Robotic Grasping of Unknown Objects

Mario Richtsfeld and Markus Vincze
Institute of Automation and Control Vienna University of Technology, Vienna Austria

1. Introduction

This work describes the development of a novel vision-based grasping system for unknown objects based on laser range and stereo data. The work presented here is based on *2.5D* point clouds, where every object is scanned from the same view point of the laser range and camera position. We tested our grasping point detection algorithm separately on laser range and single stereo images with the goal to show that both procedures have their own advantages and that combining the point clouds reaches better results than the single modalities. The presented algorithm automatically filters, smoothes and segments a *2.5D* point cloud, calculates grasping points, and finds the hand pose to grasp the desired object.

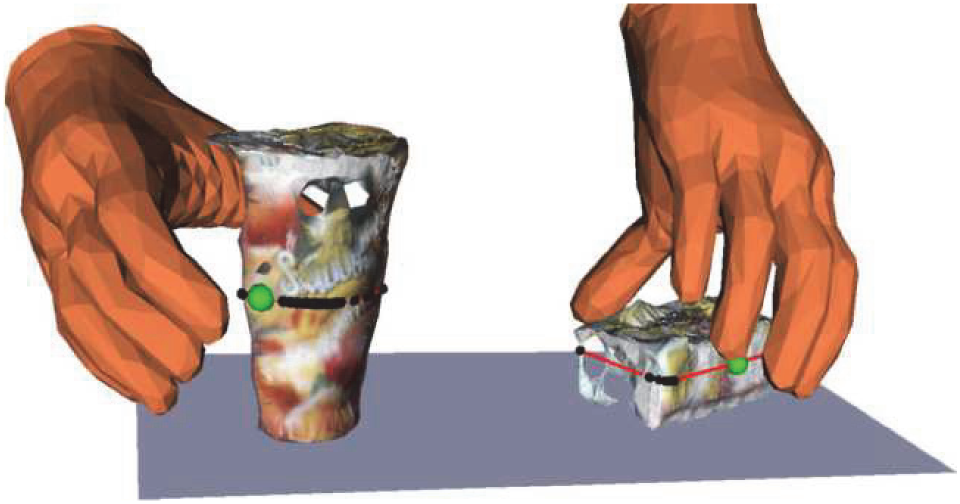

Fig. 1. Final detection of the grasping points and hand poses. The green points display the computed grasping points with hand poses.

The outline of the paper is as follows: The next Section introduces our robotic system and its components. Section 3 describes the object segmentation and details the analysis of the objects to calculate practical grasping points. Section 4 details the calculation of optimal hand poses to grasp and manipulate the desired object without any collision. Section 5 shows the achieved results and Section 6 finally concludes this work.

1.2 Problem statement and contribution

The goal of the work is to show a new and robust way to calculate grasping points in the recorded point cloud from single views of a scene. This poses the challenge that only the front side of objects is seen and, hence, the second grasp point on the backside of the object needs to be assumed based on symmetry assumptions. Furthermore we need to cope with the typical sensor data noise, outliers, shadows and missing data points, which can be caused by specular or reflective surfaces. Finally, a goal is to link the grasp points to a collision free hand pose using a full *3D* model of the gripper used to grasp the object. The main idea is depicted in Fig. 1[1].

The main problem is that *2.5D* point clouds do not represent complete *3D* object information. Furthermore stereo data includes measurement noise and outliers depending on the texture of the scanned objects. Laser range data includes also noise and outliers where the typical problem is missing sensor data because of absorption. The laser exhibits high accuracy while the stereo data includes more object information due to the better field of view. The contribution is to show in detail the individual problems of using both sensor modalities and we then show that better results can be obtained by merging the data provided by the two sensors.

