Project Acronym:	GRASP
Project Type:	IP
Project Title:	Emergence of Cognitive Grasping through Introspection, Emulation and Surprise
Contract Number:	215821
Starting Date:	01-03-2008
Ending Date:	28-02-2012

Deliverable Number:	D6
Deliverable Title :	Integrating components for synthesis merging predicted, pre-reasoning, pre-attention, and visual attention
Type (Internal, Restricted, Public):	PU
Authors	L. Szumilas, M. Vincze, M. Richtsfeld; D. Burschka, Ch. Papazov; A. Argyros, V. Papadourakis; J. Bohg, D. Kragic; T. Asfour, M. Przybylski, F. Hecht, R. Dillmann
Contributing Partners	TUW, TUM, FORTH, KTH, UniKarl

Contractual Date of Delivery to the EC:	28-02-2009
Actual Date of Delivery to the EC:	28-02-2009

Contents

Chapter 1

Executive Summary

This report presents the work of year one in WP4. WP4 is concerned with perceiving the object and hand involved in the grasp and all contextual information relevant. With grasp context we refer to the information relevant to the grasp, which at its core includes the grasp points on the objects but also the relationship to the total object, the hand, the task, and the attention on the target object. The overall objective is to perceive grasping points on unknown objects by the end of the project. Work in year one concerned

- **Task 4.1 - Acquiring (perceiving, formalising) knowledge through hand-environment interaction** The objective of this task is to obtain many cues for observing the hand to object relationship for grasping. The idea is to use these cues not only to obtain information for the observation of a human handling objects but also for the robot executing the grasping.

Results of the work in this year concern several cues and methods on how they can be used to achieve the WP goals. Work on hand and object tracking is extended to to handle occlusions (Chapter 3). In this work colour models are used for tracking. Results will be used in WP2. In work for recognition of known objects (Chapter 4) texture is used to obtain a depth image and a known object model is used to obtain the object's pose. This work is useful to start integrating in WP7.

Two works are concerned with the scene modelling and attention mechanisms as a preparation for work in WP5. Support plane extraction (Chapter 5) segments the scene based on a depth map. It is a first step towards handling the case of unknown objects. A more detailed and accurate scene modelling follows, which improves the stereo image registration to build up a better description of the parts segmented (Chapter 6).

Using this segmentation the next step is to extract potential grasp points on the yet unknown object. Chapter 7 reports work to detect grasping points in depth images from laser and/or stereo data. It shows that a simple centre of gravity assumption helps to grasp a relatively large number of objects including non convex objects such as a banana. A step further towards a general method for grasping unknown objects is the work on semi-local object shape (Chapter 8). First results on varying types of cups look promising. It is thought the most powerful approach for the ultimate goal, however, it is young and only in the first year of its development. Finally, Appendix A presents a paper submitted to a journal on taking object-relations into account for grasping (Item A) and two workshop papers on on grasping from range images (Items B and C).

In summary, the results present a wide spectrum of cues that can be exploited both for learning from observing a human as well as make a robot perceive the grasping affordances. Work in year two will be towards making the methods suitable for a wider range of objects and to start the integration towards change grounding in WP3, detection in WP5, prediction in WP6, and the integration in WP7.

Chapter 2

Introduction

Grasping is one of these problems that seem to be so simple for humans but is still unsolved in the research area of robotics. Given a specific object, an embodiment of the robot and a task, the amount of applicable grasps is huge. One difficulty here is to decide which of these potential grasps to choose. This becomes even more problematic when information, e.g., about the object geometry, is noisy or incomplete.

There is quite a big body of related work dealing with the mentioned issues. Dependent on which kinds of objects are considered, we can classify the different approaches into three categories:

- Known Objects
  The goal here is to detect a known object and its pose to retrieve a suitable grasp from an experience database. *Known* here means that, e.g., a 3D or appearance model is available.
- Unknown Objects
  Systems that fall into this category usually try to approximate the shape of an unknown object and apply rules or heuristics to reduce the number of potential grasps.
- Familiar Objects
  These approaches try to re-use grasp experience that was gathered beforehand on objects of a specific category. The assumption made here is that new objects similar to the old ones can be grasped in a similar way.

A general observation that can be made when considering the related work is that there is a trade-off between the quality of an inferred grasp and the applicability of the method in a real world scenario. The more precise, accurate and detailed an object model, the more suitable it is for grasp planning based on criteria such as for example stability. However, in general it is difficult to extract a representation like that from real world sensors. If a representation is used that is more flexible, cruder and can incorporate noise, more assumptions have to be introduced regarding object geometry and considered grasps. Thus, although applicable in a real world scenario, the quality of inferred grasps will decrease.

Emerging from this observation, we formulate the basic requirements for an object representation. First of all, it has to be extractable from real world sensors such as mobile scanners or the final target in the project - stereo cameras. Secondly, it has to be rich enough to allow for the inference of the most important grasp parameters. Finally, it needs to extend towards new objects that have never been seen before. The work in Year is ordered according to this requirement.

2.1 Overview of WP4 in Year 1

WP4 is concerned with linking knowledge from grasp examples to objects and to provide methods for perceiving the grasp context. With grasp context we refer to the information relevant to the grasp, which at its core includes the grasp points on the objects but also the relationship to the total object, the hand, the task, and the attention on the target object.

Work in year one in WP4 was carried out in Task 4.1 "Acquiring (perceiving, formalising) knowledge through hand-environment interaction". The objective of this task is to obtain a large number of cues for

observing the hand to object relationship for grasping. The idea is to use these cues not only to obtain information for the observation of a human handling objects but also for the robot executing the grasping. This places WP4 right in the middle of the GRASP work packages, see Fig. 2.1. WP4 developments are used in the learning loop as well as the mission loop.

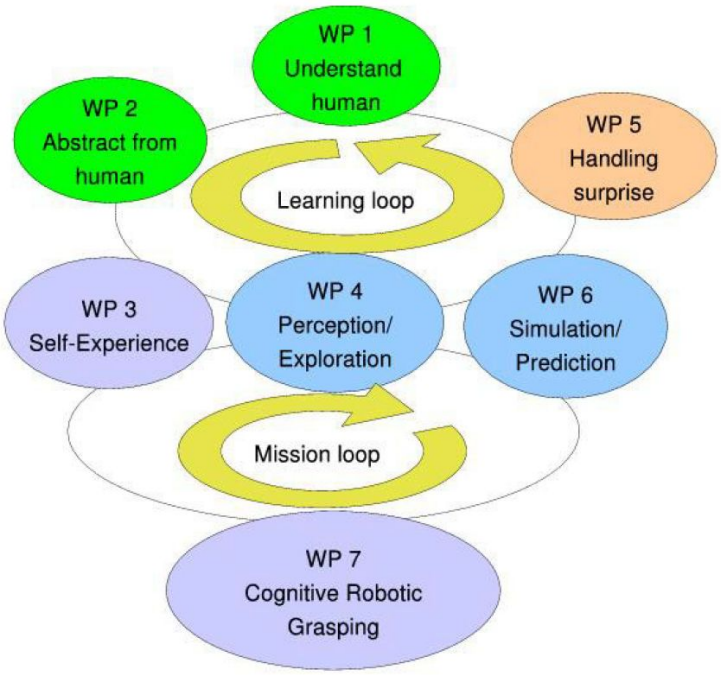


Figure 2.1: Relationship between different work packages (from Description of Work).

As part of the learning loop the task is to track hands and identify objects and their relation to it. This context-awareness (Goal 2 of the project) requires the system to maintain a representation of the surrounding world not only in its geometrical aspects but also adding additional knowledge about the world that is acquired gradually during the tutoring and later during the grounding phase (the link to WP3). The approach is to operate in the scene step by step. The system first categorises the scene entirely as a background model and structures from the background model will be moved to a more detailed foreground model once they are actually used for a grasping task (see WP5).

The tasks in year one related to the learning loop mainly concerned effort to acquiring a world model and knowledge through hand - environment interaction. The goal is to deliver the relevant information for grasping, specifically the location of the object and its relation to the hand. To this end we report in this deliverable work on

- Object tracking specifically suited to handle long term occlusions (Chapter 3), which happen frequently when grasping an object. Object tracking will also be used for reasons of efficiency, since a detailed scene analysis can not be repeated in real-time. It is rather advantageous to keep track of once segmented and detected objects.

Regarding the mission loop, the main role of WP4 is to achieve the objective of grasping any object. The hypothesis is that a system that acquires a representation of the graspable features or affordances, can handle more situations and thus allow an enhanced flexibility for grasp planning. Moreover, simply by the appropriate choice of the graspable features, it allows to integrate different task requirements which has been acknowledged as an important feature of human grasping. This direct linking of relevant perceptual and action features enables sequencing and switching between the action primitives based on sensory cues and affordances to infer a grasping strategy in a new situation.

The objectives are to combine the input image streams into a coherent perception of the scene, to deliver the relevant information for grasping, specifically the egocentric location of the object, its orientation, form, size, and cues related to grasping points. The last formulate the perceived affordance of grasping an

object. A major development needed in WP4 is to link potential grasp points with visual observations. Ultimately the task is to grasp not only known objects, but to grasp any object presented.

In year one work towards these goals is presented as follows:

- Appearance and Model-based Object Recognition and Pose Estimation (Chapter 4). Object models are taken and these objects can be recognised again and their location is obtained. This is useful for a first closing of the mission loop. Certainly, this needs to be extended towards grasping unknown objects, which will be approached in all the subsequent chapters.
- Support plane extraction (Chapter 5). When the target object is not known in advance, a good starting point is to segment the scene. This is done from fusing monocular or stereo images. We present how it is possible to extract main planes such as a support or table plane. These clusters present the constituent parts of the scene, such that subsequent processing can operate on these clusters more effectively.
- Scene modelling based on improved stereo image registration (Chapter 6). The aim is to obtain a more accurate and complete scene model. This is a preparatory step for the foreground - background detection in WP5 and can also be used for object grasp point detection in the next chapter. Since the binocular system in GRASP provides mostly incomplete views of the objects, a robust matching of partially reconstructed views is important for further processing steps.
- Detection of grasping points in depth images from laser and stereo data (Chapter 7). This work has the task of working towards grasping of unknown objects. We consider the segmented image parts and detect potential grasp points on the point cloud. This work user laser range data from single scans, because it is a currently available and a complete solution for performing early tests while other methods are still in development. We then show, that the method also works on depth images from stereo data, which is the type of sensor system that shall be mainly used in GRASP. The results show that for two-finger grasps simple symmetry can be successfully extracted. Workshop papers of this work are attached in Appendix A items B and C.
- Grasping based on Semi-Local Object Shape (Chapter 8). This work further extends grasping of unknown objects towards and open set of objects. It is an ongoing effort for object and graspable object part detection meant as complementary solution for 3D whenever 3D data is inaccurate (or occlusions happen). The first working prototypes will be ready in the second year of the project. At present work is on detection of groups of local shape as indicators of grasping affordances. As these features can be learnt, the method has the potential to be used on a wide spectrum of part forms.

The Deliverable concludes with a view on upcoming work. And finally, Appendix A (Chpater A) contains a paper submitted to a journal, which presents work on grasping based on relative object shape. This work takes into consideration the grasp points in relation to the object shape for the purpose of grasping.

Chapter 3

Object Tracking in the Presence of Long Term Occlusions

We present a robust object tracking algorithm that can handle spatially extended and temporally long object occlusions. The proposed method uses spatial and appearance based object characteristics to decide whether objects are observable or totally occluded by other objects and to successfully track them in time. The proposed approach is based on the concept of "object permanence" which suggests that an occluded object will re-emerge near its occluding object. No a priori assumption is made regarding the shape, size, colour or motion characteristics of the objects to be tracked. Instead, the method automatically builds appropriate object representations that enable robust and effective tracking and occlusion reasoning. The proposed approach has been evaluated in several video sequences where a human performs complex object manipulations in front of a colour camera. Experimental results demonstrate that the developed tracker is capable of handling several challenging situations, where the labels of objects are correctly maintained over time, despite the complex interaction among the tracked objects that leads to several layers of nested occlusions.

3.1 Introduction

Visual tracking of multiple objects is an important problem with instances in several application domains. Despite the huge amount of excellent research in the field, the effective and robust solution to the problem remains challenging in most of the realistic scenarios and settings. Part of the difficulty of the problem stems from the fact that even simple object interactions may result in occlusions that can be significant in both the temporal and the spatial dimensions. An object may totally disappear behind another object and reappear after considerable time, close to it, at a different location. Consider the example situation illustrated in Fig. 3.1 where a human grasps his keys to place them somewhere else. As soon as the keys are firmly grasped, they get totally disappeared. When the transfer is complete, the same keys reappear. Reasoning about the activities in this scene requires the capability to associate the same label to the object seen before and after hand manipulation. Clearly, the problem can become much more complicated, for example, in scenarios involving bi-manual interaction with several objects that may (or may not) differ in shape, size, appearance, etc. In this work, we present our approach to solving this kind of tracking problems. In this report we provide information on previous work, on representation issues and on results that have been obtained from the application of the proposed method in image sequences. The details of the proposed method is a theme of a scientific publication under preparation.

3.2 Previous work

A lot of approaches have already been proposed towards achieving a robust solution to the problem of object tracking in the presence of occlusions. Huang and Essa [HE05], provide a very informative overview of existing approaches. According to their classification, several existing methods handle occlusions implicitly. In the work of Khan [KS00] for people tracking, a person is segmented into classes of similar

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

Figure 3.1: Example situation where long-term occlusions need to be handled. From left to right, a human hand moves towards the keys, grasps them fully occluding them and transfers them to a different position. We are interested in a tracking framework which, without a priori information about the tracked objects, will be able to infer that the object appearing in the fourth frame is the one that disappeared in the second.

color using the EM algorithm. Then, the maximization of the a posteriori probability of these classes drive their tracking from frame to frame. McKenna [MJD+00] and Marques [MJAL03] employ appearance models of tracked regions to identify people after the occurrence of occlusions but their approach provides limited support of complex interactions. In [IM01] Isard introduces a Bayesian filter for tracking a potentially varying number of objects. A particle filter is used to perform joint inference on both the number of objects present and their configurations. Occlusion handling is achieved by incorporating the number of interacting persons into the observation model and inferring it using a Bayes Network. Jepson [JFEM03] proposes a framework for learning appearance models to be used for motion-based tracking of natural objects. The appearance model involves a mixture of stable image structure, learned over long time courses, along with two-frame motion information and an outlier process. This model is used in a motion-based tracking algorithm to provide robustness in the presence of outliers, such as those caused by occlusions.

Several other methods have been proposed that treat explicitly the problem of occlusions. Rehg [RK95] describes a framework for local tracking of self-occluding motion, in which one part of an object obstructs the visibility of another. His approach uses a kinematic model to predict occlusions and windowed templates to track partially occluded objects. Brostow [BE99] presents a method to decompose video sequences into layers that represent the relative depths of complex scenes. Activity in a scene is used to extract temporal occlusion events, which are in turn, used to classify objects on the basis of whether they occlude or they are occluded. Jojic [JF01] proposes a technique for automatically learning probabilistic 2D appearance maps and masks of moving occluding objects. The model explains each input image as a layered composition of flexible sprites. A variational expectation maximization algorithm is used to learn a mixture of sprites from a video sequence. Tao [TSK02] decomposes video frames into coherent 2D motion layers and introduces a complete dynamic motion layer representation in which spatial and temporal constraints on shape, motion and appearance are estimated using the EM algorithm. The proposed method has been applied in an airborne vehicle tracking system and examples of tracking vehicles in complex interactions are demonstrated. Zhou [ZT03] introduces the concept of background occluding layers and explicitly infer depth ordering of foreground layers. A MAP estimation framework is proposed to simultaneously update the motion layer parameters, the ordering parameters, and the background occluding layers. Wu [WYH03] proposes a dynamic Bayesian network which accommodates an extra hidden process for occlusion. The statistical inference of such a hidden process reveals the occlusion relations among different targets. In [AL04], Argyros proposes a method for tracking multiple skin coloured objects in images acquired by a possibly moving camera. The proposed method encompasses a collection of techniques that enable the modelling and detection of skin-colored objects as well as their temporal association in image sequences. Tracking over time is realized through a technique that handles multiple skin-colored objects moving in complex trajectories and occluding each other in front of a possibly moving camera. The approach of Huang [HE05] incorporates (i) a region-level association process and (ii) a object-level localization process to track objects through long periods of occlusions. Region association problem is approached as a constrained optimization problem and solved using Genetic Algorithm (GA). Objects are localized using adaptive appearance models, spatial distributions and occlusion relationships. Yu [YMC07] proposed a framework for treating the general multiple target tracking problem, which is formulated in terms of finding the best spatial and temporal association of observations that maximizes the consistency of both motion and appearance of object trajectories. Leibe et al. [LSVG07] consider multi-object tracking as a search for the globally optimal set of space-time trajectories which provides the

best explanation for the current image and for all evidence collected so far, while satisfying the constraints that no two objects may occupy the same physical space, nor explain the same image pixels at any point in time. In a recent work, Zhang [ZLN08] proposed a network flow based optimization method for data association in multiple object tracking. The maximum-a-posteriori (MAP) data association problem is mapped into a cost-flow network with a non-overlap constraint on trajectories. The optimal data association is found by a min-cost flow algorithm in the network that is augmented with an explicit occlusion model (EOM) to track long-term occlusions.

