

A Skeleton-Based Approach to Grasp Known Objects with a Humanoid Robot

Markus Przybylski, Mirko Wächter, Tamim Asfour and Rüdiger Dillmann
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT)
Adenauerring 2, 76131 Karlsruhe, Germany
Email: {markus.przybylski,mirko.waechter,asfour,dillmann}@kit.edu

Abstract—This paper is about grasping known objects of arbitrary shape with a humanoid robot. We extend our previous work, where we presented a grasp planning method using an object representation based on the medial axis transform (MAT). The MAT describes an object’s topological skeleton and contains information about local symmetry properties and thickness valuable for grasp planning. So far, our previous work was only conducted in simulation. The contribution of this paper is the transfer of our grasp planning method to the real world. We present grasping experiments with challenging arbitrarily shaped objects where we execute the grasps generated by our grasp planner on a real humanoid robot with a five-finger hand.

I. INTRODUCTION AND RELATED WORK

Grasping objects is a challenging problem in robotics. Yet, it is a capability crucial for future service robots to be able to assist people in their daily lives.

Grasp planning deals with the problem of finding a hand pose relative to a known object and a joint angle vector of the fingers such that the object can be stably grasped. Due to its complexity, grasp planning is often performed using simulation environments ([1], [2], [3]), where various constraints such as hand kinematics, forces and object shape can be taken into account. A common approach is to use various heuristics to generate candidate grasps and to assess their quality using a metric for force-closure [4]. We give a short overview on the branches of research that are most relevant to our own work.

One branch of research focuses on dimensionality reduction. In this context, Ciocarlie et al. ([5],[6]) introduced the concept of *eigengrasps* which allows to perform grasp planning in subspaces of the hand configuration space.

A second branch of work focuses on approaches based on *shape matching* to plan grasps, sometimes using databases of grasps. Li et al. [7] used motion capturing to build the database, whereas Goldfeder et al. [8] used the eigengrasp planner for that purpose and extended his approach also to partial 3D data [9]. Saut et al. [10] presented an approach that uses precomputation for the hand’s inverse kinematics.

The third branch of research is based on the *grasping by parts* paradigm, where various approaches were presented that decomposed objects into parts in order to plan grasps on the individual parts, using a forward approach and squeeze method for the hand. The first method in this context used

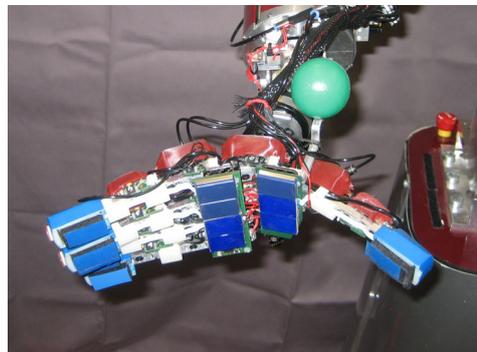


Fig. 1: ARMAR-IIIb’s right hand.



Fig. 2: Our test objects. We intentionally chose objects with challenging shapes to show the capabilities of our grasp planner.

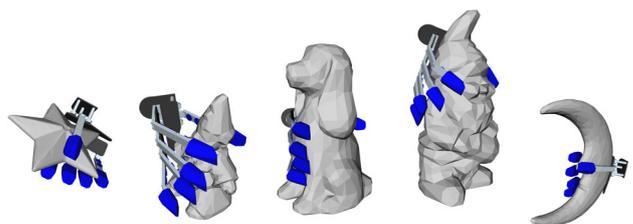


Fig. 3: Some example grasps generated by our grasp planning algorithm.

shape primitives [11] which was followed by approaches using superquadrics [12] and bounding boxes [13]. Two methods that also use the forward approach and squeeze framework but avoid shape decomposition were presented by Berenson et al. [14] who only considers surface normals of the shape and Roa et al. [15] who exploited regions of low curvature on the object’s surface for grasp planning. Aleotti et al. [16] proposed a method for grasp planning using the Reeb Graph.

In our previous work in the area of grasp planning, we followed the idea of the *grasping by parts* paradigm, but our goal was to avoid sacrificing geometrically meaningful candidate grasps due to poor shape approximation, as is the case when using shape primitives, bounding boxes or superquadrics. In order to achieve this goal, we chose to use inscribed spheres as a foundation for our grasp planning method. The first version of our grasp planner [17] analyzed slices of an object’s medial axis (MA) [18], using different heuristics to generate candidate grasps, depending on the MA slice types present in the object. While this worked well on a set of household objects, the limitation was that for objects with more complicated shapes, the user had to define additional slice types and heuristics to generate candidate grasps. In order to remove this limitation, we developed a second version of our grasp planner [19] based on the medial axis transform (MAT) which provides radius and object angle for each sphere, enabling the grasp planner to assess an object’s local thickness and to rate spheres according to their significance to grasp planning. As a result, this grasp planner can generate geometrically meaningful candidate grasps for arbitrarily shaped objects.

Yet, our previous work on grasp planning was only in simulation (see Fig. 3). In this paper, we use a real humanoid robot with a five-finger hand (see Fig. 1) to perform grasps generated by our grasp planner on objects with difficult shapes (see Fig. 2). However, in order to grasp known objects on a real robot, apart from grasp planning, some more modules need to interact. Therefore, besides the experiments on the robot, the second focus of this paper is on the integration aspects of the system, i.e. the interaction of the grasp planner, object pose estimation, inverse kinematics checks, visual servoing and hand control in order to perform a grasp.