1.3 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 and robotics. There exist several approaches on grasping quasi planar objects (Sanz et al., 1999; Richtsfeld & Zillich, 2008). (Recatalá et al., 2008) developed a framework for the development of robotic applications based on a grasp-driven multi-resolution visual analysis of the objects and the final execution of the calculated grasps. (Li et al., 2007) presented a 2D data-driven approach based on a hand model of the gripper to realize grasps. The algorithm finds the best hand poses by matching the query object by comparing object features to hand pose features. The output of this system is a set of candidate grasps that will then be sorted and pruned based on effectiveness for the intended task. The algorithm uses a database of captured human grasps to find the best grasp by matching hand shape to object shape. Our algorithm does not include a shape matching method, because this is a very time intensive step. The *3D* model of the hand is only used to find a collision free grasp.

(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. The only two integrated tactile sensors of the used robotic gripper in this work are too limited for additional information to calculate grasping points. These sensors are only used if a potential stick-slip effect occurs.

(Miller et al., 2004) developed an interactive grasp simulator "GraspIt!" for different hands and hand configurations and objects. The method evaluates the grasps formed by these hands. 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. The approach is based on a fixed relative position and

[1] All images are best viewed in colour.

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.

(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. In our work we do not use a supervised learning approach. We find grasping points according to predefined rules.

(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. We additionally analyze the calculated grasping points with a *3D* model of the hand and our algorithm obtains the required gripper pose to grasp the object. Another *3D* model based work is presented by (El-Khoury et al., 2007). 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. Our segmentation step identifies different objects in the same table scene. (Huebner et al., 2008) have applied 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.

(Stansfield, 1991) presented a system for grasping *3D* objects with unknown geometry using a Salisbury robotic hand, where 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. In our case we do not operate on a motorized and rotated table, which is unrealistic for real world use, the goal is to grasp objects when seen only from one side.

Summarizing to the best knowledge of the authors in contrast to the state of the art reviewed above our algorithm works 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 presented algorithm calculates for arbitrary objects grasping points given stereo and / or laser data from one view. The poses of the objects are calculated with a 3D model of the gripper and the algorithm checks and avoids potential collision with all surrounding objects.

2. Experimental setup

We use a fixed position and orientation between the AMTEC[2] robot arm with seven degrees of freedom and the scanning unit. Our approach is based on scanning the objects on the

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

table by a rotating laser range scanner and a fixed stereo system and the execution of the subsequent path planning and grasping motion. The robot arm is equipped with a hand prosthesis from the company Otto Bock[3], which we are using as gripper, see Fig. 2. The hand prosthesis has integrated tactile force sensors, which detect a potential sliding of an object and enable the readjustment of the pressure of the fingers. This hand prosthesis has three active fingers the thumb, the index finger and the middle finger; the last two fingers are for cosmetic reasons. Mechanically it is a calliper gripper, which can only realize a tip grasp and for the computation of the optimal grasp only 2 grasping points are necessary. The middle between the fingertip of the thumb and the index finger is defined as tool centre point (TCP). We use a commercial path planning tool from AMROSE[4] to bring the robot to the grasp location.

The laser range scanner records a table scene with a pan/tilt-unit and the stereo camera grabs two images at -4° and +4°. (Scharstein & Szeliski, 2002) published a detailed description of the used dense stereo algorithm. To realize a dense stereo calibration to the laser range coordinate system as exactly as possible the laser range scanner was used to scan the same chessboard that is used for the camera calibration. At the obtained point cloud a marker was set as reference point to indicate the camera coordinate system. We get good results by the calibration most of the time. In some cases at low texture of the scanned objects and due to the simplified calibration method the point clouds from the laser scanner and the dense stereo did not correctly overlap, see Fig. 3. To correct this error of the calibration we used the iterative closest point (ICP) method (Besl & McKay, 1992) where the reference is the laser point cloud, see Fig. 4. The result is a transformation between laser and stereo data that can now be superimposed for further processing.

