1]. It is used to calculate the homography between the reference image and the currently captured camera image.

III. VISUALLY GUIDED OBJECT GRASPING

For our grasping approach, the system perceives the environment with a camera mounted on the end-effector of the robotic arm. In the teach-in phase, the object is presented to the robot in the final grasp pose. This provides the system with an reference image in the precise configuration for the final grasping pose. After this phase, the robot arm automatically moves from any initial pose to a grasp pose such that the live image from the camera coincides with the recorded reference image. This enables the system to grasp the object (by closing the fingers of the end-effector) even if the object is moved at any time during the control procedure.

Our approach enables a set of objects to be grasped without the need for object models. As such, novel objects can be simply presented to the system for immediate grasping. Human involvement is reduced to the task of positioning an object in front of the wrist mounted camera and recording of the reference image.

This section gives an overview of the visual control and details for the steps of the robot control strategy. An extended control strategy for trajectory following in the control error space is then presented. This extension guides the end-effector motion along a coarse trajectory for smooth positioning.

A. Concept Overview

Figure 2 shows a block diagram of the conceptional overview of the robot control loop that is used for the grasping task. The first part in the control loop is the VisioTec visual tracker [10] that takes a selected bounding box from the reference image I_{ref} and performs intensity-based image registration for the tracked area in the current image I_{curr}. The output of the VisioTec tracker is the homography G between the reference target area and the corresponding area in the current image.

This homography is the input to the visual servo controller (VSC), which is described in III-B. This component takes the homography and outputs the desired end-effector velocity commands v'_k.

[1] http://wiki.ros.org/vtec_ros

Our implementation includes an interface between the VSC and the robot control due to the lack of a velocity control mode. The interface transforms the commanded velocity v'_k directly into desired joint states q.

The computed joint states are received by the robot control unit to move the robotic arm to a desired configuration. Additionally, the robot control unit returns the measured joint states q_{msr} and also the current Jacobian matrix $J(q_{msr})$. Within this block $J(q_{msr})$ is used to calculate the pseudo inverse Jacobian matrix J^{+} to calculate the joint velocities according to

$$\dot{q} = J^{+} v'_k. \tag{1}$$

The measured joint states q_{msr} and joint velocities $\dot{q}$ are used to determine the commanded joints q using the sampling time t_Δ

$$q = q_{msr} + \dot{q} t_\Delta. \tag{2}$$

The robot arm moves according to the commanded joint states and captures a different view of the environment. The latest captured image is returned to the VisioTec tracker and closes the control loop.

B. Robot Control

The purpose of the robot control is to transform the homography G_k from the VisioTec to the end-effector velocity v'_k to move the robot arm closer to the reference pose.

The implementation of the robot control requires the template location in the location matrix G_l to be set in terms of pixel coordinates (l_x, l_y). So long as the target is located in the selected area the control error ε_k will be zero. The location matrix G_l, the camera intrinsic parameters $K \in \mathbb{R}^{3\times 3}$ and the hand-eye calibration $T' \in \mathbb{SE}(3)$ are set according to

$$G_l = \begin{bmatrix} 1 & 0 & l_x \\ 0 & 1 & l_y \\ 0 & 0 & 1 \end{bmatrix}, K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, T' = \begin{bmatrix} R' & t' \\ 0 & 1 \end{bmatrix} \tag{3}$$

where f_x and f_y are the focal lengths of the camera and (c_x, c_y) is the principal point coordinate. The homography G_k from the VisioTec tracker is than transformed to the image frame with

$$G'_k = G_k G_l^{-1}. \tag{4}$$

The control point $p^* \in \mathbb{P}^2$ and the control vector $c^{*\prime} \in \mathbb{R}^3$ are used to calculate the control error

$$\varepsilon_k = \begin{bmatrix} 2I & [m^{*\prime}]_\times \\ -[c^{*\prime}]_\times & I \end{bmatrix} \begin{bmatrix} (H-I)m^{*\prime} \\ \vartheta\mu \end{bmatrix} \tag{5}$$

where

$$H = K^{-1} G'_k K, \quad m^{*\prime} = K^{-1} p^* \tag{6}$$

and

$$r = \frac{1}{2}\mathrm{vex}(H - H^\top) \tag{7}$$

$$\vartheta = \begin{cases} \arcsin(||r||) & \text{if} \quad tr(H) \geq 1, \\ \pi - \arcsin(||r||) & \text{otherwise,} \end{cases} \tag{8}$$

$$\mu = \frac{r}{||r||}. \tag{9}$$

The error from the control law is used to calculate the required camera velocity in order to reduce the control error

$$v_k = \lambda(\varepsilon_k)\varepsilon_k. \tag{10}$$

Here $\lambda(\varepsilon_k)$ is a variable gain that was used in our setting. This ensures that the gain declines with an increasing control error and will result in small end-effector velocities with high control error. This reduces velocities exponentially for very high control errors and keeps them within boundaries.

