We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Robotic Grasping of Unknown Objects

Mario Richtsfeld and Markus Vincze
Institute of Automation and Control
Vienna University of Technology
Gusshausstr. 27-29, Vienna, Austria

1. Introduction

"People have always been fascinated by the exquisite precision and flexibility of the human hand. When hand meets object, we confront the overlapping worlds of sensorimotor and cognitive functions" (Castiello, 2005). In the last few decades, the problem of grasping and manipulation of unknown objects in a fully automatic way has gained increasing importance, due mainly to the wide-spread use of service and rehabilitation robotics (Casals & Merchan, 1999), (Martens & Ruchel, 2001), (Ivlev & Martens, 2005). The grasping task has been studied from a psychological, biological and engineering focus but is still unresolved. There exist different solutions for certain cases; however there is still no general valid solution.

Automatic grasping of unknown objects from a single-view is a difficult problem because the pose and shape of the object are unknown and the possible hand configurations to realize a stable grasp are numerous. We propose a new strategy for modelling and grasping unknown objects based on 3D reconstruction. The system consists of a laser-range scanner, a simulation environment, a 7 DOF robot arm and a multi-fingered robot hand. The object to be grasped is scanned by a laser-range scanner from only one single-view and reconstructed in simulation. During the scanning process the system gets important information about the object structure, shape, place, size, and orientation of the object. We present a method for segmentation of a 2.5D point cloud into parts, assembly of rotationally symmetric parts into objects, and calculation of grasping points and poses. The algorithm checks potential collisions between the gripper, the object to be grasped, all surrounding objects and the table top. The experimental results demonstrate the effectiveness of the proposed method to grasp a wide range of objects.

The outline of the proposed work is as follows: Section 2 introduces our robotic system and its components. Section 3 describes the segmentation of 2.5D point clouds into parts, the assembly of parts into objects and details the merging of clipped rotationally symmetric objects. Section 4 details the calculation of grasping points for rotationally symmetric objects and optimal hand poses for arbitrary objects to grasp and manipulate an object without collision. Section 5 shows the achieved results and Section 6 finally concludes the work.

1.1 Problem Statement and Contribution

The goal of this work is to show a robust way of calculating possible grasps for unknown objects despite of noise, outliers and shadows. From a single-view two shadows appear: one from the camera and another one from the laser which can be caused by specular or reflective surfaces. We calculate collision free hand poses with a 3D model of the used gripper to grasp the objects, as illustrated in Fig. 1. That means that occluded objects can not be analyzed or grasped.

Fig. 1. Detected grasping points and hand poses. The green points display the grasping points for rotationally symmetric objects. The red points show an alternative grasp along the top rim. The illustrated hand poses show a possible grasp for the remaining graspable objects[1].

The problem of automatic 2.5D reconstruction to get practical grasping points and poses consists of several challenges. One of these concerns that an object might be detected as several disconnected parts, due to missing sensor data from shadows or poor surface reflectance. From a single-view the rear side of an object is not visible due to self occlusions, and the front side may be occluded by other objects. The algorithm was developed for arbitrary objects in different poses, on top of each other or side by side with a special focus on rotationally symmetric objects. If objects can not be separated because they are stacked one of each other they are considered as one object. If the algorithm detects rotationally symmetric parts (hypothesizing that the parts belong to the same object) this parts are merged, because this object class can be robustly identified and allows a cylindrical grasp as well as a tip grasp along the top rim (Schulz et al., 2005). For all other objects the algorithm calculates a tip grasp based on the top surface. To evaluate the multi-step solution procedure, we use 18 different objects presented in Fig. 2.

---

[1] All images are best viewed in colour!

Fig. 2. 18 different objects were selected to evaluate the grasp point and grasp pose detection algorithm, from left: 1. Coffee Cup (small), 2. Saucer, 3. Coffee Cup (big), 4. Cube, 5. Geometric Primitive, 6. Spray-on Glue, 7. Salt Shaker (cube), 8. Salt Shaker (cylinder), 9. Dextrose, 10. Melba Toast, 11. Amicelli, 12. Mozart, 13. Latella, 14. Aerosol Can, 15. Fabric Softener, 16. C-3PO, 17. Cat, 18. Penguin.

1.2 Related Work

In the last few decades the problem of grasping novel objects in a fully automatic way has gained increasing importance in machine vision. (Fagg & Arbib, 1998) developed the FARS model, which focuses especially on the action-execution step. Nevertheless, no robotic application has been yet developed following this path. (Aarno et al., 2007) presented an idea that the robot should, like a human infant, learn about objects by interacting with them, forming representations of the objects and their categories.

(Saxena et al., 2008) developed a learning algorithm that predicts the grasp position of an object directly as a function of its image. Their algorithm focuses on the task of identifying grasping points that are trained with labelled synthetic images of a different number of objects.

(Kragic & Bjorkman, 2006) developed a vision-guided grasping system. Their approach was based on integrated monocular and binocular cues from five cameras to provide robust 3D object information. The system was applicable to well-textured, unknown objects. A three fingered hand equipped with tactile sensors was used to grasp the object in an interactive manner. (Bone et al., 2008) presented a combination of online silhouette and structured-light 3D object modelling with online grasp planning and execution with parallel-jaw grippers. Their algorithm analyzes the solid model, generates a robust force closure grasp and outputs the required gripper pose for grasping the object. They consider the complete 3D model of one object, which will be segmented into single parts. After the segmentation step each single part is fitted with a simple geometric model. A learning step is finally needed in order to find the object component that humans choose to grasp it.

(Stansfield, 1991) presented a system for grasping 3D objects with unknown geometry using a Salisbury robotic hand, whereby every object was placed on a motorized and rotated table under a laser scanner to generate a set of 3D points. These were combined to form a 3D model. (Wang & Jiang, 2005) developed a framework of automatic grasping of unknown objects by using a laser-range scanner and a simulation environment. (Boughorbel et al.,

2007) aid industrial bin picking tasks and developed a system that provides accurate 3D models of parts and objects in the bin to realize precise grasping operations, but their superquadrics based object modelling approach can only be used for rotationally symmetric objects. (Richtsfeld & Zillich, 2008) published a method to calculate possible grasping points for unknown objects with the help of the flat top surfaces of the objects based on a laser-range scanner system. However there exist different approaches for grasping quasi planar objects, (Sanz et al., 1999). (Huebner et al., 2008) developed a method to envelop given 3D data points into primitive box shapes by a fit-and-split-algorithm with an efficient minimum volume bounding box. These box shapes give efficient clues for planning grasps on arbitrary objects. Another 3D model based work is presented by (El-Khoury et al., 2007).