This paper is organized as follows: In Section II, we present an overview of our complete system, describing the modules involved in the grasping pipeline. In Section III, we describe the humanoid robot platform we use for our experiments. In Section IV, we present experimental results. Finally, we conclude the paper in Section V with a discussion and some ideas for future research directions.

II. OUR APPROACH

In this section, we present our strategy to grasp known objects, using an offline approach based on the medial axis transform to plan grasps for mesh models. As we focus in this paper on the presentation of an integrated system, we

describe the individual modules only briefly. In each section, we refer the reader to our own previous work where more details on the algorithms and technical implementations are provided. Our system deals with the following subtasks:

- Object model acquisition, generating 3D mesh models of objects by means of a 3D laser scanner.
- Object identification and pose estimation based on computer vision methods, providing the object’s pose in a real scene. We emphasize that the focus here is not on vision, but that we want to use objects with various shapes in our experiments.
- Object shape approximation using the medial axis transform (MAT), providing information on valuable local symmetry properties of the object to be exploited by the grasp planner.
- Grasp planning, generating candidate grasps for the objects based on their MAT and testing them for force-closure.
- Grasp execution, using inverse kinematics checks and visual servoing to execute planned grasps on our humanoid robot.

A. Object model acquisition

In this paper, we perform grasping experiments on a set of objects with known geometry (see Fig. 2). For grasp planning and pose estimation, we need surface mesh models of these objects which we acquire using an interactive object modeling system ([20],[21]). The respective object is placed on a rotation plate in front of a Minolta VI-900 laser scanner which uses an active triangulation measurement method. Using various rotation angles of the plate, the laser scanner generates partial surface point clouds of the objects from different perspectives. These partial surface point clouds are registered, resulting in a triangulated mesh for each object. The object models used in this paper, among others, are also publicly available in the KIT ObjectModels Web Database [22].

B. Object identification and pose estimation

To be able to identify and estimate a 6D pose of the objects online, we use a shape-based recognition approach presented in detail in [23]. First, the object’s appearances are learned offline. We use the previously mentioned surface mesh models with a white texture to generate views of any rotation that should be recognized. The shape of any view is extracted, the data reduced using principal component analysis (PCA) and then stored in a database. For recognition, the same image processing steps as in the learning process are applied to the stereo camera images captured by the robot’s eyes to extract the shape of the single-colored object. To find the best match with the database a pattern matching with a so called universal eigenspace [24] is performed. With the stored rotation of the object in the database and the position of the object in the stereo images, the rotation and translation can be calculated to generate the needed 6D pose estimation.

C. Shape approximation using the medial axis transform

Instead of generating candidate grasps directly on an object’s surface mesh, we use an object representation based on the medial axis transform (MAT) for this purpose. In this Section and in Section II-D we briefly describe shape approximation and grasp planning based on the MAT, but we refer the interested reader to our previous work (see [19]) on this topic, where we explained these methods in greater detail.

We approximate arbitrary three-dimensional shapes by inscribing spheres of maximum diameter, which means that the spheres have to touch the original shape’s boundary at two or more different points. These spheres are called *medial spheres*. The *medial axis* (MA) denotes the union of these medial spheres’ centers. It was originally introduced by Blum [18] and provides a topological skeleton of an object. The *medial axis transform* (MAT) denotes the MA combined with the associated sphere radii. The benefit of the MAT is the fact that it is a complete shape descriptor, i.e. it can describe the original shape with arbitrary precision. This is very useful for grasp planning, as a grasp planner can easily exploit an object’s local symmetry properties contained in the MAT, while avoiding the pruning of potential high-quality candidate grasps from the search space due to poor geometry approximation, as is the case for methods based on bounding boxes [13], shape primitives [11] and superquadrics [12].

D. Candidate grasp generation and testing

For our grasp planning algorithm, the basic primitives for object geometry are the medial spheres. The parameters of a medial sphere are its center coordinates X , its radius r , the set of points P where it touches the surface and the object angle α_o (see [25]) which denotes the maximum angle included by two vectors from the sphere’s center to two different surface points $p_{si,j} \in P$. Fig. 4 shows a cross section of a box-shaped object with its medial axis (pink) and some medial spheres. The blue spheres have an object angle of $\alpha_{o,1} = 180^\circ$ and are located at the object’s central symmetry plane. The red spheres have an object angle of $\alpha_{o,2} = 90^\circ$. They are located at branches of the MA that describe the edges and corners of the object.

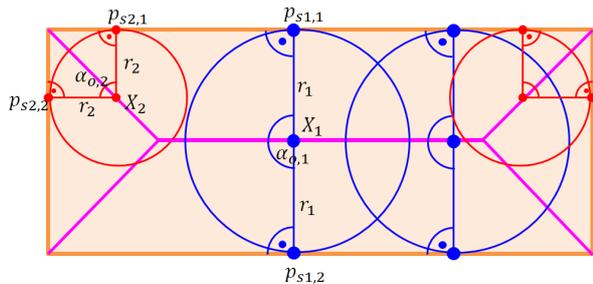


Fig. 4: Cross section of a box-shaped object with its medial axis (pink) and some medial spheres, with their centers X_i , radii r_i , object angles $\alpha_{o,i}$ and surface points $p_{si,j}$.

