The EU Framework Programme for Research and Innovation H2020
Research and Innovation Action

CENTAURO

Deliverable D6.4 Autonomous Dual Arm Pick and Place Manipulation Skills

Dissemination Level: Public

Project acronym:	CENTAURO
Project full title:	Robust Mobility and Dexterous Manipulation in Disaster Response by Fullbody Telepresence in a Centaur-like Robot
Grant agreement no.:	644839
Lead beneficiary:	UBO – University of Bonn
Authors:	D. Pavlichenko, D. Rodriguez, M. Schwarz, C. Lenz, A. S. Periyasamy and S. Behnke
Work package:	WP6 Manipulation
Date of preparation:	2018-10-09
Type:	Report
Version number:	1.0

Document History

Version	Date	Author	Description
0.1	2018-07-04	DP	First draft
0.2	2018-07-31	DP	Second draft
0.3	2018-08-24	DP	Ready for internal review
0.8	2018-09-04	DP	Reviews addressed
0.9	2018-10-09	DP	Include real-robot experiments
1.0			Submitted version

Executive Summary

This deliverable describes the pipeline for dual-arm autonomous manipulation for the Centauro robot. In order to fully make use of the capabilities of the robot (two anthropomorphic arms with 7 DOFs each) and broaden the range of tasks which can be solved, dual-arm manipulation has to be addressed. The basic concept for the CENTAURO manipulation pipeline is presented in Deliverable D6.3 – Autonomous Single-Arm Pick and Place Manipulation Skills. This deliverable focuses on the extension of grasp generation and arm trajectory optimization methods towards dual-arm tasks, which must observe kinematic closure of the manipulator chain through the manipulated object. We also report on the integration of extended methods into the CENTAURO system. The developed dual-arm manipulation pipeline is ready for the final evaluation of the integrated CENTAURO disaster-response system.

Contents

1 Introduction

Daily-life scenarios are full of objects designed to be manipulated with anthropometric arms. Thus, human-like robots are the natural solution to be used in quotidian environments. In these scenarios, many objects require two or more end-effectors 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 or operating a heavy power drill. Consequently, designing algorithms for dual-arm manipulation has attracted much interest in the research community.

In this deliverable, we describe an integrated system capable of performing autonomous dual-arm pick-and-place 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. The pipeline for solving these tasks for single-arm manipulation was developed in "Deliverable D6.3 Autonomous Single-Arm Pick and Place Manipulation Skills". The main focus of this deliverable is the extension of this pipeline towards dual-arm tasks.

First, we use semantic segmentation to detect the object. A segmented point cloud is then passed to the next step of the pipeline: deformable 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 [1][2]. 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 output trajectories of the grasp planner by applying a modified version of Stochastic Trajectory Optimization for Motion Planning (STOMP) [3], which we refer to as STOMP-New [4]. 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 challenging task due to curse of dimensionality and severe workspace constraints for joint valid configurations. We evaluate the capabilities of the designed system on the dual-arm pick-and-place task for a watering can in simulation (Fig. 1).

2 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, Krüger *et al.* [5] 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.* [6] allow dual arm robots to perform tasks, which are usually done manually by human operators in an automotive assembly plant. Stria *et al.* [7] 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. Since the software of such complex systems consist of multiple components, we further briefly review some of the noticeable works for each major software module.

2.1 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 [8], [9]. 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 [10].

2.2 Transferring Grasping Skills

Vahrenkamp *et al.* [11] 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.* [12] and Hillenbrand and Roa [1] 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

Figure 1: Pre-grasp pose of the Centauro robot for dual-arm grasping of an unknown watering can.

large execution time. The method explained here, on the other hand, is suitable for on-line scenarios.

2.3 Dual-Arm Motion Planning

Dual-arm motion planning is a challenging task, for which intensive research has been carried out. Szynkiewicz 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. The solution is found numerically. Vahrenkamp *et al.* [14] presented two different approaches for dual-arm planning: Jacobian Pseudoinverse-Based (J^+) and Inverse Kinematics Rapidly Exploring Random Tree (IK-RRT). The advantage of the first approach is that it does not need an IK solver. However, IK-RRT was shown to perform better 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] use Artificial Potential Fields (APF) in their work. The method consists of goal configuration sampling, subgoal selection and APF motion planning. It was shown that the method improves APF performance for both 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 new criteria, which we demonstrate in this work.

3 Method

3.1 System Overview

The robot is equipped with two anthropomorphic manipulators with 7 DOFs each. The right arm possesses a SCHUNK SVH 5-finger 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, the Kinect v2 sensor 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-and-place task we created the following pipeline:

- Semantic segmentation performed by using RGB-D data from the Kinect v2,
- with the segmented point cloud as input, we perform non-rigid shape registration to obtain grasping poses,
- finally, a trajectory optimization is carried out in order to obtain collision-free trajectories to reach pre-grasp poses.

The diagram of this pipeline is shown in Fig. 3.

3.2 Semantic Segmentation

For perceiving the object to be manipulated, a state-of-the-art semantic segmentation architecture RefineNet [8] 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.* [10] 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

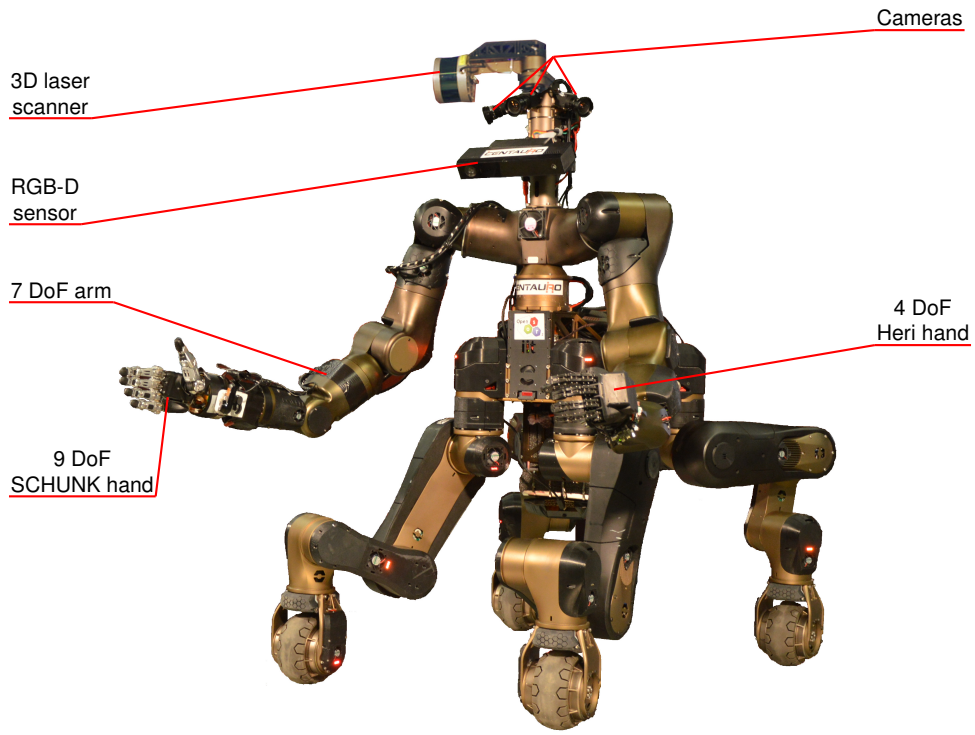


Figure 2: The Centauro robot. Main components of the upper-body are labeled.

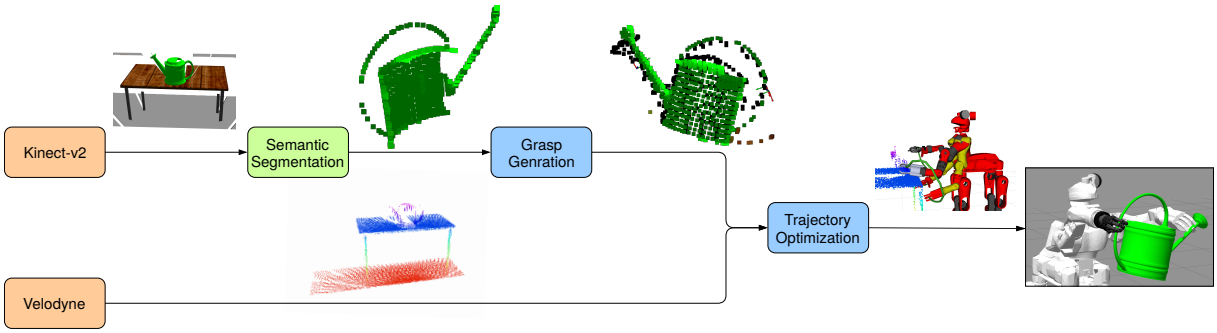


Figure 3: Simplified diagram of the system, showing the information flow between core components. Orange: sensors; Green: perception components; Blue: planning components.

consists of four ResNet blocks. After each block the features become more abstract, but also lose 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.* [10], 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.

3.3 Grasp Planning

The grasp planning is a learning-based approach that exploits the fact that objects of similar shape 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 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 point times the number of dimensions of the canonical model. This means that the variations in shape from one object to the other can be expressed by a vector of the same length across all training samples. Thus, subspace methods can be straightforward applied. Finally, the principal components of all these deformation fields are calculated by using Principal Component Analysis - Expectation Maximization (PCA-EM). They define the orthonormal 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 [18]. 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 [2]. For bimanual manipulation we associate individual

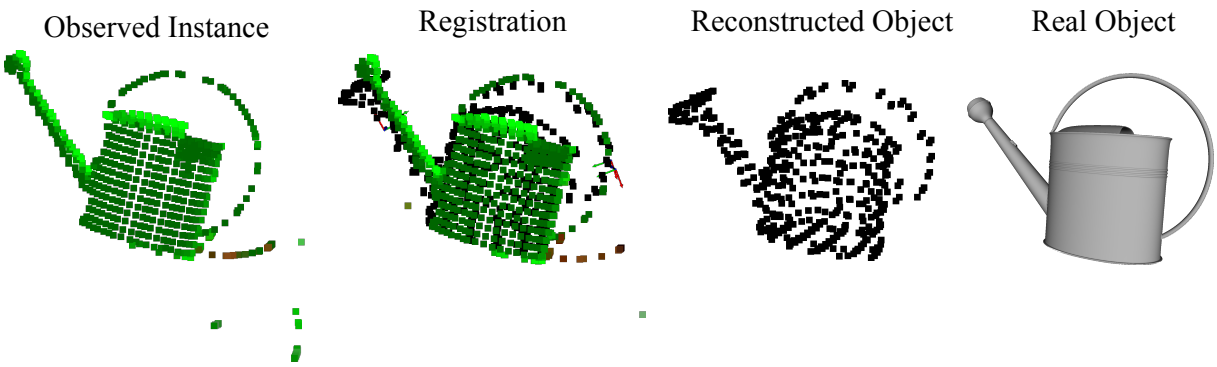


Figure 4: Shape space registration on the watering can category. Green: pointcloud of observed instance. Black: canonical model, fitted to the observed instance. The method is able to reconstruct a partially occluded instance containing noise.

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.