(Ekvall & Kragic, 2007) analyzed the problem of automatic grasp generation and planning for robotic hands where shape primitives are used in synergy to provide a basis for a grasp evaluation process when the exact pose of the object is not available. The presented algorithm calculates the approach vector based on the sensory input and in addition tactile information that finally results in a stable grasp. (Miller et al., 2004) developed an interactive grasp simulator "GraspIt!" for different hands, hand configurations and objects. The method evaluates the grasps formed by these hands. At the beginning this work uses shape primitives, by modelling an object as a sphere, cylinder, cone or box (Miller et al., 2003). Their system uses a set of rules to generate possible grasp positions. This grasp planning system "GraspIt!" is used by (Xue et al., 2008). They use the grasp planning system for an initial grasp by combining hand pre-shapes and automatically generated approach directions. Their approach is based on a fixed relative position and orientation between the robotic hand and the object, all the contact points between the fingers and the object are efficiently found. A search process tries to improve the grasp quality by moving the fingers to its neighboured joint positions and uses the corresponding contact points to the joint position to evaluate the grasp quality and the local maximum grasp quality is located. (Borst et al., 2003) show that it is not necessary in every case to generate optimal grasp positions, however they reduce the number of candidate grasps by randomly generating hand configuration dependent on the object surface. Their approach works well if the goal is to find a fairly good grasp as fast as possible and suitable. (Goldfeder et al., 2007) presented a grasp planner which considers the full range of parameters of a real hand and an arbitrary object, including physical and material properties as well as environmental obstacles and forces.

(Recatalá et al., 2008) created a framework for the development of robotic applications on the synthesis and execution of grasps. (Li et al., 2007) presented a data driven approach to realize a grasp synthesis. Their algorithm uses a database of captured human grasps to find out the best grasp by matching hand shape to object shape.

Summarizing to the best knowledge of the authors in contrast to the state of the art reviewed above our algorithm works only with 2.5D point clouds from a single-view. We do not operate on a motorized and rotated table, which is unrealistic for real world use. The segmentation and merging step identifies different objects in the same table scene. The presented algorithm works on arbitrary objects and calculates especially for rotationally symmetric objects grasping points. For all other objects the presented method calculates possible grasping poses based on the top surfaces with a 3D model of the gripper. The algorithm checks potential collision with all surrounding objects. In most cases the shape

information recovered from a single-view is too limited (missing rear side of the objects) that we do not attend to calculate force-closure grasps.

2. System Design and Architecture

The system consists of a pan/tilt-mounted red-light laser, a scanning camera and a seven degrees of freedom robot arm from AMTEC robotics[2], which is equipped with a human like prosthesis hand from OttoBock[3], see Fig. 3a.
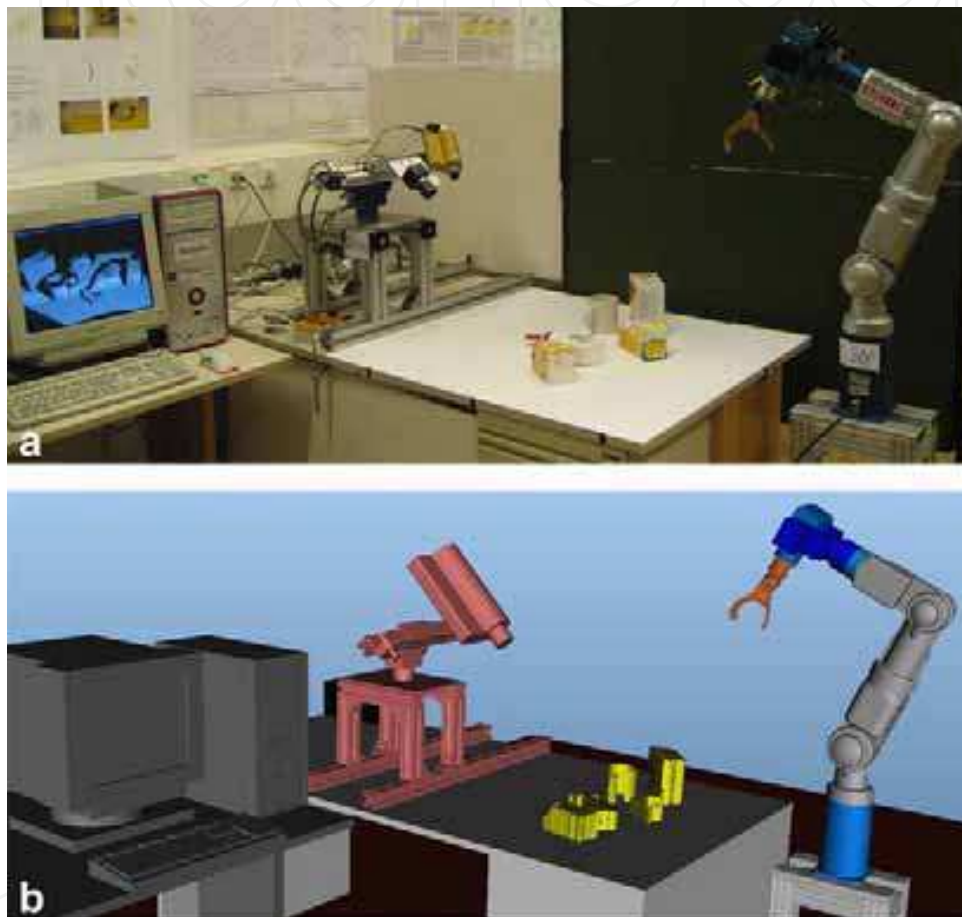

Fig. 3. **a** Overview of the system components and their interrelations. **b** Visualization of the experimental setup by a simulation tool, which is suitable to calculate the trajectory of the robot arm. The closed rear side of the objects on the table by an approximation of 2.5D to 3D is clearly visible.