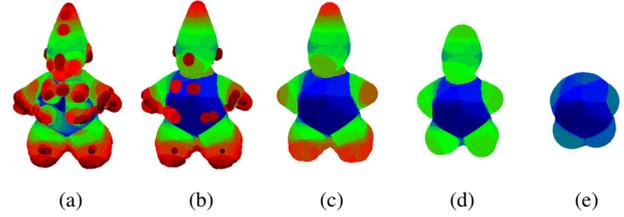


Fig. 5: Medial spheres as a representation for an object’s shape, colored according to the sphere radius. All spheres (a), only spheres with $\alpha_o \geq 120^\circ$ (b), additional lower bounds on minimum sphere radius r_{min} (c)-(e): $r_{min} = 0.3r_{max}$ (c), $r_{min} = 0.5r_{max}$ (d), $r_{min} = 0.7r_{max}$ (e)

Our algorithm is based on the idea that medial spheres with special properties should be considered for grasp planning. In order to generate candidate grasps for a medial sphere, an efficient way to access other spheres in its neighborhood is necessary. Therefore, we sort all the medial spheres into a three-dimensional grid structure, which gives us the possibility to perform spatial indexing.

The two main parameters describing a medial sphere’s significance for grasp planning are its object angle α_o and its radius r . Spheres with big object angles rather contribute to the volume of the object, while spheres with small object angles rather contribute to surface details. Fig. 5 shows one of our test objects represented by its medial spheres. The colors of the spheres range from red for the smallest spheres over yellow and green to blue for the biggest spheres.

The sphere radius is useful to determine where to grasp the object. On the one hand, spheres with big radii can often be used to generate power grasps, if their size is similar to the biggest sphere a hand can stably grasp. If a sphere is too big for the hand, it will not be considered for grasp planning, as in that case, the object is too thick to be grasped. On the other hand, small spheres may not be interesting for grasping, if bigger spheres are available somewhere else in the object. These smaller spheres often rather contribute to the surface details of the object, so sometimes it makes sense to discard them, especially if we want to generate power grasps on the bigger spheres. Yet, these spheres can be useful to generate precision grasps.

Fig. 5a shows all spheres of the object. Fig. 5b shows only spheres with object angles $\alpha_o \geq 120^\circ$. In addition to the restriction on α_o , Fig. 5c depicts only spheres with radii bigger than 30% of the biggest sphere radius in the object. Similarly, the threshold is 50% and 70% in Fig. 5d and Fig. 5e, respectively. As illustrated in Fig. 5, applying a lower bound of $\alpha_o \geq 120^\circ$ and discarding smaller spheres gradually reduces the object’s surface details, but preserves the object’s skeleton which is very useful for grasp planning, as we will see in the following.

Our grasp planning algorithm uses only spheres with object angles $\alpha_o \geq 120^\circ$. For each remaining medial sphere

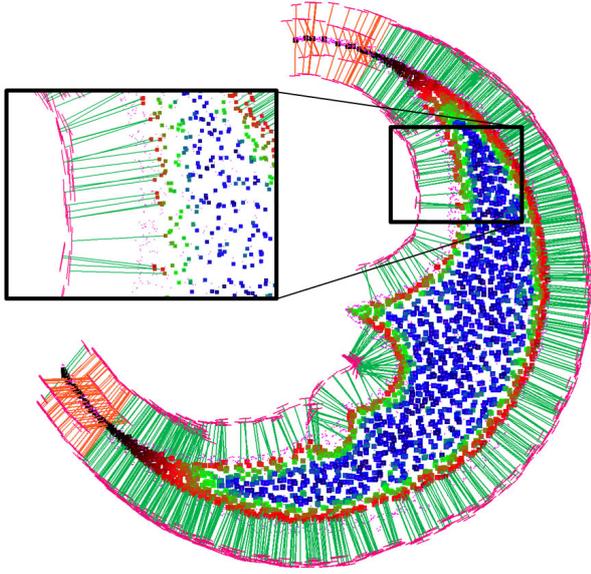


Fig. 6: Centers of the inscribed spheres and candidate grasps generated for the moon object. The dots indicate the sphere centers, colored according to their respective ρ_N values. Green and orange lines indicate hand approach directions toward the object; short magenta lines indicate hand orientation vectors.

s , we examine a local neighborhood N within a search radius r_N around s . We perform principal component analysis (PCA) on the sphere centers of s_N which yields the first two eigenvectors e_1, e_2 and the corresponding eigenvalues λ_1, λ_2 . Depending on the ratio of the eigenvalues

$$\rho_N = \frac{\lambda_2}{\lambda_1} \quad (1)$$

we classify each medial sphere to be located on a local symmetry axis, at the rim of a local symmetry plane or inside a local symmetry plane. For spheres on a local symmetry axis, we generate approach directions for the hand perpendicular to the symmetry axis, i.e. e_1 , and a hand orientation that enables the fingers to wrap around the symmetry axis. For spheres at a local symmetry plane's rim, we generate approach directions for the hand perpendicular to the rim (i.e. parallel to e_2), with hand orientations that make it likely that the fingers will establish contact with the object at opposing sides of the local symmetry plane, where e_1 is aligned to the local symmetry plane's rim. Spheres inside a local symmetry plane may not be reachable for grasping. Therefore, we do not generate any candidate grasps for these spheres. In all cases, the respective sphere's center is the grasp target point to be approached by the hand during grasping.