3.4 Trajectory Optimization

Given pre-grasp poses for both arms, it is key to plan a collision-free trajectory to reach them. We use our modification of Stochastic Trajectory Optimization for Motion Planning (STOMP) [3]: STOMP-New, which showed better performance in previous experiments [4]. 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 $\boldsymbol{\theta}_i \in \mathbb{R}^J$ in joint space with J joints. A naïve initial trajectory that is often used is the linear interpolation between the given start and goal configurations $\boldsymbol{\theta}_{start}$ and $\boldsymbol{\theta}_{goal}$. Start and goal configurations are unchanged during the optimization, the algorithm then outputs an optimized trajectory.

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) for obstacle cost computation as well as for orientation constraints. These upgrades allow us to optimize trajectories of two independent arms simultaneously. Moreover, in case dual arm manipulation is required, a kinematic chain closure constraint has to be satisfied. In order to obtain trajectories which satisfy this constraint, we add an additional term $q_{cc}(.,.)$ to the cost function:

$$\begin{aligned} q(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) =& q_o(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) + q_l(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) + q_c(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) \\ &+ q_d(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) + q_t(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) + q_{cc}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}), \end{aligned} \tag{1}$$

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

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

$$q_{cc}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) = \frac{1}{2} \max_j q_{ct}(\boldsymbol{\theta}_j) + \frac{1}{2} \max_j q_{co}(\boldsymbol{\theta}_j), j \in i...i+1, \tag{2}$$

where $q_{ct}(.)$ is the term which penalizes deviations in the translation between the end-effectors along the transition and the term $q_{co}(.)$ penalizes deviations of the relative orientation of the end-effectors, respectively.

Given two end-effectors ee_1 and ee_2, the initial translation $t_{desired} \in \mathbb{R}^3$ between them is measured in the very first configuration $\boldsymbol{\theta}_0$ of the trajectory. Then, for every evaluated configuration $\boldsymbol{\theta}_j$, the corresponding translation t_j between ee_1 and ee_2 is measured. We can now measure the deviation from the desired translation: $\delta t = |t_{desired} - t_j|$. Finally, we select the largest component $t_{dev} = \max_{x,y,z} \delta t | \delta t = \langle x, y, z \rangle$ and compute the translation cost:

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

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 $q_{co}(.)$ for penalizing deviations in the orientation. The initial relative orientation $o_{desired} \in \mathbb{R}^3$ between ee_1 and ee_2 is measured in the very first configuration $\boldsymbol{\theta}_0$. For every configuration $\boldsymbol{\theta}_j$, the corresponding relative orientation o_j is measured. The deviation from the desired orientation is computed: $\delta o = |o_{desired} - o_j|$. We select the largest component $o_{dev} = \max_{r,p,y} \delta o | \delta o = \langle r, p, y \rangle$ and compute the orientation cost:

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

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 the trajectories for moving objects which are held with two hands.

Table 1: 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%

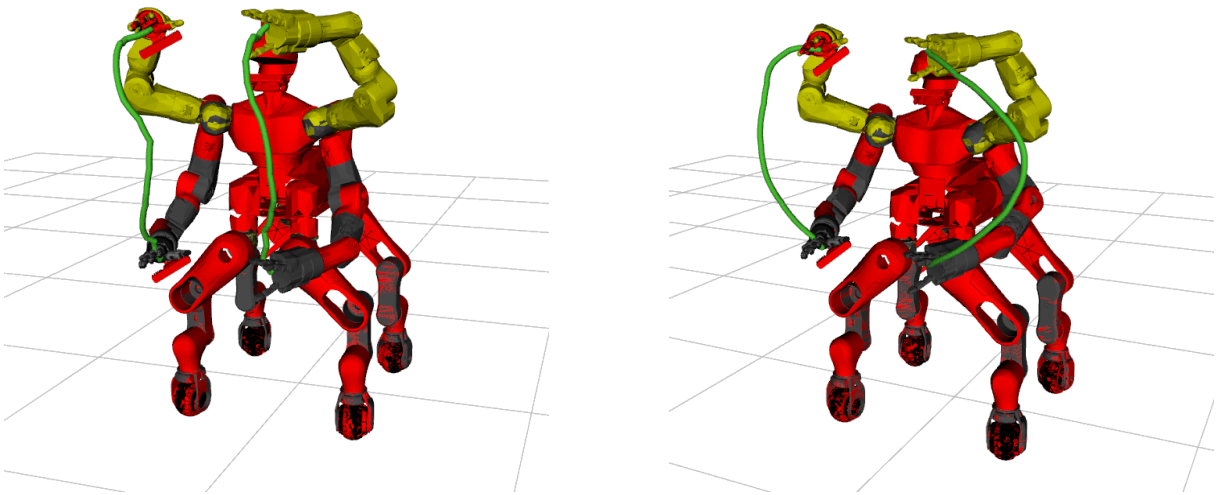
4 Evaluation

First, we present the evaluation of the arm trajectory optimization method alone. In the latter subsection, we evaluate the performance of the developed pipeline by picking a watering can with two hands.

4.1 Trajectory Optimization

In this subsection we present the results of the evaluation of the trajectory optimization in isolation. Experiments were performed using the Gazebo simulator. Both 7 DOFs arms were used simultaneously, resulting in a total of 14 DOFs. We performed the experiments on an Intel Core i7-6700HQ CPU, 16 GB of RAM, 64 bit Kubuntu 16.04 with 4.13.0-45 kernel using ROS Kinetic. The algorithm ran on a single core with 2.60 GHz.

Since the main extension we have made to the trajectory optimization algorithm in this work is the introduction of the closed kinematic chain constraint, we investigate how it 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 1.

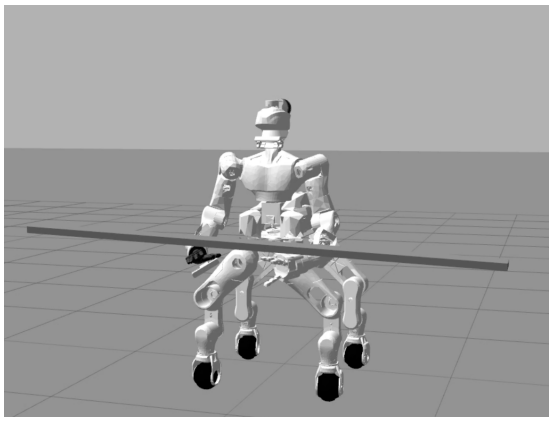


(a) (b)

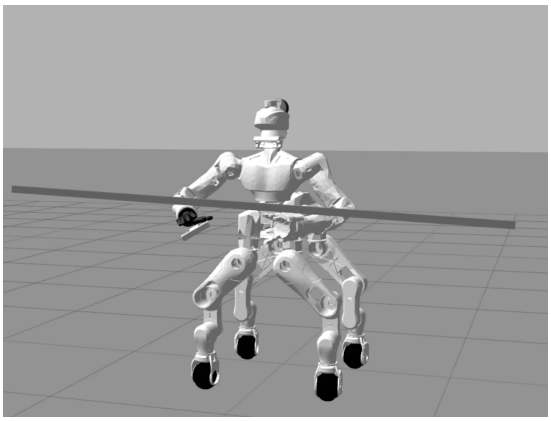
Figure 5: Comparison of the trajectories obtained with/without kinematic chain closure constraint. Red: start configuration; Yellow: goal configuration; Green: paths of the end-effectors. **(a)** Closure constraint enabled. The robot has to follow the kinematically difficult path. **(b)** Closure constraint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

One can observe that 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%. This happens because the space of valid configurations is reduced by several orders of magnitude 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. 5 one can also observe the difference between two typical trajectories for this task. With enabled closure constraint the robot has to follow a kinematically complicated path. Meanwhile without this constraint the arms can be moved easily widely by the sides of the robot.

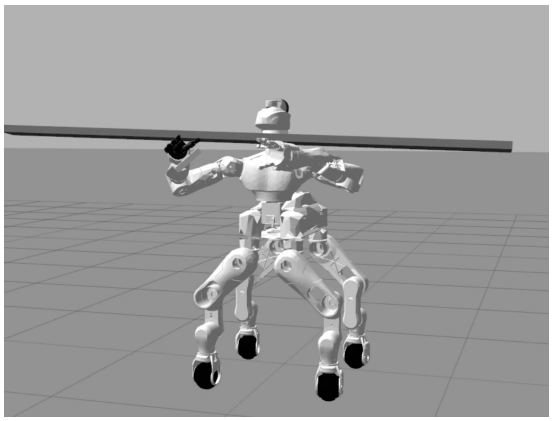
We also demonstrate the optimization with closure constraints enabled for a practical task. The robot has a long bulky bar laying on its wrists (Fig. 6 (a)) and the task is to lift it up. Since the bar is not secured in any way, it is essential to preserve the closure constraint, and also to maintain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



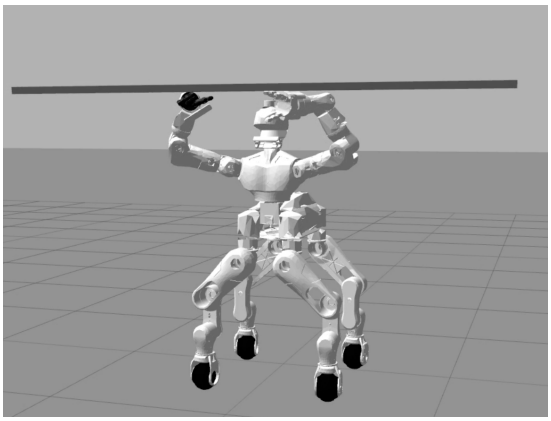
(a)



(b)



(c)



(d)

Figure 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 2: 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)	

4.2 Dual-Arm Picking of Watering Can

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. To speed up the simulation, only the upper-body was actuated. Moreover, the collision models of the fingers were modeled as primitive geometries: capsules and boxes. The laser scanner and the RGB-D 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 with a explained variance equal to 98%.