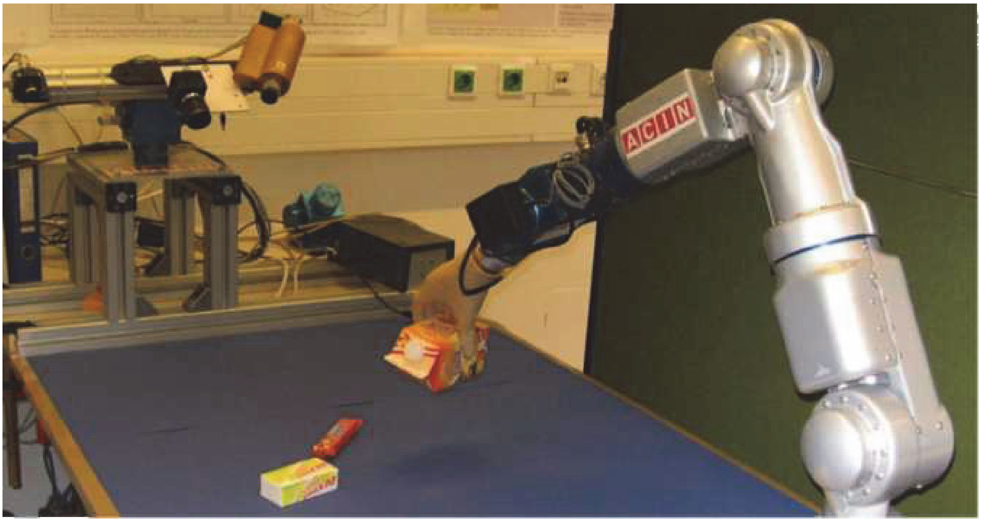

Fig. 2. Overview of the system components and their interrelations.

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

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

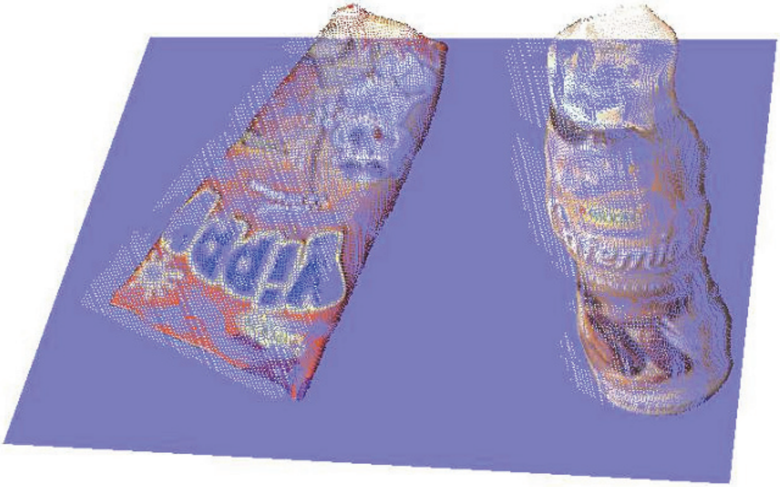

Fig. 3. Partially overlapping point clouds from the laser range scanner (white points) and dense stereo (coloured points). A clear shift between the two point clouds shows up.