As an example, consider Fig. 6. It shows the centers of the inscribed medial spheres and the resulting candidate grasps. The sphere centers are depicted as dots with colors ranging from black over red, yellow and green to blue for increasing values of ρ_N . The green and orange rays pointing

towards the object are hand approach directions, where the green rays indicate approach directions originating from local symmetry planes and orange rays indicate approach directions originating from local symmetry axes. The short magenta lines at the end of the approach directions indicate the hand orientation vectors. As illustrated in Fig. 6, the hand approach directions and orientations are aligned with the object's central symmetry plane, so it is likely that the fingers of the robot hand will touch the object at opposing sides when closing the hand.

As a second example, consider Fig. 7 which shows approach directions of candidate grasps for the clown object, depending on the choice of the search radius r_N and the sphere radii to be considered for grasp planning. The clown object in Fig. 7 is 14.0cm tall. The top row of Fig. 7 shows approach directions for $r_N = 1.0cm$, whereas the bottom row shows approach directions for $r_N = 3.0cm$. In the leftmost column, spheres for candidate grasp generation were selected without regarding the sphere radius. In the other columns (from left to right), only spheres with $r \geq kr_{max}$, $k \in \{0.3, 0.5, 0.7\}$ were considered for candidate grasp generation, where $k = 0.7$ means that only spheres with at least 70% of the radius of the biggest sphere in the object were considered for candidate grasp generation. As can be seen from Fig. 7, increasing the value of k leads to thin parts of the object like the head, the hand and the feet being ignored and thick parts of the torso being preferred during candidate grasp generation. Comparing the upper and the lower row of Fig. 7, the difference is mainly in the approach directions. In case of the bigger search radius in the lower row, a bigger part of the object is considered for PCA, in the extreme case (lower right) resulting in various approach directions towards the left and the right side of the object.

In order to test the candidate grasp for force-closure, we use a testing procedure similar to the one proposed by Berenson et al. [14]. For each candidate grasp, we first set the hand into the grasp target point, with its palm facing the object, and a start orientation corresponding to the candidate grasp's approach direction and hand orientation vector. Then, we retract the hand from the object along the approach direction until there is no more inter-penetration between the object and the hand. Finally, we close the fingers around the object, determine the contact points and normals and compute the common force-closure measure [4] of the candidate grasp. We keep candidate grasps with force-closure for actual execution on the robot. For more details on the algorithm for candidate grasp generation based on the MAT, we refer the reader to our previous work in [19].

E. Grasp Execution

Now we have a collection of force-closure grasps generated by our grasp planner. Due to environmental restrictions and reachability constraints, not all of these grasps can be executed. The reachability of a specific grasp depends on the object's pose in the scene which we determine using the