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

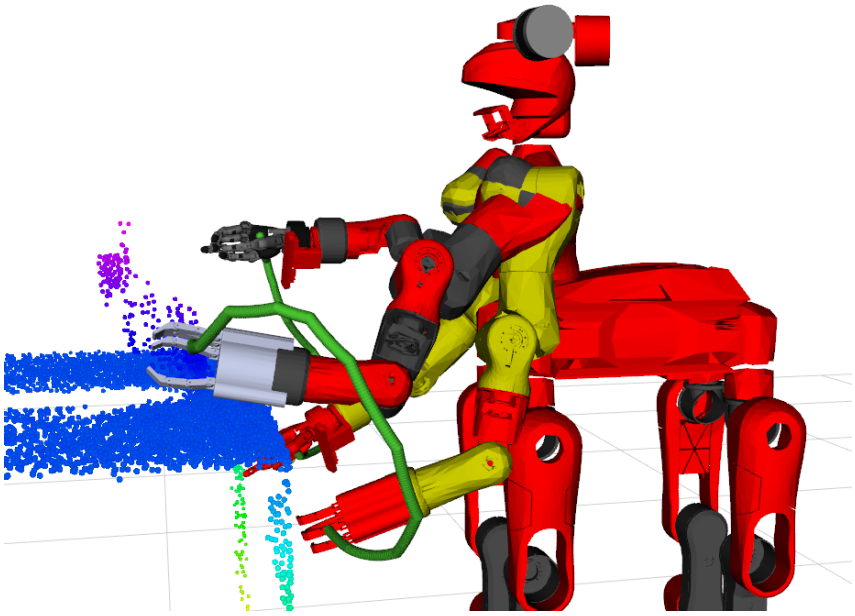
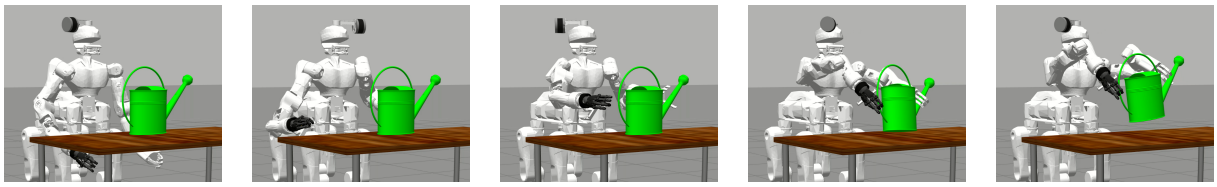
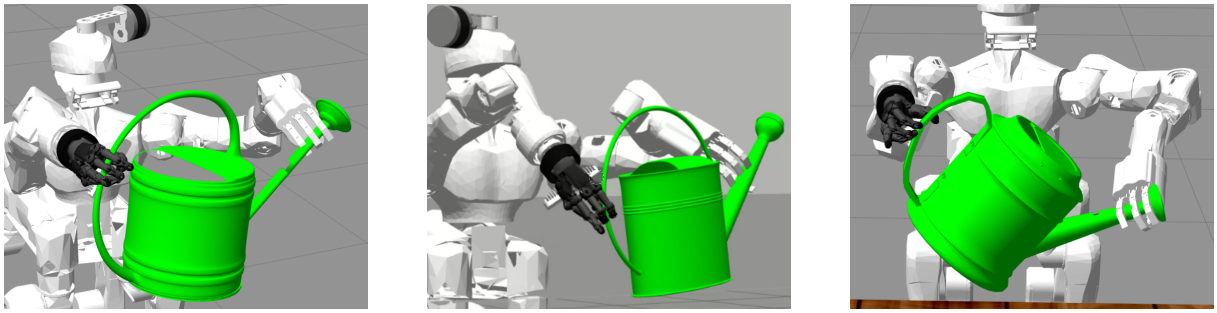


Figure 7: Dual-arm trajectory for reaching pre-grasp poses. Yellow: initial pose; Black and grey: goal pose; Green: paths of the end-effectors. The arms have to retract back in order to avoid collisions with the table.



(a) (b) (c) (d) (e)

Figure 8: Centauro performing a dual-arm functional grasp of the watering can. **(a)** Initial pose. **(b)** - **(d)** Reaching the pre-grasp pose. **(e)** Can is grasped. **(f)** Can is lifted.



(a) (b) (c)

Figure 9: 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.

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 the task is considered solved. In this manner, the can is rotated around its Z-axis for +0.25, 0 and -0.25 radians. We used ground truth for the 6D pose of the can. 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 2.

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. 7. The robot performing the experiment with Can 2 is depicted in Fig. 8. All three cans forming our test set, successfully grasped, are shown in Fig 9.

4.3 Real-Robot Experiments

In order to evaluate the system 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, where it was possible. Obtained average runtimes and success rates are

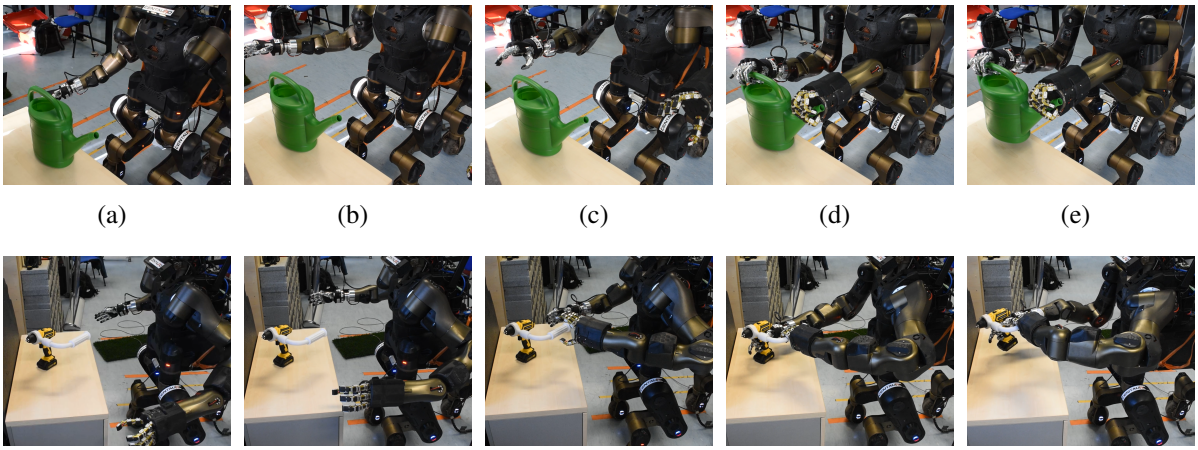
Table 3: Average runtime and success rate of each component of the pipeline.

Component	Runtime [s]	Success rate
Semantic segmentation	0.72 ± 0.21	100%
Pose estimation	0.13 ± 0.06	—
Grasp generation	4.51 ± 0.69	—
Trajectory optimization	0.96 ± 0.29	100%
Complete pipeline	6.32 ± 1.25	80%

shown in Table 3.

We do not provide the success rate for the pose estimation, since the ground truth is not available. Consequently, it is hard to access the success rate of grasp generation as it may fail due to the previous step of the pipeline. Overall, the pipeline took 6-7son average with a success rate of 80%. One of the attempts failed on the stage of grasping the can. This happened because the approaching (goal) pose of the trajectory optimizer was not close enough to the object which results 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, Centauro 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 of both tools is shown in Fig 10. Footages of all described experiments can be found online[1].



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 10: Centauro performing a dual-arm functional grasp of the watering can and drill. **(a)** Initial pose. **(b)** - **(c)** Reaching the pre-grasp pose. **(d)** Can is grasped. **(e)** Can is lifted. Same procedure applies to the drill.

[1]Experiment video: `http://www.ais.uni-bonn.de/videos/Humanoids_2018_Bimanual_Manipulation`

5 Conclusions

We extended a single-arm pipline to perfom tasks using two hands. The approach begins with the perception modules, which segment 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.

In order to evaluate our integrated system, we performed a set of experiments in simulation and on the real Centauro robot, which has two arms with 7 DOFs each. In the main experiment, the robot successfully grasped three previously unseen watering cans with two hands from different poses. The latent space for non-rigid registration was built using only eight watering can instances. In this experiment, the trajectory optimization for dual-arm setup showed success rate of 100% and average runtime of 0.9 seconds. The experiment on trajectory optimization showed that our method can solve reliably and fast the tasks of planning for two arms which act independently. However, with introduction of the closure constraint, the runtime grew significantly. Nevertheless, we demonstrated that the method is capable of producing feasible trajectories even under multiple complex constraints.

We performed a series of experiments on the real robot to grasp a watering can and a two-handed drill. These experiments demonstrated that proposed pipeline can solve real-world tasks. The developed dual-arm manipulation pipeline is ready for the final evaluation of the integrated CENTAURO disaster-response system.

References

[1] 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.

[2] 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.

[3] 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.

[4] D. Pavlichenko and S. Behnke, "Efficient stochastic multicriteria arm trajectory optimization", in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[5] J. Krüger, G Schreck, and D. Surdilovic, "Dual arm robot for flexible and cooperative assembly", *Cirp Annals-manufacturing Technology*, vol. 60, pp. 5–8, 2011.

[6] P. Tsarouchi, S Makris, G. Michalos, M. Stefos, K. Fourtakas, K. Kaltsoukalas, D. Kontrovrakis, and G. Chryssolouris, "Robotized assembly process using dual arm robot", *Procedia CIRP*, vol. 23, 2014.

[7] 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 *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] T. Stouraitis, U. Hillenbrand, and M. A. Roa, "Functional power grasps transferred through warping and replanning", in *2015 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[13] W. Szynkiewicz 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", *2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 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] 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.

Transferring Category-based Functional Grasping Skills by Latent Space Non-Rigid Registration

Diego Rodriguez and Sven Behnke

***Abstract*—Objects within a category are often similar in their shape and usage. When we—as humans—want to grasp something, we transfer our knowledge from past experiences and adapt it to novel objects. In this paper, we propose a new approach for transferring grasping skills that accumulates grasping knowledge into a category-level canonical model. Grasping motions for novel instances of the category are inferred from geometric deformations between the observed instance and the canonical shape. Correspondences between the shapes are established by means of a non-rigid registration method that combines the Coherent Point Drift approach with subspace methods. By incorporating category-level information into the registration, we avoid unlikely shapes and focus on deformations actually observed within the category. Control poses for generating grasping motions are accumulated in the canonical model from grasping definitions of known objects. According to the estimated shape parameters of a novel instance, the control poses are transformed towards it. The category-level model makes our method particularly relevant for on-line grasping, where fully-observed objects are not easily available. This is demonstrated through experiments in which objects with occluded handles are successfully grasped.**

***Index Terms*—Dexterous manipulation, Grasping, Multi-fingered hands.**

I. Introduction

WHILE transferring grasping skills within a category happens frequently and effortless in humans, obtaining that generalization in robots is still an open problem. People can be shown objects that they never saw before, and they often will immediately know how to grasp and operate them. This happens by transferring knowledge from their learned model of the object category, e.g., screw drivers, to novel instances. Although the manipulation of known objects can be planned offline, many open-world applications require the manipulation of unknown instances. Our approach accumulates manipulation knowledge of known instances in category-level models and transfers manipulations skills to novel instances (Fig. 1).

The method presented in this paper focuses on functional grasping, i.e., on motions that allow not only to grasp the object but also to use it. We use the term *grasping* to refer to the process of bringing the object into the hand, and not only to the final configuration of hand and object. We propose a method for generating grasping motions for novel instances by making use of category-level shape information represented by a learned latent shape space. Our method aggregates object shape and grasping knowledge from multiple known instances of a category in a canonical model. The learned latent space of shape variations enables a category-specific shape-aware non-rigid registration procedure that establishes correspondences between a view of a novel object instance and the canonical model. Our method finds a transformation from the canonical model to the view in the latent shape space—linearly interpolating and extrapolating from other transformations found within the category—which best matches the observed 3D points. This estimates the shape parameters of the novel instance and allows for inference of its occluded parts. By the non-rigid transformation and the aggregated manipulation knowledge, control poses for the novel instance are inferred. The grasping motion is finally generated by using those control poses.