3.3 Proposed method

The method proposed in this work uses two types of information regarding the scene. The first, is the result of scene background subtraction and produces a map showing "where" action takes place in the scene. The second type of information comes from the estimation of several (one per object) Gaussian Mixture Models of color that represent "what" is the appearance of moving objects. The proposed method does not make any assumptions about the shape, the appearance the number or the motion characteristics of the tracked objects. On the contrary, such information is automatically derived and properly updated in time. Much of the success of the method depends on a mechanism inspired by [AL04] that properly associates foreground pixels to different objects. Thus, models of object appearance can be properly maintained and tracked. Occlusion handling is treated through a method founded on the principle of "object permanence" [Pia54, BSW85], studied in the context of human psychology. Object permanence refers to the ability of children to realize that an object exists even when it cannot be seen. Piaget [Pia54] believed that most infants grasp the object permanence concept when they are at the age of about eight or nine months old. Other, more recent studies [BSW85], indicate that infants can reach the object permanence stage at the age of five months, showing the fundamental role of the concept in visual perception.

The methods that are closest to our approach are the ones proposed by [AL04] and by Huang et al [HE05]. The approach in [AL04], handles skin-colored objects and, in general, requires prior training to the color model of the objects to be tracked. Our approach may handle objects of completely different appearances for which no a priori information is assumed to be known. In addition to the more complete appearance models, the exploitation of the concept of "object permanence" makes the proposed method much more competent in handling long term occlusions.

Huang [HE05] also used the concept of "object permanence" to successfully handle long term occlusions of a varying number of objects over extended image sequences. However, their approach relies heavily on the correct association of blobs between frames. As a side effect, their method fails to handle objects of similar appearance even in the case of limited interaction between them.

Figure 3.3 illustrates the information flow of the proposed algorithm. Each frame of the input image sequence is first background subtracted [Z.Z04] to detect foreground pixels and to form distinct blobs, i.e regions of connected foreground pixels. Assuming a still camera, background subtraction gives rise to a change mask that can be attributed to the moving objects. A set of objects that must be correctly associated to the pixels of the detected foreground blobs is also maintained. Clearly, even in the simple case of partial occlusions, there is no one-to-one mapping between objects and blobs. Therefore, the goals of the proposed method is to exploit spatial and photometric object information to (a) associate foreground blob pixels with objects, (b) investigate occlusion relationships between objects, (c) update the object models and, (d) use all extracted information to enable tracking.

With respect to object modelling, no a priori knowledge regarding the object's 2D or 3D shape, appearance or motion is assumed. To achieve tracking, simple, generic object models need to be automatically built and maintained. Each object is represented with a parametric model that takes into account both the spatial layout and the photometric appearance of objects. The object model consists of an ellipse that describes the position and spatial distribution of an object and a mixture of Gaussians that describes its color distribution.

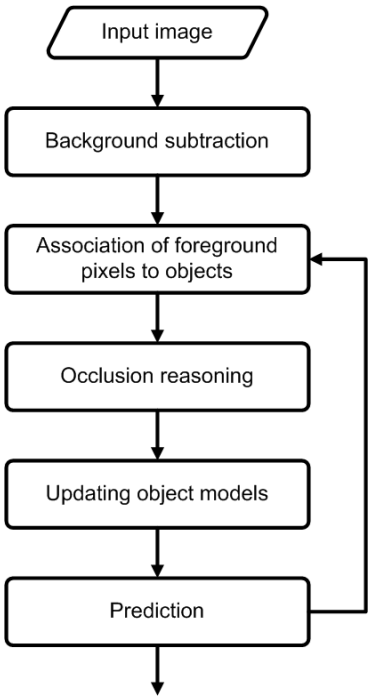


Figure 3.2: The flow diagram of the proposed method for tracking multiple objects in the presence of long term occlusions.

3.4 Results

The proposed method has been tested and evaluated in a series of image sequences demonstrating challenging tracking scenarios. Results from two representative video sequences are presented in this work. In all reported experiments, input sequences consisted of standard VGA resolution images (640×480) acquired at 20Hz.

The first test image sequence ("Objects" sequence) is 1280 frames long and shows a human manipulating several objects on a table desktop. Characteristic snapshots with results from this sequence are shown in Fig.3.3. Initially, the human brings into the scene a basket containing several objects. Then, he empties the basket, interacts with the objects, fills the basket again and finally empties it once more. At the start of the experiment, the system has no a priori knowledge about the type, size, color, shape or motion of the objects to be observed. At the end of the experiment the proposed method has been able to track individual objects and has built a model of their color appearance.

More specifically, Fig. 3.3(a) shows the empty desktop on which the experiment is performed and of which a background model has been built. Since background subtraction is not the main focus of the current work, the background has been intentionally kept simple to simplify the corresponding process. In Fig. 3.3(b), the human hand has already brought into the scene a box containing a few objects. Having no a priori knowledge about the scene other than a background model of it, the system identifies the constellation of the hand, the blue box and the rest of the objects as a single multicolour object, for which it builds a single object model[1]. As soon as the hand leaves the box on the table (Fig. 3.3(c)), the originally connected set of pixels becomes disconnected. The original object hypothesis (blue color) is assigned to the blue box object because of the color relevance. Another object (hand, red contour) is automatically generated. For the next frames, the hand color appearance model is updated. The same happens also to the appearance model of the blue box, in which the components corresponding to the previously joined hand, now vanish. The hand interacts with the box again (Fig. 3.3(d)). Now, the color models built assist the method in correctly assigning the pixels of the single connected blob to the two object hypotheses (hand, box). In Fig. 3.3(e), the hand has taken the pincer off the blue box and

[1]Individual objects are identified through the use of different colours for their contours and through object arithmetic labels printed on object centroid. Thus, an object is successfully tracked if it maintains the same color and label in all of its occurrences.

moves it to another position on the table. For the moment, the method interprets this as a change in the appearance of the hand and, at that stage, the pincer appears as part of the hand object. This is because the pincer has never been observed in isolation but only as a part of another object (box). As soon as the hand leaves the pincer on the table, the pincer is understood as an individual object (Fig. 3.3(f), purple contour). The identity of the pincer object is not lost even when the hand passes several times over it, grasps it and moves it to another place on the table (Fig. 3.3(g)-(j)). In a similar manner, the hand empties the rest of the basket's objects. As shown in Fig. 3.3(k), the hand, the box and the pincer maintain their original identity, while the two other objects have acquired their own object identities. In Fig. 3.3(l), the hand has grasped the green object and has used it to completely occlude the yellow one. The full occlusion has been signalled and both object hypotheses still live under the observed region of the occluding object. Both objects are transferred to a new position, the hand removes the green occluding object (Fig. 3.3(m)) and the correct identity for the yellow object is still maintained. The green object is again brought on top of the yellow one, fully occluding it once more. This time, the blue box is also brought on top of the green object creating a nested occlusion (Fig. 3.3(o)). When the hand brings the green object again in sight dragging it under the blue box, the green object still maintains its original identity (Fig. 3.3(p)). The same happens to the yellow object (Fig. 3.3(q)). The manipulation of objects continues; the hand brings all objects again into the blue basket and starts roving the latter around (Fig. 3.3(r),(s)). The experiment ends with the hand emptying the basket once more (Fig. 3.3(t)). Correct object identities are still maintained.

A second experiment was performed on the "lemons" sequence (presented in Fig. 3.4), demonstrating that the method succeeds in handling occlusions when tracking objects of similar appearance. In a setting that is similar to the previous one, two hands appear in front of a camera (Fig. 3.4(a)) and are assigned two different object identities. The hand appearing at the left (red contour) holds two lemons which are integral part of the hand object as long as the hand is holding them. As soon as lemons appear in isolation (Figs. 3.4(b),(c)) they get their own object labels. In Fig. 3.4(d), each hand grasps a lemon, fully occludes it (Fig. 3.4(e)) and then reveals it (Fig. 3.4(f)). Lemon identities have been maintained. The two hands grasp the two lemons totally occluding them and then cross (Fig. 3.4(g)). Hands reveal what they carry (Fig. 3.4(h),(i)), showing that despite the complex interaction of two similar looking objects with two other similar looking objects and the simultaneous presence of two full occlusions, the identities of the lemons are correctly tracked. The experiment ends after the hands leave the objects they hold on table (Figs. 3.4(j)-(l)).

3.5 Discussion

In this work, we presented a method for tracking multiple objects in the presence of occlusions with long temporal duration and large spatial extends. The proposed method can cope successfully with multiple objects dynamically entering and exiting the field of view of a camera and interacting in complex patterns. No a priori information is assumed for the object's structure, appearance or motion. All information that is required for tracking and occlusion reasoning is dynamically collected, maintained and updated. Tracking is performed by systematically assigning pixels of foreground blobs to simple geometrical models of objects, taking into account object's appearance. Occlusion reasoning is based on the concept of "object permanence". Experimental results demonstrate the capability of the proposed approach in handling challenging situations involving multiple interacting objects even in cases of long-term and nested occlusion relationships. Future directions include the study of more elaborate spatial and appearance models that could provide with more accurate object representations permitting the handling of even more demanding tracking scenarios.

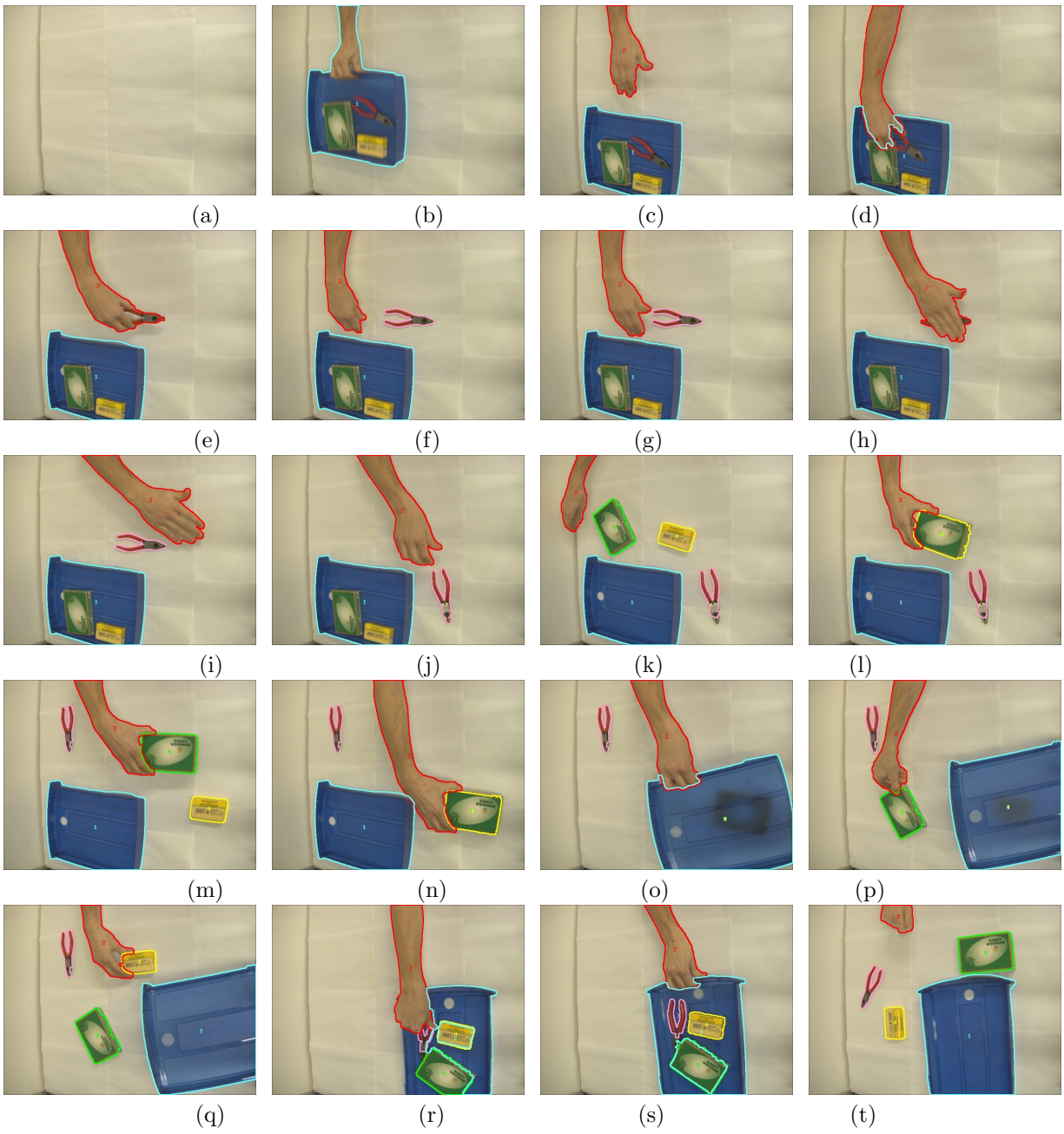


Figure 3.3: Characteristic snapshots from the tracking experiment on the "objects" image sequence.

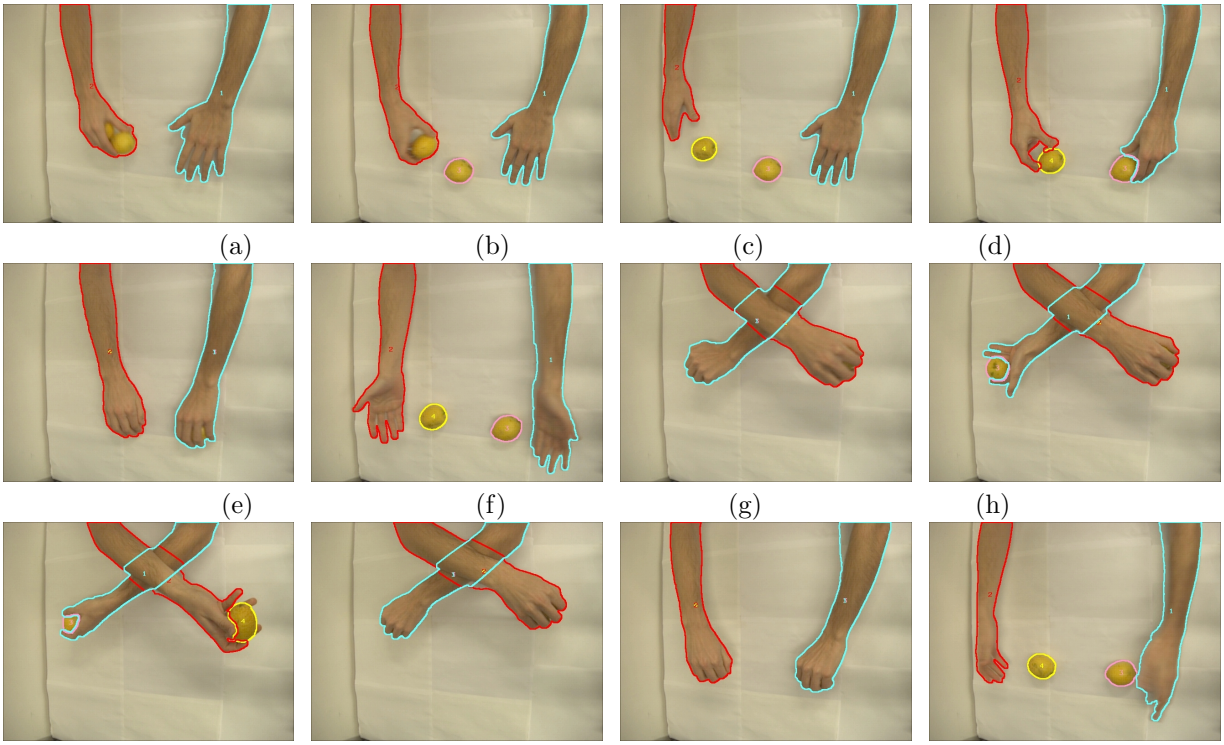


Figure 3.4: Characteristic snapshots from the tracking experiment on the “lemons” image sequence.

Chapter 4

Vision for Grasping of known Objects

4.1 Appearance-based and Model-based Object Recognition and Pose Estimation

In our previous work, we have developed approaches for the robust recognition and accurate 6D pose estimation of single-colored and textured objects [AAD07]. The approaches allow for frame rate tracking and can thus be used within closed-loop visual servoing tasks. On the humanoid robot ARMAR III, implementations of these approaches are successfully applied for scene analysis and grasping using a visual servoing approach, as shown in [AAV+08].

4.1.1 Single-colored Objects

For single-colored objects, stereo triangulation, matching of global object views and on-line projection of a 3D model of the object are combined. The requirement for the approach is global segmentation of the objects, which is accomplished by colour segmentation. For training, a 3D model of the object is used to produce a view set in simulation. This view set is compressed by using the Principal Component Analysis (PCA). Along with each view, the orientation with which that view was produced is stored.

For recognition, each region candidate obtained by the segmentation routine is matched against the database. An initial orientation estimate is given by orientation information that was stored together with the matched view. An initial position estimate is given by the stereo triangulation result of the segmented regions in the left and right camera image.