First, the laser-range system scans the table scene and delivers a 2.5D point cloud. A high resolution sensor is needed in order to detect a reasonable number of points of the objects with sufficient accuracy. We use a red-light LASIRIS laser from StockerYale[4] with 635nm

---

[2] http://www.amtec-robotics.de/

[3] http://www.ottobock.de/

[4] http://www.stockeryale.com/index.htm

and a MAPP2500 CCD-camera from SICK-IVP[5] mounted on a PowerCube Wrist from AMTEC robotics.

The prosthesis hand has three active fingers: the thumb, the index finger, and the middle finger; the last two fingers are just for cosmetic reasons. The integrated tactile sensors are used to detect the sliding of objects to initialize a readjustment of the pressure of the fingers. It is thought that people will accept this type of gripper rather than an industrial gripper, due to the form and the optical characteristics. The virtual centre between the fingertip of the thumb, the index and the last finger is defined as tool centre point (TCP). The seventh degree of freedom of the robot arm is a rotational axis of the whole hand and is required to enable complex object grasping and manipulation and to allow for some flexibility for avoiding obstacles. There is a defined pose between the AMTEC robot arm and the scanning unit. A commercial path planning tool by AMROSE[6] calculates a collision free path to grasp the object. Before the robot arm delivers the object, the user can check the calculated trajectory in a simulation sequence, see Fig. 3b. Then the robot arm executes the off-line programmed trajectory. The algorithm is implemented in C++ using the Visualization Tool Kit (VTK)[7].

2.1 Algorithm Overview

The grasping algorithm consists of six main steps, see Fig. 4:

• Raw Data Pre-Processing: The raw data points are pre-processed with a smoothing filter to reduce noise to reduce noise.

• Range Image Segmentation: This step identifies different objects on the table or parts of an object based on a 3D DeLaunay triangulation.

• Pairwise Matching: Find high curvature points, which indicate the top rim of an object part, fit a circle to these points, and merge rotationally symmetric objects.

• Approximation of 2.5D Objects to 3D Objects: This step is only important to detect potential collisions by the path planning tool.

    -Rotationally Symmetric Objects: Add additional points by using the main axis information.

    -Arbitrary Objects: The non-visible range will be closed with planes, normal to the table plane.

• Grasp Point and Pose Detection:

    -Grasp Point Detection: Rotationally Symmetric Objects.

    -Grasp Pose Detection: Arbitrary Objects.

• Collision Detection: Considering all surrounding objects and the table surface as obstacles, to evaluate the calculated hand pose.

---

[5] http://www.sickivp.se/sickivp/de.html

[6] http://www.amrose.dk/

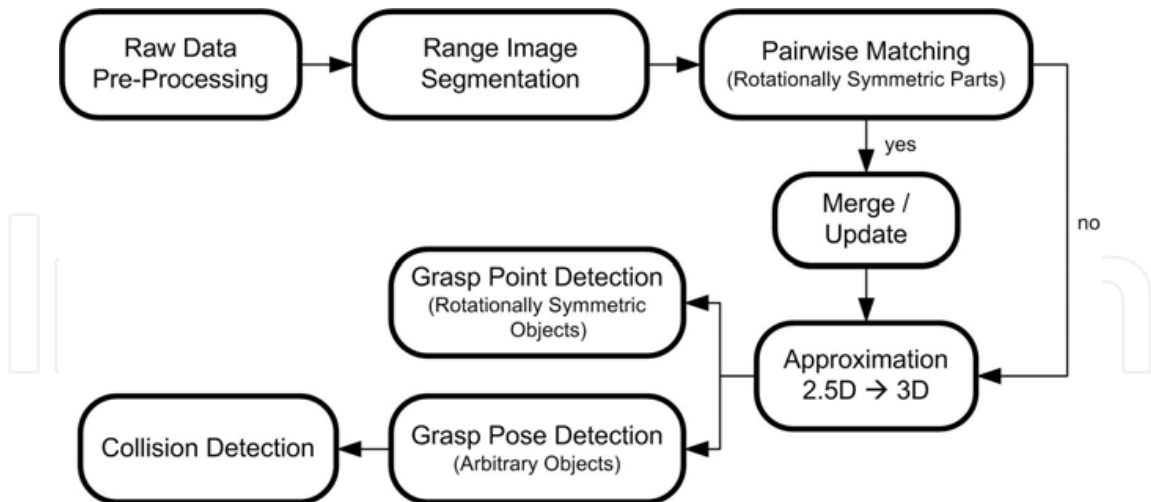[7] Freely available open source software, http://public.kitware.com/vtk

Fig. 4. Overview of the presented grasping algorithm.

3. Range Image Segmentation

The range image segmentation starts by detecting the surface of the table with a RANSAC (Fischler et al. 1981) based plane fit (Stiene et al., 2002). We define an object or part as a set of points with distances between neighbors. For that we build a kd-tree (Bentley, 1975) and calculate the minimum d_{min}, maximum, d_{max}, and average distance d_a between all neighboring points as input information for the mesh generation step. (Arya et al., 1998). The segmentation of the 2.5D point cloud is achieved with the help of a 3D mesh generation based on the triangles calculated by a 3D DeLaunay triangulation (O'Rourke, 1998). Then all segments of the mesh are extracted by a connectivity filter (Belmonte et al., 2004). This step segments the mesh into different components (objects or parts). An additional cut refinement was not arranged. The result may contain an over- or an under segmentation depending on the overlap of the objects as illustrated in Fig. 5.

