

Autonomous Dual-Arm Manipulation of Familiar Objects

Dmytro Pavlichenko, Diego Rodriguez, Max Schwarz, Christian Lenz, Arul Selvam Periyasamy and Sven Behnke

Abstract—Autonomous dual-arm manipulation is an essential skill to deploy robots in unstructured scenarios. However, this is a challenging undertaking, particularly in terms of perception and planning. Unstructured scenarios are full of objects with different shapes and appearances that have to be grasped in a very specific manner so they can be functionally used. In this paper we present an integrated approach to perform dual-arm pick tasks autonomously. Our method consists of semantic segmentation, object pose estimation, deformable model registration, grasp planning and arm trajectory optimization. The entire pipeline can be executed on-board and is suitable for on-line grasping scenarios. For this, our approach makes use of accumulated knowledge expressed as convolutional neural network models and low-dimensional latent shape spaces. For manipulating objects, we propose a stochastic trajectory optimization that includes a kinematic chain closure constraint. Evaluation in simulation and on the real robot corroborates the feasibility and applicability of the proposed methods on a task of picking up unknown watering cans and drills using both arms.

I. INTRODUCTION

Daily-life scenarios are full of objects optimized to fit anthropometric sizes. Thus, human-like robots are the natural solution to be used in quotidian environments. In these scenarios, many objects require two or more grasping affordances in order to be manipulated properly. Such objects may have complex shapes involving multiple degrees of freedom (DOF), be partially or completely flexible or simply be too large and/or heavy for single-handed manipulation, for instance, moving a table and operating a heavy power drill.

In this paper, we describe an integrated system capable of performing autonomous dual-arm pick tasks. Such tasks involve the consecutive accomplishment of several sub-tasks: object recognition and segmentation, pose estimation, grasp generation, and arm trajectory planning and optimization. Each of these subproblems is challenging in unstructured environments when performed autonomously—due to the high level of uncertainty coming from noisy or missing sensory measurements, complexity of the environment, and modeling imperfection. Thus, designing and combining software components which solve these sub-problems into one integrated pipeline is challenging.

We use semantic segmentation to detect the object. A segmented point cloud is then passed to the next step of the



Fig. 1. The Centauro robot performing bimanual grasping of a novel watering can.

pipeline: deformable model registration and grasp generation. Since instances of the same object category are similar in their usage and geometry, we transfer grasping skills to novel instances based on the typical variations of their shape. Intra-classes shape variations are accumulated in a learned low-dimensional latent shape space and are used to infer new grasping poses.

Finally, we optimize the resulting trajectories of the grasp planner by applying a modified version of Stochastic Trajectory Optimization for Motion Planning (STOMP) [1], which we refer to as STOMP-New [2]. We extend our previous work by adding an additional cost component to preserve the kinematic chain closure constraint when both hands hold an object. For typical human-like upper-body robots, the dual-arm trajectory optimization problem with closure constraint is a non-trivial task due to curse of dimensionality and severe workspace constraints for joint valid configurations. We perform experiments to investigate the influence of the new constraint on the performance of the algorithm.

The main contribution of this paper is the introduction of a complete software pipeline capable of performing autonomous dual-arm manipulation. The pipeline was demonstrated with the Centauro robot [3]. Even though the robot base is quadruped, the upper-body is anthropomorphic with a torso, two arms, and a head. We evaluate the capabilities of the designed system on the dual-arm pick task in simulation and on the real robot (Fig. 1).

II. RELATED WORK

Robotic systems which perform dual-arm manipulation are widely used for complex manipulation tasks. Many of such systems are applied in industrial scenarios. For instance,

All authors are with the Autonomous Intelligent Systems (AIS) Group, Computer Science Institute VI, University of Bonn, Germany. Email: pavlichenko@ais.uni-bonn.de. This work was supported by the European Union's Horizon 2020 Programme under Grant Agreement 644839 (CENTAURO) and the German Research Foundation (DFG) under the grant BE 2556/12 ALROMA in priority programme SPP 1527 Autonomous Learning.

Krüger *et al.* [4] present a dual arm robot for an assembly cell. The robot is capable of performing assembly tasks both in isolation and in cooperation with human workers in a fenceless setup. The authors use a combination of online and offline methods to perform the tasks. Similarly, Tsarouchi *et al.* [5] allow dual arm robots to perform tasks, which are usually done manually by human operators in a automotive assembly plant. Stria *et al.* [6] describe a system for autonomous real-time garment folding. The authors introduce a new polygonal garment model, which is shown to be applicable to various classes of garment. However, none of the previously mentioned works present a complete and generic pipeline, [4] and [5] do Stria *et al.* [6] was proposed a very specific and limited use-case. To the best knowledge of the authors, there are no significant recent works, which present a complete autonomous robotic system for dual-arm manipulation. In the following subsections we briefly review some of the noticeable works for each of the core components of our pipeline.