Splitting up position and orientation computation in this way leads to an error-prone pose estimate, since both the position and the orientation of the object influence the appearance of the object in the image. Furthermore, the triangulation result of the centroids depends on the view of the object and thus cannot serve as a constant reference point. In order to solve these problems, a pose correction algorithm is applied, which make use of on-line projection of the 3D model. This pose correction algorithm is an iterative procedure, which in each iteration corrects the position vector by computing the triangulation error in simulation and correcting the orientation estimate on the basis of the updated position estimate.

In order to achieve maximum recognition robustness, each stored object representation is treated as a separate hypothesis and is verified, rather than computing the best matching view from all views of the database i.e. determining the object identity on the basis of the pure 2D appearance. For each object hypothesis, the pose is computed and the on-line simulation result using the estimated pose is used as input to the verification procedure. In this way, two similar views of two different objects can be distinguished reliably, since for a correct hypothesis the estimated pose must produce the same view in simulation as in the real view, in terms of shape and size.

4.1.2 Textured Objects

For textured objects, a two-step approach using local features is applied. First, the object is recognized including 2D localization, which is accomplished on the basis of 2D feature correspondences. In order to achieve frame rate tracking performance, the SIFT descriptor is combined with the Harris corner detector, including an extension to achieve scale-invariance *without* a time-consuming scale space analysis. On the basis of the 2D localization result, a 6D pose estimate of the object is computed by making use of the stereo system.

The 2D localization is computed by a homography, which is computed with a pipeline that consists of a Hough transform, a RANSAC method, iterative affine transformation estimation, and final full homography estimation. As before, each object hypothesis is verified separately, i.e. the pipeline is applied for each stored object representation. Each hypothesis that successfully passes the verification pipeline yields an instance of the object.

The conventional approach to 6D pose estimation within such a framework is based on 2D-3D point correspondence, e.g. by using the POSIT algorithm or more recent variants. In contrast, we apply a stereo-based method which does not suffer from instabilities caused by inaccurate homography estimates. For this purpose, interest points within in the localized 2D area of the object are collected and correlated with the right camera image, yielding a sparse depth map. The resulting point cloud is registered with the object model. For planar objects, this is accomplished by computing the regression plane and intersecting the view rays through the corner points.

Results of exemplary applications of the system are shown in Fig. 4.1.

Figure 4.1: Results of application of the object recognition and pose estimated system. The pose estimates are applied to the wire frame models of the objects and projected into the left camera image.

4.2 Detection and Grasping based on point clouds

4.2.1 Incorporating Laser Range Data for Stereo-based Grasping with a Humanoid Robot

For grasp analysis on the basis of high-quality 3D representations of objects that are acquired by a Laser range scanner, the box decomposition method presented in [HK08] is applied. The box decomposition method yields a set of potential grasp approach directions and grasp starting points, which are determined by the surfaces of suitable extracted boxes. The object models are acquired by the object modelling center presented in [**?**] (`http://wwwiaim.ira.uka.de/ObjectModels`), which uses the Laser range scanner Minolta Vivid VI-900.

The problem for applying the results of such an off-line grasp analysis for on-line grasp execution with a humanoid robot is that the object representation for recognition and pose estimation and the Laser range data are given in two different coordinate systems. The relationship between these two coordinate systems is given by a *fixed* rigid body transformation, which must be determined only once.

For on-line recognition and pose estimation, the approach for textured objects as described in Section 4.1.2

is adopted. In order to calibrate the mentioned rigid body transformation, a tool has been developed. This tool computes for one given scene the pose of the object of interest by using the recognition system. At the same time the scanned model is mapped into the same stereo image pair of the scene, and its pose is adjusted manually so that the model projection matches the stereo views. The searched rigid body transformation is then given by the transformation between the automatically computed pose estimate and the manually adjusted pose.

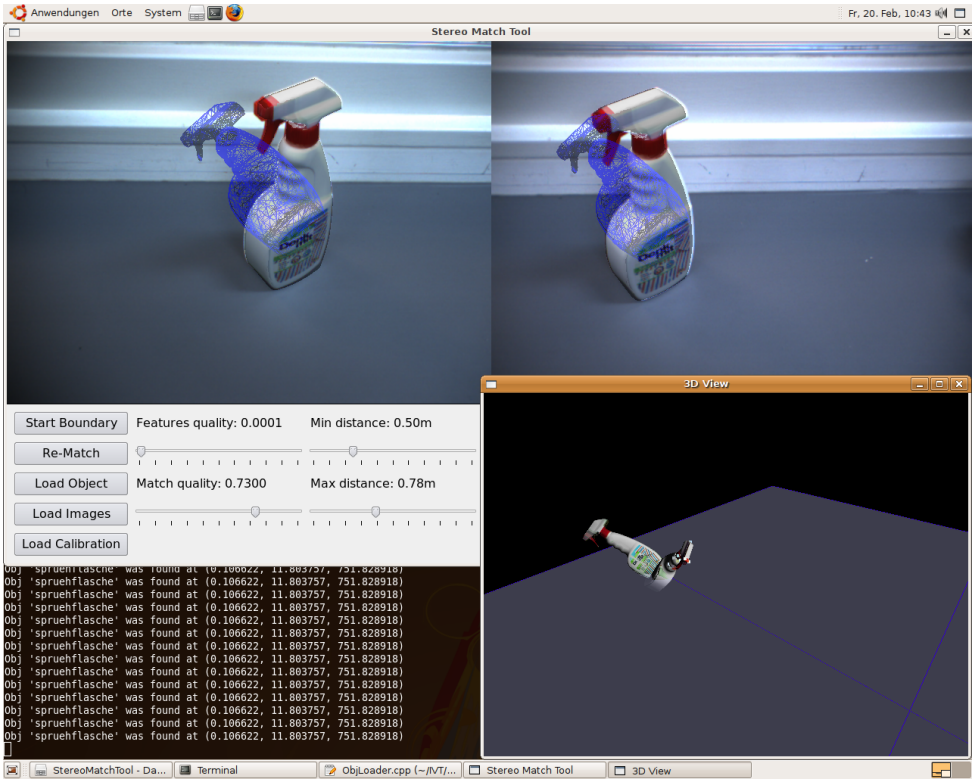


Figure 4.2: Screenshot of the tool for manual calibration of the searched rigid body transformation. The manually adjusted pose of the high-quality object model is visualized as a transparent, textured mesh. If applying the result of the textured-based object pose estimation system directly to this representation, the pose illustrated by the blue wire frame model is obtained. The searched transformation is the rigid body transformation between these two poses.

By applying this approach, the benefits of a Laser range scan can be exploited at runtime. The recognition system delivers pose estimates of this high-quality representation by applying the calibrated rigid body transformation to the pose estimate of the internal representation. The processing rate amounts to approx. 20 Hz. An exemplary result of the final pose estimate is shown in Fig. 4.3.

Figure 4.3: Result of the final pose estimate for an example scene, after application of the calibrated rigid body transformation.

Chapter 5

Point-Based 3D Clustering of the Scene

The manipulation tasks require a knowledge about the 3D structures in the environment to define grasping points on mission relevant objects and to avoid collisions with obstacles. The 3D reconstruction was part of our activities to generate the input data for further processing. We use real-time correlation-based algorithms for dense 3D reconstruction from binocular stereo and added monocular reconstruction to complete the object representations in front of the robot.

5.1 Monocular Reconstruction

While for binocular reconstruction we relied on correlation-based approaches providing dense 3D reconstructions of the environment, an interesting question is how to complete the 3D structure of an object without the necessity of moving the robot around the scene to resolve the self-occlusions. Our approach is to use an additional camera in the wrist of the robot that can provide additional information to reconstruct the missing 3D structure. Such a camera is common in robotic manipulation systems. It is used to implement visual servoing approaches and to increase the accuracy while manipulating objects.

The goal is to estimate the motion of the camera to define the epipolar geometry between images in the motion sequence. Once the epipolar geometry is known, the system can reconstruct the 3D information from the optical flow in the images. We use our algorithm presented in [BH04] to localize the monocular camera in the scene. A known reference pattern on the table simplifies the processing as depicted in Fig. 5.1.

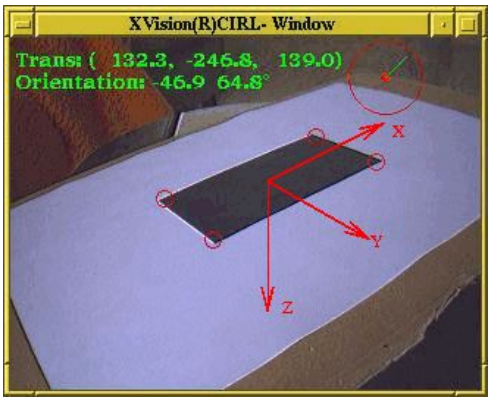


Figure 5.1: Pose estimation from known reference structure.

We extended the OpenVis3D code based on [OA05] to calculate dense optical flow fields from two images. Results of this computation are shown in Fig. 5.2.

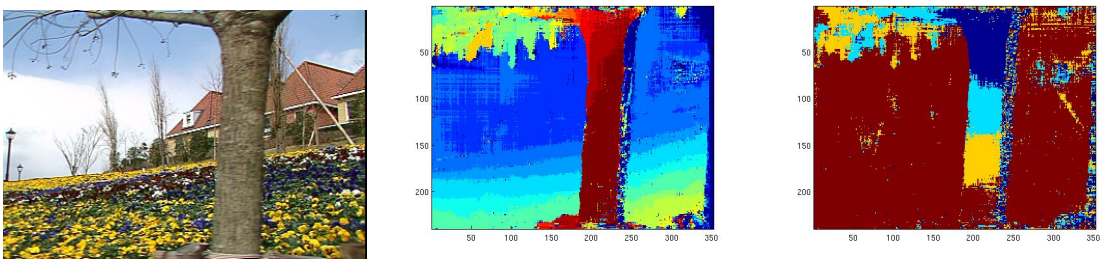


Figure 5.2: An original motion sequence is used to calculate horizontal and vertical optical flow.

We define a *spherical disparity* equation to calculate the distances along the rays of observations (n'_i, n_i) in two camera images. The baseline B of the system corresponds to the distance $\frac{T}{m}$ traveled by the camera and it is "divided" by the *spherical disparity* s

$$s = (n'_i \quad - R \cdot n_i)\,, \tag{5.1}$$

which represents a difference vector between the two projections (n'_i, n_i) rotated to the coordinate frame of n'_i in which $\frac{T}{m}$ is defined. Since there is no significant plane as it is the case for the image plane of a coplanar binocular system, a normal distance definition of Z does not make any sense and it is replaced by the radial distance to the focal points of both projections $(\frac{D_i}{m}, \frac{D'_i}{m})$. The reconstructed depths are scaled down to the same scale as T.

We plan to continue the implementation of this promising monocular approach in the second funding period to complete the dense reconstruction of the scene based without the necessity to move the robot around the table with the objects to be manipulated.

5.2 Supporting Plane Subtraction

Reconstruction algorithms generate *background representations* (see D7 from WP5) that do not distinguish between mission relevant objects and the scene. An initial clustering of 3D point clouds can be achieved by subtracting a supporting plane. In our experiments such a supporting plane represents a table, where the objects are placed (Fig. 5.3).

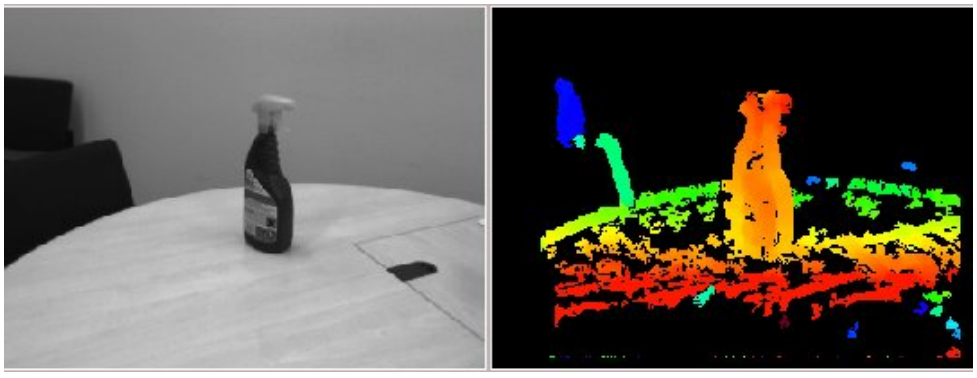


Figure 5.3: 3D information from a typical table setup.

Many algorithms are based on RANSAC to fit planes into the reconstructed 3D data. These approaches are sensitive to calibration errors of the extrinsic parameters of a stereo rig. We use an approach working directly on disparity data omitting the necessity of explicit 3D reconstruction of the points. This algorithm is integrated into the stereo reconstruction algorithm and uses the disparity images to extract the plane equations.

Following the derivation in [BH02], given a plane $\mathcal{P}_r$ in R^3,

$$\mathcal{P}_r : a_r x + b_r y + c_r z = d_r \tag{5.2}$$

the equivalent disparity plane is given by

$$\begin{aligned} \forall z \neq 0 : \quad a_r \frac{x}{z} + b_r \frac{y}{z} + c_r &= \frac{d_r}{z} \\ a_r u + b_r v + c_r &= k \cdot D(u,v) \\ \text{with } u = \frac{x}{z},\ v = \frac{y}{z}, \quad k &= \frac{d_r}{B}. \end{aligned} \tag{5.3}$$

where $D(u,v)$ represents the disparity at image coordinates (u,v). Clearly, (5.3) describes a plane in UVD space. We can write (5.3) in the following form

$$\begin{aligned} D(u,v) &= \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{pmatrix} \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{n_r^*} \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \\ \text{with} \qquad & \rho_1 = \frac{a_r}{k},\ \rho_2 = \frac{b_r}{k},\ \rho_3 = \frac{c_r}{k} \end{aligned} \tag{5.4}$$

In this form, it is clear that plane membership can now be checked simply by computing the dot product between the normal vector to the plane and the image coordinates, then comparing this to the disparity value at that location.

Discontinuities in the disparity are natural boundaries for all surfaces. Thus, the first step in plane detection is to compute the gradient magnitude of the disparity image. Locations with large gradient are then set to a negative value; locations with low gradient are set to 1, and unreconstructed locations are set to zero. This prevents the seed selection step described below from selecting points too close to a boundary or in regions with significant clutter or high aspect ratio.

We then remove, one by one, surfaces from the original disparity image $\mathcal{I}_\mathcal{D}$. We first convolve the disparity image with a box filter $\mathcal{M}$ of size $N \times N$. The kernel size N is chosen to match the minimum expected size of the imaged traffic sign in the image. This convolution results in a new pseudo-image $\mathcal{K}$

$$\mathcal{K} = \mathcal{M} * \mathcal{I}_\mathcal{D} \tag{5.5}$$

where the image values describe the size of the homogeneous region around them. All points with a value $\mathcal{K}(u,v) > 0.7 \cdot N^2$ are now considered good seeds for the surface estimation. In other words, we allow 30% of the reconstructed pixels in a region to be drop-outs or 15% to be on a boundary.

We now start at the selected seed points and we estimate the direction vector $\mathbf{n_r^*}$ from (5.4) by solving the following linear equation for $\mathbf{n_r^*}$:

$$\begin{aligned} &\forall D_i > 0 : \\ &\begin{pmatrix} \sum u_i \cdot D_i \\ \sum v_i \cdot D_i \\ \sum D_i \end{pmatrix} = \begin{pmatrix} \sum u_i^2 & \sum u_i v_i & \sum u_i \\ \sum u_i v_i & \sum v_i^2 & \sum v_i \\ \sum u_i & \sum v_i & \sum 1 \end{pmatrix} \cdot \mathbf{n_r^*} \end{aligned} \tag{5.6}$$

We use the result from (5.6) in (5.4) to verify the planarity of the contiguous region in the image. Pixels associated with a plane are zeroed in the disparity image $\mathcal{I}_\mathcal{D}$ and the pseudo-image $\mathcal{K}$ calculated in (5.5). If all pixels in the region fulfill the requirement from (5.4) then we save the resulting region with its 3D parameters as a center point m and 2 vectors (r_1, r_2) along the principal axes of the region. The lengths of these vectors specify the size of the region.

This process is repeated until no pixel satisfies $\mathcal{K}(u,v) > 0.7 \cdot N^2$. After subtracting of the supporting plane the objects stay as separated point clouds in space that can be categorized and, in case of an interaction with the human hand, moved to the list of foreground objects (Fig. 5.4).

The other alternative that we want to implement in the next step is a segmentation of point clouds based on motion in the scene. Human induced motion of objects moves point clouds in the world. These point clouds define interesting, mission-relevant objects that need to be inspected by the system. This classifies them as foreground objects. After subtracting of the supporting plane the objects stay as separated point

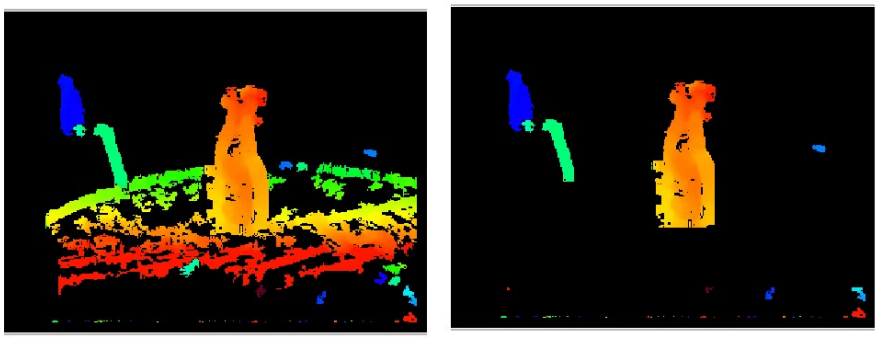


Figure 5.4: Result of the supporting plane subtraction.

clouds in space that can be categorized and, in case of an interaction with the human hand, moved to the list of foreground objects.