Fig. 5. Results after the first segmentation step. Object no. 1 is cut into two parts and objects no. 5 and 7 are overlapping. The not perfectly segmented objects are red encircled

After the object segmentation step the algorithm finds the top surfaces of all objects using a RANSAC based plane fit and generates a 2D DeLaunay triangulation, with this 2D surface information the top rim points and top feature edges of every object can be detected, as illustrated in Fig. 6. For the top surface detection the algorithm uses a pre-processing step to find out all vertices[8] of the object with a normal vector in x-direction bigger than in y- or z-direction, $n[x] > n[y]$ and $n[x] > n[z]$, the x-direction is normal to the table plane. The normal vectors of all vertices are calculated with the faces (triangles) of the generated mesh.
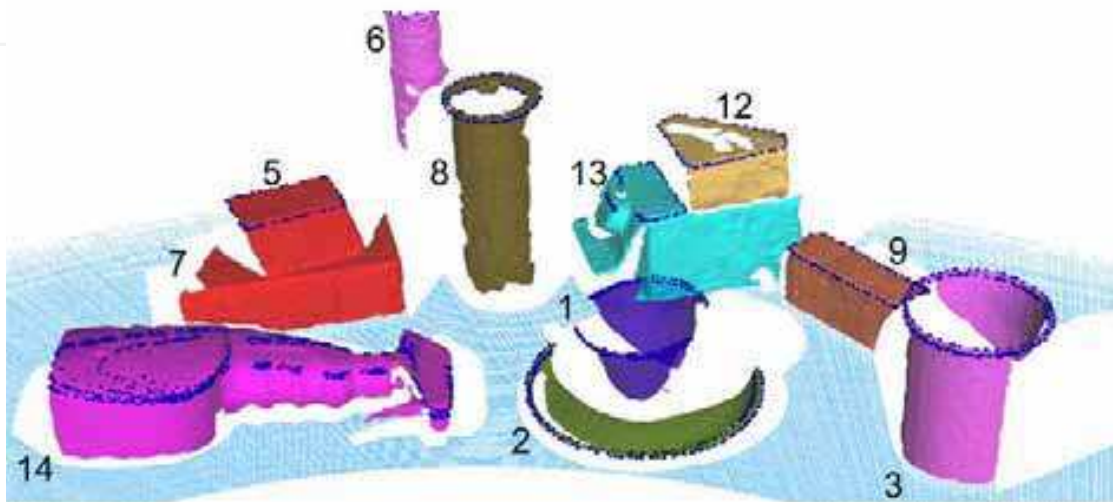
Fig. 6. Results after the merging step. The wrongly segmented rotationally symmetric parts of object no. 1 are successfully merged to one object. The blue points represent the top rim of the objects.

3.1 Pairwise Matching

We developed a matching method, which is specifically for rotationally symmetric objects, because this objects can be stable segmented, detected and merged in a point cloud with unknown objects. To detect the top rim circle of rotationally symmetric objects a RANSAC based circle fit (Jiang & Cheng, 2005) with a range tolerance of 2mm is used.

Several tests have shown that this threshold provides good results for our currently used laser-range scanner. For an explicit description, the data points are defined as (p_{xi}, p_{yi}, p_{zi}) and (c_x, c_y, c_z) is the circle's centre with a radius r. The error must be smaller than a defined threshold:

$$\left| \left\| \vec{p} - \vec{c} \right\| - r \right| \leq 2 \tag{1}$$

This operation will be repeated for every point of the top rim. The run with the maximum number n of included points wins.

$$n = \left| \left\{ p \, \middle| \, \left| \left\| \vec{p} - \vec{c} \right\| - r \right| \leq 2 \right\} \right| \tag{2}$$

---

[8] In geometry, a vertex is a special kind of point which describes the corners or intersections of geometric shapes and a polygon is a set of faces.

If more than 80% of the rim points of both parts (rotationally symmetric parts) lie on the same circle, the points of both parts are examined more closely with the fit. For that we calculate the distances of all points of both parts to the rotation axis, see Equ. 3, the yellow lines represent the rotation axis, see Fig. 1, object no. 3. If more than 80% of all points of both parts agree, both parts are merged to one object, see Fig. 6, object no. 1.

$$d = \left(\vec{p} - \vec{c} \right) \times \vec{n} \tag{3}$$

3.2 Approximation of 3D Objects

This step is important to detect potential collisions by the path planning tool from AMROSE. In order to avoid wrong paths and collisions with other objects, due to missing model information, because in 2.5D point clouds every object is seen from only one view, but the path planning tool needs full information to calculate a collision free path. During the matching step the algorithm detected potential rotationally symmetric objects and merged clipped parts. With this information, the algorithm rotates only points along the axis by 360° in 5° steps, which fulfil the necessary rotation constraint. This means that only points will be rotated, which have a corresponding point on the opposite side of the rotation axis (Fig. 5, object no. 1) or build a circle with the neighbouring points along the rotation axis, as illustrated in Fig. 1, object no. 3 and Fig. 7, object no. 1. By this relatively simple constraint object parts such as handles or objects close to the rotationally symmetric object will not be rotated. For all other arbitrary objects, every point will be projected to the table plane and with a 2D DeLaunay triangulation the rim points can be detected. These points correspond with the rim points of the visible surfaces. So, the non-visible surfaces can be closed, these surfaces will be filled with points between the corresponding rim points, as illustrated in Fig. 7. Filling the non-visible range with vertical planes may lead to incorrect results, especially when the rear side of the objects is far from vertical, but this step is only to detect potential collisions by the path planning tool.
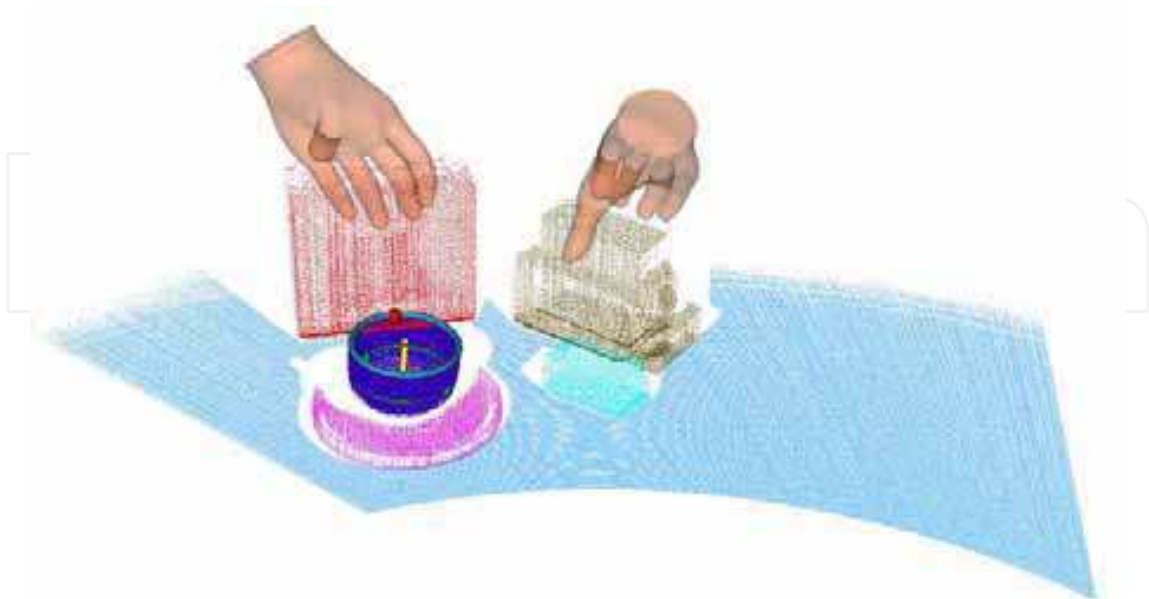