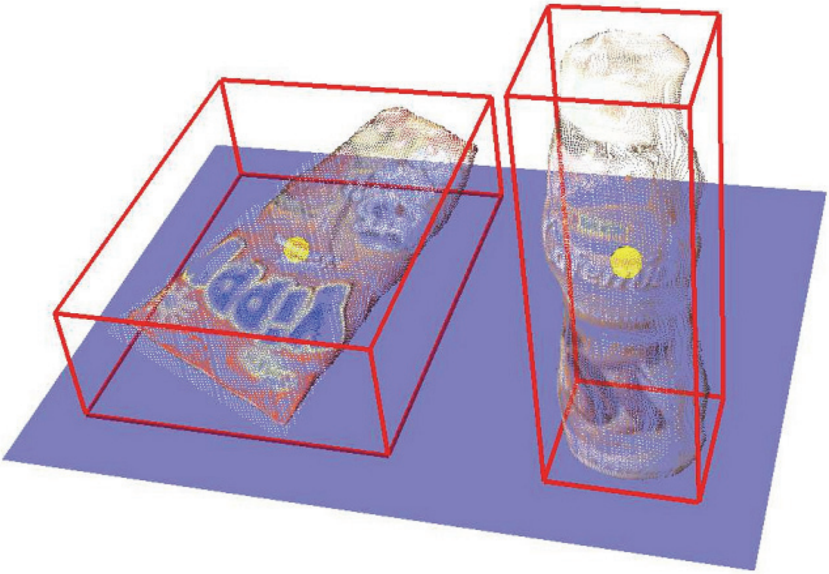

Fig. 4. Correction of the calibration error applying the iterative closest point (ICP) algorithm. The red lines represent the bounding boxes of the objects and the yellow points show the approximation to the centre of the objects.

3. Grasp point detection

The algorithm to find grasp points on the objects consists of four main steps as depicted in Fig. 5:

- Raw Data Pre-processing: The raw data points are pre-processed with a geometrical filter and a smoothing filter to reduce noise and outliers.
- Range Image Segmentation: This step identifies different objects based on a 3D DeLaunay triangulation, see Section 4.
- Grasp Point Detection: Calculation of practical grasping points based on the centre of the objects, see Section 4.
- Calculation of the Optimal Hand Pose: Considering all objects and the table surface as obstacles, find an optimal gripper pose, which maximizes distances to obstacles, see Section 5.

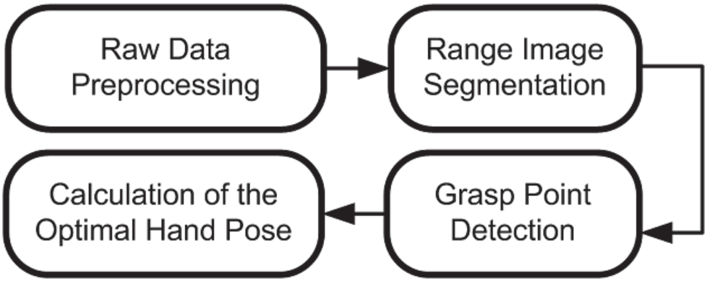

Fig. 5. Overview of our grasp point and gripper pose detection algorithm.

4. Segmentation and grasp point detection

There is no additional segmentation step for the table surface needed, because the red light laser of the laser range scanner is not able to detect the surface of the blue table and the images of the stereo camera were segmented and filtered directly. However, plane segmentation is a well known technique for ground floor or table surface detection and could be used alternatively, e.g., (Stiene et al., 2006).

The segmentation of the unknown objects will be achieved with a *3D* mesh generation, based on the triangles, calculated by a DeLaunay triangulation [10]. After mesh generation we look at connected triangles and separate objects.

In most grasping literature it is assumed that good locations for grasp contacts are actually at points of high concavity. That's absolutely correct for human grasping, but for grasping with a robotic gripper with limited DOF and only two tactile sensors a stick slip effect occurs and makes these grasp points rather unreliable.

Consequently to realize a possible, stable grasp the calculated grasping points should be near the centre of mass of the objects. Thus, the algorithm calculates the centre c of the objects based on the bounding box, Fig. 4, because with a *2.5D* point cloud no accurate centre of mass can be calculated. Then the algorithm finds the top surfaces of the objects with a RANSAC based plane fit (Fischler & Bolles, 1981). We intersect the point clouds with horizontal planes through the centre of the objects. If the object does not exhibit a top plane, the normal vector of the table plane will be used. From these n cutting plane points p_i we calculate the (planar) convex hull V, using Equ. 1 and illustrated in Fig. 6.

$$V = ConvexHull\left(\bigcup_{i=0}^{n-1} p_i\right) \tag{1}$$

With the distances between two neighbouring hull points to the centre of the object c we calculate the altitude d of the triangle, see Equ. 2. $\vec{v}$ is the direction vector to the neighbouring hull point and $\vec{w}$ is the direction vector to c. Then the algorithm finds the shortest normal distance d_{min} of the convex hull lines, illustrated in Fig. 6 as red lines, to the centre of the object c, where the first grasping point is located.