At the current stage of the project, we define objects separated by the supporting plane subtraction which were touched by the human hand as mission relevant objects (*foreground*).

Chapter 6

Stochastic Optimisation for Rigid Point Set Registration

6.1 Motivation

Stereo reconstruction provides only three-dimensional data in areas, where sufficient texture is present. Therefore, typical reconstruction results are usually incomplete with holes in the areas of poor texture. Additionally, backfaces of objects that are not observed by the cameras usually stay unexplored. Objects need to be fitted based on a noisy, partial, and incomplete view. Instead of using the reconstruction data directly, we propose a 3D model fitting approach to replace the noisy reconstruction by the ideal model description from the knowledge database developed in WP5 as part of the ontology. This approach solves the problem of accurate and complete 3D reconstruction and object localization in 3D.

We propose a new method for pairwise correspondence free rigid point set registration. We pay special attention to outlier robustness and globally optimal alignment. The problem of registering two point clouds in space is converted to a minimization of a nonlinear cost function. We propose a novel cost function, that aims to reduce the impact of noise—common for real world data. Its definition is based on the input point sets and is directly related to the quality of a concrete rigid transform between them. Compared to least squares like methods, which are known to be very sensitive to outliers, our cost function leads to greater robustness in this regard. In order to achieve a global optimal registration, we develop a new stochastic approach for global optimization. Tests on both artificial and real world data show the robustness of the proposed registration algorithm to occlusions and noise.

Point set registration is a fundamental problem in computational geometry with applications in the fields of computer vision, computer graphics, image processing and many others.

The problem can be formulated as follows. Given two finite point sets $\mathbf{M} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subset \mathbb{R}^3$ and $\mathbf{S} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\} \subset \mathbb{R}^3$ find a mapping $T : \mathbb{R}^3 \to \mathbb{R}^3$ such that the point set $T(\mathbf{M}) = \{T(\mathbf{x}_1), \ldots, T(\mathbf{x}_m)\}$ is optimally aligned in some sense to the set $\mathbf{S}$. $\mathbf{M}$ is referred to as the model point set or just the model and $\mathbf{S}$ is termed the scene point set or just the scene. Points from $\mathbf{M}$ and $\mathbf{S}$ are called model points respectively scene points.

If T is a rigid transform, i.e. $T(\mathbf{x}) = R(\mathbf{x}) + \mathbf{t}$ for a rotation R and a translation $\mathbf{t}$ we have the problem of a rigid point set registration. This special case is of major importance for the task of object recognition, tracking, localization and mapping, object modeling, just to name a few.

The rest of the chapter is organized as follows. Section 6.2 gives an overview over previous work on rigid point set registration and stochastic optimization. In section 6.3, we define the task of aligning two point sets as a functional minimization problem over the set of rigid transforms in three dimensional space. In this special case, the more general problem of minimizing a functional is equivalent to the minimization of a continuous cost function over a compact subset of $\mathbb{R}^6$. In section 6.4, we introduce our novel stochastic approach for global optimization. Section 6.5 presents experimental results obtained by our registration method. Conclusions and future work are drawn in the final section 6.6 of this chapter.

6.2 Related Work

6.2.1 Rigid Point Set Registration

The Iterative Closest Point (ICP) algorithm is doubtlessly the most popular point set registration method. Since its introduction by Chen and Medioni [CM91] and Besl and McKay [BM92] a variety of improvements has been proposed in the literature. A good summary as well as new results on acceleration of ICP like algorithms has been given by Rusinkiewicz and Levoy [RL01]. A major drawback of all these ICP variants is that they assume a good initial guess for the orientation of the model point set (with respect to the scene point set). This orientation is improved in an iterative fashion until an optimal rigid transform is found. Whether the solution is globally the optimal one or not depends heavily on the initial guess. Another disadvantage of the methods compared by Rusinkiewicz and Levoy [RL01] is that they use local surface features like surface normals which can not be computed very reliably in the presence of noise.

The formulation of the registration task as the minimization of a cost function has already been introduced in the literature. Blais and Levine [BL95] define a cost function which measures the quality of registration between two data sets and minimize it by the Very Fast Simulated Reannealing (VFSR) algorithm [Ing89]. Although VFSR is suitable for the task of global optimization, i.e. the "good initial guess" assumption could be removed, the definition of the cost function used by Blais and Levine [BL95] is based on this assumption. Furthermore their method is applicable to range images only and is not suitable for general point sets. Other methods designed for range images are proposed in [CHC99, DWJ97].

All of the registration methods cited above rely on some kind of model to scene correspondence. Since its establishment between two arbitrary oriented general point sets is not trivial at all, some authors try to solve the registration problem without a correspondence estimation. The approach we use is most related to the ones proposed by Mitra *et al*let@tokeneonedot[MGPG04] and Pottmann *et al*let@tokeneonedot[PHYH06]. They express the registration problem as a minimization of a cost function, whose definition is not based on a correspondence between the model and the scene. For its minimization however a local optimization method is used. This results in the already mentioned strong dependence on a good initial transform estimation.

The major difference between our algorithm and the ones introduced in [MGPG04, PHYH06] is that we propose a new *noise-resistant* cost function and develop a novel approach for its *global* minimization.

6.2.2 Stochastic Optimization

Stochastic optimization has received considerable attention in the literature over the last three decades. Much work has been devoted to the theory and applications of simulated annealing (SA in what follows) as an optimization technique. A comprehensive overview over this field is given in [HP95, PR02]. As our optimization approach is inspired by an SA algorithm (the one proposed in [BS91]) we shall outline the structure of a typical SA method and briefly discuss advantages and disadvantages. A general SA algorithm can be described as follows [PR02]:

1. Let $\mathbf{x}_0$ be a given starting point in the search space, $\mathbf{Z}_0 := \{\mathbf{x}_0\}$ and $k := 0$.
2. Sample a point $\mathbf{y}_{k+1}$ from a distribution $D(\cdot, \mathbf{Z}_k)$.
3. Sample a uniform random number $p \in [0, 1]$ and set

$$\mathbf{x}_{k+1} := \begin{cases} \mathbf{y}_{k+1} & \text{if } p \leq A(\mathbf{x}_k, \mathbf{y}_{k+1}, t_k) \in [0, 1] \\ \mathbf{x}_k & \text{otherwise,} \end{cases}$$

   where A is called the acceptance function and t_k is a parameter called the temperature at iteration k.
4. Set $\mathbf{Z}_{k+1} := \mathbf{Z}_k \cup \{\mathbf{y}_{k+1}\}$. $\mathbf{Z}_k$ contains all the information collected up to iteration k.
5. Set $t_{k+1} := U(\mathbf{Z}_{k+1}) \geq 0$, where U is called the cooling schedule.
6. If a stopping criterion fails set $k := k + 1$ and go to step 2, otherwise break.

A major property of SA algorithms is their "willingness" to explore points in search space, at which the objective function takes values greater then the current minimum [BW98] (see step 3). This is what makes SA algorithms able to escape from local minima and thus makes them suitable for the task of global minimization. A known drawback of SA algorithms is the fact that they waste a lot of iterations in generating candidate points, evaluating the objective function at these points and finally reject them [PR02] (see step 3). In order to reduce the number of rejections, Bilbro and Snyder [BS91] select candidate points from "promising" regions of the search space, i.e. from regions in which the objective function is likely to have low values. They achieve this by adapting the distribution $D(\cdot, \mathbf{Z}_k)$ at every iteration at which a candidate point $\mathbf{y}_{k+1}$ is accepted (see steps 2 and 3). If however $\mathbf{y}_{k+1}$ is not accepted, then $D(\cdot, \mathbf{Z}_k)$ remains unchanged. This is—in the case of candidate rejection—a considerable waste of computational time, since the information gained by the (expensive) evaluation of the objective function is not used at all. [1]

6.3 Registration as a Minimization Problem

Consider we are given a model point set $\mathbf{M} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subset \mathbb{R}^3$ and a scene point set $\mathbf{S} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\} \subset \mathbb{R}^3$. Suppose we have a continuous function $S : \mathbb{R}^3 \to \mathbb{R}$, called the scene scalar field, which takes small values when evaluated at or near the scene points $\mathbf{y}_i,\ i \in \{1, \ldots, n\}$ and increases with increasing distance between the evaluation point and the nearest scene point. The scene scalar field S will be precisely defined in section 6.3.1. Consider for now it is given and it has the above mentioned property. Our aim is to find a rigid transform $T : \mathbb{R}^3 \to \mathbb{R}^3$ of the form $T(\mathbf{x}) = R \cdot \mathbf{x} + \mathbf{t}$ for a rotation matrix $R \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$ such that the functional

$$\mathcal{F}(T) = \sum_{j=1}^{m} S(T(\mathbf{x}_j)), \quad \mathbf{x}_j \in \mathbf{M}. \tag{6.1}$$

gets minimized. The definition of $\mathcal{F}$ in equation (6.1) is based on the following quite natural idea, which is common for the most registration algorithms including ICP like methods: We seek for a rigid transform, which brings the model points as close as possible to the scene points. The major difference between the proposed method and ICP like algorithms is the introduction of the scalar field $S : \mathbb{R}^3 \to \mathbb{R}$ which allows the definition of the functional $\mathcal{F}$ not based on point correspondences between model and scene.

6.3.1 Definition of the Scene Scalar Field

It is a widely unsolved problem to establish pointwise model $\leftrightarrow$ scene correspondence which relates points that semantically belong to each other. The reason why we introduce the above mentioned scene scalar field, is that it can be used to omit this correspondence establishment step (common to ICP like methods).

Given the scene point set $\mathbf{S} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ we want to have a function $S : \mathbb{R}^3 \to \mathbb{R}$ which takes its minimal value at the scene points, i.e.

$$S(\mathbf{y}_i) = s_{\min} \in \mathbb{R}, \quad \forall i \in \{1, \ldots, n\} \tag{6.2}$$

and takes greater values for all other points in $\mathbb{R}^3$, i.e.

$$S(\mathbf{y}) > s_{\min}, \quad \forall \mathbf{y} \in \mathbb{R}^3 \setminus \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}. \tag{6.3}$$

If we set

$$S(\mathbf{y}) := \min_{\mathbf{y}_i \in \mathbf{S}} \|\mathbf{y} - \mathbf{y}_i\| \tag{6.4}$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^n$ we get an unsigned distance field for the scene point set $\mathbf{S}$, which is implicit used by ICP. It is obvious, that this choice for S fulfills both criteria (6.2) and (6.3).

Mitra *et al*let@tokeneonedot[MGPG04] and Pottmann *et al*let@tokeneonedot[PHYH06] consider in their work more sophisticated scalar fields. They assume, that the scene point set $\mathbf{S}$ consists of points sampled

[1] The algorithm we introduce in section 6.4 avoids this, by adapting a tree like data structure to the objective function at every iteration, regardless of the fact if an improvement of the current minimum value has been achieved or not.

from some underlying surface Φ. The scalar field S at a point $\mathbf{y} \in \mathbb{R}^3$ is set to be the squared distance from $\mathbf{y}$ to the surface Φ. In this context, S is called the squared distance function to the surface Φ.

Given the unit normal vector $\vec{\mathbf{n}}$ along with the principal curvature directions $\vec{\mathbf{e}}_1$ and $\vec{\mathbf{e}}_2$ for each point on the surface Φ a local coordinate system, called the principal frame, can be formed. Let ρ_i be the principal radius of curvature in the direction $\vec{\mathbf{e}}_i$ and let d denote the signed distance from a point $\mathbf{y} \in \mathbb{R}^3$ to the closest point $\mathbf{y}' \in \Phi$ on the surface. The sign of d is positive if $\mathbf{y}$ and the centers of the osculating circles at $\mathbf{y}'$ with radii ρ_1 respectively ρ_2 lie on the same side of the surface around $\mathbf{y}'$. Pottmann and Hofer [PH03] show that for a point $\mathbf{y} \in \mathbb{R}^3$, with coordinates y_1, y_2, y_3 expressed in the principal frame at $\mathbf{y}' \in \Phi$ the second order Taylor approximant of the squared distance function is given by

$$F_d(\mathbf{y}) = F_d(y_1, y_2, y_3) = \frac{d}{d-\rho_1} y_1^2 + \frac{d}{d-\rho_2} y_2^2 + y_3^2. \tag{6.5}$$

In order to have an overall non-negative scalar field the following modified version of the above equation is used

$$F^+(\mathbf{y}) = \hat{\delta}_1 y_1^2 + \hat{\delta}_2 y_2^2 + y_3^2 \tag{6.6}$$

for

$$\hat{\delta}_i = \begin{cases} d/(d-\rho_i) & \text{if } d < 0, \\ 0 & \text{otherwise} \end{cases}. \tag{6.7}$$

A transformation of F^+ to the global coordinate system yields the desired scalar field S

$$S(\mathbf{y}) := \hat{\delta}_1(\vec{\mathbf{e}}_1 \cdot (\mathbf{y} - \mathbf{y}'))^2 + \hat{\delta}_2(\vec{\mathbf{e}}_2 \cdot (\mathbf{y} - \mathbf{y}'))^2 + (\vec{\mathbf{n}} \cdot (\mathbf{y} - \mathbf{y}'))^2. \tag{6.8}$$

For point sets rather then surfaces Mitra *et al*let@tokeneonedot[MGPG04] approximate the foorpoint $\mathbf{y}'$ by the closest point from $\mathbf{S}$ to $\mathbf{y} \in \mathbb{R}^3$. We refer to [MGPG04] for more details on the estimation of $\vec{\mathbf{e}}_1$, $\vec{\mathbf{e}}_2$, $\vec{\mathbf{n}}$, $\hat{\delta}_1$ and $\hat{\delta}_2$ aswell on efficient techniques for approximating the squared distance function for whole point sets.

We refer to [MGPG04] for details on computing the squared distance function and its approximation for point sets.

The version of S given in equation (6.4) and the one used by Mitra *et al*let@tokeneonedot[MGPG04] are both essentially distance fields. This means that $\lim_{\|\mathbf{y}\| \to \infty} S(\mathbf{y}) = \infty$, i.e. $S(\mathbf{y})$ approaches to infinity as the point $\mathbf{y}$ gets infinitely far from the point set. This has the practical consequence, that a registration technique based on an unbounded scalar field S will be sensitive to outliers or will not perform well on partially visible objects in the scene, as model points lying far away from the scene point set, will have great impact on the functional value in equation (6.1) and thus will prevent the minimization algorithm from converging towards the global optimal alignment.

To avoid this problem we propose to use a bounded scalar field satisfying equations (6.2), (6.3) and having the additional property

$$\lim_{\|\mathbf{y}\| \to \infty} S(\mathbf{y}) = 0. \tag{6.9}$$

We set

$$S(\mathbf{y}) := -\varphi\left(\min_{\mathbf{y}_i \in \mathbf{S}} \|\mathbf{y} - \mathbf{y}_i\|\right), \tag{6.10}$$

where $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$, for $\mathbb{R}^+ := \{x \in \mathbb{R} : x \geq 0\}$ is a strictly monotonically decreasing continuous function with

$$\max_{x \in \mathbb{R}^+} \varphi(x) = \varphi(0), \tag{6.11}$$

$$\lim_{x \to \infty} \varphi(x) = 0. \tag{6.12}$$

In our implementation we use a rational function of the form $1/(1 + \alpha x^2)$ (see figure 6.1) because it is computationally efficient to evaluate and can be easily controlled by a single parameter α which gives us the possibility to adapt the scalar field S to the scene point set.

Setting $\varphi(x) := 1/(1 + \alpha x^2)$ results in the following radially symmetric scalar field

$$S_\alpha(\mathbf{y}) = -\frac{1}{1 + \alpha \left(\min_{\mathbf{y}_i \in \mathbf{S}} \|\mathbf{y} - \mathbf{y}_i\|\right)^2}, \quad \alpha > 0. \tag{6.13}$$

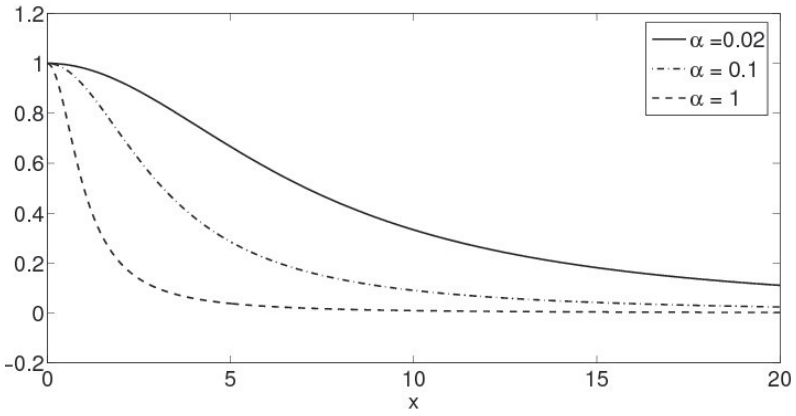


Figure 6.1: Strictly monotonically decreasing continuous functions of the form $\varphi(x) = 1/(1 + \alpha x^2)$ for three different α values.