Fig. 7. Detection of grasping points and hand poses. The green points illustrate the computed grasping points for rotationally symmetric objects. The red points show an

alternative grasp along the top rim, thereby one grasping point is enough for an open object. The illustrated hand poses show a possible grasp for the remaining graspable objects.

4. Grasp Point and Pose Detection

The algorithm for grasp point detection is limited to rotationally symmetric objects and the grasp poses will be calculated for arbitrary objects. After the segmentation step we find out if the object is open or closed, for that we fit a sphere into the top surface. If there is no point of the object in this sphere we consider the object is opened. Then the grasping points of all cylindrical objects can be calculated. For every rotationally symmetric object we calculate two grasping points along the rim in the middle of the object (green coloured points, as illustrated in Fig. 8, object no. 1 and no. 6). If the path planner is not able to detect a possible grasp, the algorithm calculates alternative grasping points along the top rim of the object near the strongest curvature, as illustrated in Fig. 8, object no. 6 as red points. If it is an open object one grasping point is enough to realize a stable grasp near the top rim, as illustrated in Fig. 8 object no. 1. The grasping points should be calculated in such a way that they are next to the robot arm, which is mounted on the opposite side of the laser-range scanner. The algorithm detects the strongest curvature along the top rim with a Gaussian curvature filter (Porteous, 1994).
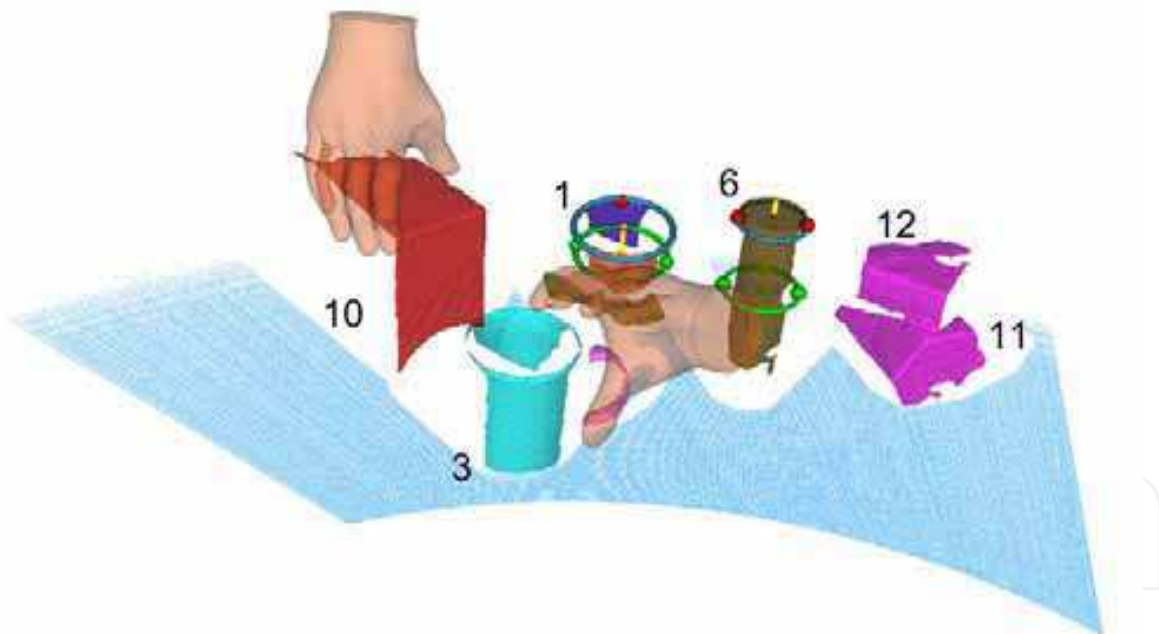

Fig. 8. Detection of grasping points and hand poses. The computed grasping points for rotationally symmetric objects. The red points show an alternative grasp along the top rim, thereby one grasping point is enough for an open object. The illustrated hand poses show a possible grasp for the remaining graspable objects.

To successfully grasp an object it is not always sufficient to locally find the best grasping pose. The algorithm should calculate an optimal grasping pose to realize a good grasp without collision as fast as possible. In general, conventional multidimensional "brut force"

search methods are not practical to solve this problem. (Li et al., 2007) show a practical shape matching algorithm, where a reduced number of 38 contact points are considered. Most shape matching algorithms need an optimization step through that the searched optimum can be efficiently computed.