$$\lambda(\varepsilon_k) = \alpha \mathrm{e}^{-\gamma||\varepsilon_k||} \tag{11}$$

The control parameters $\alpha > 0$ and $\gamma > 0$ can be tuned such that a higher α increases the velocity and a higher γ increases the damping of the gain.

Finally, the velocity from the camera frame is transformed to the tool center point (TCP) frame with

$$v'_k = \begin{bmatrix} R' & [t']_\times R' \\ 0 & R' \end{bmatrix} v_k \tag{12}$$

where $[t']_\times$ is the skew symmetric matrix of the translation vector between the camera and TCP frame and R' is the rotation matrix for the transform.

C. Trajectory Following

The control strategy as described above moves the robot to a reference pose. For a more reliable grasping, the system can be adapted for smoother and more stable end-effector movement. This first requires the introduction of a desired control error trajectory $\varepsilon^*(t)$. The actual control error is derived from the error trajectory $\varepsilon^*(t)$ according to

$$\varepsilon'(t) = \varepsilon(t) - \varepsilon^*(t) \tag{10}$$

Stable trajectories are then achieved by adapting the control law from Equation (13) to

$$v_k = \lambda(\varepsilon')\varepsilon'(t) + \frac{\partial \varepsilon^*(t)}{\partial t} \tag{14}$$

Specific details of the desired control error is presented in Section V.

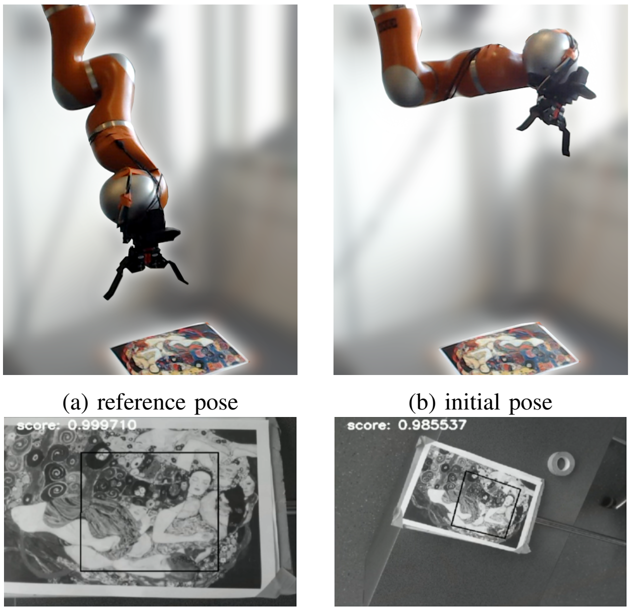
IV. EXPERIMENTAL SETUP

In our experiments, we use a KUKA LWR IV [1] robotic arm with the provided control unit. The robot arm has 7 degrees of freedom (DoF) and is controlled with position commands for the joints. A Logitech HD C920 webcam and a dynamixel AX-12A Dual Gripper are mounted on the end-effector of the arm with a 3D printed support structure.

The VisioTec, the VSC and the interface depicted in the block diagram of Figure 2 run on a remote PC. The remote PC and the KUKA control unit communicate via Ethernet. Communication between the remote PC and the robot control unit is enabled with the kuka-lwr-ros package[2], which uses the fast research interface[3] (FRI) [12].