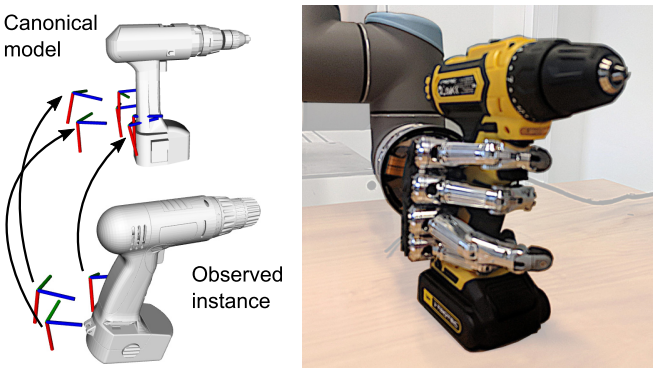


Figure 1: Shape information and grasping knowledge for known object instances are aggregated in a category-level canonical model. Grasping control poses are transferred to novel instances of the category for generating the grasping motion.

In this paper, we extend our previous work [1] by accumulating grasping knowledge in the canonical model in addition to the shape information, which enriches our transferring skill model.

II. Related Work

A. Non-Rigid Registration and Shape Spaces

Most of the non-rigid registration methods proposed so far differ mostly by the prior restrictions or regularization on the deformation that the points can undergo. Several restrictions

Manuscript received: November, 21, 2018; Revised January, 26, 2018; Accepted March, 19, 2018.

This paper was recommended for publication by Editor Han Ding upon evaluation of the Associate Editor and Reviewers' comments. 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.

All authors are with the Autonomous Intelligent Systems (AIS) Group, Computer Science Institute VI, University of Bonn, Germany {rodriguez, behnke}@ais.uni-bonn.de

Digital Object Identifier (DOI): see top of this page.

such as conformal maps [2]–[4], isometry [5]–[7], thin-plate splines [8], [9], elasticity [10] and Motion Coherence Theory [11] have been used to encourage or constrain different types of transformations.

For surface reconstruction, many methods use non-rigid registration [12]–[15]. Approaches such as presented by Li *et al.* [12] and Zollhöfer *et al.* [16] sequentially add higher frequency details coming from new depth camera frames to a low-resolution 3D capture through non-rigid registration.

For category-based shape spaces, several methods have been proposed. Hasler *et al.* [17] generate a shape space of human bodies with poses using 3D markers and human scans. Burghard *et al.* [18] developed a shape space of varying geometry based on dense correspondences. Engelmann *et al.* [19] define a shape manifold which models intra-class shape variance; this method is robust with noisy or occluded regions.

B. Transferring Grasping Skills

Based on segmented objects according to their RGB-D appearance, Vahrenkamp *et al.* [20] transfer grasp poses from a set of template grasps. Ficuciello *et al.* [21] developed an approach to confer grasping capabilities based on a reinforcement learning technique and postural synergies. In [22] and [23], functional grasp poses are warped such that distance between correspondences is minimized, then the warped poses are replanned in order to increase the functionality of the grasp. In [24] a similar contact warping is combined with motor synergies to generalize human grasping. Stueckler *et al.* [25] transfer manipulation skills using a non-rigid registration method based on multi-resolution surfel maps. The non-rigid registration serves as the mechanism to warp available grasping poses.

C. Discussion

Although current state-of-the-art methods for non-rigid registration yield good results, they have some limitations. Newcombe *et al.* [15] use optical flow constraints and thus this approach does not perform well with large deformations or changes in color and illumination. Moreover, several captures of the object are required. The method by Burghard *et al.* [18] accurately estimates dense correspondences, but does not perform well with incomplete scans or noisy data. To solve these problems, we incorporate category-level information in our approach, such that we are able to register partially-occluded novel instances using a single capture of the object. Methods such as Engelmann *et al.* [19] deal with minor misalignments and occlusions, but do not offer correspondences between points and do not give any kind of transformation. Our method, on the other hand, offers a transformation for each point of the novel instance and even points that do not belong to the object which allows us to transform grasp poses.

Regarding transferring grasping skills, we tackle the problem of requiring a fully observed [22] or a non-occluded [20] object by exploiting the geometrical information residing in our learned categorical model. Unlike [25] we model shape and grasping not for single known instances, but for object categories, which gives us the possibility to learn typical shape variations and to infer grasping information even when parts of the object are not observed. More importantly, none of previous approaches is able to accumulate and to use knowledge from *several* previous successfully experiences, which is the main focus of this paper.

III. Method

Our approach is composed of a learning phase and an inference phase (Figs. 2 and 3). In the learning phase, a category-specific linear model of the transformations that a category of objects can undergo is built. In this manner, poses in the space of the canonical shape can be transformed into the space of an observed instance. These poses can be added even after the learning phase. The category-specific linear model is learned as follows: First, we select a single instance from the training dataset to be the canonical model of the category. Then, we find the transformations relating this instance to all other instances of the category using Coherent Point Drift (CPD) [11]. Finally, we find a linear latent subspace of these transformations, which becomes our transformation model for the category. For each instance in the training set, an associated grasping descriptor $\boldsymbol{\varsigma}$ (vector representation of the grasping motion) is also transformed into the canonical space. In this manner, multiple experiences can be aggregated in the canonical model.

In the inference phase, given a novel observed instance, our method searches in the subspace of transformations to find the transformation which best relates the canonical shape to the observed instance. Depending on the resulting latent shape variables and the aggregated grasping knowledge accumulated in the canonical model, a grasping descriptor for the novel instance is inferred.

A. Categories and Shape Representation

A category is composed by a set of objects which share the same topology and have a similar shape. Each category has a canonical shape $\mathbf{C}$ that will be deformed to fit the shape of the training and testing sample shapes. To represent a shape, we use point clouds, which can be generated from meshes by ray-casting from several viewpoints on a tessellated sphere and then down-sampling with a voxel grid filter. Each category specifies a canonical pose and reference frame, used for initial alignments.

B. Coherent Point Drift

Here, we shortly describe the Coherent Point Drift (CPD) [11] and how we use it for our non-rigid registration.

CPD estimates a deformation field mapping between a template point set $\mathbf{S}^{[t]} = (\mathbf{s}_1^{[t]}, ..., \mathbf{s}_M^{[t]})^T$ and a reference point set $\mathbf{S}^{[r]} = (\mathbf{s}_1^{[r]}, ..., \mathbf{s}_N^{[r]})^T$. The points in $\mathbf{S}^{[t]}$ are modeled as centroids of a Gaussian Mixture Model (GMM) from which the points in $\mathbf{S}^{[r]}$ are drawn. CPD maximizes the likelihood of the GMM while imposing constraints on the motion of the centroids such that points near each other should move coherently and have a similar motion to their neighbors [26]. The likelihood of

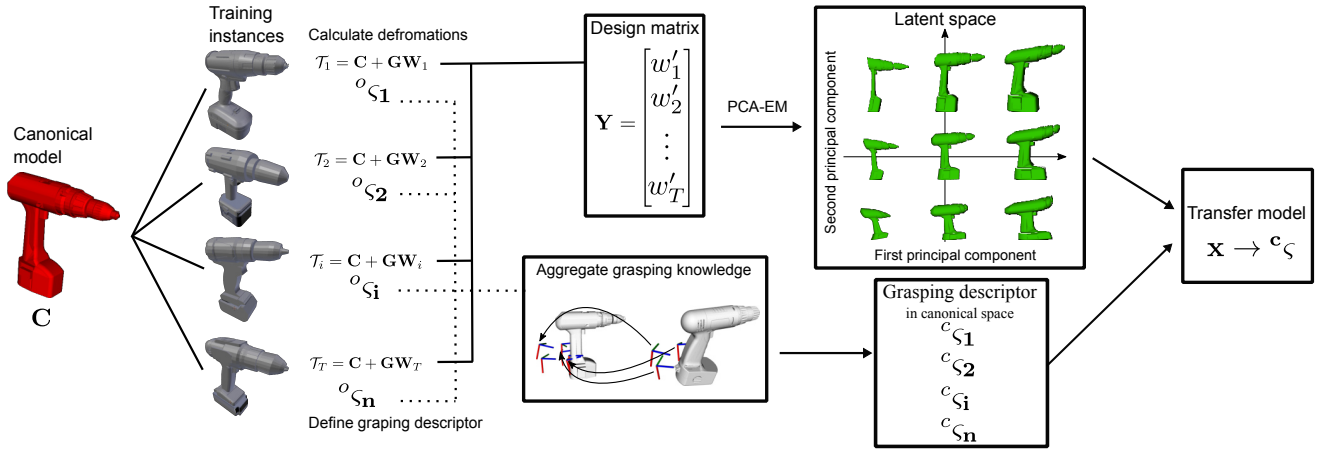


Figure 2: Training phase. The deformations between each instance and the canonical model are calculated using CPD. These deformations are assembled into the design matrix $\mathbf{Y}$. Using PCA-EM, the principal components which constitute the latent space are extracted. The grasping descriptor for each training sample is aggregated in the canonical model. The latent variables serve as feature vector while the grasping descriptor is the desired output for the grasping transfer model.

the GMM is not directly maximized, but instead its equivalent negative log-likelihood function is minimized:

$$E(\boldsymbol{\psi}, \sigma^2) = -\sum_{n=1}^{N} \log \sum_{m=1}^{M} \exp^{-\frac{1}{2\sigma^2}\left\|\mathbf{s}_n^{[r]} - \mathcal{T}(\mathbf{s}_m^{[t]}, \boldsymbol{\psi})\right\|^2}, \quad (1)$$

where $\mathcal{T}(\mathbf{s}_m^{[t]}, \boldsymbol{\psi})$ is a parametrized transformation from the template point set to the reference set, and σ^2 is the covariance of the Gaussian density. The transformation $\mathcal{T}$, for the non-rigid registration, is defined as the initial position plus a displacement function v:

$$\mathcal{T}(\mathbf{S}^{[t]}, v) = \mathbf{S}^{[t]} + v(\mathbf{S}^{[t]}). \quad (2)$$

The constraints on the motion of the centroids are realized by regularizing the displacement function v. Adding this regularization $\phi(v)$ to the negative log-likelihood Eq. (1), we obtain

$$f(v, \sigma^2) = E(\sigma^2, v) + \frac{\lambda}{2}\phi(v), \quad (3)$$

where λ is a trade-off parameter between the goodness of maximum likelihood fit and regularization. A particular choice of $\phi(v)$ leads to the following displacement function $v(\mathbf{Z})$ [11]:

$$v(\mathbf{Z}) = G(\mathbf{S}^{[\mathbf{t}]}, \mathbf{Z})\mathbf{W}, \quad (4)$$

for any set of D-dimensional points $\mathbf{Z}_{N\times D}$. $G(\mathbf{S}^{[\mathbf{t}]}, \mathbf{Z})$ is defined as a Gaussian kernel matrix composed element-wise by:

$$g_{ij} = G(\mathbf{s}_i^{[t]}, \mathbf{z}_j) = \exp^{-\frac{1}{2\beta^2}\left\|\mathbf{s}_i^{[t]} - \mathbf{z}_j\right\|^2}, \quad (5)$$

$\mathbf{W}_{M\times D}$ is a matrix of kernel weights, and β is a scalar that controls the strength of interaction between points. An additional interpretation of $\mathbf{W}$ is as a set of D-dimensional deformation vectors, each associated with one of the M points of $\mathbf{S}^{[\mathbf{t}]}$. For convenience in the notation, $\mathbf{G}_{M\times M}$ will denote $G(\mathbf{S}^{[t]}, \mathbf{S}^{[t]})$. Note that $G(\cdot,\cdot)$ can simply be computed by Eq. (5), but the matrix $\mathbf{W}$ needs to be estimated.