$$d = \frac{\left\|\vec{v} \times \vec{w}\right\|}{\left\|\vec{v}\right\|} \tag{2}$$

In *2.5D* point clouds it is only possible to view the objects from one side, however we assume a symmetry of the objects. Hence, the second grasping point is determined by a reflection of the first grasping point using the centre of the object. We check a potential lateral and above grasp of the object on the detected grasping points with a simplified *3D* model of the hand. If no accurate grasping points could be calculated with the convex hull of the cutting plane points p_i the centre of the object is displaced in *1mm* steps towards the top surface of the object (red point) with the normal vector of the top surface until a positive grasp could be detected. Another method is to calculate the depth of indentation of the gripper model and to calculate the new grasping points based on this information.

Fig. 6 gives two examples and shows that the laser range images often have missing data, which can be caused by specular or reflective surfaces. Stereo clearly correct this disadvantage, see Fig. 7.

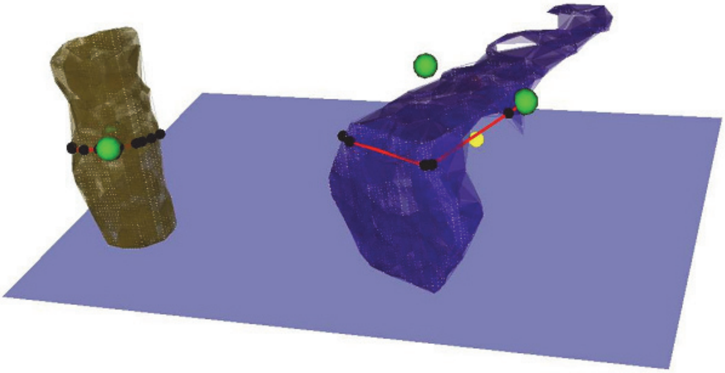

Fig. 6. Calculated grasping points (green) based on laser range data. The yellow points show the centre of the objects. If, through the check of the 3D gripper no accurate grasping points could be calculated with the convex hull (black points connected with red lines) the centre of the objects is displaced towards the top surface of the objects (red points).

Fig. 7 illustrates that with stereo data alone there are definitely better results possible then with laser range data alone given that object appearance has texture. This is also reflected in Tab. 2. Fig. 8 shows that there is a smaller difference between the stereo data alone (see Fig. 7) and the overlapped laser range and stereo data, which Tab. 2 confirms.

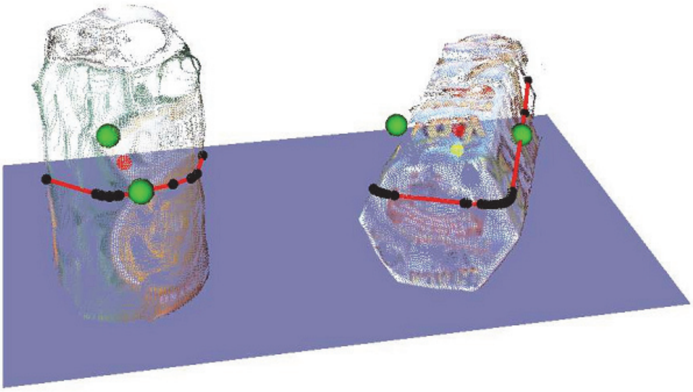

Fig. 7. Calculated grasping points (green) based on stereo data. The yellow points show the centre of mass of the objects. If, through the check of the 3D gripper no accurate grasping points could be calculated with the convex hull (black points connected with red lines) the centre of the objects is displaced towards the top surface of the objects (red points).