A. Semantic Segmentation

The field of semantic segmentation experienced much progress in recent years due to the availability of large datasets. Several works showed good performance using complex models that require extensive training on large data sets [7], [8]. In contrast, in this work we use a transfer learning method that focuses on fast training, which greatly increases the flexibility of the whole system [9].

B. Transferring Grasping Skills

Vahrenkamp *et al.* [10] transfer grasp poses from a set of pre-defined grasps based on the RGB-D segmentation of an object. The authors introduced a transferability measure which determines an expected success rate of the grasp transfer. It was shown that there is a correlation between this measure and the actual grasp success rate. In contrast, Stouraitis *et al.* [11] and Hillenbrand and Roa [12] warp functional grasp poses such that the distance between point correspondences is minimized. Subsequently the warped poses are replanned in order to increase the functionality of the grasp. Those methods can be applied only in off-line scenarios, though, because of their large execution time. The method explained here, on the other hand, is suitable for on-line scenarios.

C. Dual-Arm Motion Planning

Dual-arm motion planning is a challenging task, for which intensive research has been carried out. Szykiewicz and Błaszczyk [13] proposed an optimization-based approach to path planning for closed-chain robotic systems. The path planning problem was formulated as a function minimization problem with equality and inequality constraints in terms of the joint variables. Vahrenkamp *et al.* [14] presented two different approaches for dual-arm planning: J^+ and IK-RRT. Although the first one does not require an inverse kinematics (IK) solver, IK-RRT was shown to perform better

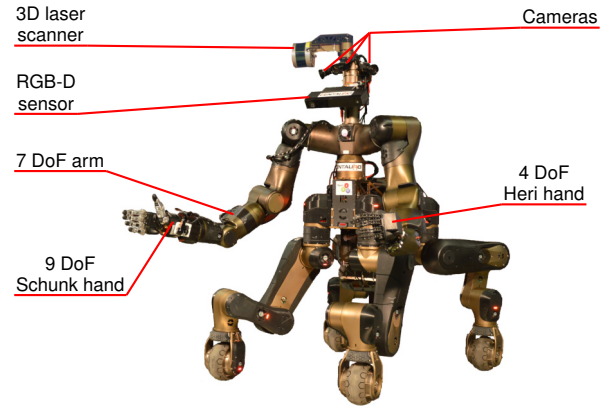


Fig. 2. The Centauro robot. The main components of the upper-body are labeled.

on both single and dual-arm tasks. In contrast, a heuristic-based approach was proposed by Cohen *et al.* [15]. The method relies on the construction of a manipulation lattice graph and an informative heuristic. Even though the success of the search depends on the heuristic, the algorithm showed good performance in comparison with several sampling-based planners. Byrne *et al.* [16] proposes a method consisting of goal configuration sampling, subgoal selection and Artificial Potential Fields (APF) motion planning. It was shown that the method improves APF performance for independent and cooperative dual-arm manipulation tasks. An advantage of our approach to arm trajectory optimization is the flexibility of the prioritized cost function which can be extended to support different criteria, which we demonstrate in this work.

III. SYSTEM OVERVIEW

In this work we test our software pipeline on a centaur-like robot, developed within the CENTAURO project¹. The robot has a human-like upperbody, which is mounted on the quadrupedal base. It is equipped with two anthropomorphic manipulators with 7 DOF each. The right arm possesses a SVH Schunk hand as an end-effector, while the left arm is equipped with a Heri hand [17]. The sensor head has a Velodyne Puck rotating laser scanner with spherical field of view as well as multiple cameras. In addition, a Kinect v2 [18] is mounted on the upper part of the chest. The Centauro robot is depicted in Fig. 2.

In order to perform an autonomous dual-arm pick tasks we propose the following pipeline (Fig. 3):

- *Semantic Segmentation* performed by using RGB-D data from the Kinect v2,
- *Pose Estimation* on the resulting segmented point cloud,
- non-rigid *Shape Registration* to obtain grasping poses,
- and finally, *Trajectory Optimization* to obtain collision-free trajectories to reach pre-grasp poses.