[2] https://github.com/epfl-lasa/kuka-lwr-ros

[3] https://cs.stanford.edu/people/tkr/fri/html/



(a) reference pose

(b) initial pose

(c) reference image

(d) initial image

Fig. 3: Positioning Experiment

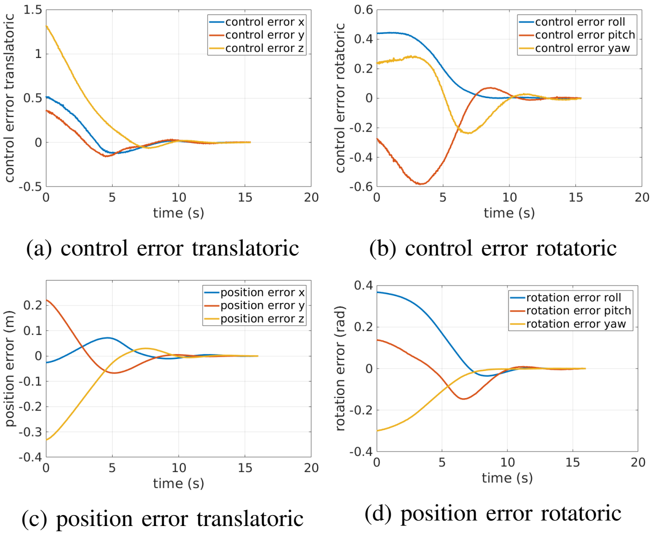
V. RESULTS

A number of different experiments were executed to evaluate the performance of our approach. First, we perform positioning experiments to establish a base line to evaluate the benefit of the trajectory following adaptation. In these experiments, the robot arm is tasked to position itself with respect to a known planar textured surface. Second, we perform grasping performance to showcase the capability of the system to grasp novel objects in clutter. The grasping task is restricted to rigid textured objects with a planar surface, where for each experiments the object is presented to the system in its final grasping pose. Finally, we conduct experiments to test the robustness under camera parameter errors.

A. Positioning Experiment

For this experiments, a plain textured image is used as target. After a reference image is recorded from a reference pose the robot arm is moved to an initial pose. The goal is to control the robot arm in such a way that it moves back to the reference pose while reducing the control error. Figure 3 shows the setup of the positioning experiment with the robot arm in the reference pose and initial pose. The reference image with the selected target area and the initial image with the tracked target area are also shown.

For the evaluation, the robot state is recorded along the whole trajectory from the initial pose to the final pose. Figure 4 plots the pose and control errors over the duration of the experiment. We can see that the errors reduce and eventually go to zero after 15 seconds. Although the final pose is reached, the movement shows unintended behaviour as seen by the significant overshoot in Figure 4c and Figure 4d.



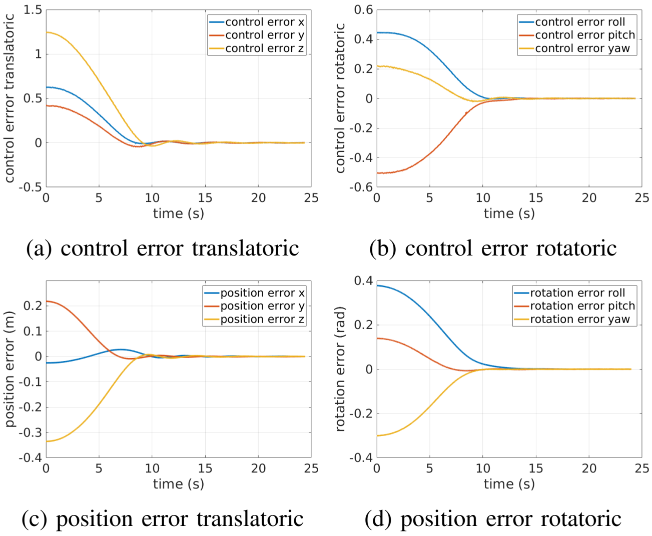
(a) control error translatoric

(b) control error rotatoric

(c) position error translatoric

(d) position error rotatoric

Fig. 4: Results from the positioning experiment



(a) control error translatoric

(b) control error rotatoric

(c) position error translatoric

(d) position error rotatoric

Fig. 5: Results from the trajectory following experiment

B. Trajectory Following

The desired control error is set in such a way that the initial error is gradually reduced along a smooth function with the property of being continuously differentiable. A Lipschitz continuous desired control error is made possible by setting

$$\varepsilon^* = \begin{cases} \varepsilon_0(1+2(\frac{t}{T})^3 - 3(\frac{t}{T})^2) & t \le T \\ 0 & t > T \end{cases} \tag{15}$$

where ε_0 is the initial control error from the initial end-effector pose. This choice of control error guarantees a Lipschitz continous commanded velocity.