To minimize Eq. (3), CPD uses an Expectation Maximization (EM) algorithm. In the E-step, the posterior probabilities matrix $\mathbf{P}$ is estimated using past parameter values. This matrix $\mathbf{P}$ is composed element-wise by:

$$p_{mn} = \frac{e^{-\frac{1}{2\sigma^2}\left\|\mathbf{s}_n^{[r]} - (\mathbf{s}_m^{[t]} + G(m,\cdot)\mathbf{W})\right\|^2}}{\sum_{m=1}^{M} e^{-\frac{1}{2\sigma^2}\left\|\mathbf{s}_n^{[r]} - (\mathbf{s}_m^{[t]} + G(k,\cdot)\mathbf{W})\right\|^2} + \frac{\omega}{1-\omega}\frac{(2\pi\sigma^2)^{\frac{D}{2}}}{N}} \quad (6)$$

where ω reflects the assumption on the amount of noise.

In the M-step, the matrix $\mathbf{W}$ is estimated by:

$$(\mathbf{G} + \lambda\sigma^2 d(\mathbf{P1})^{-1})\mathbf{W} = d(\mathbf{P1})^{-1}\mathbf{P}\mathbf{S}^{[r]} - \mathbf{S}^{[t]} \quad (7)$$

where $\mathbf{1}$ represents a column vector of ones and $d(\cdot)^{-1}$ is the inverse diagonal matrix. For a more detailed description of the CPD algorithm, please refer to [11].

In our method, we use the canonical shape $\mathbf{C}$ for the deforming template shape $\mathbf{S}^{[\mathbf{t}]}$ and each training example $\mathbf{T}_i$ as the reference point set $\mathbf{S}^{[\mathbf{r}]}$. Therefore, the transformations $\mathcal{T}_i$ are defined as

$$\mathcal{T}_i(\mathbf{C}, \mathbf{W}_i) = \mathbf{C} + \mathbf{G}\mathbf{W}_i \quad (8)$$

where $\mathbf{W}_i$ is the $\mathbf{W}$ matrix computed by taking training example $\mathbf{T}_i$ as the reference point set $\mathbf{S}^{[\mathbf{r}]}$.

C. Latent Space

CPD allows us to define a feature vector representing the deformation field. This vector has the same length for all training examples; additionally, elements in this vector correspond with the same elements in another. This allows us to learn a latent lower-dimensional space.

We observe from Eq. (8) that the deformation field between the canonical and an observed instance is fully determined by $\mathbf{G}$ and $\mathbf{W}$. Moreover, we see that $\mathbf{G}$ only requires the points of the canonical shape and it remains constant for all training examples. Therefore, the entire uniqueness of the deformation field for each training example is captured by its matrix $\mathbf{W}$.

We construct a row vector $\mathbf{y}_i \in \mathbb{R}^{p=M\cdot D}$ from each matrix $\mathbf{W}_i$ of each training example $\mathbf{T}_i$, that characterizes the

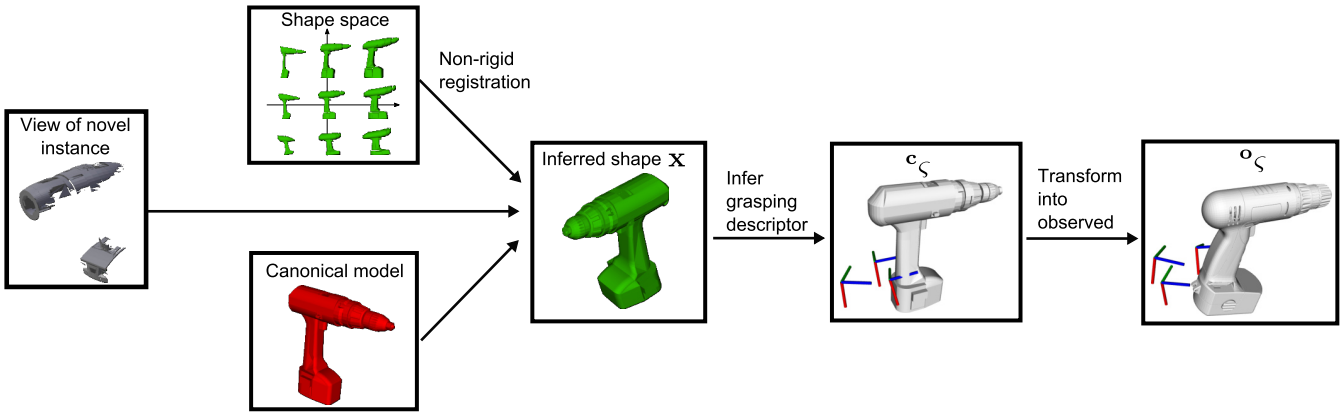


Figure 3: The canonical shape (red) is matched against a partially-occluded target shape (leftmost) by finding its latent shape parameters. The grasping descriptor ${}^{c}\boldsymbol{\varsigma}$ is inferred from $\mathbf{x}$. Finally, the descriptor is transformed to the observed space.

corresponding deformation field. The vectors are normalized to have zero-mean and unit-variance and are then assembled into a design matrix $\mathbf{Y}$. Finally, we find a lower-dimensional manifold of deformation fields for the category by applying the Principle Component Analysis Expectation Maximization (PCA-EM) algorithm on the matrix $\mathbf{Y}$.

Much like with CPD, we alternate between an E- and M-step. The E-step is given by:

$$\mathbf{X} = \mathbf{Y}\mathbf{L}^T(\mathbf{L}\mathbf{L}^T)^{-1} \tag{9}$$

whereas the M-step is defined by:

$$\mathbf{L} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{10}$$

$\mathbf{L}_{p\times q}$ is the resulting matrix of principle components. So, for a new normalized set of observations $\mathbf{Y_o}$, the latent variables can be found by postmultiplying $\mathbf{Y_o}$ by $\mathbf{L}$. In this manner, a deformation field is now described by only q latent parameters. Similarly, any point $\mathbf{x}$ in the latent space can be converted into a deformation field transformation by first postmultiplying $\mathbf{x}$ by $\mathbf{L}^T$ and by converting the result into a $\mathbf{W}_{M\times D}$ matrix after the respective denormalization. Thus, moving through the q-dimensional space linearly interpolates between the deformation fields.

D. Grasping Knowledge Aggregation

We aggregate grasping knowledge from different instances into the canonical model in two steps: first, by generating the grasping motion in the observed space and, second, by transforming its grasping descriptor into the canonical space.

A grasping motion is represented as a sequence of parametrized primitives each of them defined by a control pose expressed in the same coordinate system of the shape of the object. The generation of grasping motions can be performed manually for each instance in the training set, which favors accuracy over time and wear off of the system (on real robotic platforms). This imposes however a limit on the number of samples of the training dataset mostly because of time constraints. In order to overcome this limit, we adopt a constrained sample-based motion generation approach.

A sampled motion is created by generating constrained random 6D poses around the control poses of the canonical grasping motion as depicted in Figure 4. Each component of the translation is sampled from a normal distribution. For the rotation, a quaternion is build out of three uniformed points following the approach described in [27]. These orientations are filtered by specific functional constraints of each category, in the case of drills, for example, rotations that occlude or impede the use of the trigger are discarded. If the sampled grasping motion leads to collisions with other objects in the environment including the robotic arm, the motion is discarded as well. Finally, the sampled motion is executed and evaluated. If the object is functionally grasped successfully, the grasping control poses are transformed into the canonical space.

Finding the transformation from the observed space into the canonical space is equal to finding the inverse transformation of Eq. (2) or equivalently to finding the inverse transformation of Eq. (4). However, the inverse function v^{-1} is not directly available. It can nonetheless be estimated for a point $\mathbf{o}$ in the space of the observed shape using a set of points $\mathbf{Z} = (\mathbf{z}_1, ..., \mathbf{z}_M)^T$ in the canonical space which deform close to $\mathbf{o}$ by the equation:

$$v^{-1}(\mathbf{o}) = -\frac{\sum_{i=1}^{M} G(\mathbf{o}, \mathbf{z}_i + v(\mathbf{z}_i))v(\mathbf{z}_i)}{\sum_{i=1}^{M} G(\mathbf{o}, \mathbf{z}_i + v(\mathbf{z}_i))}. \tag{11}$$

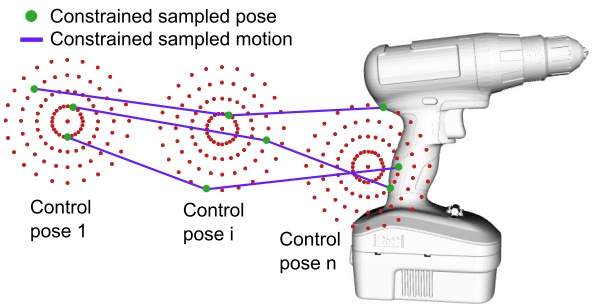


Figure 4: Sampled-based grasping motion generation. 6D constrained random poses are sampled around control poses of the canonical grasping motion.

For transforming the orientation, we apply Eq. (11) to the rotational vector base of each pose and orthonormalize it.

For each instance in our training dataset, we have so far a latent vector $\mathbf{x}_i$ that represents the shape deformations from the canonical instance to the observed instance and a grasping descriptor $\boldsymbol{\varsigma}_i$ transformed into the canonical space. We set the latent vector $\mathbf{x}_i$ as a feature vector and the grasping descriptor $\boldsymbol{\varsigma}_i$ as the corresponding target output and train a linear regression model. In other words, grasping knowledge is aggregated in the canonical model by serving as a training label of a regression model (Fig. 5). Algorithm 1 summarizes the training phase (Figure 2).

E. Shape Inference

A shape transformation is specified by the q parameters of the latent vector $\mathbf{x}_i$ plus additional seven parameters of a rigid transformation $\boldsymbol{\theta}_i$. The rigid transformation is meant to account for minor misalignments between the observed shape and the canonical shape at the global level.

We concurrently optimize for the latent parameters and the rigid transformation using gradient descent. As CPD and ICP, our method requires an initial coarse alignment of the observed shape because of the expected local minima. We want to find an aligned dense deformation field which when applied to the canonical shape $\mathbf{C}$ minimizes the distance to corresponding points in the observed shape $\mathbf{O}$. Specifically, we want to minimize the energy function:

$$E(\mathbf{x}, \boldsymbol{\theta}) = -\sum_{m=1}^{M} \log \sum_{n=1}^{N} \exp^{\frac{1}{2\sigma^2}\|\mathbf{O}_n - \Theta(\mathcal{T}_m(\mathbf{C}_m, \mathbf{W}_m(\mathbf{x})), \boldsymbol{\theta})\|^2} \tag{12}$$

where the function Θ applies the rigid transformation given parameters $\boldsymbol{\theta}$.

When a minimum is found, we can transform any point or set of points into the observed space by applying the deformation field using Eq. (4) and Eq. (2) and then applying the rigid transformation Θ. Moreover, CPD provides a dense deformation field, allowing us to find deformation vectors for novel points, even those added after the field is created.