IV. PERCEPTION

For perceiving the object to be manipulated, a state-of-the-art semantic segmentation architecture [7, RefineNet]

¹<https://www.centauro-project.eu/>

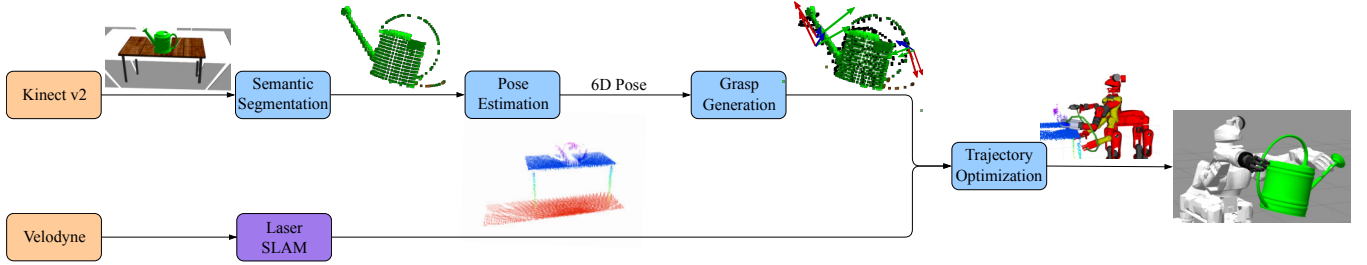


Fig. 3. Simplified diagram of the system, showing the information flow between core components. Orange: sensors; Blue: main components of the pipeline; Purple: external modules.

is trained on synthetic scenes. Those are composed of a small number of captured background images which are augmented randomly with inserted objects. This approach follows Schwarz *et al.* [9] closely, with the exception that the inserted object segments are rendered from CAD meshes using the open-source Blender renderer. The core of the model consists of four ResNet blocks. After each block the features become more abstract, but also lose the resolution. So, the feature maps are upsampled and merged with the map from the next level, until the end result is at the same time high-resolution and highly semantic feature map. The final classification is done by a linear layer followed by a pixel-wise SoftMax.

At inference time, also following Schwarz *et al.* [9], we postprocess the semantic segmentation to find individual object contours. The dominant object is found using the pixel count and is extracted from the input image for further processing.

The 6D pose of the object is estimated as follows: the translation component is computed by projecting the centroid of the object contour into 3D by using the depth information; the orientation component is calculated from the principle components on the 3D object points of the object and incorporating prior knowledge of a canonical model defined for each category. This initial pose estimate is refined by the shape space registration described in Sec. V-A.

V. MANIPULATION PLANNING

A. Grasp Planning

The grasp planning is a learning-based approach that exploits the fact that objects similar to each other can be grasped in a similar way. We define a category as a set of models with related extrinsic geometries. In the training phase of the method, a shape (latent) space of the category is built. This is done by computing the deformation fields of a canonical model C towards the other models in the category. This is carried out by using the Coherent Point Drift (CPD) non-rigid registration method. CPD provides a dense deformation field, thus new points can be warped even after the registration. Additionally, the deformation field of each object in the training set can be expressed in a vector whose dimensionality equals the number of points times the number of dimensions of the canonical model. This means that the variations in shape from one object to the other

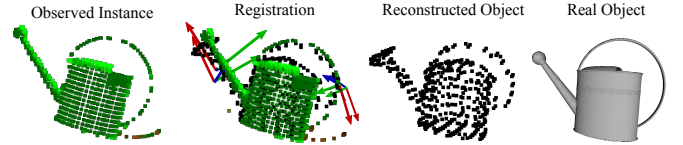


Fig. 4. Shape space registration on the watering can category. The method is able to reconstruct a partially occluded instance containing noise.

can be expressed by a vector of the same length across all training samples. Thus, subspace methods can be straightforwardly applied. Finally, the principal components of all these deformation fields are calculated by using Principal Component Analysis - Expectation Maximization (PCA-EM) which define the basis of the shape space.

Once the shape space is constructed, new instances can be generated by interpolating and extrapolating in the subspace. In the inference phase, we search in the latent space in a gradient-descent fashion for an instance which relates to the observed model at best. We do this by optimizing a non-linear function that minimizes a weighted point distance. An additional rigid registration is also incorporated in the cost function to account for misalignments. Furthermore, the latent variables are regularized which has shown to provide numerical stability. Once the descriptor in the latent space is known, it is transformed back to obtain the deformation field that best describes the observation. In this process, partially occluded shapes are reconstructed. The registration is robust against noise and misalignments to certain extent [19]. Fig. 4 shows a partially observed instance with noise and the reconstructed object after the shape registration.

The canonical model has associated control poses that describe the grasping motion. These control poses are warped using the inferred deformation field. More details about the shape space registration can be found in [20]. For bimanual manipulation we associate individual grasping control frames to each arm and warp them according to the observed model. Because each of the control poses is independent, simultaneous arm motions are possible. The control poses contain the pre-grasp and final grasp poses.