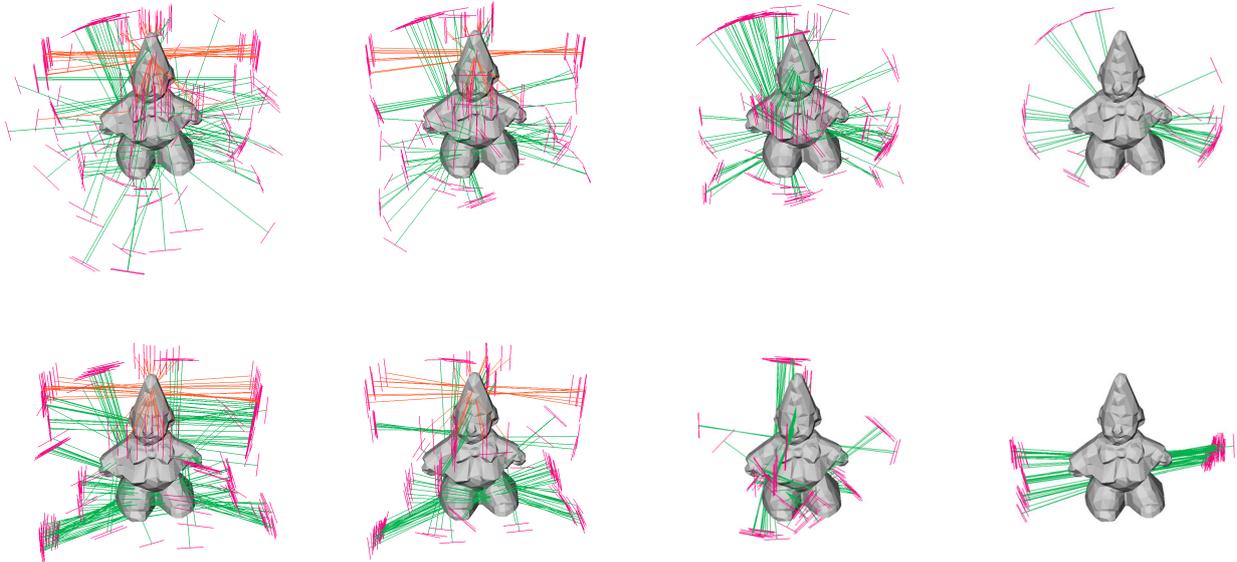


Fig. 7: Candidate grasps generated for the clown object for different parameter choices of r_N and r_{min} . Green and orange lines indicate approach directions of the hand, magenta lines indicate associated hand orientation vectors. Top row: $r_N = 1.0cm$. Bottom row: $r_N = 3.0cm$. 1st column: no lower bound on sphere radius. 2nd column: $r_{min} = 0.3r_{max}$. 3rd column: $r_{min} = 0.5r_{max}$. 4th column: $r_{min} = 0.7r_{max}$.

method described in Section II-B. Once the object pose is known, the grasps that were originally in an object-centered coordinate frame are transformed to the robot’s platform coordinate frame. Executing individual grasps on the robot consists of three steps: moving the hand to a pre-grasp pose, then moving it to the final pose, and finally closing the fingers of the hand. The pre-grasp pose is necessary to guarantee a collision-free movement of the hand to the final grasp pose, without pushing or toppling the object caused by undesired contacts between hand and object. The grasp pose is the final pose of the hand with respect to the object which was found by the grasp planner. We generate the pre-pose from the grasp pose by moving the hand a small distance along the approach direction away from the grasp pose. We use the Reachability Spaces method proposed by Vahrenkamp et al. [26] to filter out unreachable grasps. Reachable grasps can then be selected for actual execution. In order to compensate for inaccuracies in pose estimation and in the execution of the arm movements, we use a visual servoing approach as described in [27] where we observe both the object pose and the end effector pose to move the hand to the pre-pose and to the final grasp pose. At the grasp pose, we close the fingers according to the joint angles determined by the grasp planner and lift the object.

III. EXPERIMENTAL PLATFORM

We used the humanoid robot ARMAR-IIIb for our experiments (see [28] and Fig. 8). From a kinematics point of view,

ARMAR-IIIb consists of the following subsystems: The head, the torso, the two arms, and the platform. ARMAR-IIIb’s head has seven degrees of freedom (DoF). There are two cameras in each eye. The eyes have independent pan joints and a common tilt joint. The torso has one DoF which enables the robot to turn its upper body. Each arm has seven DoF: three DoF in the shoulder, two DoF in the elbow and two DoF in the wrist. The hand consists of five fingers with 8 DoF driven by pneumatic actuators [29]. For our experiments, we used ARMAR-IIIb’s right hand, which additionally provides joint encoders and pressure sensors, allowing for force position control of each DoF [30]. The hand has a total of eight DoF; one DoF for flexion of the palm, two DoF for flexion in the thumb, the index, and the middle finger, respectively, and one DoF for combined flexion of the ring and pinky finger.

IV. EXPERIMENTAL EVALUATION

In this section, we will demonstrate our method on the humanoid robot ARMAR-IIIb, grasping a set of test objects. We present experimental results for the set of objects depicted in Fig. 2. We chose these test objects due to their irregular shapes, and some of them have also considerable surface details. This makes grasp planning for these objects more difficult than it is in the case of common household objects that are typically used for grasping experiments. Our goal is to show that our grasp planning algorithm is able to generate feasible grasps even for objects with very irregular shapes.