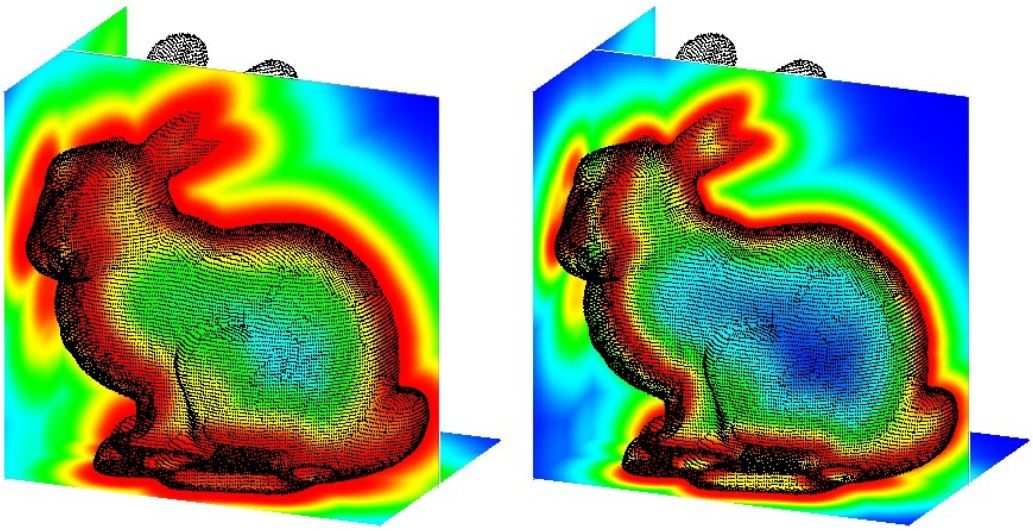


Figure 6.2: The scene scalar field S_α as defined in (6.13) computed for the Stanford bunny using $\alpha = 2262.932083$ (left) and $\alpha = 8728.45232$ (right). S_α is evaluated at a number of points lying on the three planes and the resulting scalars are visualized using a standard color mapping technique.

It is easy to see, that (6.2), (6.3) and (6.9) hold for $S_\alpha(\mathbf{y})$:

$$S_\alpha(\mathbf{y}_i) = -1, \quad \forall i \in \{1, \ldots, n\} \tag{6.14}$$

$$S_\alpha(\mathbf{y}) > -1, \quad \forall \mathbf{y} \in \mathbb{R}^3 \setminus \mathbf{S} \tag{6.15}$$

$$\lim_{\|\mathbf{y}\| \to \infty} S_\alpha(\mathbf{y}) = 0. \tag{6.16}$$

The scalar field S_α is easily adapted to the scene point set by relating α to the mean of the distances from every point $\mathbf{y}_i \in \mathbf{S}$ to its next point $\mathbf{y}_j \in \mathbf{S}, j \neq i$. Figure 6.2 shows how different values of α impact S_α.

6.3.2 Parametrization of the Euclidean Group

A rigid transform $T : \mathbb{R}^3 \to \mathbb{R}^3$ consists of a rotation and a translation (both in $\mathbb{R}^3$). More precisely, we have:

$$T(\mathbf{x}) = R \cdot \mathbf{x} + \mathbf{t}, \quad R \in \mathbb{R}^{3\times 3},\ \mathbf{t} \in \mathbb{R}^3. \tag{6.17}$$

The vector $\mathbf{t}$ is the translational and the matrix R the rotational part of T. In order to be a rotation R has to fulfill two conditions:

$$R^T R = I, \text{ and } \det(R) = 1, \tag{6.18}$$

where R^T denotes the transpose of R, I is the identity matrix and $\det(R)$ is the determinant of R. We use Euler angles θ, ϕ and ψ to set up the matrix R as a rotation by θ about the x-axis, followed by a rotation by ϕ about the y-axis and a rotation by ψ about the z-axis. We will denote the resulting matrix by $R_{\theta,\phi,\psi}$. Its explicit form can be found in [Kan90].

The set of all mappings of the form (6.17) meeting the requirements in (6.18) is called the Euclidean group of three dimensional space, often denoted by $SE(3)$. A parametrization of $SE(3)$ is given by

$$T_{\mathbf{x}} : \mathbb{R}^6 \to SE(3), \tag{6.19}$$

$$T_{\mathbf{x}}(\theta, \phi, \psi, t_x, t_y, t_z) := R_{\theta,\phi,\psi} \cdot \mathbf{x} + (t_x, t_y, t_z). \tag{6.20}$$

For the angles we have $\theta \in [0, \pi],\ \phi \in [0, 2\pi),\ \psi \in [0, 2\pi)$ and (t_x, t_y, t_z) is an arbitrary vector in $\mathbb{R}^3$. In the next section of the chapter we will use the parametrization of $SE(3)$ to convert the minimization of the functional defined in (6.1) into the minimization of a cost function.

Note that $\mathbf{x}$ in equation (6.20) is kept constant. We emphasize this by augmenting T with a subscript $\mathbf{x}$. This is opposite to (6.17) where $\mathbf{x}$ is variable and R and $\mathbf{t}$ are constant.

6.3.3 Cost Function Definition

At the beginning of section 6.3 we have formulated the rigid point set registration problem as a functional minimization problem: minimize $\mathcal{F}$ (see equation (6.1)) over the Euclidean group $SE(3)$. Using the parametrization of $SE(3)$ defined in section 6.3.2 we convert $\mathcal{F}$ to a real-valued scalar field $F : \mathbb{R}^6 \to \mathbb{R}$ of the form

$$F(\theta, \phi, \psi, x, y, z) = \sum_{j=1}^{m} S_\alpha(T_{\mathbf{x}_j}(\theta, \phi, \psi, x, y, z)). \tag{6.21}$$

Substituting $T_{\mathbf{x}_j}$ for (6.20) yields

$$F(\theta, \phi, \psi, x, y, z) = \sum_{j=1}^{m} S_\alpha(R_{\theta,\phi,\psi} \cdot \mathbf{x}_j + (x, y, z)), \tag{6.22}$$

for the model points $\mathbf{x}_1, \ldots, \mathbf{x}_m$ and for S_α defined by equation (6.13). A global minimizer $\mathbf{p}^* \in \mathbb{R}^6$ of F defines a rigid transform that brings the model points as close as possible to the scene points $\mathbf{y}_1, \ldots, \mathbf{y}_n$. Note that the scene points are not explicitly given in (6.22). Instead they are used for the definition of the scene scalar field S_α as described in section 6.3.1.

What makes the proposed cost function robust to outliers is the fact, that outlier object points $\mathbf{x}_{k_1}, \ldots, \mathbf{x}_{k_p}$ will have a marginal contribution to the sum in equation (6.22). This is due to property (6.16) of the scene scalar field: it returns values close to zero as evaluated at points far away from the scene point set (i.e. at outlier points).

The cost function given in (6.22) is highly nonlinear and in general nonconvex. This is due to the fact that the rotation matrix $R_{\theta,\phi,\psi}$ is defined by sine and cosine functions and the scene scalar field S is nonlinear and in general noncovex. This results in a great number of local minima of F over the search space. Using a local optimization procedure—common for the most point registration methods in the literature—will lead in the most cases to a local minimizer of F and thus will not give the best alignment between model and scene.

We employ a novel stochastic approach for global optimization, described in the next section of this chapter. We seek for the global minimum of F over the search space

$$\mathbf{X} := [-\pi/2, \pi/2] \times [-\pi, \pi] \times [-\pi, \pi] \times BB(\mathbf{S}), \tag{6.23}$$

where $BB(\mathbf{S})$ denotes the axis-aligned minimum bounding box for the point set $\mathbf{S}$. The first three intervals in (6.23) build the search space for the rotational part and the bounding box for the translational part of the rigid transform.

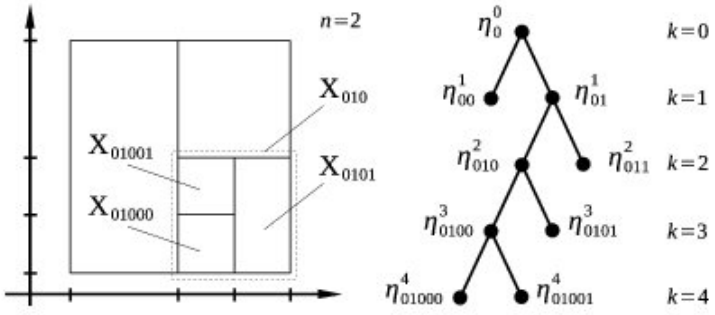


Figure 6.3: Example of a four level deep tree structure we use in our minimization algorithm.

6.4 Adaptive Branch and Bound Search for Global Optimization

In this section, we present a novel stochastic approach for global minimization inspired by the work of Bilbro and Snyder [BS91]. Our algorithm shares two properties with the one presented in [BS91]: (i) we use the same data structure (a k-d like tree) to represent the search space and (ii) we adapt the tree during the search process to the objective function.

In contrast to [BS91], where the tree is updated only when a new candidate point is accepted (see section 6.2.2), we update it at every iteration, so we use *all* the information gained by the evaluation of the objective function.

6.4.1 Problem Definition

We call a set $\mathbf{X} \subset \mathbb{R}^n$ an n-dimensional (or n-d) interval in $\mathbb{R}^n$ iff

$$\mathbf{X} = [a_0, b_0] \times \ldots \times [a_{n-1}, b_{n-1}], \quad a_i, b_i \in \mathbb{R}, \tag{6.24}$$

for the intervals $[a_i, b_i]$. Given an n-d interval $\mathbf{X} \subset \mathbb{R}^6$ and a bounded continuous function $f : \mathbf{X} \to \mathbb{R}$ our aim is to find a global minimizer $\mathbf{x}^* \in \mathbf{X}$ of f, i.e. we seek for an $\mathbf{x}^* \in \mathbf{X}$ satisfying

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbf{X}. \tag{6.25}$$

6.4.2 Algorithm Specification

We use a k-d tree like data structure to represent the search space $\mathbf{X}$. The root η_0^0 is at the 0-th level of the tree and represents the whole set $\mathbf{X}_0 := \mathbf{X}$. The root has two children η_{00}^1 and η_{01}^1 at the first level of the tree. They represent the n-d intervals $\mathbf{X}_{00}$ respectively $\mathbf{X}_{01}$ resulting from bisecting the 0-th interval (i.e. $[a_0, b_0]$) of $\mathbf{X}_0$ and assigning the first half to $\mathbf{X}_{01}$ and the second half to $\mathbf{X}_{11}$. In general a node η_s^k (where $k > 0$ and s is a binary string of length $k+1$) is at the k-th level of the tree and has two children η_{s0}^{k+1} and η_{s1}^{k+1} at the next level. The children nodes represent the same n-d interval as η_s^k (i.e. $\mathbf{X}_s$) except for that the (k mod n)-th interval of $\mathbf{X}_s$ is bisected and the first respectively second half is assigned to η_{s0}^{k+1} respectively η_{s1}^{k+1} (see figure 6.3).

6.4.2.1 Continuous Minimization as a Tree Search

We build the tree in an iterative fashion beginning by the root. During the search process more resolution is added to promising regions in the search space, i.e. the tree is build with higher resolution in the vicinity of points in $\mathbf{X}$ for which the objective function f has low value.

For every tree node η_s^k the following items are stored: (i) an n-d interval $\mathbf{X}_s \subset \mathbf{X}$ and (ii) a pair $(\mathbf{x}_s, f(\mathbf{x}_s))$ consisting of a point $\mathbf{x}_s$ uniformly distributed over $\mathbf{X}_s$ and the corresponding function value $f(\mathbf{x}_s)$. To

see how the algorithm works on the tree, consider for now that the tree has been partially build and that for every node the above mentioned items are available.

Selecting a Leaf At every iteration the search begins at the root and proceeds down the tree until a leaf (i.e. node without children) is found. In order to reach a leaf we have to choose a concrete path from the root down to this leaf, i.e. at each node we have to decide, whether to take its left or right child as the next station. This decision is made probabilistically. For every node two numbers $p_0, p_1 \in (0,1)$ are computed in a way that $p_0 + p_1 = 1$. Arriving at a node we choose to descent via either its left or right child with probability p_0 respectively p_1. The idea is to compute the probabilities in such a way, that the "better" child, i.e. the one with the lower function value, has greater chance to be selected. In section 6.4.2.2, we specify how to compute p_0 and p_1. We make these left/right decisions until we encounter a leaf.

Expanding the Tree After reaching a leaf η_s^k, the n-d interval associated with it gets bisected in the way described at the beginning of section 6.4.2, which results in the creation of two n-d intervals $\mathbf{X}_{s0}$ and $\mathbf{X}_{s1}$ associated with two new children η_{s0}^{k+1} and η_{s1}^{k+1}. In this way we add more resolution in this region of the search space. Next we evaluate the new children, i.e. we assign to the left and right one a pair $(\mathbf{x}_{s0}, f(\mathbf{x}_{s0}))$ respectively $(\mathbf{x}_{s1}, f(\mathbf{x}_{s1}))$.

Note that the parent node η_s^k already stores a pair $(\mathbf{x}_s, f(\mathbf{x}_s))$. As we have $\mathbf{X}_s = \mathbf{X}_{s0} \cup \mathbf{X}_{s1}$ it follows that $\mathbf{x}_s$ is contained either in $\mathbf{X}_{s0}$ or $\mathbf{X}_{s1}$ or in both (as $\mathbf{X}_{s0} \cap \mathbf{X}_{s1} \neq \emptyset$ is an n-1 dimensional interval). Thus we set

$$(\mathbf{x}_{s0}, f(\mathbf{x}_{s0})) := (\mathbf{x}_s, f(\mathbf{x}_s)) \text{ for } \mathbf{x}_s \in \mathbf{X}_s \setminus \mathbf{X}_{s1}, \tag{6.26}$$

$$(\mathbf{x}_{s1}, f(\mathbf{x}_{s1})) := (\mathbf{x}_s, f(\mathbf{x}_s)) \text{ for } \mathbf{x}_s \in \mathbf{X}_{s1}. \tag{6.27}$$

To compute the other pair we generate a uniformly distributed point over the appropriate n-d interval ($\mathbf{X}_{s0}$ or $\mathbf{X}_{s1}$) and evaluate the function at this point.

Updating the Tree During the search we want to compute the random paths from the root down to a certain leaf such that promising regions—leafs with low function values—are visited more often then non-promising ones. Thus after evaluating a new created leaf, we propagate its (possibly very low) function value as close as possible to the root. This is done by the following updating process. Suppose that the parent point $\mathbf{x}_s$ is contained in the set $\mathbf{X}_{s0}$ belonging to the new left child η_{s0}^{k+1}. Thus we randomly generate $\mathbf{x}_{s1} \in \mathbf{X}_{s1}$ (uniformly distributed), compute $f(\mathbf{x}_{s1})$, and assign the pair $(\mathbf{x}_{s1}, f(\mathbf{x}_{s1}))$ to the new right child. Ascend from η_{s1}^{k+1} to the root, comparing at every node η_u^j the function value $f(\mathbf{x}_{s1})$ with the function value of that node, i.e. with $f(\mathbf{x}_u)$. If $f(\mathbf{x}_{s1}) < f(\mathbf{x}_u)$ we update the current node by setting $(\mathbf{x}_u, f(\mathbf{x}_u)) := (\mathbf{x}_{s1}, f(\mathbf{x}_{s1}))$. If $f(\mathbf{x}_{s1}) \geq f(\mathbf{x}_u)$ then no improvement for η_u^j is possible and we break up the updating process.

Note that if $f(\mathbf{x}_{s1})$ is the lowest function value found so far, it will be propagated up to the root, otherwise it will be propagated only up to a certain level $m \in \{1, \ldots, k+1\}$. Thus every node contains the minimum function value (and the point at which f takes this value) found in the n-d interval associated with this node. As the root represents the whole search space, it contains the point we are interested in, namely the point at which f takes the lowest value found up to the current iteration.

Initializing the Tree The tree is initialized by storing the following information in the root: (i) the bounds of the whole search space $\mathbf{X}$ and (ii) a pair $(\mathbf{x}_0, f(\mathbf{x}_0))$, consisting of a point $\mathbf{x}_0$ uniformly sampled over $\mathbf{X}$ and the corresponding function value $f(\mathbf{x}_0)$.

6.4.2.2 Probability assignment

As already pointed out in the last section, the two numbers p_0 and p_1 represent the probability for selecting the left respectively right child of a node η_s^k. We compute p_0 and p_1 for each node based on the function values associated with its children η_{s0}^{k+1} and η_{s1}^{k+1}. Let f_{s0} and f_{s1} be the function value associated with η_{s0}^{k+1} respectively η_{s1}^{k+1}. The probabilities should fulfill the following criteria

$$p_0 > p_1 \quad \text{for} \quad f_{s0} < f_{s1}, \tag{6.28}$$

i.e. the "better" child has greater chance to be selected. Thus the regions represented by better children are explored in greater detail.