At the beginning the internal centre and the principal axis of the top surface are calculated with a transformation that fits and transforms a sphere inside, see Fig. 9b the blue top surfaces. After the transformation, this sphere has an elliptical form in alignment of the top surface points, hereby also the principal axis is founded. The algorithm transforms the rotation axis of the gripper (defined by the fingertip of the thumb, the index finger, and the last finger, as illustrated in Fig. 9a) along the principal axis of the top surface and the centre (calculated with the fingertips) of the hand c_h will be translated to the centre of the top surface c_{top}, whereby $c_h = c_{top}$ results. The hand will be rotated, so the normal vector of the hand aligns in reverse direction with the normal vector of the top surface. Afterwards the hand is shifted along the normal vectors up to a possible collision with the grasping object.
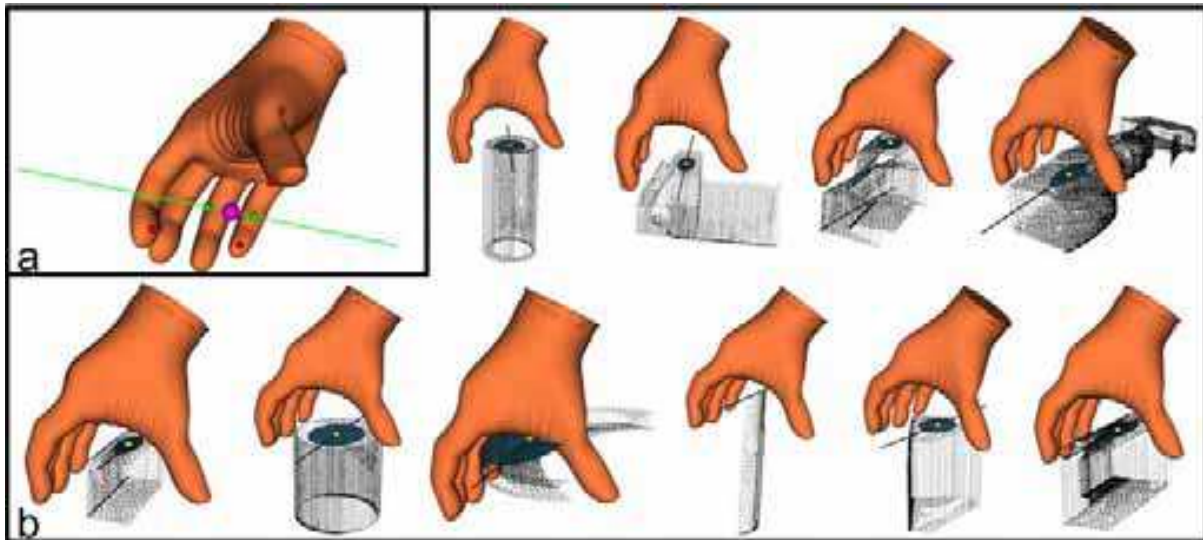

Fig. 9. Detection of grasping poses. **a** The rotation axis of the hand must be aligned with the principal axis of the top surface. **b** First grasping result: The hand was transformed and rotated along the principal axis of the top surface. After this step the algorithm checks potential collisions with all surrounding objects.

4.1 Collision Detection

The calculated grasping pose will be checked for a potential collision with the remaining objects and the table, as illustrated in Fig. 8. The algorithm determines if it is possible to grasp the object with an obb-tree. This method verifies possible points of the objects inside the hand by the calculated pose. If the algorithm detects a potential collision, the calculated pose will not be accepted.

5. Experiments and Results

We evaluated the detected grasping points and poses directly on the objects with an AMTEC robot arm and gripper. The object segmentation, merging, grasp point, pose detection, and collision detection is performed on a PC with 3.2GHz dual core processor and takes an average time of 35sec., depending on the number of the surrounding objects on the table, see Tab.1. The algorithm is implemented in C++ using the Visualization Tool Kit (VTK). Testing of 5 different point clouds for every object in different combination with other objects, the algorithm shows positive results. A remaining problem is that in some cases, interesting parts of shiny objects are not visible for the laser-range scanner, hence our algorithm is neither able to calculate correct grasping points nor the pose of the object. Another problem is that the quality of the point cloud is sometimes not good enough to guarantee a successful grasp, as illustrated in Fig. 10. The success of our grasping point and grasping pose algorithm depends on the ambient light, object surface properties, laser-beam reflectance, and absorption of the objects. For object no. 2 (saucer) the algorithm cannot detect possible grasping points or a possible grasping pose, because of shadows of the laser-range scanner and occlusion with the coffee cup, as illustrated in Fig. 1. In addition this object is nearly impossible to grasp with the used gripper. The algorithm cannot calculate possible grasping poses for object no. 16 (C-3PO), because of inadequate scan data. Finally the used gripper was not able to grasp object no. 15 (fabric softener), because of a slip effect. For all tested objects we achieved an average grasp rate of 71.11%.

In our work, we demonstrate that our grasping point and pose detection algorithm with a 3D model of the used gripper for unknown objects, performs practical grasps, as illustrated Tab. 2.

Calculation Steps	Time[sec]
Plane Fit	1.4sec
Mesh Generation	11sec
Mesh Segmentation	0.7sec
Top Surface Detection	0.9sec
Merging Rotationally Symmetric Objects	2.0sec
Approximation of 3D Objects	6sec
Grasp Point Detection	3.5sec
Grasp Pose Detection	6.5sec
Collision Detection	10sec
Sum	**35sec**

Table 1. Duration of Every Calculation Step.

No.	Objects	Rate[%]
1	Coffee Cup (small)	100%
2	Saucer	0%
3	Coffee Cup (big)	60%
4	Cube	100%
5	Geometric Primitive	100%
6	Spray on Glue	100%
7	Salt Shaker (cube)	100%
8	Salt Shaker (cylinder)	100%
9	Dextrose	80%

10	Melba Toast	100%
11	Amicelli	80%
12	Mozart	60%
13	Latella	100%
14	Aerosol Can	80%
15	Fabric Softener	0%
16	C-3PO	0%
17	Cat	20%
18	Penguin	100%
Overall		**71.11%**