B. Trajectory Optimization

The grasp planner provides pre-grasp poses for both arms, the trajectory optimizer plans a collision-free trajectory to reach them. We use STOMP-New, which showed better

performance in previous experiments [2]. It has a cost function consisting of five cost components: collisions, joint limits, end-effector orientation constraints, joint torques and trajectory duration. The input is an initial trajectory Θ which consists of N keyframes $\theta_i \in \mathbb{R}^J$, $i \in \{0, \dots, N-1\}$ in joint space with J joints. Normally, a naïve linear interpolation between the given start and goal configurations θ_{start} and θ_{goal} is used. The start and goal configurations are not modified during the optimization.

Since the optimization is performed in joint space, extending the algorithm to use two arms instead of one is straightforward. We extended the approach to support multiple end-effectors (two in the context of this work), so trajectories of two independent arms are simultaneously optimized.

However, for moving an object grasped with two hands, a kinematic chain closure constraint has to be satisfied. Thus, the following term $q_{cc}(\cdot, \cdot)$ is added to the cost function:

$$q(\theta_i, \theta_{i+1}) = q_o(\theta_i, \theta_{i+1}) + q_l(\theta_i, \theta_{i+1}) + q_c(\theta_i, \theta_{i+1}) + q_d(\theta_i, \theta_{i+1}) + q_t(\theta_i, \theta_{i+1}) + q_{cc}(\theta_i, \theta_{i+1}), \quad (1)$$

where $q(\theta_i, \theta_{i+1})$ is a cost for the transition from the configuration θ_i to θ_{i+1} . The cost function now consists out of six terms, where the first five are coming from our original implementation of STOMP-New. By summing up costs $q(\cdot, \cdot)$ of the consecutive pairs of transitions θ_i, θ_{i+1} of the trajectory Θ , we obtain the total cost.

The new term $q_{cc}(\cdot, \cdot)$ for the kinematic chain closure constraint is formulated as:

$$q_{cc}(\theta_i, \theta_{i+1}) = \frac{1}{2} \max_j q_{ct}(\theta_j) + \frac{1}{2} \max_j q_{co}(\theta_j), j \in \{i, \dots, i+1\} \quad (2)$$

where $q_{ct}(\cdot)$ penalizes deviations in translation between the end-effectors along the transition and $q_{co}(\cdot)$ penalizes deviations of the relative orientation of the end-effectors.

Given two end-effectors, ee_{f1} and ee_{f2} , the initial translation $t_{desired} \in \mathbb{R}^3$ between them is measured in the first configuration θ_0 of the trajectory. Then, for each evaluated configuration θ_j , the corresponding translation t_j between ee_{f1} and ee_{f2} is computed. The deviation from the desired translation is thus defined as: $\delta t = |t_{desired} - t_j|$. Finally, we select the largest component $t_{dev} = \max_{x,y,z} \delta t$ and compute the translation cost:

$$q_{ct}(\theta_j) = \begin{cases} C_{ct} + C_{ct} \cdot t_{dev} & \text{if } t_{dev} \geq t_{max} \\ \frac{t_{dev}}{t_{max}}, & \text{otherwise} \end{cases}, \quad (3)$$

where t_{max} is the maximum allowed deviation of the translation component and $C_{ct} \gg 1$ is a predefined constant. Thus, $q_{ct} \in [0, 1]$ if the deviation of the translation is below the allowed maximum and $q_{ct} \gg 1$ otherwise.

Similarly, we define the term $q_{co}(\cdot)$ for penalizing deviations in the orientation. The initial relative orientation $o_{desired} \in \mathbb{R}^3$ between ee_{f1} and ee_{f2} is calculated in the first configuration θ_0 . For each configuration θ_j , the corresponding relative orientation o_j is measured. The deviation from the desired orientation is computed as: $\delta o = |o_{desired} - o_j|$. We select the largest component

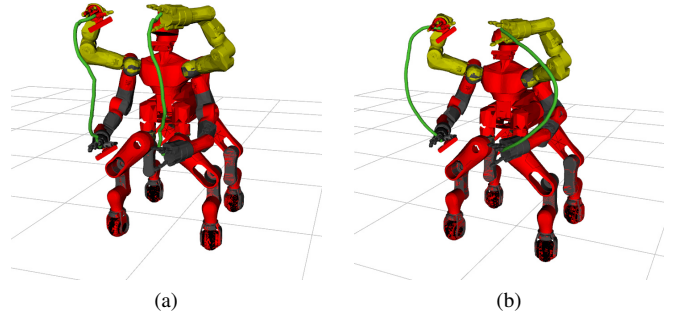


Fig. 5. Comparison of the trajectories obtained with/without kinematic chain closure constraint. Red: start configuration; Yellow: goal configuration; Green: paths of the end-effectors. (a) Closure constraint enabled. The robot has to follow the kinematically difficult path. (b) Closure constraint disabled. The arms can be moved easily to the sides of the robot.