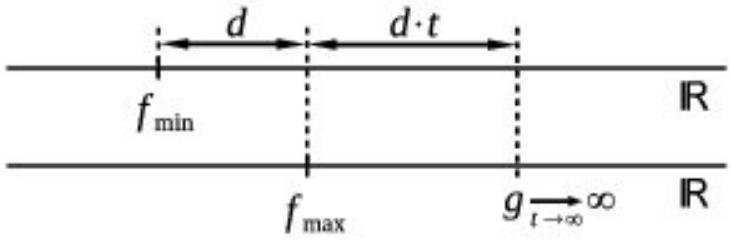


Figure 6.4: Assigning probabilities according to the values $g - f_{\min}$ and $g - f_{\max}$.

Let $f_{\min} := \min\{f_{s0}, f_{s1}\}$ and $f_{\max} := \max\{f_{s0}, f_{s1}\}$. We set the probability $p_{\min}$ (corresponding to $f_{\min}$, i.e. the probability assigned to the better child) to be proportional to $g - f_{\min}$ and the probability $p_{\max}$ (corresponding to $f_{\max}$) to be proportional to $g - f_{\max}$, where g is a real number greater then $f_{\max}$ (see figure 6.4). Expressing this through a parameter $t \geq 0$ yields

$$p_{\min} = \frac{d + dt}{d + 2dt} = \frac{t}{1 + 2t} + \frac{1}{1 + 2t}, \tag{6.29}$$

$$p_{\max} = \frac{dt}{d + 2dt} = \frac{t}{1 + 2t}, \tag{6.30}$$

where $d = f_{\max} - f_{\min}$. For $t \to \infty$ we have $p_{\min} = p_{\max} = \frac{1}{2}$ and our optimization algorithm becomes a pure random search. Setting $t := 0$ results in $p_{\min} = 1$ and $p_{\max} = 0$ and makes the algorithm choosing strictly the "better" child for every node, which leads to the exclusion of a great portion of the search space and in general prevents the algorithm from finding a global minimum. Obviously, t should be chosen from the interval $(0, \infty)$. For our algorithm the parameter t plays a similar role as the temperature parameter for a simulated annealing algorithm, so we will refer to t as temperature as well. Like in simulated annealing, the search begins on a high temperature level, so the algorithm samples the search space quite uniformly. The temperature is decreased gradually during the search process, so that the promising regions of the search space are explored in greater detail. More precisely, we update t according to the following cooling schedule:

$$t = t_{\max} \cdot \exp\left(-v \left\lfloor \frac{j}{m} \right\rfloor\right), \tag{6.31}$$

where $t_{\max} > 0$ is the temperature at the beginning of the search, $v > 0$ determines how fast the temperature decreases, $j \in \mathbb{N}$ is the current iteration number, $m \in \mathbb{N} \setminus \{0\}$ states that a temperature update takes place at every m-th iteration and $\lfloor x \rfloor$ denotes the greatest integer less or equal to $x \in \mathbb{R}$.

6.4.2.3 Stopping rule

We break the search, if for the last $N \in \mathbb{N}$ iterations the absolute difference between the last sample of the objective function and the sample before is less than a predefined $\epsilon > 0$. At the beginning of the search the objective function is quite uniformly sampled over the search space, so we have large differences between subsequent samples and the stopping criterion will not be met. As the temperature is decreased the computation of a path from the root down to a certain leaf becomes more and more "deterministic" (although it will never be strictly deterministic, as we always have $t > 0$). Thus the search should stop as it gets too restricted to a certain path and it makes no significant progress, i.e. as the absolute differences between the last N samples are less then ϵ.

6.5 Experimental Results

In this section we present the experimental results we have obtained after testing our method on two point sets at different noise levels.

6.5.1 Implementation Issues

In order to speed up the testing process we use a discrete version S_α^d of the scene scalar field defined in equation (6.13). We divide the bounding box of the scene point set in a number of pairwise disjoint axis–aligned boxes $\mathbf{B}_j$ such that $\cup_{j=1}^{p} \mathbf{B}_j = BB(\mathbf{S})$. The function value of S_α at the center of every box $\mathbf{B}_j$ is associated to the portion of space occupied by $\mathbf{B}_j$. To evaluate S_α^d at a point $\mathbf{y} \in \mathbb{R}^3$ one has to perform three divisions to determine the box $\mathbf{y}$ is lying in and to return the value associated with this box. We set $S_\alpha^d(\mathbf{x}) := 0$ for $\mathbf{y} \notin BB(\mathbf{S})$.

In order to locate a global minimum of the cost function more precisely we run the optimization procedure described in section 6.4 three times: the first time over the whole search space (see (6.23)) and the next two times over the six-dimensional interval centered at the minimum found in the run before and having side lengths four times shorter then the last search interval.

Case	Method#1	Method#2	Method#3
1	50	837	970
2	47	877	230
3	31	25	415
4	35	144	2356
5	45	300	556

Table 6.1: The parameter values used for testing.

6.5.2 Registration results

We ran hundred registration trials for every pair of input point sets: (i) noiseless bunny point sets (see figure 6.5), (ii) noise–corrupted bunny point sets (see figure 6.5), and (iii) noiseless bottle point sets (see figure 6.6). The results can be seen in the figures 6.7 – 6.12. The abscissas show the deviation from the global optimal registration. The translation error in the x-, y- and z-axis is given in percent from the length of the bounding box of the scene point set in x-, y- and z-direction respectively.

6.6 Conclusions and Future Work

We introduced a new technique for pairwise correspondence free rigid registration of point sets. Our method is based on a noise robust cost function and on a novel stochastic approach for global optimization. Characteristic to the proposed algorithm is that it does not rely on an initial estimation of the global optimal rigid transform between the point sets and that it is robust against outliers. We experimentally demonstrated that the proposed algorithm performs good on noise corrupted and incomplete point sets.

Since the binocular system in GRASP provides mostly incomplete views of the objects, a robust matching of partially reconstructed views was of a great importance for the project.

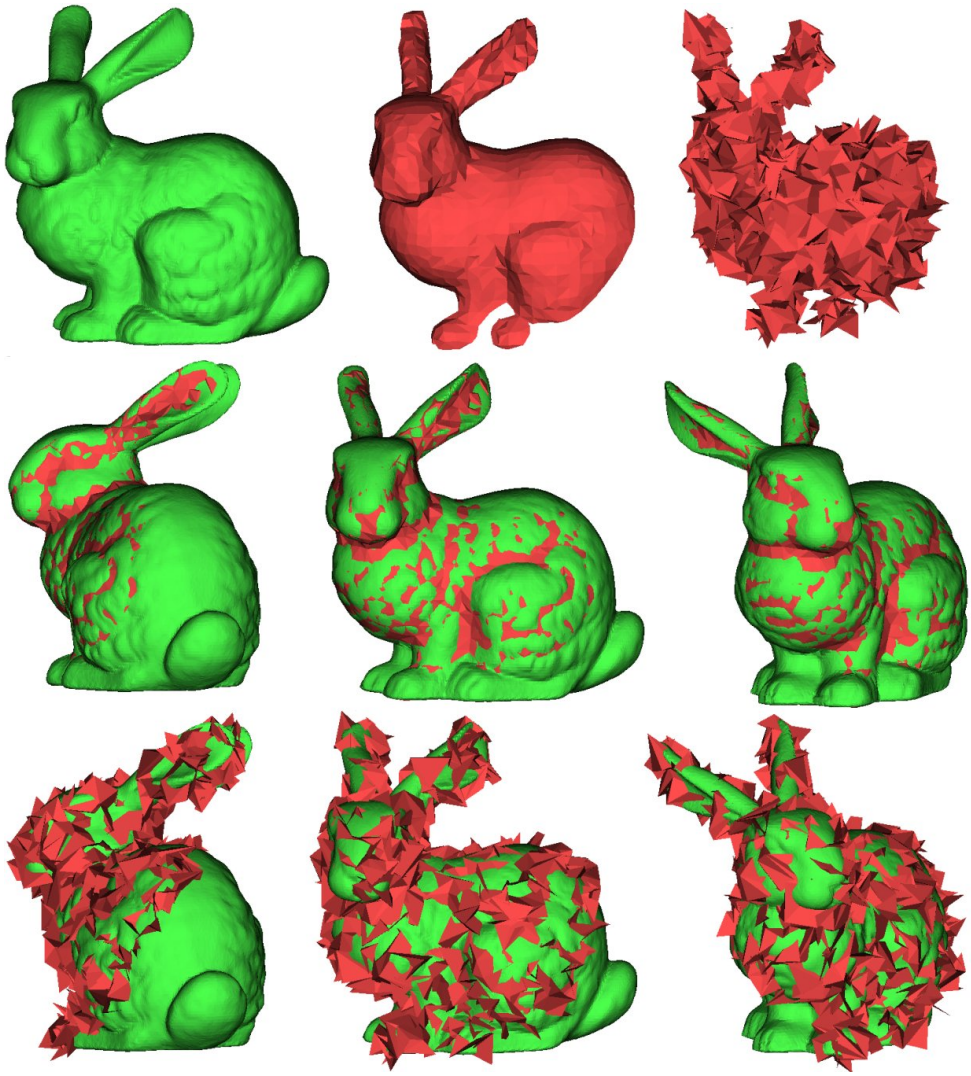


Figure 6.5: First row: input point sets (with and without noise). Second and third row: Registrations results.

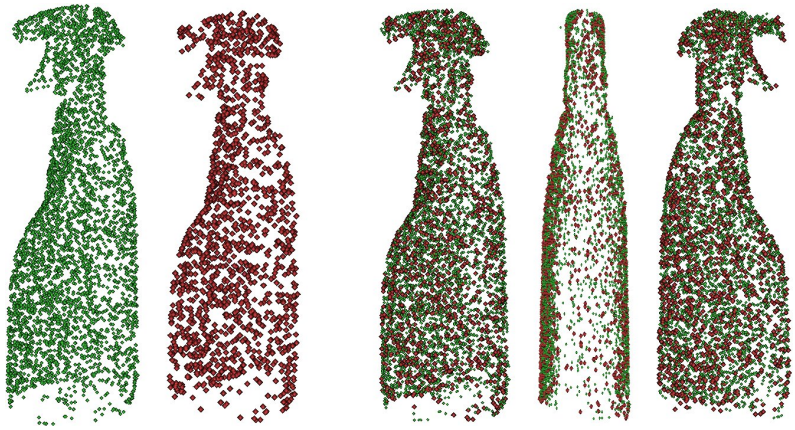


Figure 6.6: Left side: input point sets. Right side: registration results.

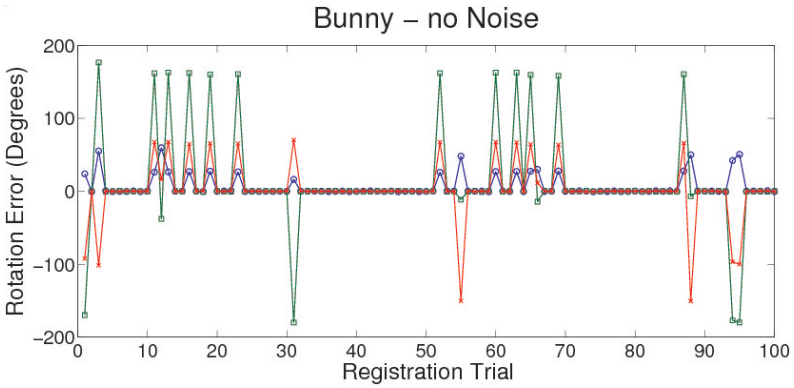


Figure 6.7: Rotation error for the noiseless bunny point sets.

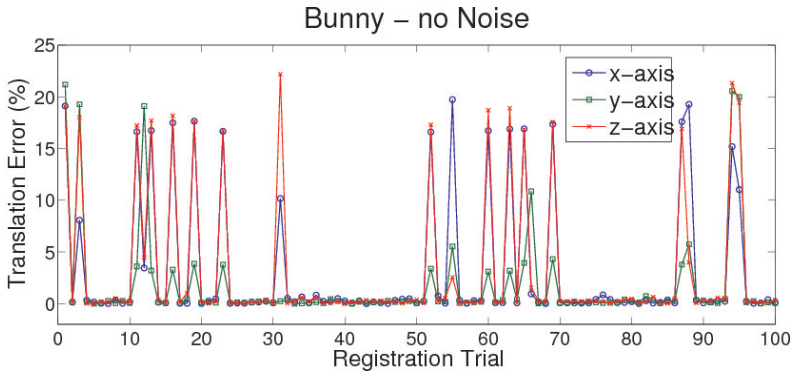


Figure 6.8: Translation error for the noiseless bunny point sets.

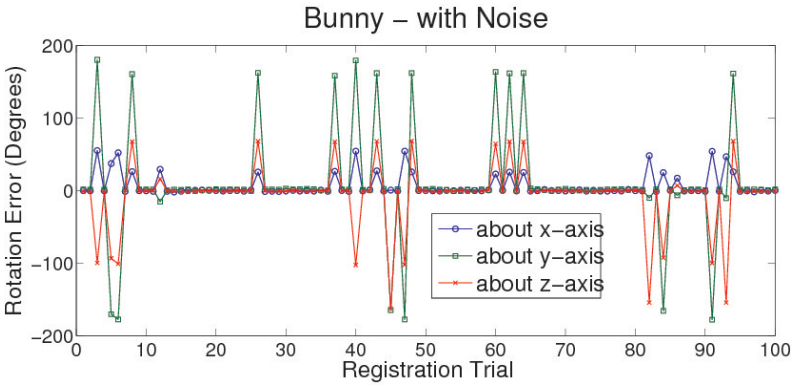


Figure 6.9: Rotation error for the noise–corrupted bunny point sets..

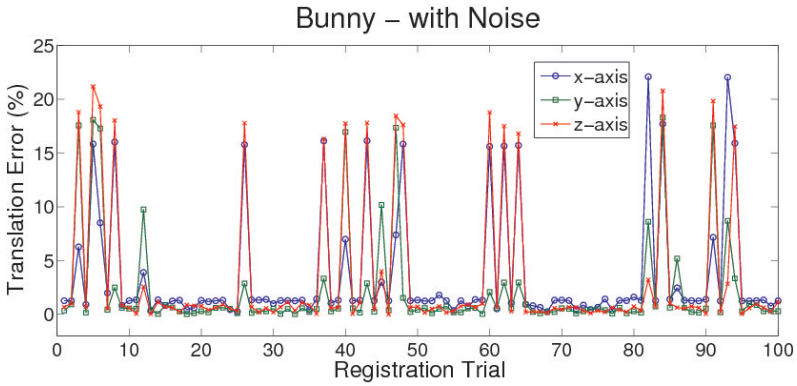


Figure 6.10: Translation error for the noise–corrupted bunny point sets.

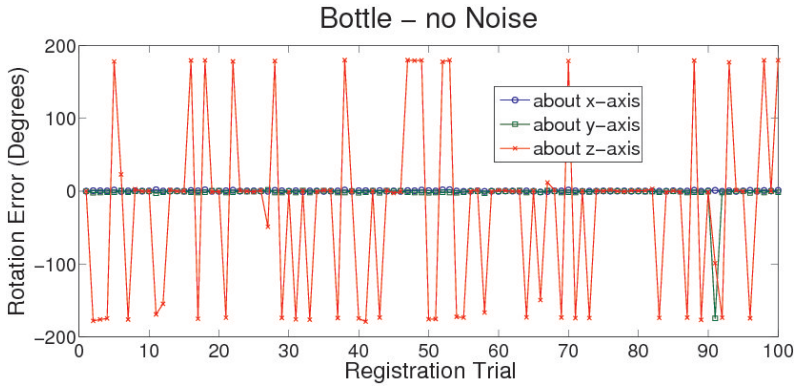


Figure 6.11: Rotation error for the bottle point sets.

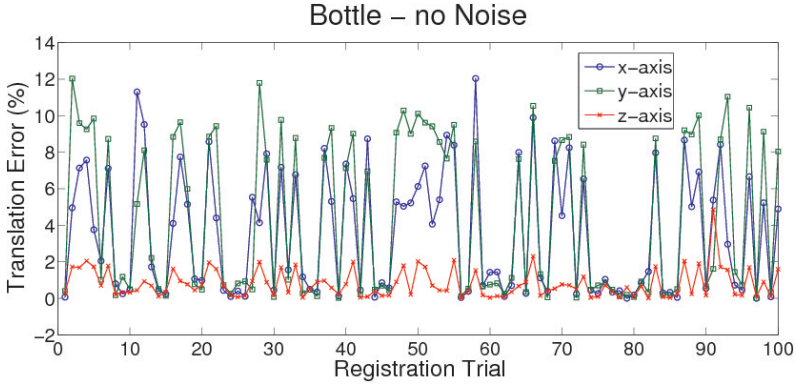


Figure 6.12: Translation error for the bottle point sets.

Chapter 7

Detection and Grasping based on Point Clouds using a Combination of Laser and Stereo Data

The chapter presents a solution for grasping novel objects with the help of a laser range scanner and a stereo camera. This includes a first step towards autonomous object detection and grasp motion planning. The system used to accomplish this task consists of a fixed working station equipped with a laser range scanner, a camera, a seven degrees of freedom manipulator and a hand prosthesis as gripper. This work user laser range data from single scans, because it is a currently available and a complete solution for performing early tests while other methods are still in development. We also show, that the very method can be applied to stereo vision (see Section 7.4), which is the modality we want to use in the final demonstration.