Table 2. Grasping rate of different, unknown objects (each object was tested 5 times).
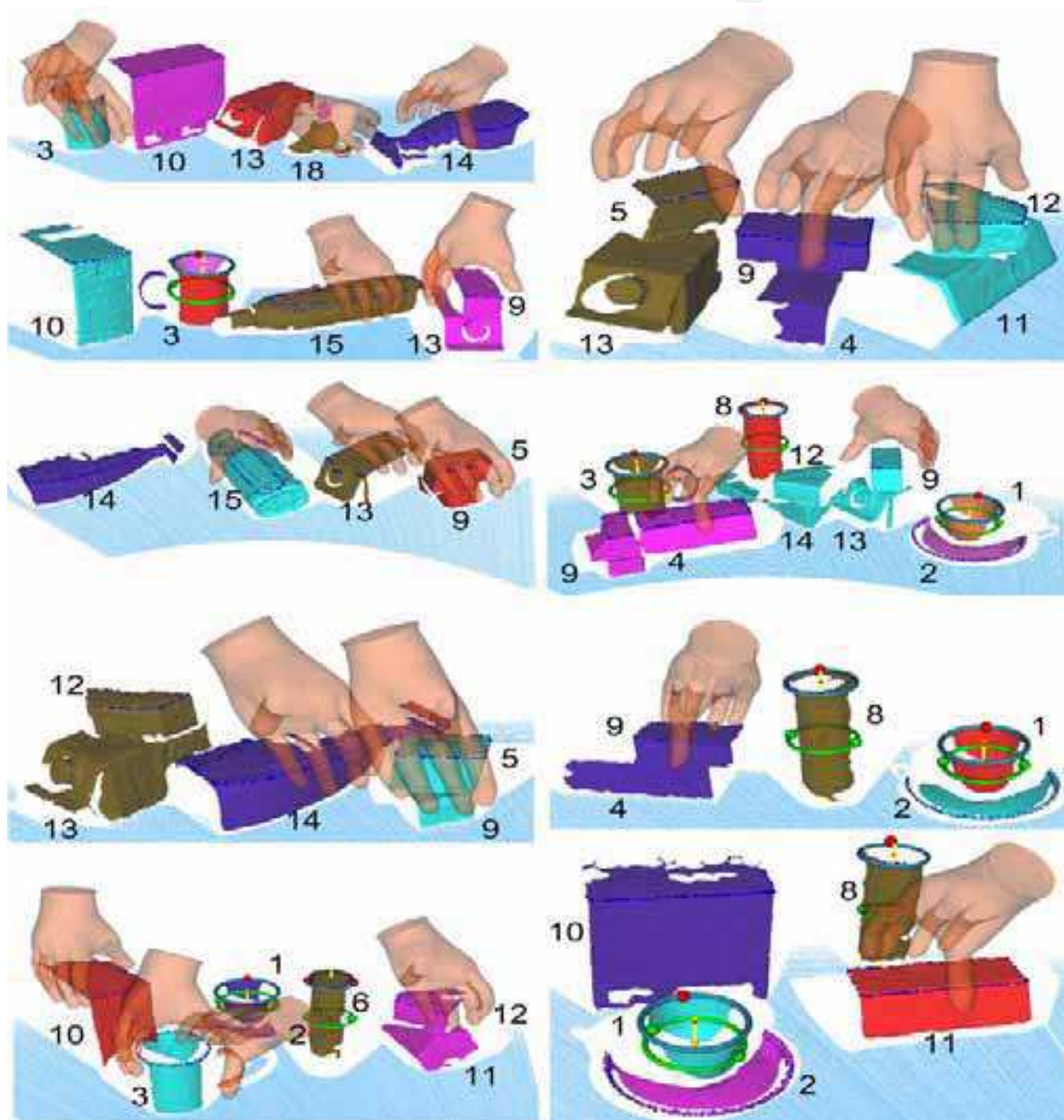
Fig. 10. Examples of detection results.

7. Conclusion and Future Work

We present a method for automatic grasping of unknown objects with a prosthesis hand, by incorporating a laser-range scanner which shows a high reliability. The approach for object grasping is well suited for use in related applications under different conditions and can be applied to a reasonable set of objects.

From a single-view the rear side of an object is not visible due to self occlusions, and the front side may be occluded by other objects. The algorithm was developed for arbitrary objects in different poses, on top of each other or side by side with a special focus on rotationally symmetric objects. If objects can not be separated because they are stacked one of each other they are considered as one object. If the algorithm detects rotationally symmetric parts this parts are merged, because this object class can be robustly identified and allows a cylindrical grasp as well as a tip grasp along the top rim. For all other objects the algorithm calculates a tip grasp based on the top surface, also for on top of each other arranged objects.

In the near future we plan to use a deformable hand model to reduce the opening angle of the hand, so we can model the closing of a gripper in the collision detection step. We will also use geometric hashing (Wolfson, 1997) for the matching step in order to be able to unite faster, several parts of the point cloud.

8. References

Aarno, D., Sommerfeld, J., Kragic, D., Pugeault N., Kalkan, S., Wörgötter, F., Kraft, D., Krüger, N. (2007). Early Reactive Grasping with Second Order 3D Feature Relations. IEEE International Conference on Robotics and Automation, Workshop: From features to actions - Unifying perspectives in computational and robot vision.

Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R. (1998). An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions. Journal of the ACM, Vol. 45, No. 6, pp. 801-923.

Bentley, J.L. (1975). Multidimensional Binary Search Trees Used for Associative Searching. Communications of the ACM, Vol. 18, No. 19, pp. 509-517.

Belmonte, Ó., Remolar, I., Ribelles, J., Chover, M., Fernández, M. (2004). Efficiently using connectivity information between triangles in a mesh for real-time rendering. Elsevier Science, Vol. 20, No. 8, pp. 1263-1273.

Besl, P.J., McKay, H.D. (1992). A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 239-256.

Bone, G.M., Lambert, A., Edwards, M. (2008). Automated modelling and robotic grasping of unknown three-dimensional objects. IEEE International Conference on Robotics and Automation, pp. 292-298.

Borst, C., Fischer, M., Hirzinger, G. (2003). Grasping the dice by dicing the grasp. IEEE/RSJ International Conference on Robotics and Systems, pp. 3692-3697.

Boughorbel, F., Zhang, Y. (2007). Laser ranging and video imaging for bin picking. Assembly Automation, Vol. 23, No. 1, pp. 53-59.

Casals, A., Merchan, R. (1999). Capdi: A robotized kitchen for the disabled and elderly people. Proceedings of the 5th European Conference for the Advancement Assistive Technology / AAATE, pp 346–351.

Castiello, U. (2005). The neuroscience of grasping. Nature Reviews Neuroscience, Vol. 6, No. 9, pp. 726-736.

Ekvall, S., Kragic, D. (2007). Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning. IEEE International Conference on Robotics and Automation, pp. 4715-4720.

El-Khoury, S., Sahbani A., Perdereau, V. (2007). Learning the Natural Grasping Component of an Unknown Object. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2957-2962.

Fagg, A.H., Arbib, M.A. (1998). Modeling parietal-premotor interactions in primate control of grasping. Neural Networks, Vol. 11, pp. 1277-1303.