$o_{dev} = \max_{r,p,y} \delta o$ and compute the orientation cost:

$$q_{co}(\theta_j) = \begin{cases} C_{co} + C_{co} \cdot o_{dev} & \text{if } o_{dev} \geq o_{max} \\ \frac{o_{dev}}{o_{max}}, & \text{otherwise} \end{cases}, \quad (4)$$

where o_{max} is the maximum allowed deviation of the orientation component and $C_{co} \gg 1$ is a predefined constant. Extending the algorithm with this constraint allows to optimize trajectories, maintaining the kinematic chain closure constraint, and, hence, plan trajectories for moving objects which are held with two hands.

VI. EVALUATION

First, we present the evaluation of the arm trajectory optimization alone. In the latter subsection, we evaluate the performance of the developed pipeline by picking a watering can with two hands in simulation. Finally, we present the experiments performed with the real robot: dual-arm picking of watering can and drill.

A. Trajectory Optimization

Experiments were performed using the gazebo simulator with the Centauro robot. Both 7 DOF arms were used simultaneously, resulting in a total of 14 DOF. We performed the experiments on an Intel Core i7-6700HQ CPU, 16 GB of RAM, 64 bit Ubuntu 16.04 with 4.13.0-45 kernel using ROS Kinetic. The algorithm ran on a single core with 2.60 GHz.

We investigate how the introduction of the close chain kinematic constraint influences the performance of the algorithm. We compared the performance of the algorithm with and without the constraint in an obstacle-free scenario, where the robot had to lift both arms upwards (Fig. 5). We solved the problem 50 times with enabled/disabled closure constraint, each. The time limit for the algorithm was set to 10 s. The obtained runtimes and success rates are shown in the Table I.

When the algorithm performs optimization without closure constraint, the runtime is relatively short with a very small standard deviation and 100% success rate. On the other hand, with enabled closure constraint, the runtime grew significantly by 1267% and the success rate dropped to 83%.

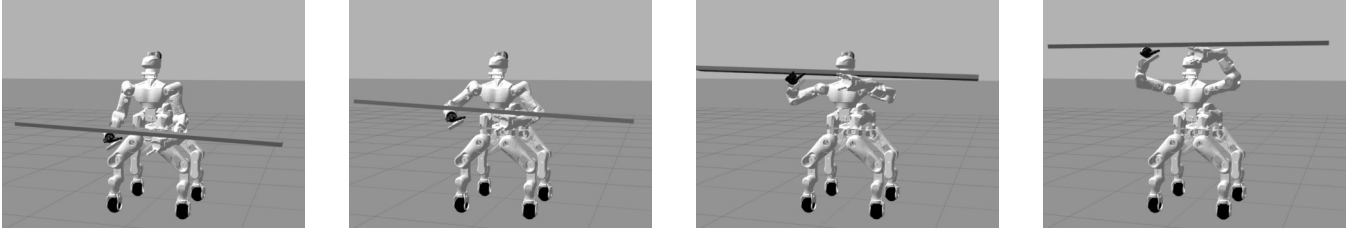


Fig. 6. The Centauro robot lifting a long bulky bar. As the bar is laying on the wrists unsecured, not only the closure constraint has to be preserved, but also the orientation of the end-effectors has to remain the same during the whole trajectory.

TABLE I
COMPARISON OF THE AVERAGE RUNTIME AND SUCCESS RATE
WITH/WITHOUT CLOSURE CONSTRAINT.

	Without closure constraint	With closure constraint
Runtime [s]	0.34 ± 0.01	4.31 ± 2.42
Success rate	100%	83%
Runtime growth	—	1267%

This happens because the space of valid configurations is largely reduced when enforcing the closure constraint and the sampling-based algorithm struggles to converge to a valid solution. This also explains the large standard deviation for the case when the closure constraint is enabled. In Fig. 7 the error between desired and actual pose of the end-effectors, observed during one of those trajectories, is shown.

We also demonstrate the optimization with closure constraints enabled for a practical task. The robot has a long bulky bar laying on its wrists (Fig. 6 (a)) and the task is to lift it up. Since the bar is not secured in any way, it is not only necessary to preserve the closure constraint, but also to maintain the exact orientation of the end-effectors along the whole trajectory (Fig. 6).

B. Dual-Arm Picking in Simulation

We evaluate the proposed system by picking a watering can with two arms in a functional way, i.e., that the robot can afterwards use it. The experiments were performed in the Gazebo simulator with the Centauro robot. To speed up the simulation, only the upper-body was actuated. Moreover,

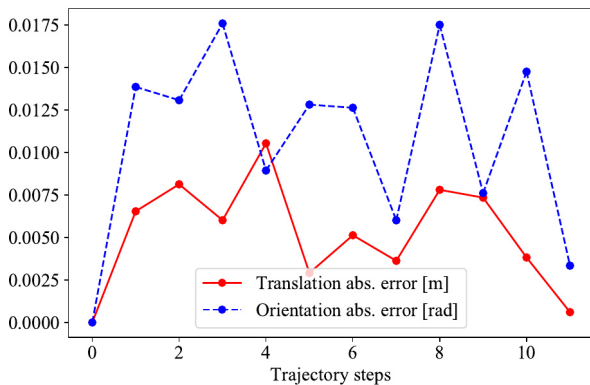


Fig. 7. Error between desired and observed end-effectors relative pose for trajectories shown in Fig. 5.