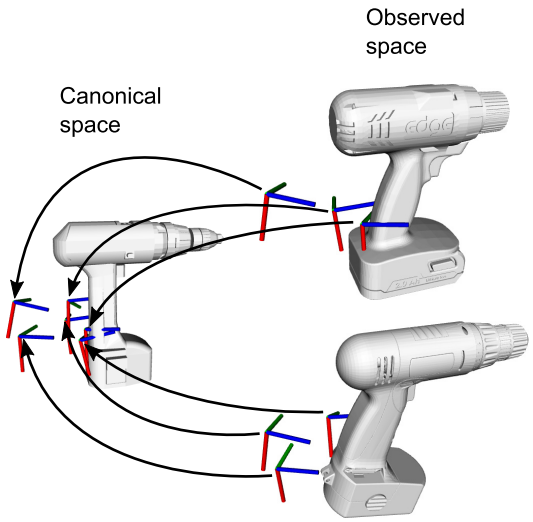


Figure 5: Grasping knowledge aggregation. Grasping descriptors of observed instances are transformed and aggregated in the canonical model by Eq. (11)

F. Transferring Grasping Skills

The transfer of grasping skills for novel instances is performed as follows. A latent vector $\mathbf{x}$ describing the shape deformation of the object from the canonical instance is calculated as explained in Section III-E. This vector constitutes a test sample of the linear regression, whose inference is a grasping descriptor ${}^{c}\boldsymbol{\varsigma}$. Then, ${}^{c}\boldsymbol{\varsigma}$ is transformed into the observed space. This transformation is performed in two steps. First, the control poses of the grasping motion are warped using Eq. (2) replacing $\mathbf{S}^{[t]}$ by the translational part and the rotational vector base of the control poses. Because the warping process can violate the orthogonality of the orientation, we orthonormalize the warped orientation. Second, we apply the rigid transformation Θ defined by the parameters $\boldsymbol{\theta}$.

The resulting transformed control poses ${}^{o}\boldsymbol{\varsigma}$ are expressed in the frame of the object. Thus, for executing the motion each of the poses has to be adapted relative to the pose of the observed object by premultiplying the control poses by the pose of the object w.r.t. the base of the manipulator. Algorithm 2 summarizes the inference of grasping skills.

IV. Setup and Evaluation

In this section, we evaluate only the grasping skill transfer because the latent space non-rigid registration method was already evaluated in [1]. We tested our method on two categories: *Drill* a *Spray Bottle*, containing 13 and 17 instances respectively. We obtained the object models from two online CAD databases: GrabCad [1] and 3DWarehouse[2]. The CAD models were converted into meshes in order to generate the input point clouds for our method. They were obtained by

Algorithm 1 Training phase

Input: A set of training shapes in their canonical pose with corresponding grasping descriptors ${}^{o}\boldsymbol{\varsigma}$.

1: Select a canonical shape $\mathbf{C}$ via heuristic or pick the one with the lower reconstruction energy.
2: Estimate the deformation fields between the canonical shape and the other training examples using CPD.
3: Concatenate the resulting set of $\mathbf{W}$ matrices from the deformation fields into a design matrix $\mathbf{Y}$.
4: Perform PCA-EM on the design matrix $\mathbf{Y}$ to compute the latent space of deformation fields $\mathbf{x}$.
5: Transform the grasping descriptors ${}^{o}\boldsymbol{\varsigma}$ into the canonical space ${}^{c}\boldsymbol{\varsigma}$.
6: Train the Linear Regressor $\mathcal{R} : \mathbf{x} \rightarrow {}^{c}\boldsymbol{\varsigma}$.

Output: A canonical shape $\mathbf{C}$, a latent space of deformation fields $\mathbf{L}$ and a trained model for inferring grasping descriptors $\mathcal{R}$.

[1]https://grabcad.com/library
[2]https://3dwarehouse.sketchup.com/

Algorithm 2 Grasping Skills Inference

Input: Transformation model ($\mathbf{C}$, $\mathbf{L}$), trained regressor $\mathcal{R}$ and observed shape $\mathbf{O}$

1: Use gradient descent to estimate the parameters of the underlying transformation ($\mathbf{x}$ and $\boldsymbol{\theta}$) until the termination criteria is met. To calculate the value of the energy function, in each iteration:
   - Using the current values of $\mathbf{x}$ and $\boldsymbol{\theta}$:
   1) Create vector $\hat{\mathbf{Y}}$ and convert it into matrix $\mathbf{W}$.
   2) Use Eq. (4) and Eq. (2) to deform $\mathbf{C}$.
   3) Apply the rigid transformation Θ to the deformed $\mathbf{C}$.
2: Use the resulting $\mathbf{x}$ to infer a grasping descriptor ${}^{c}\boldsymbol{\varsigma}$ inferred by $\mathcal{R}$.
3: Transform the grasping descriptor into the observed space.

Output: Grasping descriptor in observed space ${}^{c}\boldsymbol{\varsigma}$.

ray-casting from several viewpoints on a tessellated sphere and down-sampling with a voxel grid filter.

We use the five-fingered Schunk hand with a total of 9 fully actuated Degrees of Freedom (DoF) and 11 mimic joints. The experiments were carried out in the Gazebo simulation environment. The collision model of the finger links were modeled by capsules using an automatic ROS optimal capsule generator based on the Roboptim library [28] as shown in Fig. 6. The inertia tensors of the graspable objects were approximated using Meshlab. For building the shape latent spaces, we parametrized CPD with $\beta=1$, $\lambda=3$ and $\sigma^2 = 0.01$. The number of latent variables was set to capture at least 95% of the variance of each class. The grasping motions for each object in the training set were sampled as described in Section III-D with a maximum distance of 0.04 m and a maximum angular deviation of 0.2.

For each category, we select the canonical model manually. We use cross validation leaving two samples out. We trained six *drill* and seven *spray bottle* grasping transfer models. Because our method is able to infer category-alike geometries, we also evaluated our method with partially-observed point clouds. For this, we generate a single view of the test objects of each cross validation model. In total, we evaluated the method on 12 fully observed and 12 partially observed *drills* and 14 fully observed and 14 partially observed spray bottles. For each instance, one simulation trial was performed because the execution of the generated motion is fully deterministic in simulation.

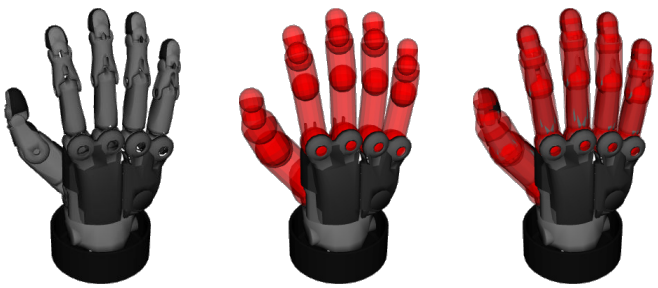


Figure 6: Visual and collision model of the robotic hand. At rightmost both models are displayed simultaneously to show the goodness of the capsule approximation.

TABLE I
RATIO OF SUCCESSFULLY TRANSFERED GRASPS.

	Drill		Spray Bottle	
	Grasp	Func. Grasp	Grasp	Func. Grasp
Fully observed	7/12	4/12	8/14	3/14
Partially observed	6/12	3/12	9/14	6/14

From the 52 instances to be grasped 30 were successfully grasped; that yields a success rate of 57.7%. Note, however, that a successfully grasped instance in our approach considers the entire motion, not only the last grasp configuration.

Regarding functional grasps, i.e., the index finger is able to trigger the tools, 16 instances were successfully grasped which results in a 32% success rate. The results are presented in Table I. Compared to the results presented in [22], although the success rate of our method is lower, our method is able to handle partially-occluded objects and an inference takes in average 7 s compared to the 12.6 min which is only suitable for offline applications. Figure 7 shows for each category two different—a fully observed and a partially-observed—samples that were successfully grasped.

Our method was also tested in real-robot experiments. We created only one latent transformation model for the *drill* category using all the 12 available meshes plus the canonical model. The observed object was inferred from one single view captured by the Kinect v2 sensor [29]. The tests were carried out on two different platforms: a UR10 arm and the CENTAURO robot. The hand was controlled by a PID position-current cascade controller, such that the joint position controller defines the desired joint currents. The saturation values of the current controller together with the PID values of the position controller were set to provide a certain level of compliance which contributed mainly at the last stage of the grasping motion. Using the UR10 robotic arm, our method was able to grasp two different drills twice without any failure. Similarly, with the CENTAURO robot, our approach grasped one instance of a drill twice without any failure (Fig. 8).

A video illustrating our approach is available online[3].

A. Discussion

Real experiments with two different robotic arms demonstrate that our method does not depend on the kinematics of the arm holding the hand. We assume however that the kinematics of the arm is able to reach 6D poses in its workspace. Our method is also agnostic to the robotic hand; a canonical grasping motion that is suitable to the hand is the only requirement for applicability.

Most of the grasping motions that failed exhibited a high deviation with respect to the canonical control poses which indicates a large variance in the learned transfer model. This suggests a need for more sample-efficient inference methods and the need for more training data.