Figure 5 shows the pose and control errors with trajectory following activated. Compared to Figure 4, we see that the control error is now reduced in a more controlled way, which results in a more direct and smooth movement. It can be seen in Figure 5c that the pose error is gradually reduced with less overshoot along the x-axis compared to Figure 4c.

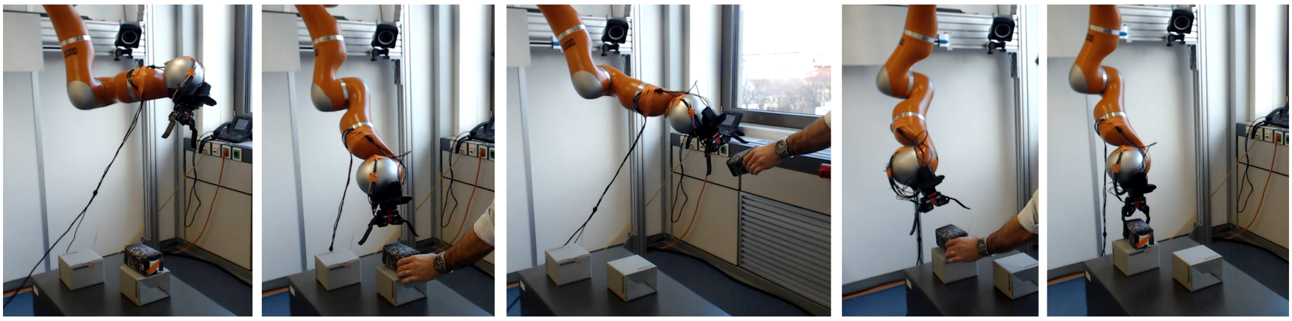
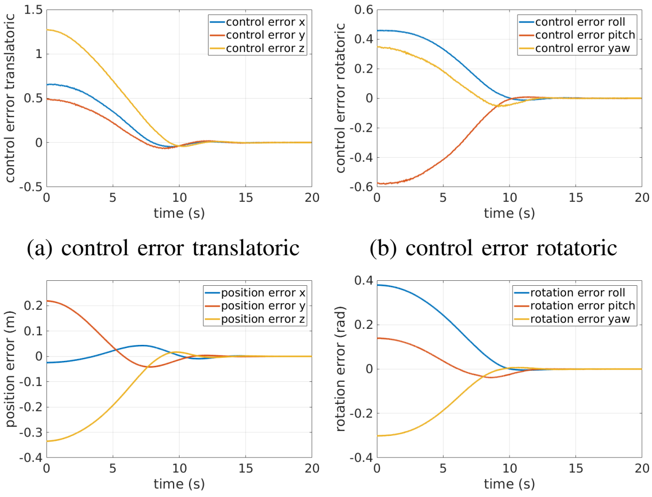


Fig. 6: Grasping an object while repositioning the target object



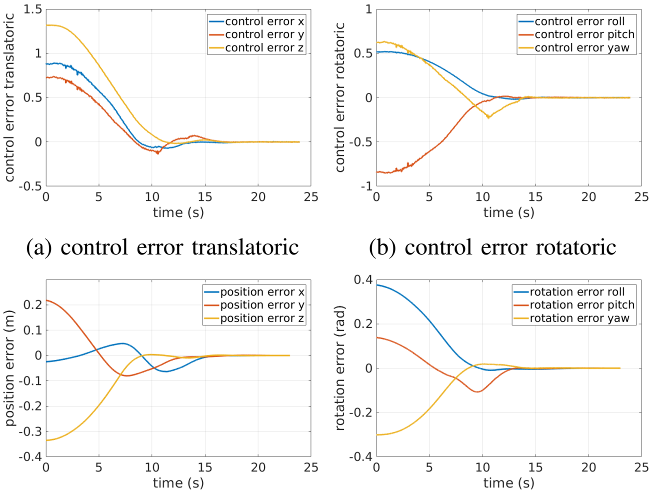
(a) control error translatoric

(b) control error rotatoric

(c) position error translatoric

(d) position error rotatoric

Fig. 7: Results from the trajectory following with 37.40% errors in camera parameters



(a) control error translatoric

(b) control error rotatoric

(c) position error translatoric

(d) position error rotatoric

Fig. 8: Results from the trajectory following with 113.74% errors in camera parameters

C. Object Grasping

The setting from the trajectory following experiments were used to grasp objects. Qualitative results are presented in Figure 6. This shows not only a successful grasp but also the fact that the system successfully compensates for the movement of the target during the process of positioning to the final pose. So long as the target object remains in the camera's field of view the robot is able to successfully follow the target. Grasping is achieved by closing the fingers of the gripper when the pose error is below a predifined threshold.