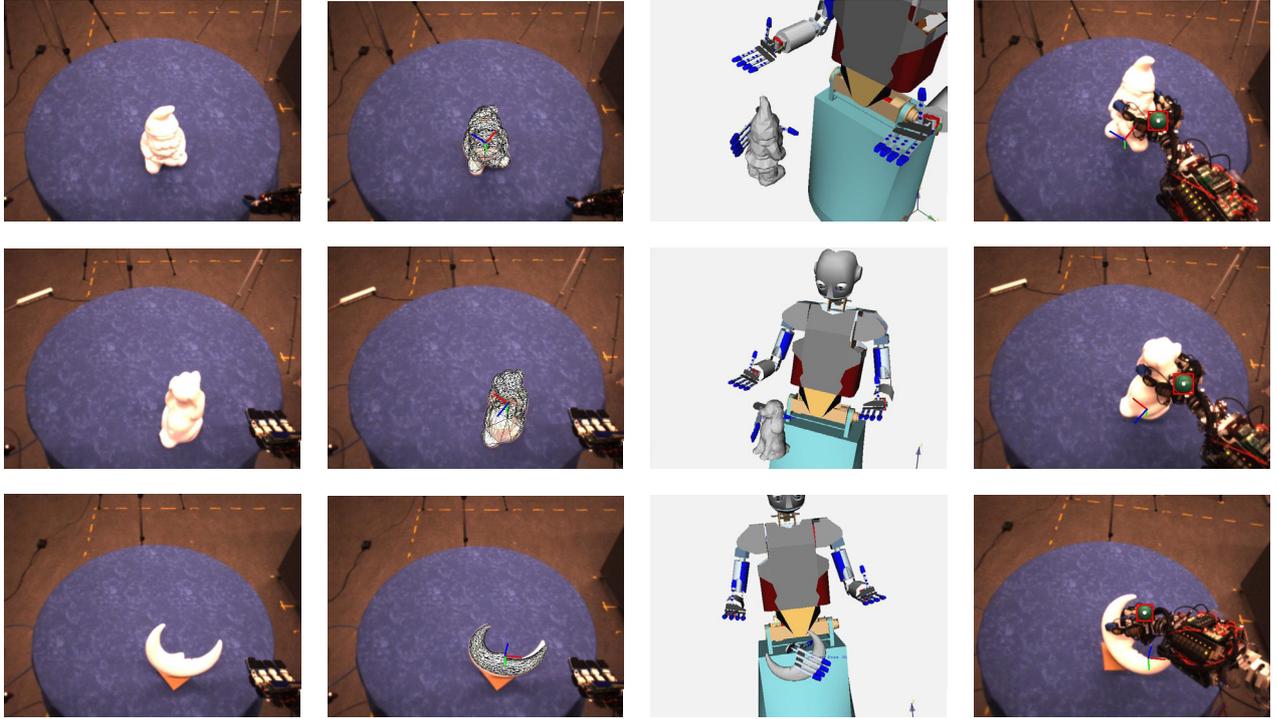


Fig. 9: Steps of the grasp execution. 1st column: Test object on the table. 2nd column: Object localization with object mesh and coordinate frame superimposed. 3rd column: selected candidate grasp. 4th column: ARMAR grasps and lifts the object.

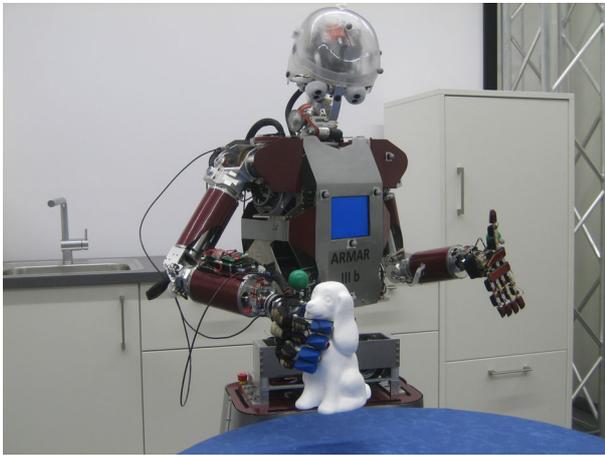


Fig. 8: ARMAR-IIIb grasps an object.

The test objects are small toy figures made of styrofoam: a dog, a lawn gnome, a moon, a star and a clown. The dog and the lawn gnome are big relative to ARMAR’s hand, with varying levels of thickness. The clown and the star have sizes comparable to the hand. The moon has an intermediate size and a smooth surface. The ways these objects can be grasped are restricted. It matters where to grasp the objects, from which direction to approach the hand and which hand orientation to use in order for the fingers to be able to wrap around the object.

During the preparation phase for the experiments, we

generated surface meshes of the objects with the 3D laser scanner, as explained in Section II-A and registered these meshes with the recognition system. For grasp planning, we computed the MAT of each object model as follows: Based on the surface point cloud of the mesh, we first generated the MA using the Tight Cocone tool [31]. Then, we reconstructed the MAT from the MA and the surface point cloud, using a simple search-based method [19]. Using each object’s MAT and surface mesh, we generated a set of candidate grasps as described in Section II-D and computed their force-closure ϵ score, keeping only the force-closure grasps for execution. For this purpose, we used a model of the ARMAR-IIIb hand and the OpenRAVE [2] simulator. The steps of the grasp execution are illustrated in Fig. 9, showing ARMAR’s camera images during the execution of candidate grasps and a virtual representation of the scene. For the online execution of the candidate grasps on the robot, we placed each object in front of the robot on a table (Fig. 9, 1st column). In case of the moon and the star, we fixed the object with wire on a small piece of cardboard in order to ensure an upright pose of the object. In case of the clown, we put the object on a small pedestal in order to reduce the risk of collision between the hand and the table. We localized the object using the object recognition and pose estimation system, transforming the previously object-centered candidate grasps to the actual scene in ARMAR’s platform coordinate frame (Fig. 9, 2nd column). Using the Reachability Spaces IK test, we checked the reachability of each candidate grasp’s pre