[3]http://www.ais.uni-bonn.de/videos/RA-L_2018_Rodriguez

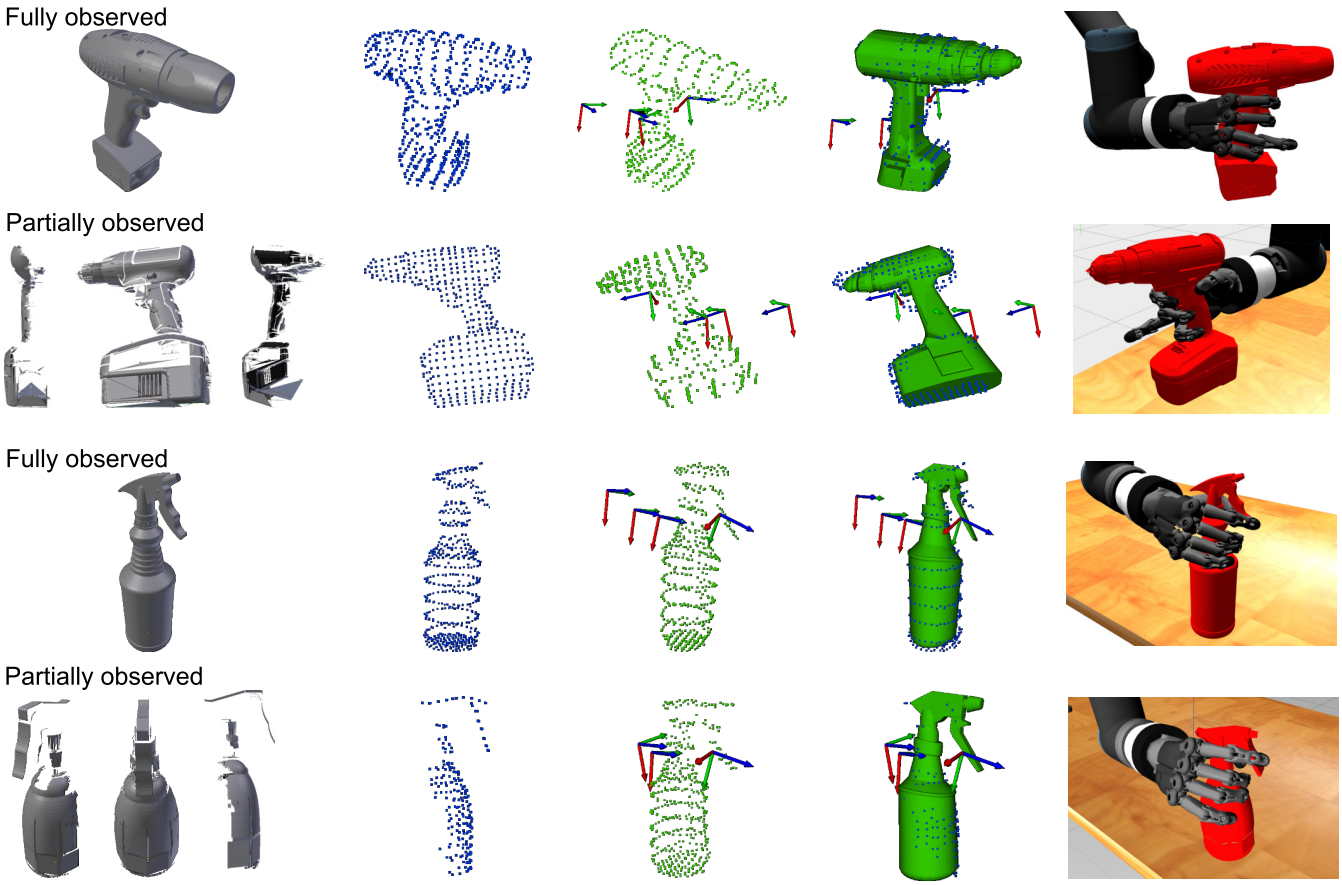


Figure 7: At the leftmost the meshes are shown. For illustration purposes we show additional perspectives of the same single view of the partially observed objects. The respective point clouds are shown in blue. The inferred instances (green point clouds) together with the transformed control points that define the motion are also displayed. In order to observe how good the inference matches the observed points, the mesh of the canonical models is transformed and displayed (green meshes) together with the observed data (blue points). Finally, the resulting grasped object in Gazebo is also depicted at the rightmost.

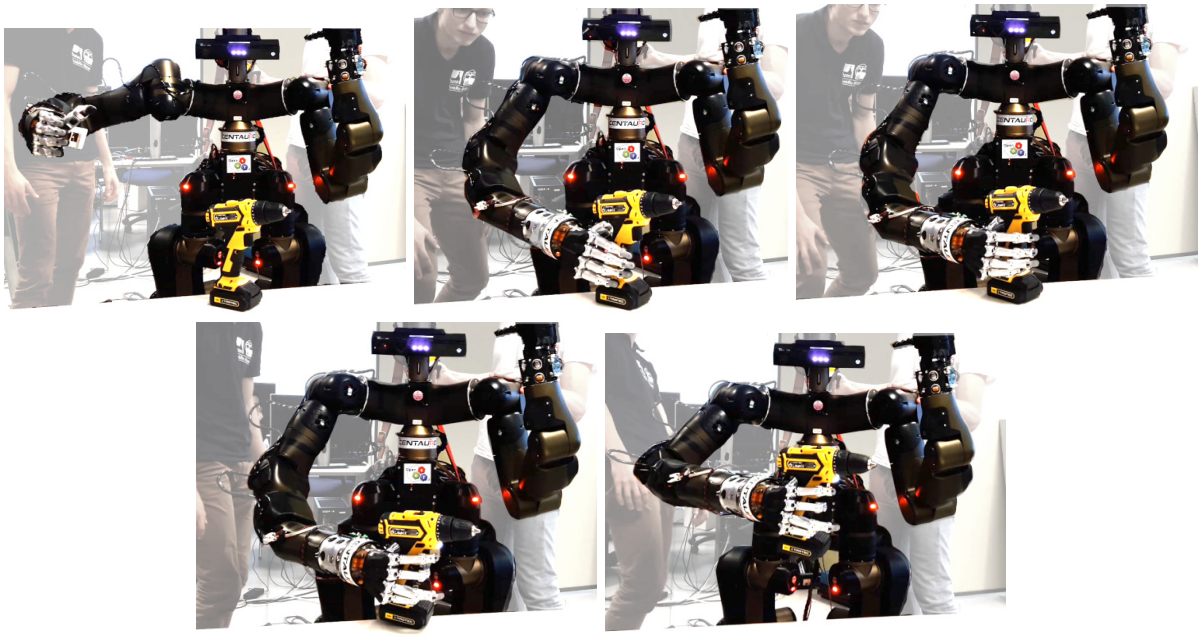


Figure 8: Experiments performed with the Centauro robot grasping autonomously a novel instance of a drill.

V. Conclusion

In this paper, we proposed a new approach of transferring grasping skills between objects within a category that is based on the knowledge aggregation of different training samples into a canonical model. Thanks to the learned latent shape space, our method is capable of completing missing or occluded object surfaces from partial views. Our method was able to transfer grasping skills with real robotic platforms from experiences collected only in simulation. This demonstrates the feasibility regarding the available sensory data (single-view point clouds) and runtime of our approach.

For future work, we want to consider more complex categories that impose higher variations in the joint configuration of the hand. So, more dimensionality reduction will be expected. As we realized the reduced number of training samples limits the presented approach, we start looking into automatic generation of plausible meshes from the canonical model. We also want to explore variants of the CPD algorithm in order to speed our current implementation. Finally, we would like also to exploit additional sensory modalities such as joint currents and force-torque sensors.

References

[1] D. Rodriguez, C. Cogswell, S. Koo, and S. Behnke, "Transferring grasping skills to novel instances by latent space non-rigid registration", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[2] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation", in *ACM Transactions on Graphics (TOG)*, vol. 21, 2002, pp. 362–371.

[3] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios, "Dense non-rigid surface registration using high-order graph matching", in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 382–389.

[4] V. G. Kim, Y. Lipman, and T. Funkhouser, "Blended intrinsic maps", in *ACM Transactions on Graphics (TOG)*, vol. 30, 2011, p. 79.

[5] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Efficient computation of isometry-invariant distances between surfaces", *SIAM Journal on Scientific Computing*, vol. 28, no. 5, pp. 1812–1836, 2006.

[6] A. Tevs, M. Bokeloh, M. Wand, A. Schilling, and H.-P. Seidel, "Isometric registration of ambiguous and partial data", in *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pp. 1185–1192.

[7] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas, "One point isometric matching with the heat kernel", in *Computer Graphics Forum*, Wiley Online Library, vol. 29, 2010, pp. 1555–1564.

[8] B. Allen, B. Curless, and Z. Popović, "The space of human body shapes: reconstruction and parameterization from range scans", in *ACM Transactions on Graphics (TOG)*, vol. 22, 2003, pp. 587–594.

[9] B. J. Brown and S. Rusinkiewicz, "Global non-rigid alignment of 3-d scans", in *ACM Transactions on Graphics (TOG)*, vol. 26, 2007, p. 21.

[10] D. Hahnel, S. Thrun, and W. Burgard, "An extension of the ICP algorithm for modeling nonrigid objects with mobile robots", in *18th International Joint Conference on Artificial Intelligence (IJCAI)*, 2003, pp. 915–920.

[11] A. Myronenko and X. Song, "Point set registration: Coherent point drift", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 12, pp. 2262–2275, 2010.

[12] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction", in *ACM Transactions on Graphics (TOG)*, vol. 28, 2009, p. 175.

[13] J. Süßmuth, M. Winter, and G. Greiner, "Reconstructing animated meshes from time-varying point clouds", in *Computer Graphics Forum*, Wiley Online Library, vol. 27, 2008, pp. 1469–1476.

[14] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling, "Efficient reconstruction of nonrigid shape and motion from real-time 3D scanner data", *ACM Transactions on Graphics (TOG)*, vol. 28, no. 2, p. 15, 2009.

[15] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 343–352.

[16] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, *et al.*, "Real-time non-rigid reconstruction using an RGB-D camera", *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 156, 2014.

[17] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel, "A statistical model of human pose and body shape", in *Computer Graphics Forum*, Wiley Online Library, vol. 28, 2009, pp. 337–346.

[18] O. Burghard, A. Berner, M. Wand, N. Mitra, H.-P. Seidel, and R. Klein, "Compact part-based shape spaces for dense correspondences", *ArXiv preprint arXiv:1311.7535*, 2013.

[19] F. Engelmann, J. Stückler, and B. Leibe, "Joint object pose estimation and shape reconstruction in urban street scenes using 3D shape priors", in *German Conference on Pattern Recognition (GCPR)*, 2016, pp. 219–230.

[20] N. Vahrenkamp, L. Westkamp, N. Yamanobe, E. E. Aksoy, and T. Asfour, "Part-based grasp planning for familiar objects", in *16th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 919–925.

[21] F. Ficuciello, D. Zaccara, and B. Siciliano, "Synergy-based policy improvement with path integrals for anthropomorphic hands", in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1940–1945.

[22] T. Stouraitis, U. Hillenbrand, and M. A. Roa, "Functional power grasps transferred through warping and replanning", in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4933–4940.

[23] U. Hillenbrand and M. A. Roa, "Transferring functional grasps through contact warping and local replanning", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 2963–2970.

[24] H. B. Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters, "Generalization of human grasping for multi-fingered robot hands", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 2043–2050.

[25] J. Stueckler, R. Steffens, D. Holz, and S. Behnke, "Real-time 3D perception and efficient grasp planning for everyday manipulation tasks.", in *European Conference on Mobile Robots (ECMR)*, 2011, pp. 177–182.

[26] A. L. Yuille and N. M. Grzywacz, "The motion coherence theory", in *Computer Vision, 2nd International Conference on (ICCV)*, IEEE, 1988, pp. 344–353.

[27] K. Shoemake, "Uniform random rotations", in *Graphics Gems*, Morgan Kaufmann, 1992, pp. 124–132.

[28] A. E. Khoury, F. Lamiraux, and M. Taix, "Optimal motion planning for humanoid robots", in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3136–3141.

[29] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling", in *International Conference on Advanced Robotics (ICAR)*, 2015, pp. 388–394.

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 onboard 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 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.

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.

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

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,

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. Szynkiewicz 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

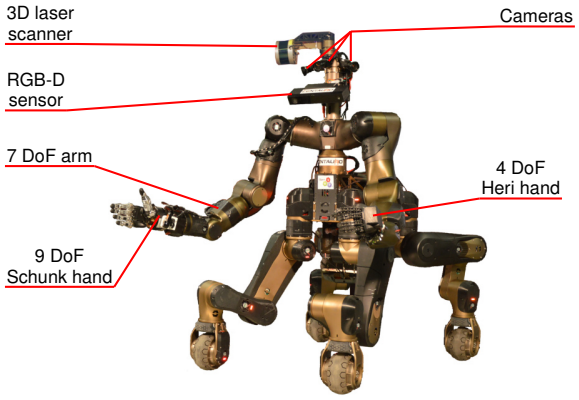


Fig. 2. The Centauro robot. The main components of the upper-body are labeled.