In this Chapter we present different methods for segmentation of a $2^1/_2$D point cloud into parts, assembly of parts into objects and calculation of grasping points based on laser range data, which works for cylindrical objects and arbitrary objects. This Section includes also an alignment to combine laser range and stereo data to get more object information. We successfully demonstrate this approach by grasping a variety of different shapes and present a step towards full automation.

7.1 System Approach

The main challenges to solve are the robust segmentation, the detection of edges and surfaces and their interpretation to predict grasping points. Our approach is based on scanning the objects by a rotating laser range scanner and execution of subsequent path planning and grasping motion. Hence the system consists of a pantiled-mounted red-light laser, a scanning camera and a seven degrees of freedom robot arm, which is equipped with a human like prosthesis hand (see Fig. **??**). By scanning the table scene we get a $2^1/_2$D point cloud.

The laser range scanner records a table scene with the help of a pan/tilt-unit. A high resolution sensor is needed in order to detect a reasonable number of points of the objects with the required accuracy. The laser range scanner consists of a red-light LASIRIS laser from StockerYale[1] with 635nm and a MAPP2500 CCD-camera from SICK-IVP[2] mounted on a pan/tilt-unit (PowerCube Wrist from AMTEC[3] robotics).

Additionally we use a stereo camera mounted on the pan/tilt unit. The stereo camera grabs two images at $-4°$ and $+4°$. Scharstein and Szeliski [SS] published a detailed description of the used dense stereo algorithm. To realize a dense stereo calibration to the laser range coordinate system as exact as possible the laser range scanner was used to scan the same chessboard as used for the camera calibration. At the obtained point cloud a marker was set as reference point to the camera coordinate system. Section 7.4

[1] http://www.stockeryale.com/index.htm
[2] http://www.sickivp.se/sickivp/de.html
[3] http://www.amtec-robotics.com/

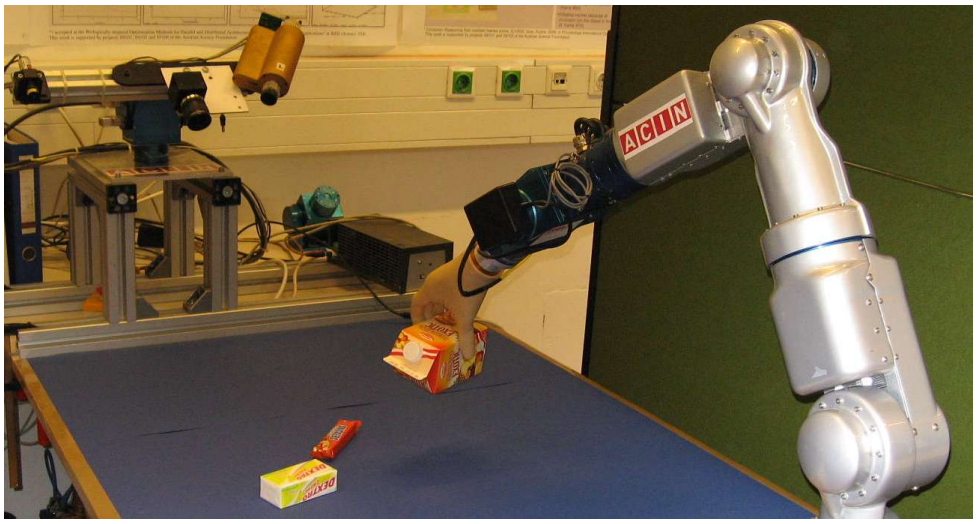


Figure 7.1: Overview of the robot system at TUW. In the background left the scanner and camera. To the right the Amtec arm with an Otto Bock hand prosthesis as gripper.

details the dense stereo calibration and Section 7.5 illustrates a combination of laser range and dense stereo point clouds.

We use the "Light Weight Arm 7 DOF" from AMTEC robotics and a hand prosthesis from Otto Bock as gripper. The seventh degree of freedom is required to enable complex object grasping and manipulation and allow for some flexibility to avoid obstacles. The prosthesis as end effector is selected due to the integrated force sensors. It has three active fingers, the thumb, the index finger and the middle finger. The last two fingers are just for cosmetic reasons. As a huge advantage the integrated tactile sensors are used to detect a potential sliding of objects, which initializes a readjustment of the fingers. A commercial path planning tool from AMROSE[4] calculates the trajectory to grasp the object. Before the robot arm delivers the object, the user can check the calculated trajectory in a simulation sequence (see Fig. 7.10). Then the robot arm executes the off-line programmed trajectory. The algorithm is implemented in C++ using the Visualization Tool Kit (VTK)[5].

7.2 Grasp-Point Detection Based on Top-Surfaces

The geometric entities we start with is a $2^1/_2$D recorded point cloud of a typical table scene as shown in Fig. 7.2. The main goal is to find a robust way to detect the grasp points of any kind of object in the recorded point cloud, see Fig. 7.3. Robustness includes the positive detection of grasp points despite of noise, outliers and shadows and missing data points, which can be caused by specular surfaces.

The problem of automatic $2^1/_2$D reconstruction to get grasping points consists of several challenging parts. Objects can be broken into disconnected parts, due to missing sensor data from shadows (see e.g. the self occlusion of the coffee cup) or poor surface reflectance. In order to calculate the correct grasping points, we need to identify complete objects and therefore reassemble parts belonging to the same object. Fig. 7.4 gives an overview of our segmentation and merging algorithm and Fig. 7.5 shows the outline of our multi-step solution procedure.

The main steps of the grasping algorithm are:

- Raw Data Preprocessing and Normal Vector Estimation: The raw data points are preprocessed with a low pass filter to reduce noise and the normal vectors are calculated with a Principal Component Analysis (PCA) [HJBJ+] based on a local neighbourhood of 5mm.

[4]http://www.amrose.dk/
[5]Freely available open source software, http://public.kitware.com/vtk.

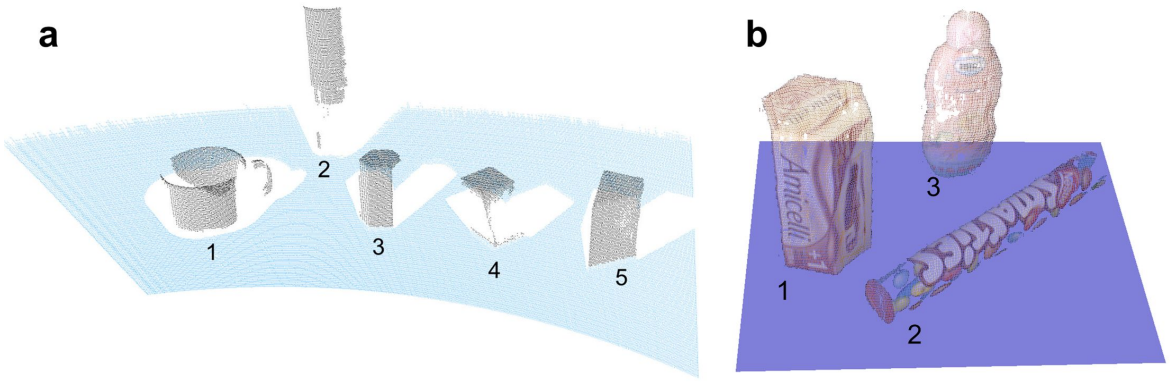


Figure 7.2: **a** Exposure of a raw laser point cloud. The two shadows from laser and camera are clearly visible (from left to right: 1. coffee cup, 2. spray-on glue, 3. five-corner object, 4. quadrangle rhombic object, 5. abnormal quadrangle object). **b** Exposure of a dense stereo point cloud. (from left to right: 1. five-corner object, 2. cylinder, 3. shampoo).

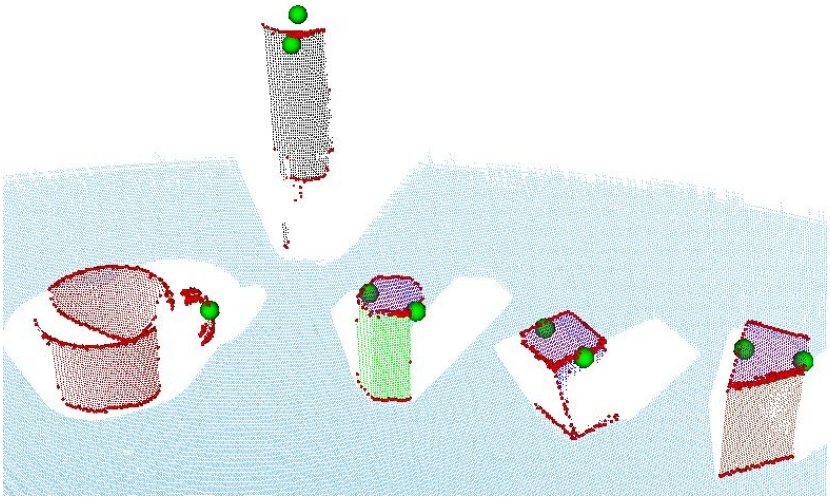


Figure 7.3: Final detection of the grasping points. The green points display the computed grasp points. Images are best viewed in color.

- Range Image Segmentation: This step identifies complete objects or parts of objects. In the latter case we need to find matching parts and reassemble complete objects before calculating grasp points.
- Pairwise Matching:
  - Cylindrical Parts: Finding high curvature points which indicate the top rim of objects and fit circle to these points. Matching of cylinders using the circle information.
  - Arbitrary Parts: Projection of parts into the ground plane.
- Analysing of Object Properties: Determine if the segmented object is open or closed and if there is a potential handle to define grasp type.
- Grasp Point Detection: Calculation of possible grasping points with the help of the gained features.
- Transmission of the Calculated Object Position to the Path Planning Tool: The calculated grasp point position in the actual environment model for collision avoidance has to be transmitted to the path planning tool.

7.2.1 Range Data Segmentation

The range data segmentation starts by detecting the surface of the table with a RANSAC [FB81] based plane fit. There exist different ways to realize a stable object segmentation. In our work we tested two

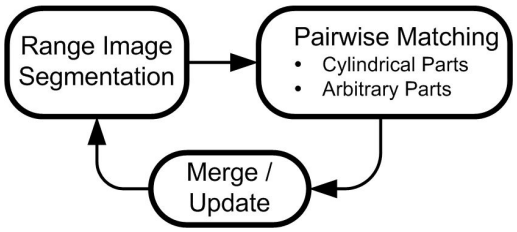


Figure 7.4: Overview of our segmentation and merging algorithm.

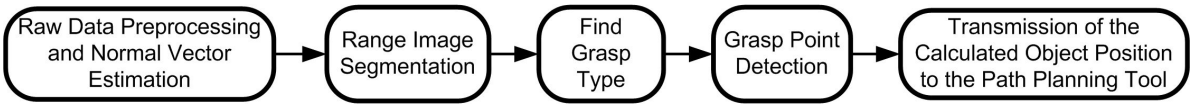


Figure 7.5: High level overview of our algorithm.

different methods. The first one is based on recursive flood-filling and the other one is based on mesh segmentation.

Recursive Flood-Filling: We define an object (part) as a set of points, with distances between neighbours below a threshold d_a. we build a kd-tree [Ben75] to find neighbours and use recursive flood-filling function [BB07] to identify connected point sets. d_a is the average distance between the neighbouring points. Distance d is defined as euclidean distance with an additional weighting factor w_g derived from the angle between normal vectors n of neighbouring points.

$$d = \sqrt{(x_i - x_m)^2 + (y_i - y_m)^2 + (z_i - z_m)^2} \tag{7.1}$$

$$\cos\alpha = \frac{\vec{n_i} \bullet \vec{n_m}}{\|\vec{n_i}\|\|\vec{n_m}\|} \tag{7.2}$$

$$w_g = 1 - |\cos(\alpha)| \tag{7.3}$$

$$d \cdot w_g < d_a \tag{7.4}$$

For our example in Fig. 7.2 we find seven parts, when there are five objects. The wrongly segmented parts are red encircled in Fig. 7.6. Parts with less than 30 points are regarded noise and are discarded. The occlusions are not a problem as long as the top surface of the occluded object is visible.

Mesh Segmentation: By this method the segmentation of the remaining points after the plane fit will be achieved with the help of a 3D mesh generation, based on the triangles calculated by a De-Launay triangulation [O'R98]. The necessary settings for the mesh generation are already determined with the distances d_{min} and d_{max} of the closest neighbouring point for all points of the complete point cloud, [AMNS98], as illustrated in Fig. 7.7.

7.2.1.1 Pairwise Matching

We developed two different matching methods. While the first method *"Cylindrical Parts"* is limited to cylindrical objects the second method *"Arbitrary Parts"* can be used generally.

Cylindrical Parts We want to find the top rim circle of cylindrical objects. We analyse the curvature of points to filter neighbouring points with an angle difference between $\pm 78°$ and $\pm 90°$. In Fig. 7.6 the high curvature points are coloured red.

For comparison, Jiang et al. [JC] published a method for 3D circle fitting. They reduce the number of local minima, but the error function is no more Euclidean. We use a RANSAC based circle fit with a

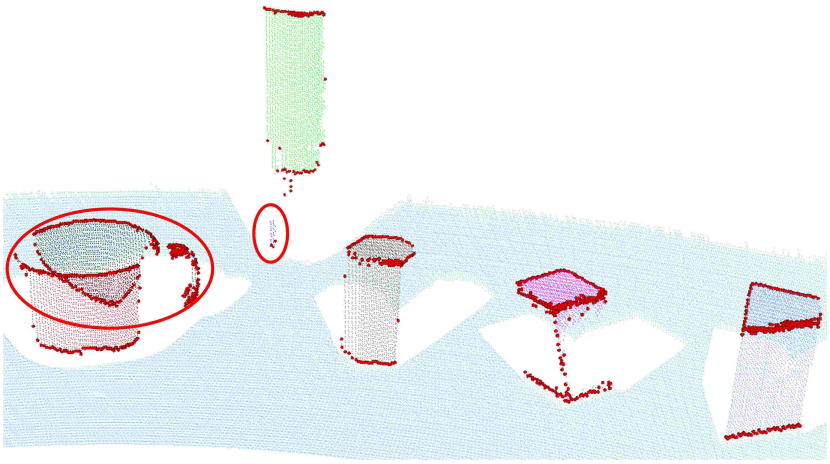


Figure 7.6: Results after the first segmentation step. Seven objects are detected, where in reality only five are. Object number 1 and 2 split into two parts. Object number 1 because of shadows and object number 2 because of pure reflectance. The wrongly segmented objects are red encircled. Images are best viewed in colour.

Figure 7.7: Table scene with five different objects (from left: 1. adhesive tape, 2. adhesive "Uhu", 3. plate, 4. cheese "blue", 5. spoon). The right image shows the generated mesh. The two shadows from laser and camera and the grasping points (green coloured) and the rim points (blue coloured) are displayed. Best viewed in colour.

range of tolerance of $2mm$. For an explicit description, the data points are defined as (p_i, p_i, p_i) and (p_m, p_m, p_m) is the circle's center with a radius r. The error must be smaller than a defined threshold:

$$|\|\vec{p} - \vec{p_m}\| - r| \leq 2 \tag{7.5}$$

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

$$n = |\{p|\|\vec{p} - \vec{p_m}\| - r| \leq 2\}| \tag{7.6}$$

If more than 50% of the rim points of both parts lie on the circle, the points of both parts are examined more closely with the cylinder fit (see Fig. 7.6, object number 1 (coffee cup)).

We use RANSAC again, where we use rim points for selecting models and all points of both parts for checking models. For that we calculate the distances of all points of both parts to the rotation axis, see Equ. 7.7 with a defined error distance of $2mm$ (see Fig. 7.8, the yellow lines represent the rotation axis). If more than 50% of all points of both parts agree with the cylinder model, both parts are merged to one object (see Fig. 7.8, especially object number 1 and 2).

$$d = (\vec{x} - \vec{m}) \times \vec{n} \tag{7.7}$$

$$|d - r| \leq 2mm \tag{7.8}$$

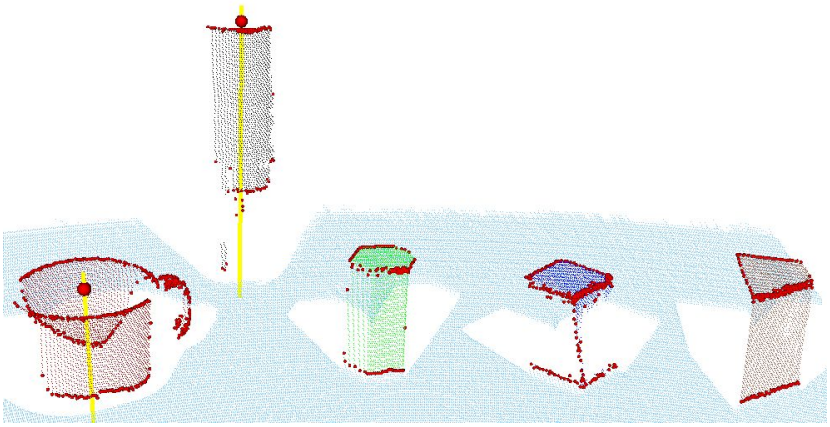


Figure 7.8: Result after the pairwise matching step. Five objects are detected, the yellow lines represent the rotation axis. Images are best viewed in colour.