TABLE II
SUCCESS RATE OF PICKING WATERING CANS FROM THE TEST SET AND
PERFORMANCE OF THE TRAJECTORY OPTIMIZATION METHOD.

	Success rate (attempts to solve)	Traj. opt. runtime [s] Success rate
Can 1	75% (4)	0.9 ± 0.24 100%
Can 2	100% (5)	
Can 3	60% (3)	

the collision model of the fingers were modeled as primitive geometries: capsules and boxes. The laser scanner and the RGBD sensor were also incorporated in the simulation. We trained the semantic segmentation model using synthetic data. We used 8 CAD models of the watering can to render 400 frames. Additional training data with semantic labeling is obtained by placing the frames onto multiple backgrounds and generating the ground truth labels.

For constructing the shape space we define a training set composed of the same watering cans used to train the semantic segmentation model. The test set consisted out of three different watering cans. For the registration, the objects were represented as point clouds generated by ray-casting operations on meshes obtained from 3D databases. The shape space contained 8 principal components.

The task of the experiment is to grasp and to lift upwards all three cans from the test set. Each trial starts with the robot standing in front of the table, on which the watering can is placed. The arms of the robot are located below the surface of the table, so that a direct approach (straight line) to the object will result in a collision. Each can had to be successfully grasped three times with different orientation so that the

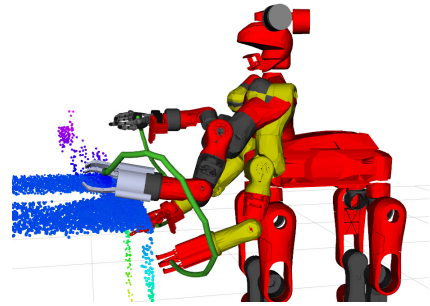


Fig. 8. Dual-arm trajectory for reaching pre-grasp poses. Yellow: initial pose; Black and grey: goal pose; Green: paths of the end-effectors. The arms have to retract back in order to avoid collisions with the table.

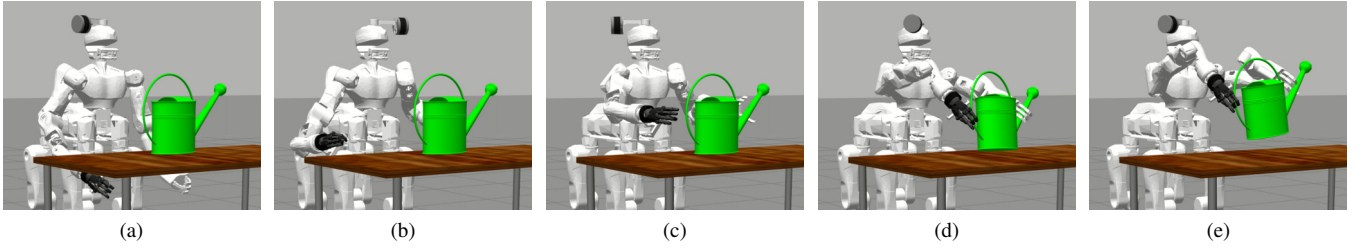


Fig. 9. Centauro performing a dual-arm functional grasp of the watering can in simulation. (a) Initial pose. (b) - (c) Reaching the pre-grasp pose. (d) Can is grasped. (e) Can is lifted.

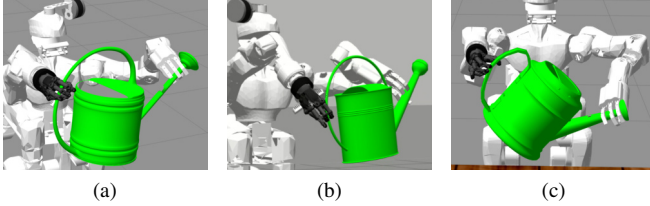


Fig. 10. Three cans from the test set successfully grasped. (a) - (c) Can 1, Can 2, Can 3, respectively. Note that all the cans have different geometry.

task is considered solved. In this manner, the can is rotated around its Z-axis for $+0.25$, 0 and -0.25 radians. In order to evaluate the performance of the non-rigid registration against misalignments, noise in range ± 0.2 radians was added to the yaw component of the 6D pose. The trials were performed until each of the three grasps succeeded once. Obtained success rates and measured average runtime of the trajectory optimization method are presented in Table II.