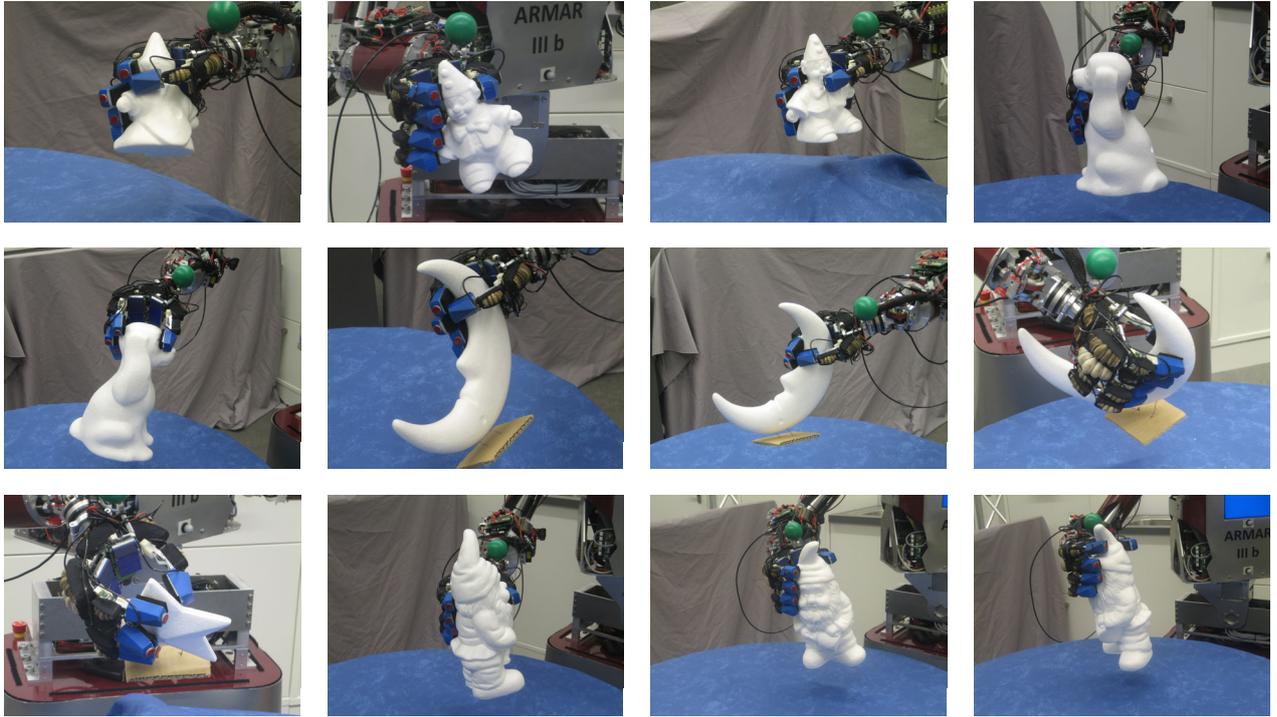


Fig. 10: Some example grasps we executed during our experiments.

pose and grasp pose, discarding candidate grasps outside ARMAR’s workspace. We manually selected some of the reachable candidate grasps for execution on ARMAR (Fig. 9, 3rd column). In order to grasp the objects, we first moved the hand to a collision-free configuration above the table. Then, using visual servoing, we moved the hand to the pre-pose, and finally to the grasp pose. At the grasp pose, we closed the fingers and lifted the object (Fig. 9, 4th column). In our experiments, ARMAR was able to grasp all of our test objects. Some of the resulting grasps are shown in Fig. 10. It has to be noted, that despite the use of visual servoing techniques, a residual uncertainty remains in the hand pose relative to the object, as both hand pose and object pose rely on vision data. Also, due to the pneumatic actuation of the fingers, the fingers’ final poses and therefore the contacts between the fingers and the object vary from the values predicted by the grasp planning simulation. As a combined effect, sometimes the object was pushed a bit by the palm or the fingers before grasping, and during closing the hand, the object sometimes turned a bit. Yet, our approach to grasp planning takes this into account, as we can focus on generating grasps for big spheres of the object, ignoring thin parts and surface details. This way, despite the pose uncertainties, the hand with its relatively high number of DoF is able to wrap around the object and successfully grasp it in most cases, because the approach direction and the hand orientation with respect to the object are geometrically meaningful in the sense that the fingers touch the object at opposing sides [32] and the hand is able

to squeeze the object. Also, in case of objects with obvious symmetry properties and a smooth surface, like the moon and the star, the grasps are aligned with the central symmetry planes, effectively favoring finger contacts in regions of low curvature during grasping.

V. DISCUSSION AND CONCLUSION

In this paper, we presented a strategy to grasp known objects based on the medial axis transform (MAT), a topological skeleton representation based on inscribed spheres, that contains information about an object’s local symmetry properties. We extended our previous work on grasp planning [19] and performed grasping experiments on a real humanoid robot. In order to execute the planned grasps, we integrated object pose estimation and visual servoing components, and obtained a system that can grasp objects with difficult shapes even in the presence of imperfect sensor data and limited actuator precision. The strength of our grasp planning approach is the fact that it can deal with arbitrarily shaped objects, and the usage of a complete shape descriptor avoids sacrificing potential high-quality candidate grasps due to poor geometry approximation. The grasp planner is able to identify parts of the object where the object’s local thickness is suitable for grasping and generates candidate grasps accordingly. The possibility to discard spheres depending on their contribution to the object’s shape enables the planner to focus on the graspable parts of the object and to generate grasps even for objects with many surface details. Possible future research directions may include the execution of precision grasps.

This may be possible by the use of fingertip tracking [33] in order to increase the precision of finger placement.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement no. 270273 (Xperience). We wish to thank Professor Tamal Dey from The Ohio State University for providing his Tight Cocone software and Rosen Diankov from Carnegie Mellon University for excellent support in all questions regarding the OpenRAVE simulator. We wish to thank Alexander Kasper for his help during scanning the test objects.