5. Grasp pose

To successfully grasp an object it is not always sufficient to find locally the best grasping points, the algorithm should also decide at which angle it is possible to grasp the selected object. For this step we rotate the *3D* model of the hand prosthesis around the rotation axis, which is defined by the grasping points. The rotation axis of the hand is defined by the fingertip of the thumb and the index finger of the hand, as illustrated in Fig. 9. The algorithm checks for a collision of the hand with the table, the object that shall be grasped and all obstacles around it. This will be repeated in *5°* steps to a full rotation by *180°*. The algorithm notes with each step whether a collision occurs. Then the largest rotation range where no collision occurs is found. We find the optimal gripper position and orientation by an averaging of the maximum and minimum largest rotation range. From this the algorithm calculates the optimal gripper pose to grasp the desired object.

The grasping pose depends on the orientation of the object itself, surrounding objects and the calculated grasping points. We set the grasping pose as a target pose to the path planner, illustrated in Fig. 9 and Fig. 1. The path planner tries to reach the target object on his part. Fig. 10 shows the advantage to calculate the gripper pose. The left Figure shows a collision free path to grasp the object. The right Figure illustrates a collision of the gripper with the table.

6. Experiments and results

To evaluate our method, we choose ten different objects, which are shown in Fig. 11. The blue lines represent the optimal positions for grasping points. Optimal grasping points are

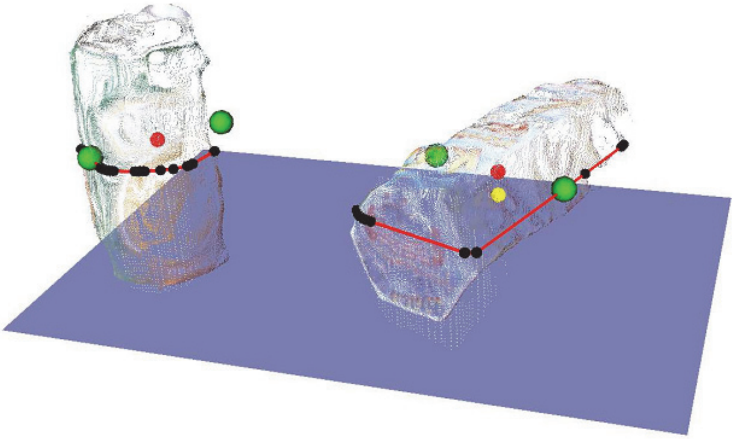

Fig. 8. Calculated grasping points (green) based on the combined laser range and stereo data.

required to be placed on parallel surfaces near the centre of the objects. To challenge the developed algorithm we included one object (Manner, object no. 6), which is too big for the used gripper. The algorithm should calculate realistic grasping points for object no. 6 in the pre-defined range, however it should recognize that the object is too large and the maximum opening angle of the hand is too small.

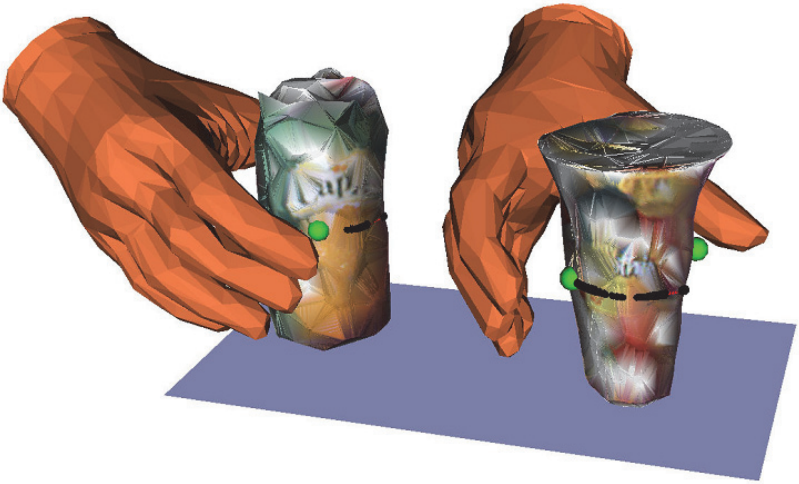

Fig. 9. The rotation axis of the hand is defined by the fingertip of the thumb and the index finger of the gripper. This rotation axis must be aligned with the axis defined by the grasping points. The calculated grasping pose of the gripper is by object no. 8 (Cappy) -32.5° and object no. 9 (Smoothie) -55°.