Our system solved the task Can 2 with no issues, whereas Can 1 and especially Can 3 were more difficult. For Can 1, there was a minor misalignment of the grasp pose for the right hand, which did not allow us to grasp the can successfully. Can 3 had the most distinctive appearance among the cans in our dataset, that is why it caused the most difficulties. During the experiment we often had to run the non-rigid registration several times because it was stuck in local minima. STOMP-New showed consistent success rate and satisfactory runtime of around one second. Typical trajectories for reaching pre-grasp poses are shown in Fig. 8. The Centauro robot performing the experiment with Can 2 is depicted in Fig. 9. All three cans forming our test set, successfully grasped, are shown in Fig 10.

C. Real-Robot Experiments

On the real Centauro robot we performed the same experiment, as described above for a single orientation of the watering can. The pipeline was executed five times in attempt to grasp the can with two hands in a functional way. The method succeeded four times out of five. We measured the average runtime for each component of the system as well as the success rate (Table III).

We do not provide the success rate for the pose estimation, since the ground truth was not available. Consequently, it is hard to assess the success rate of grasp generation as it

TABLE III
AVERAGE RUNTIME AND SUCCESS RATE OF EACH COMPONENT OF THE PIPELINE.

Component	Runtime [s]	Success rate
Semantic segmentation	0.74	100%
Pose estimation	0.12	—
Grasp generation	4.51 ± 0.69	—
Trajectory optimization	0.96 ± 0.29	100%
Complete pipeline	6.27 ± 0.98	80%

may fail due to the previous step of the pipeline. Trajectory optimization method shown a consistent average runtime of around 1 s and a 100% success rate. Overall, the pipeline took around 6 s on average with a success rate of 80%. One of the attempts failed on the stage of grasping the can, because the approaching (goal) pose of the trajectory optimizer was not close enough to the object which resulted in a collision between the hand and the watering can while reaching the pregrasp pose. Consequently, the object moved away from the estimated pose. This suggests that the approaching pose given to the trajectory optimizer should be closer to the object.

In addition to the watering can, the Centauro robot also grasped a two-handed drill to demonstrate that our pipeline can be applied to different types of objects. The process of grasping and lifting both tools is shown in Fig 11. Footages of the experiments can be found online².

VII. CONCLUSIONS

We have developed an integrated approach for autonomous dual-arm pick tasks of unknown objects of a known category. The manipulation pipeline starts with the perception modules, which are capable of segmenting the object of interest. Given the segmented mesh, we utilize a non-rigid registration method in order to transfer grasps within an object category to the observed novel instance. Finally, we extended our previous work on STOMP in order to optimize dual-arm trajectories with kinematic chain closure constraint.

We performed a set of experiments in simulation and with the real robot to evaluate the integrated system. The experiment on trajectory optimization showed that our method can solve the tasks of planning for two arms reliably and fast. However, with introduction of the closure constraint, the

²Experiment video: http://www.ais.uni-bonn.de/videos/Humanoids_2018_Bimanual_Manipulation