REFERENCES

- [1] A. T. Miller and P. K. Allen, "Graspit! A versatile simulator for robotic grasping," *Robotics Automation Magazine, IEEE*, vol. 11, no. 4, pp. 110–122, 2004.
- [2] R. Diankov and J. Kuffner, "OpenRAVE: A Planning Architecture for Autonomous Robotics," Robotics Institute, Tech. Rep. CMU-RI-TR-08-34, 2008. [Online]. Available: <http://openrave.programmingvision.com>
- [3] [Online]. Available: <http://www.sourceforge.net/projects/simox>
- [4] C. Ferrari and J. Canny, "Planning optimal grasps," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1992, pp. 2290–2295 vol.3.
- [5] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 3270–3275.
- [6] M. Ciocarlie and P. Allen, "Hand posture subspaces for dexterous robotic grasping," *Int. Journal of Robotics Research*, vol. 28(7), pp. 851–867, 2009.
- [7] Y. Li, J. L. Fu, and N. S. Pollard, "Data-Driven Grasp Synthesis Using Shape Matching and Task-Based Pruning," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 4, pp. 732–747, 2007.
- [8] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The Columbia grasp database," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 1710–1716.
- [9] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen, "Data-driven grasping with partial sensor data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 1278–1283.
- [10] J.-P. Saut and D. Sidobre, "Efficient models for grasp planning with a multi-fingered hand," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 347–357, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889011001515>
- [11] A. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2003, pp. 1824–1829 vol.2.
- [12] C. Goldfeder, C. Lackner, R. Pelossof, and P. K. Allen, "Grasp Planning via Decomposition Trees," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 4679–4684.
- [13] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 1628–1633.
- [14] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2007, pp. 42–48.
- [15] M. A. Roa, M. J. Argus, D. Leidner, C. Borst, and G. Hirzinger, "Power Grasp Planning for Anthropomorphic Robot Hands," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- [16] J. Aleotti and S. Caselli, "Grasp Synthesis by 3D Shape Segmentation Using Reeb Graphs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.
- [17] M. Przybylski, T. Asfour, and R. Dillmann, "Unions of Balls for Shape Approximation in Robot Grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.
- [18] H. Blum, *Models for the Perception of Speech and Visual Form*. Cambridge, Massachusetts: MIT Press, 1967, ch. A transformation for extracting new descriptors of shape, pp. 362–380.
- [19] M. Przybylski, T. Asfour, and R. Dillmann, "Planning Grasps for Robotic Hands using a Novel Object Representation based on the Medial Axis Transform," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.
- [20] A. Kasper, R. Becher, P. Steinhaus, and R. Dillmann, "Developing and Analyzing Intuitive Modes for Interactive Object Modeling," in *International Conference on Multimodal Interfaces*, 2007.
- [21] R. Becher, P. Steinhaus, R. Zöllner, and R. Dillmann, "Design and Implementation of an Interactive Object Modelling System," in *International Symposium on Robotics*, 2006.
- [22] "KIT ObjectModels Web Database: Object Models of Household Items." [Online]. Available: <http://i61p109.ira.uka.de/ObjectModelsWebUI>
- [23] P. Azad, T. Asfour, and R. Dillmann, "Accurate Shape-based 6-DoF Pose Estimation of Single-colored Objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.
- [24] H. Murase and S. Nayar, "Learning and Recognition of 3D Objects from Appearance," in *IEEE Workshop on Qualitative Vision*, 1993, pp. 39–50.
- [25] B. Miklos, J. Giesen, and M. Pauly, "Discrete scale axis representations for 3D geometry," in *ACM SIGGRAPH 2010 papers*, ser. SIGGRAPH '10. New York, NY, USA: ACM, 2010, pp. 101:1–101:10.
- [26] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid Motion Planning for Dual-Arm Manipulation and Re-Grasping Tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 2464–2470.
- [27] N. Vahrenkamp, S. Wieland, P. Azad, D. Gonzalez-Aguirre, T. Asfour, and R. Dillmann, "Visual Servoing for Humanoid Grasping and Manipulation Tasks," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2008, pp. 406–412.
- [28] T. Asfour, K. Regensteiner, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2006.
- [29] I. Gaiser, S. Schulz, A. Kargov, H. Klosek, A. Bierbaum, C. Pylatiuk, R. Oberle, T. Werner, T. Asfour, G. Bretthauer, and R. Dillmann, "A New Anthropomorphic Robotic Hand," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Daejeon, Korea, 2008.
- [30] A. Bierbaum, J. Schill, T. Asfour, and R. Dillmann, "Force Position Control for a Pneumatic Anthropomorphic Hand," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Paris, France, 2009, pp. 21–27.
- [31] T. Dey and S. Goswami, "Tight cocone: a water-tight surface reconstructor," in *Proceedings of the eighth ACM symposium on Solid modeling and applications*. ACM, 2003, pp. 127–134.
- [32] T. Iberall, "Human Prehension and Dexterous Robot Hands," *The International Journal of Robotics Research*, vol. 16, no. 3, pp. 285–299, 1997. [Online]. Available: <http://dx.doi.org/10.1177/027836499701600302>
- [33] M. Do, T. Asfour, and R. Dillmann, "Particle Filter-Based Fingertip Tracking with Circular Hough Transform Features," in *IAPR Machine Vision Applications (MVA'11)*, Nara, Japan, 2011.