These results show that our visual servoing approach can be used to grasp objects in unpredictable conditions. Further experiments show that the inclusion of trajectory following enables grasping in cluttered scenes, where collisions are possible near the target object, i.e. obstacles sitting on the same plane as the target object. As shown in Figure 1, grasping is successful as the gripper moves directly towards the object without colliding with surrounding objects.

D. Robustness against errors in the Camera Parameters

The positioning experiment is repeated with an added error in the focal lengths f_x and f_y if the camera intrinsic matrix K.

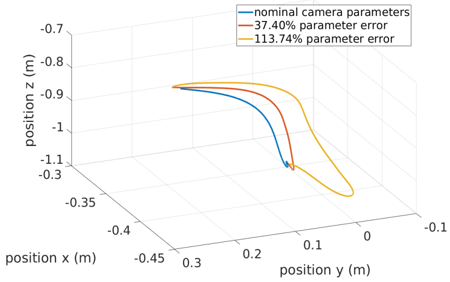


Fig. 9: End-effector trajectory for different camera parameters

In order to obtain the nominal parameters K, a checkerboard and the ROS camera calibration package[4] is used.

Figure 7 shows the pose and control error over time with an error of 37.40% in the focal length of the camera parameters and Figure 8 shows the same plots with 113.74% of error. The results show that even with 113.74% error the robot still manages to reach the reference pose. Further increase in the error of the camera parameters results in a

[4] http://wiki.ros.org/camera_calibration

failed experiments because the target shifts out of the field of view. In this case the reference image can longer be tracked.

Although the reference pose is still reached with a very high error the resulting end-effector trajectory is not feasible for grasping experiments. Figure 9 shows the trajectories of the end-effector from the initial pose to the the reference pose with different settings of the camera parameters. We can see that the trajectories with the nominal parameters and 37.40% error are similar and follow a very direct path. But it is visible that the trajectory with 113.74% error of camera parameters follows a more complicated path that would result in a collision with surrounding objects.

VI. Conclusion

The results show that a reliable grasp is possible with our method even if we add errors to the camera intrinsic parameters. We show that newly taught objects can be tracked and grasped with the system.

Future work will exploit the redundancy of the KUKA LWR IV. The 7 DoF of the robot arm has one additional degree of freedom compared to the workspace of the robot. This can be used to avoid singularities, joint limits or keeping distance between the joints and obstacles.

Further plans will improve the system in such a way that the robot can detect the target in a newly presented image even if the target is completely lost in between. This can be beneficial for a mobile platform which can exploit the room and than plan the object manipulation accordingly. This would allow a robot to grasp or manipulate previously learned objects in an novel environment.

References

[1] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, *et al.*, "The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing," in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE, 2010, pp. 1–8.

[2] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*. Springer, 1998, pp. 66–78.

[3] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[4] ——, "Visual servo control. ii. advanced approaches [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[5] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 507–515, 2001.

[6] D. Fischinger, M. Vincze, and Y. Jiang, "Learning grasps for unknown objects in cluttered scenes," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 609–616.

[7] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[8] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[9] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 87–100, 2007.

[10] L. Nogueira, E. de Paiva, and G. Silveira, "VISIOTEC intensity-based homography optimization software: Basic theory and use cases," CTI, Brazil, Tech. Rep. CTI-VTEC-TR-01-2017, 2017.

[11] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[12] G. Schreiber, A. Stemmer, and R. Bischoff, "The fast research interface for the kuka lightweight robot," in *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)*. Citeseer, 2010, pp. 15–21.

[13] G. Silveira, "On intensity-based nonmetric visual servoing," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 1019–1026, 2014.

[14] G. Silveira and E. Malis, "Direct visual servoing: Vision-based estimation and control using only nonmetric information," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 974–980, 2012.

[15] G. Silveira, L. Mirisola, and P. Morin, "Decoupled intensity-based nonmetric visual servo control," *IEEE Transactions on Control Systems Technology*, 2018.

[16] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: keeping the object in the field of vision," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1624–1629.

[17] W. J. Wilson, C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.