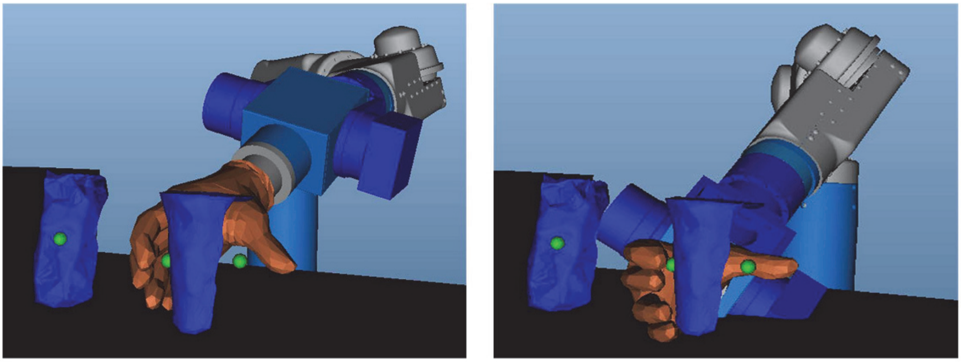

Fig. 10. The left Figure shows the calculated grasping points with an angle adjustment, where as the right Figure shows a collision with the table and a higher collision risk with the left object no. 8 (Cappy) as the left Figure with an angle adjustment of -55°.

In our work, we demonstrate that our grasping point detection algorithm and the validation with a *3D* model of the used gripper for unknown objects shows very good results, see Tab. 2. All tests were performed on a PC with *3.2GHz* Pentium dual-core processor and the average run time is about 463.78sec and the calculation of the optimal gripper pose needs about *380.63sec*, see Tab. 1 for the illustrated point cloud, see Fig. 9. The algorithm is implemented in C++ using the Visualization ToolKit (VTK)[5].

Calculation Steps	**Time [sec]**
Filter (Stereo Data)	*14sec*
Smooth (Stereo Data)	*4sec*
Mesh Generation	*58.81sec*
Segmentation	*2sec*
Grasp Point Detection	*4.34sec*
Grasp Angle	*380.63sec*
Overall	*463.78sec*

Table 1. Duration of calculation steps.

Tab. 2 illustrates the evaluation results of the detected grasping points by comparing them to the optimal grasping points as defined in Fig. 11. For the evaluation every object was scanned four times in combination with another object in each case. This analysis shows that a successful grasp based on stereo data with *82.5%* is considerably larger than with laser range data with *62.5%*. The combination of both data sets with *90%* definitely wins.

We tested every object with four different combined point clouds, as illustrated in Tab. 3. In no case the robot was able to grasp the test object no. 6 (Manner), because the size of the object is too big for the used gripper. This fact could be determined before with the computation of the grasping points, however the calculated grasping points are in the

[5] Open source software, http://public.kitware.com/vtk.

defined range of object no. 6. Thus the negative test object, as described in Section 4 was successfully tested.

No.	Objects	Laser [%]	Stereo [%]	Both [%]
1	Dextro	*100%*	*100%*	*100%*
2	Yippi	*0%*	*0%*	*25%*
3	Snickers	*100%*	*100%*	*100%*
4	Cafemio	*50%*	*100%*	*100%*
5	Exotic	*100%*	*100%*	*100%*
6	Manner	*75%*	*100%*	*100%*
7	Maroni	*75%*	*50%*	*75%*
8	Cappy	*25%*	*75%*	*100%*
9	Smoothie	*100%*	*100%*	*100%*
10	Koala	*0%*	*100%*	*100%*
Overall		*62.5%*	*82.5%*	*90%*

Table 2. Grasping rate of different objects on pre-defined grasping points.