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 centraur-like robot, developed within the CENTAURO project[1]. The robot has a human-like upperbody, which is mounted on the qudrupedal 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/

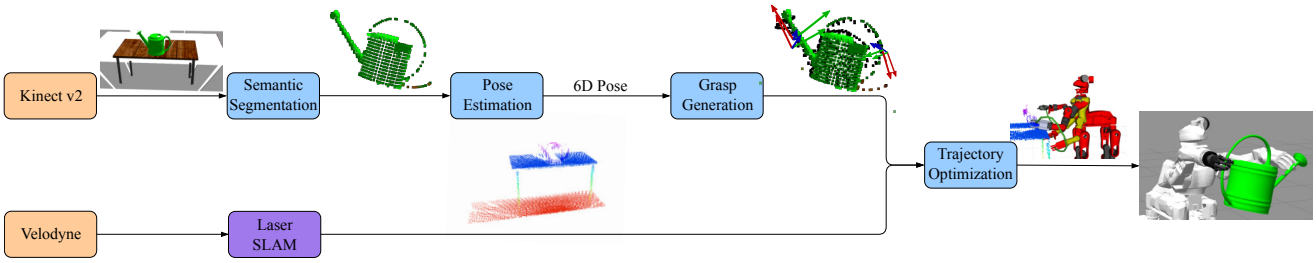


Fig. 3. Simplified diagram of the system, showing the information flow between core components. Orange: sensors; Blue: main components of the pipeline: Purple: external modules.

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 $\mathbf{C}$ 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 mean that the variations in shape from one object to the other

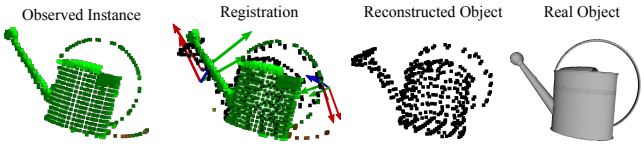


Fig. 4. Shape space registration on the watering can category. The method is able to reconstruct a partially occluded instance containing noise.

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 $\boldsymbol{\theta}_i \in \mathbb{R}^J, i \in \{0, \ldots, N-1\}$ in joint space with J joints. Normally, a naïve linear interpolation between the given start and goal configurations $\boldsymbol{\theta}_{start}$ and $\boldsymbol{\theta}_{goal}$ 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 $q_{cc}(.,.)$ is added to the cost function:

$$\begin{aligned} q(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) =& q_o(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) + q_l(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) + q_c(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) \\ &+ q_d(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) + q_t(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) + q_{cc}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}), \end{aligned} \tag{1}$$

where $q(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1})$ is a cost for the transition from the configuration $\boldsymbol{\theta}_i$ to $\boldsymbol{\theta}_{i+1}$. 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 $\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}$ 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:

$$q_{cc}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) = \tfrac{1}{2} \max_j q_{ct}(\boldsymbol{\theta}_j) + \tfrac{1}{2} \max_j q_{co}(\boldsymbol{\theta}_j), j \in \{i, \ldots, i+1\} \tag{2}$$

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, eef_1 and eef_2, the initial translation $t_{desired} \in \mathbb{R}^3$ between them is measured in the first configuration $\boldsymbol{\theta}_0$ of the trajectory. Then, for each evaluated configuration $\boldsymbol{\theta}_j$, the corresponding translation t_j between eef_1 and eef_2 is computed. The deviation from the desired translation is thus defined as: $\delta t = |t_{desired} - t_j|$. Finally, we select the largest component $t_{dev} = \max_{x,y,z} \delta t | \delta t = \langle x, y, z \rangle$ and compute the translation cost:

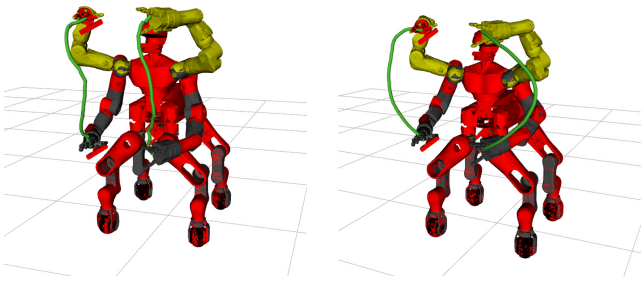
$$q_{ct}(\boldsymbol{\theta}_j) = \begin{cases} C_{ct} + C_{ct} \cdot t_{dev} & \text{if } t_{dev} \geq t_{max} \\ \frac{t_{dev}}{t_{max}}, & \text{otherwise} \end{cases}, \tag{3}$$

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 eef_1 and eef_2 is calculated in the first configuration $\boldsymbol{\theta}_0$. For each configuration $\boldsymbol{\theta}_j$, 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 $o_{dev} = \max_{r,p,y} \delta o | \delta o = \langle r, p, y \rangle$ and compute the orientation cost:

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

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.



(a) (b)

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.

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 Kubuntu 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%.

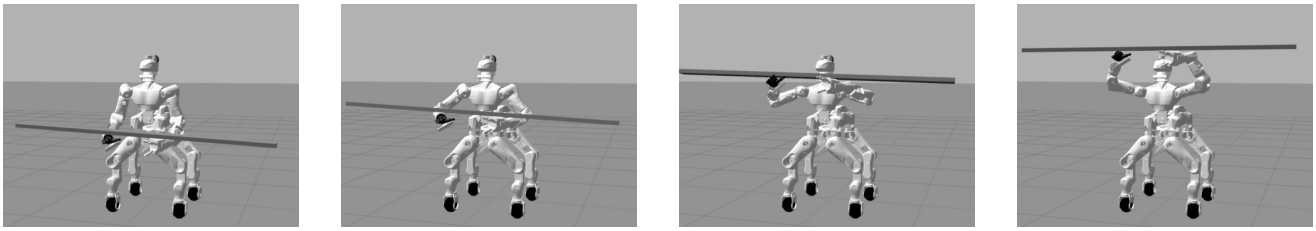


Fig. 6. The Centauro robot lifting a long bulky bar. As the bar is laying on the wrists unsecured, not only the closure constraint has to be preserved, but also the orientation of the end-effectors has to remain the same during the whole trajectory.

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%

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)	

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, 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

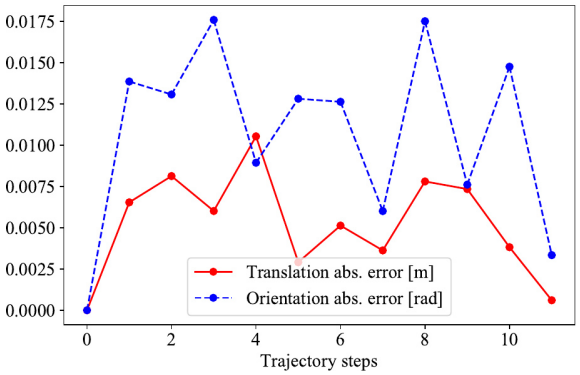


Fig. 7. Error between desired and observed end-effectors relative pose for trajectories shown in Fig. 5.

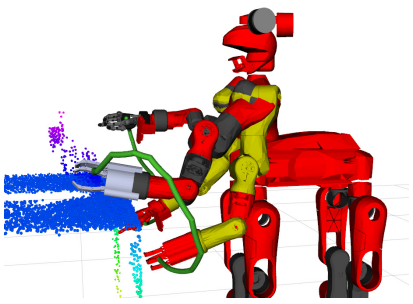
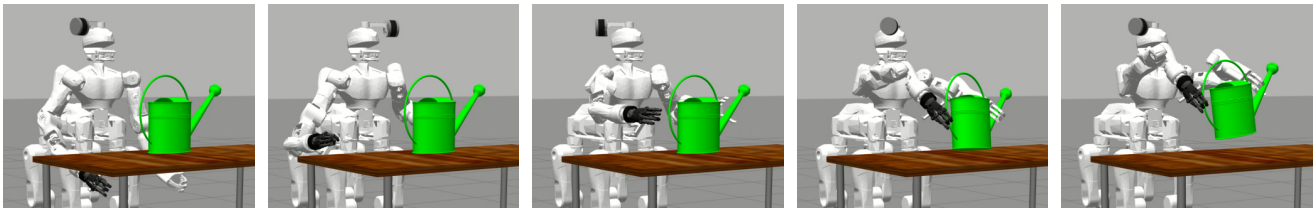


Fig. 8. Dual-arm trajectory for reaching pre-grasp poses. Yellow: initial pose; Black and grey: goal pose; Green: paths of the end-effectors. The arms have to retract back in order to avoid collisions with the table.



(a) (b) (c) (d) (e)

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.

(a) (b) (c)

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.

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 $\pm$ 0.69	—
Trajectory optimization	0.96 $\pm$ 0.29	100%
Complete pipeline	6.27 $\pm$ 0.98	80%

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. Consequetly, it is hard to assess the success rate of grasp generation as it 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.

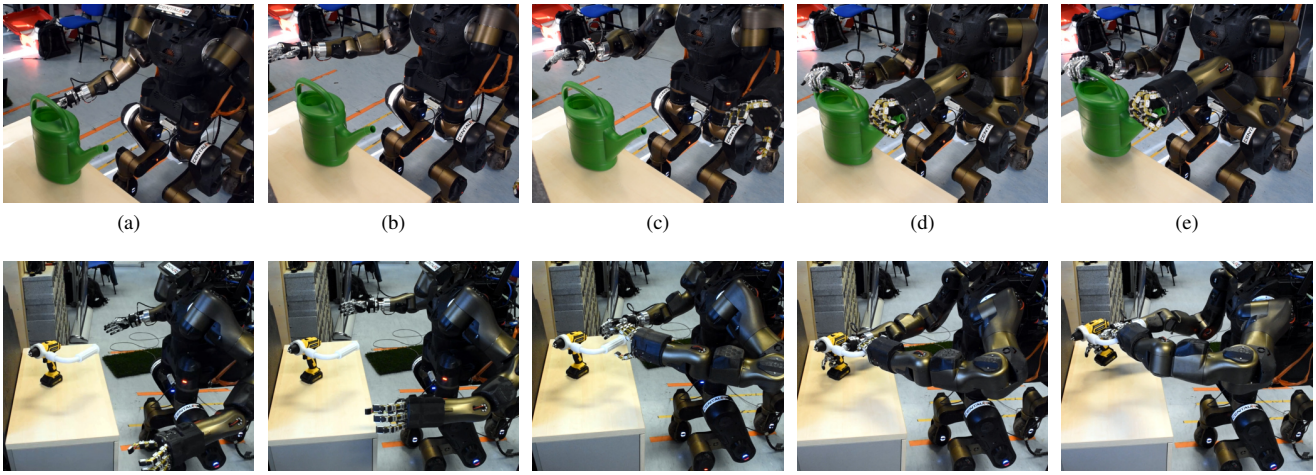
In addition to the watering can, the Centauro robot also grasped a two-handed drill to demonstrate that our pipeline can be applied to different types of objects. The process of grasping and lifting both tools is shown in Fig 11. Footages of the experiments can be found online[2].

VII. CONCLUSIONS

We have developed an integrated approach for autonomous dual-arm pick tasks of unknown objects of a know 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

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



(a) (b) (c) (d) (e)

(a) (b) (c) (d) (e)

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. Droeschel, 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. Stefos, K. Fourtakas, K. Kaltsoukalas, D. Kontrovrakis, 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. Szynkiewicz 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.