Fischler, M.A., Bolles, R.C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, Vol. 24, No. 6, pp. 381-395.

Goldfeder, C., Allen, P.K., Lackner, C., Pelossof, R. (2007). Grasp Planning via Decomposition Trees. IEEE International Conference on Robotics and Automation, pp. 4679-4684.

Huebner, K., Ruthotto, S., Kragic, D. (2008). Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping. IEEE International Conference on Robotics and Automation, pp. 1628-1633.

Ivlev, O., Martens, C. (2005). Rehabilitation robots friend-i and friend-i with the dexterous lightweight manipulator. IOS Press, Vol. 17, pp. 111–123.

Jiang, X., Cheng, D.C. (1999). Fitting of 3D circles and ellipses using a parameter decomposition approach. Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling, pp. 103–109.

Kragic, D., Bjorkman, M. (2006). Strategies for object manipulation using foveal and peripheral vision. IEEE International Conference on Computer Vision Systems, pp. 50-55.

Li, Y., Fu, J.L., Pollard, N.S. (2007). Data-Driven Grasp Synthesis Using Shape Matching and Task-Based Pruning. IEEE Transactions on Visualization and Computer Graphics, Vol. 13, No. 4, pp. 732-747.

Martens, C., Ruchel, N. (2001). A friend for assisting handicapped people. IEEE Robotics & Automation Magazine, Vol. 8, pp. 57–65.

Miller, A.T., Allen, P.K. (2004). GraspIt! A Versatile Simulator for Robotic Grasping. IEEE Robotics & Automation Magazine, Vol. 11, No. 4, pp. 110-112.

Miller, A.T., Knoop, S. (2003). Automatic grasp planning using shape primitives. IEEE International Conference on Robotics and Automation, pp. 1824-1829.

O'Rourke, J. (1998). Computational Geometry in C. Univ. Press, Cambridge, 2nd edition.

Porteous, I.R. (1994). Geometric Differentiation. Univ. Press, Cambridge.

Recatalá, G., Chinellato, E., Del Pobil, Á.P., Mezouar, Y., Martinet, P. (2008). Biologically-inspired 3D grasp synthesis based on visual exploration. Autonomous Robots, Vol. 25, No. 1-2, pp. 59-70.

Richtsfeld, M., Zillich, M. (2008). Grasping Unknown Objects Based on $2^1/_2$D Range Data. IEEE International Conference on Automation Science and Engineering, pp. 691-696.

Sanz, P.J., Iñesta, J.M., Del Pobil, Á.P. (1999). Planar Grasping Characterization Based on Curvature-Symmetry Fusion. Applied Intelligence, Vol. 10, No. 1, pp. 25-36.

Saxena, A., Driemeyer, J., Ng, A.Y. (2008). Robotic Grasping of Novel Objects using Vision. International Journal of Robotics Research, Vol. 27, No. 2, pp. 157-173.

Stansfield, S.A. (1991). Robotic grasping of unknown objects: A knowledge-based approach. International Journal of Robotics Research, Vol. 10, No. 4, pp. 314-326.

Schulz, S., Pylatiuk, C., Reischl, M., Martin, J., Mikut, R., Bretthauer, G. (2005). A hydraulically driven multifunctional prosthetic hand. Robotica, Cambridge University Press, Vol. 23, pp. 293-299.

Stiene, S., Lingemann, K., Nüchter, A., Hertzberg, J. (2006). Contour-based Object Detection in Range Images. Third International Symposium on 3D Data Processing, Visualization, and Transmission, pp. 168-175.

Wang, B., Jiang, L. (2005). Grasping unknown objects based on 3d model reconstruction. Proceedings of International Conference on Advanced Intelligent Mechatronics, pp. 461-466.

Wolfson, H.J. (1997). Geometric Hashing: An Overview. IEEE Computational Science and Engineering, Vol. 4, No. 4, pp. 10-21.

Xue, Z., Zoellner, J.M., Dillmann, R. (2008). Automatic Optimal Grasp Planning Based On Found Contact Points. IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1053-1058.
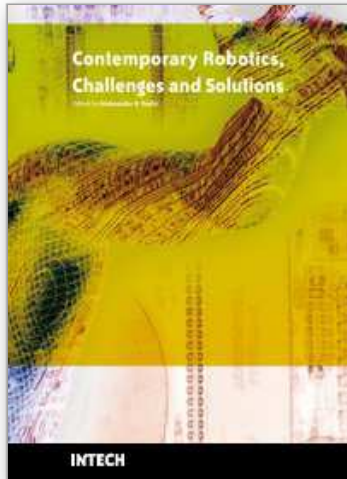
Contemporary Robotics - Challenges and Solutions

Edited by A D Rodi

This book book is a collection of 18 chapters written by internationally recognized experts and well-known professionals of the field. Chapters contribute to diverse facets of contemporary robotics and autonomous systems. The volume is organized in four thematic parts according to the main subjects, regarding the recent advances in the contemporary robotics. The first thematic topics of the book are devoted to the theoretical issues. This includes development of algorithms for automatic trajectory generation using redudancy resolution scheme, intelligent algorithms for robotic grasping, modelling approach for reactive mode handling of flexible manufacturing and design of an advanced controller for robot manipulators. The second part of the book deals with different aspects of robot calibration and sensing. This includes a geometric and treshold calibration of a multiple robotic line-vision system, robot-based inline 2D/3D quality monitoring using picture-giving and laser triangulation, and a study on prospective polymer composite materials for flexible tactile sensors. The third part addresses issues of mobile robots and multi-agent systems, including SLAM of mobile robots based on fusion of odometry and visual data, configuration of a localization system by a team of mobile robots, development of generic real-time motion controller for differential mobile robots, control of fuel cells of mobile robots, modelling of omni-directional wheeled-based robots, building of hunter- hybrid tracking environment, as well as design of a cooperative control in distributed population-based multi-agent approach. The fourth part presents recent approaches and results in humanoid and bioinspirative robotics. It deals with design of adaptive control of anthropomorphic biped gait, building of dynamic-based simulation for humanoid robot walking, building controller for perceptual motor control dynamics of humans and biomimetic approach to control mechatronic structure using smart materials.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds

51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821