Tab. 2 shows that the detected grasping points of object no. 2 (Yippi) are not ideal to grasp it. The *75%* in Tab. 3 were possible due to the rubber coating of the hand and the compliance of the object. For a grasp to be counted as successful, the robot had to grasp the object, lift it up and hold it without dropping it. On average, the robot picked up the unknown objects *85%* of the time, including the defined test object (Manner, object no. 6), which is too big for the used gripper. If object no. 6 is not regarded success rate is *95%*.

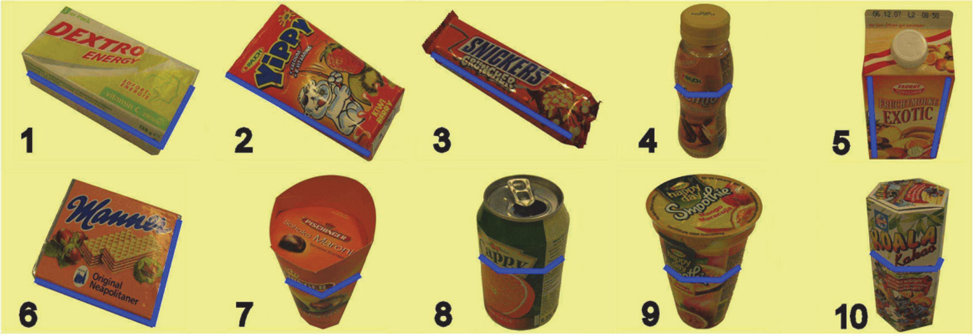

Fig. 11. Ten test objects. The blue lines represent the optimal positions for grasping points near the centre of the objects, depending on the used gripper. From left top: 1. Dextro, 2. Yippy, 3. Snickers, 4. Cafemio, 5. Exotic, 6. Manner, 7. Maroni, 8. Cappy, 9. Smoothie, 10. Koala.

For objects such as Dextro, Snickers, Cafemio, etc., the algorithm performed perfectly with a *100%* grasp success rate in our experiments. However, grasping objects such as Yippi or Maroni is more complicated, because of the strongly curved surfaces, and so its a greater challenge to successfully detect possible grasping points, so that even a small error in the grasping point identification, resulting in a failed grasp attempt.

No.	Objects	Grasp-Rate [%]
1	Dextro	*100%*
2	Yippi	*75%*
3	Snickers	*100%*
4	Cafemio	*100%*
5	Exotic	*100%*
6	Manner	*0%*
7	Maroni	*75%*
8	Cappy	*100%*
9	Smoothie	*100%*
10	Koala	*100%*
Overall		*85%*

Table 3. Successfully grasps with the robot based on point clouds from combined laser range and stereo data.

7. Conclusion and future work

In this work we present a framework to successfully calculate grasping points of unknown objects in *2.5D* point clouds from combined laser range and stereo data. The presented method shows high reliability. We calculate the grasping points based on the convex hull points, which are obtained from a plane parallel to the top surface plane in the height of the visible centre of the objects. This grasping point detection approach can be applied to a reasonable set of objects and for the use of stereo data textured objects should be used. The idea to use a *3D* model of the gripper to calculate the optimal gripper pose can be applied to every gripper type with a suitable *3D* model of the gripper. The presented algorithm was tested to successfully grasp every object with four different combined point clouds. In *85%* of all cases, the algorithm was able to grasp completely unknown objects.

Future work will extend this method to obtain more grasp points in a more generic sense. For example, with the proposed approach the robot could not figure out how to grasp a cup whose diameter is larger than the opening of the gripper. Such a cup could be grasped from above by grasping the rim of the cup. This method is limited to successfully convex objects. For this type of objects the algorithm must be extended, but with more heuristic functions the possibility to calculate wrong grasping points will be enhanced.

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.

8. References

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.

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.

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.

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.

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

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

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.

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

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 *2.5D* Range Data. IEEE 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.

Scharstein, D., Szeliski, R. (2002). A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms, International Journal of Computer Vision, Vol. 47, No. 1-3, pp. 7-42.

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.

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.

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.