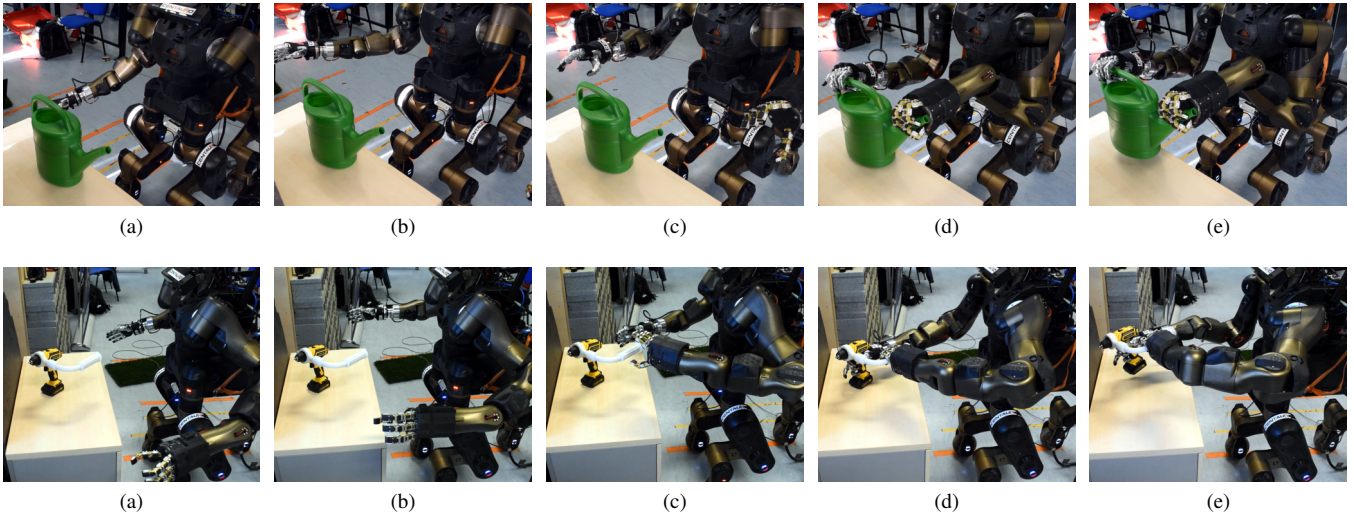


Fig. 11. Centauro performing a dual-arm functional grasp of the watering can and a two-handed drill. (a) Initial pose. (b) - (c) Reaching the pre-grasp pose. (d) Can/drill is grasped. (e) Can/drill is lifted.

runtime grew significantly. Nevertheless, we demonstrated that the method is capable of producing feasible trajectories even under multiple complex constraints. In the simulation experiment, the robot successfully grasped three previously unseen watering cans with two hands from different poses.

On real-robot experiments, our pipeline successfully grasped and lifted several times a watering can and a two-handed drill. These experiments demonstrated that our system can be successfully applied to solve tasks in the real world in an on-line fashion.

REFERENCES

- [1] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.
- [2] D. Pavlichenko and S. Behnke, "Efficient stochastic multicriteria arm trajectory optimization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.
- [3] T. Klamt, D. Rodriguez, M. Schwarz, C. Lenz, D. Pavlichenko, D. Droschel, and S. Behnke, "Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018.
- [4] J. Krüger, G. Schreck, and D. Surdilovic, "Dual arm robot for flexible and cooperative assembly," *CIRP Annals-Manufacturing Technology*, pp. 5–8, 2011.
- [5] P. Tsarouchi, S. Makris, G. Michalos, M. Stefanos, K. Fourtakas, K. Kaltsoukalas, D. Kontrovakis, and G. Chryssolouris, "Robotized assembly process using dual arm robot," *5th CIRP Conference on Assembly Technologies and Systems (CATS)*, vol. 23, 2014.
- [6] J. Stria, D. Průša, V. Hlaváč, L. Wagner, V. Petrík, P. Krsek, and V. Smutný, "Garment perception and its folding using a dual-arm robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.
- [7] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [8] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.
- [9] M. Schwarz, C. Lenz, G. M. García, S. Koo, A. S. Periyasamy, M. Schreiber, and S. Behnke, "Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing," in *Int. Conf. on Robotics and Automation (ICRA)*, 2018.
- [10] N. Vahrenkamp, L. Westkamp, N. Yamanobe, E. E. Aksoy, and T. Asfour, "Part-based grasp planning for familiar objects," in *2016 IEEE-RAS 16th Int. Conf. on Humanoid Robots (Humanoids)*, 2016.
- [11] T. Stouraitis, U. Hillenbrand, and M. A. Roa, "Functional power grasps transferred through warping and replanning," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.
- [12] U. Hillenbrand and M. A. Roa, "Transferring functional grasps through contact warping and local replanning," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.
- [13] W. Szykiewicz and J. Błaszczyk, "Optimization-based approach to path planning for closed chain robot systems," *Int. Journal of Applied Mathematics and Computer Science*, vol. 21, no. 4, pp. 659–670, 2011.
- [14] N. Vahrenkamp, D. Berenson, T. Asfour, J. J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2464–2470, 2009.
- [15] B. Cohen, S. Chitta, and M. Likhachev, "Single- and dual-arm motion planning with heuristic search," *The Int. Journal of Robotics Research*, vol. 33, no. 2, pp. 305–320, 2014.
- [16] S. Byrne, W. Naeem, and S. Ferguson, "Improved APF strategies for dual-arm local motion planning," *Transactions of the Institute of Measurement and Control*, vol. 37, no. 1, pp. 73–90, 2015.
- [17] Z. Ren, C. Zhou, S. Xin, and N. Tsagarakis, "Heri hand: A quasi dexterous and powerful hand with asymmetrical finger dimensions and under actuation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017, pp. 322–328.
- [18] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in *Int. Conf. on Advanced Robotics (ICAR)*, 2015, pp. 388–394.
- [19] D. Rodriguez, C. Cogswell, S. Koo, and S. Behnke, "Transferring grasping skills to novel instances by latent space non-rigid registration," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2018.
- [20] D. Rodriguez and S. Behnke, "Transferring category-based functional grasping skills by latent space non-rigid registration," in *IEEE Robotics and Automation Letters (RA-L)*, 2018, pp. 2662–2669.