Arbitrary Objects Another simple way of pairwise matching can be used for arbitrary objects (see Fig. 7.6 object number 2 (spray-on glue) the small red encircled segmented points). You can see that for the points projected to the ground plane, the projected point clouds of both parts should overlap. Se we check the pairwise distance of projected points of one part to another.

7.2.2 Grasp Features and Grasp Point Detection

After the segmentation step we find out if the object is open or closed. We fit a circle with radius r into the top rim points of each object and then place a sphere with radius $r\frac{2}{3}$ into the center of this circle. If there is no point of the object in this sphere we consider the object open. Now, the grasping points of all cylindrical objects can be calculated. If the diameter is smaller than the maximum opening angle of the hand prosthesis, the gripper can grasp the object at the outside surface (see Fig. 7.3, object number 2).

If the maximum opening angle of the hand prosthesis is too small and we found it is an open object than we try to a find a possible grasping point at the outside edge.

The algorithm finds that object number 3, 4 and 5 are closed and they have no cylindrical form. The algorithm finds the top surface of these objects, see Fig. 7.3 (magenta coloured planes) and Fig. 7.9. From these planes (magenta coloured in Fig. 7.9) we calculate the convex hull V (see Equ. 7.9) of all points n of the plane.

$$V = ConvexHull\left(\bigcup_{i=0}^{n-1} p_i\right) \tag{7.9}$$

The corner points (see Fig. 7.9, cyan coloured) are calculated with the average angle γ between the V hull points (magenta coloured), where r are the direction vectors between the points.

$$\alpha_i = \arccos\left(\frac{\vec{r}_{i-1} \bullet \vec{r}_{i+1}}{\|\vec{r}_{i-1}\|\|\vec{r}_{i+1}\|}\right) \tag{7.10}$$

$$\gamma = \frac{\sum_{i=1}^{|V|} \alpha_i}{n} \tag{7.11}$$

In the following step all angles are calculated over again and if the angle α_i is smaller than the average angle γ (see Equ. 7.12) the hull point is defined as corner point, [BA00]. Another more general way to find the top rim points is to generate a 2D DeLaunay triangulation based on the top surface points, so

the rim points and feature edges are detected.

$$\alpha_i < \gamma \tag{7.12}$$

Of the polygon formed by these corner points, we now find the longest line c. We than look for a parallel line on the opposite side or if we can not find a parallel line a corner point on the opposite side. With a and b the distances to the opposite point we calculate the altitude h of the triangle abc, where β is the angle between a and b. We check the lines left and right of the furthest point for parallelism with an angle tolerance of 5° (see Fig. 7.9, green lines). If the angle difference is larger than 5° and there are several remaining points we analyse the next largest distances. If no suitable line can be found, we just take the furthest point. If the distance of this corner point is bigger than the maximum opening angle (110mm) of the hand prosthesis, no suitable grasp point can be found.

$$\beta = \arccos\left(\frac{a^2 - b^2 - c^2}{-2 \cdot b \cdot c}\right) \tag{7.13}$$

$$h = b \cdot \sin(\beta) \tag{7.14}$$

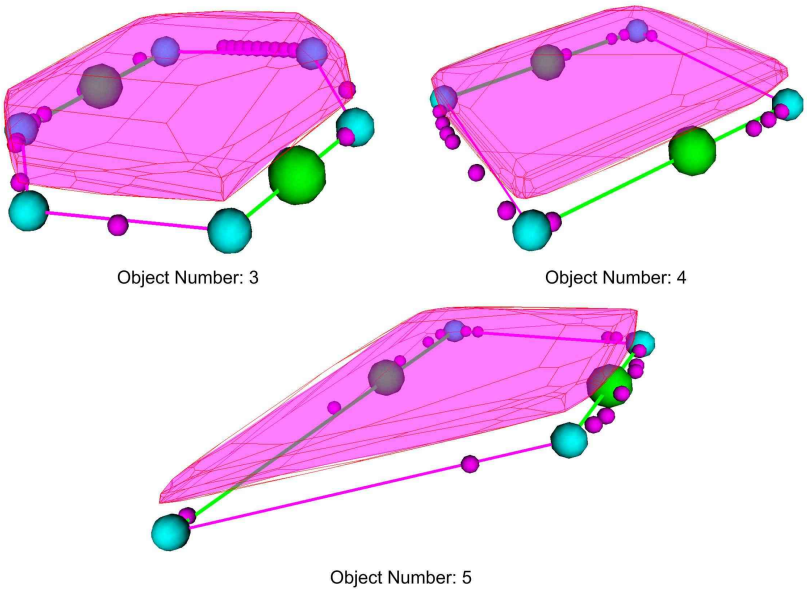


Figure 7.9: Representation of the top surface of the last three objects. The magenta coloured points represent the convex hull. The green points represent the computed grasp points, where the grasp surfaces are represented by the green lines. The corner points, cyan coloured, are computed from the convex hull and the magenta lines show the surfaces which are rather unsuitable to grasp the object. Images are best viewed in colour.

The 1^{st} grasping point in the center of the shorter line is regarded and the 2^{nd} grasping point on the opposite line is calculated with the help of the altitude of the triangle (see Fig. 7.9, especially object number 5). These guarantees a stable grasp for every polygon shape

Now we displace the grasping points $5mm$ towards the table (see Fig. 7.3). That has several reasons, we don't want to grasp on the rim of the object (which would be unstable). On the other hand the computed grasp points are also suitable for inclined top surfaces. Moving the grasp point near to the object center of mass (nearer to the table) would result in a more difficult trajectory for the path planning tool.

7.2.3 Object Grasping

Here we calculate a collision free robot trajectory and execute the grasping activity safely. In the last step of Fig. 7.5 we calculate the object position in the actual environment model and transmit it to the path planning tool. For that a 3D mesh will be generated by using all objects including the target object, based on the triangles calculated by a DeLaunay triangulation [O'R98]. In Fig. 7.10 the mesh generated objects on the table are coloured green. The robot path is calculated by the path planning tool from AMROSE. The input is the detected object pose, the environment model, the grasping points and a transformation between the robot coordinate system and the laser range scanner coordinate system. The output is a collision free robot trajectory to the desired object. We prefer to grasp the different objects from above, which simplifies planning of collision free paths.

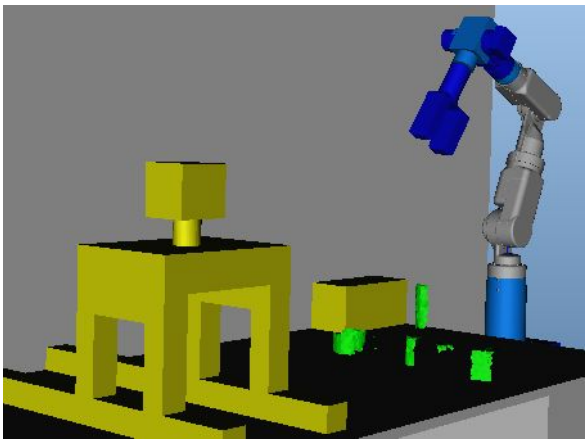


Figure 7.10: Visualization of the experimental set-up by a simulation tool, which is suitable to calculate the trajectory of the robot arm. Images are best viewed in colour.

Before the robot executes the trajectory, the user can check a simulation of the calculated trajectory and decide whether it is safe enough to handle the object or not (see Fig. 7.10). After the robot approaches the user can initiate the closing of the gripper. As soon as the gripper encloses the object, the robot motion to the transfer point starts. Finally the desired object can be placed at a defined position or directly handed over to the user.

The calculation of the object segmentation and grasp point detection is performed by a PC with 1.8GHz Pentium IV processor. The reliability depends on the ambient light, object surface properties, laser beam reflections and vibrations. Therefore, the laser range scanner must be configured to the respective environment. By using an additional red-light filter the impact of light or reflections can be minimized.

7.3 Detection of optimal gripper pose for grasping based on 3D features

Another goal of our work is to analyse the calculated grasp points with the help of a 3D model of the hand prosthesis, which we are using as gripper, see Fig. 7.11. The 3D model of the gripper is realized with a Minolta VIVID 700 range scanner. This 3D model enables it to calculate the optimal position and orientation of the gripper to successfully grasp the desired object. Furthermore it affords to consider all surrounding objects to identify potential obstacles. As well the opening angel can be observed to detect a possible collision with the table. All these information is important for the path planner to calculated a successfully path to grasp the desired object.

We simulate the complete grasping process with a commercial path planning tool from AMROSE[6]. The input is the detected object pose, the gripper pose, the environment model, the grasp points and a transformation between the robot coordinate system and the laser range scanner coordinate system. The

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

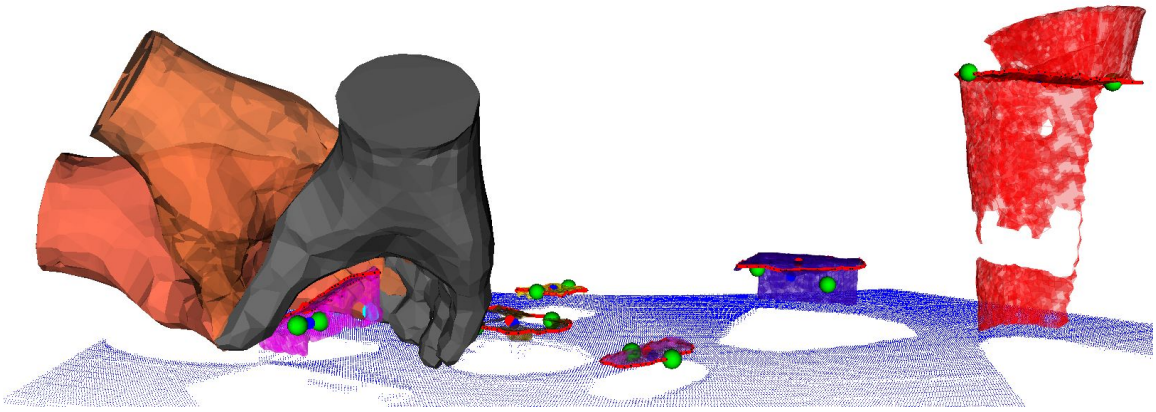


Figure 7.11: This figure shows on the left side three different hand configurations to grasp the stapler. The left 3D model of the hand (red coloured) shows the maximum positive hand orientation by 90°, the right hand (black coloured) shows the maximum negative hand orientation by −30° and the hand model in the middle (orange coloured) shows the optimum orientation by 60°.

output is a collision free robot trajectory to the desired object. Before the robot executes the trajectory, the user can check a simulation of the calculated trajectory.

In order to successfully grasp an object it is not sufficient to find locally the best grasp points, the algorithm must also decide like humans from which angle it is possible to grasp it. Moreover the algorithm checks the validity of the grasp points. For that approach we rotate the 3D model of the hand prosthesis around the rotation axis, which is defined through the grasp points. The rotation axis of the hand is defined by the thumb and the index finger of the hand as illustrated in fig. 7.12 with the cyan coloured points. At the beginning the hand is placed accurately over the grasping object. This start position is defined with a grasping angle of 0°. Furthermore the opening angle of the hand is set to its maximum. The algorithm checks for a collision of the hand with the table or other objects. If there is no collision our approach calculates the maximum and minimum possible rotation angles. We find the best gripper position and orientation by an averaging of the maximum and minimum possible rotation angles. Through that, the algorithm calculates the best gripper pose to grasp the desired object for the path planning tool. If there is a collision the grasp point detection algorithm calculates new grasp points for the desired object. Then the algorithm takes for the first grasp point (GP1) the second shortest euclidean distance between the center of mass and the rim line and all other calculations are repeated.

We decide to use the power crust algorithm for the surface reconstruction [ACK01] of the 3D model of the hand prosthesis, because this algorithm delivers very good results and is quite fast. It realizes a construction which takes a sample of points from the surface of a 3D object and produces a surface mesh and an approximate medial surface axis. The approach approximates the medial axis transform (MAT) of the object. Then it uses an inverse transform to produce the surface representation from the MAT.

This approach allows it to change the start position and orientation of the gripper on-line depending on the grasping object. The grasping pose depends on the grasping object itself, surrounding objects and the calculated grasp points. The advantage of this novel implementation is that it realizes a alleviation for the path planner to grasp an object fast and successfully, as illustrated in Fig. 7.12. More details are in the paper attached in Appendix A item C.

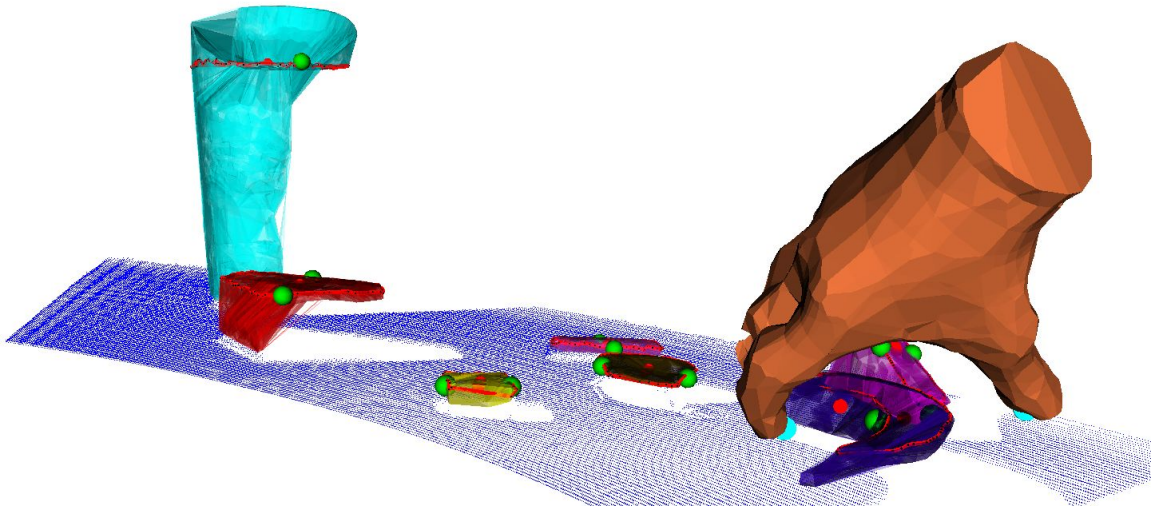


Figure 7.12: The rotation axis of the hand is defined through the thumb and the index finger of the hand with the cyan coloured points. This rotation axis must be aligned with the axis defined by the grasp points.

7.4 Dense Stereo Analysis Overview

Stereo reconstruction is a 3D-reconstruction method using the environment images of a pair of cameras, referred to as stereo cameras. In this method two 2D-images of an object are processed to compute its 3D-coordinates in the real world. Stereo cameras are observing objects from a different point of view, producing images with a different object position - on the images they appear shifted to a certain extend. The amount of the object shift, measured in pixels, is called disparity. Disparity is the key information used to determine the special position of an object in the real world. There is a direct link between the position of an object with respect to the stereo cameras and its disparity value. With the disparity made available, one can compute the objects 3D-coordinates by simply applying triangulation. In addition to determining the objects 3D-geometry, stereo images are supplying the colour information about the environment as well, making it possible to produce 3D-models with the corresponding grey shading or true colour (depending on whether black and white or colour cameras have been used). In order to perform a 3D-reconstruction, some basic data describing the properties of the stereo camera set-up is required. This necessary data is obtained by the camera calibration, witch is the first step in the implementation of a stereo reconstruction system. An efficient algorithm for finding disparities may require additional image processing, such as rectification. Rectification of stereo images is derived from the epipolar geometry of the stereo set-up and it can be used to increase the reliability of the computed disparity results by introducing the epipolar constraint. Further image processing, such as chroma keying (Colorado segmentation of the image objects) and filtering of the disparity values may be used to isolate the objects of interest from their surrounding and to reduce disparity errors. There is a variety of implementations of a stereo reconstruction process with different methods utilized in solving individual tasks within the process [SS]. A disparity search method suitable for obtaining fast 3D-reconstructions of the well-textured objects is area block matching with sliding window [MMHM]. It relies on using a local cost function to evaluate the similarity level of potentially matching pixels. Applying the consistency constraint, that requires the matching process to produce the same correspondence pair of pixels when the matching direction is reversed from left-to-right to right-to-left, increases the result's reliability. The advantage of low processing time is, in return, limited by difficulties to properly reconstruct poorly textured objects, representing the biggest drawback of this method.

In order to obtain good reconstruction results, the emphasis should be set at providing accurate camera calibration parameters as the precondition, and at the proper choice of the disparity search method. However, a more robust behaviour regarding insufficient object texture, presented by some advanced methods, is coupled with increased processing time. Making a fast and accurate method, suitable for real-time applications, remains a challenge in the development of the stereo reconstruction algorithms.

7.5 Combination of Laser-Range and Stereo-Data

One of the primary problems is that $2^1/_2$D point clouds do not represent complete 3D object information. Furthermore stereo data includes a lot of noise and outliers, depending on the texture of the scanned objects. Laser range data typically includes less noise and outliers, but also produces incomplete data because of absorption (e.g. dark surfaces, non dispersive reflectance). The laser exhibits a very high accuracy and the stereo data includes more object information, due to the better field of view. The work contains a comparison of the results of both systems. It shows that better results can be obtained with a combination of both.

Fig. 7.13 illustrates the combination of stereo and laser data using a local shape alignment.

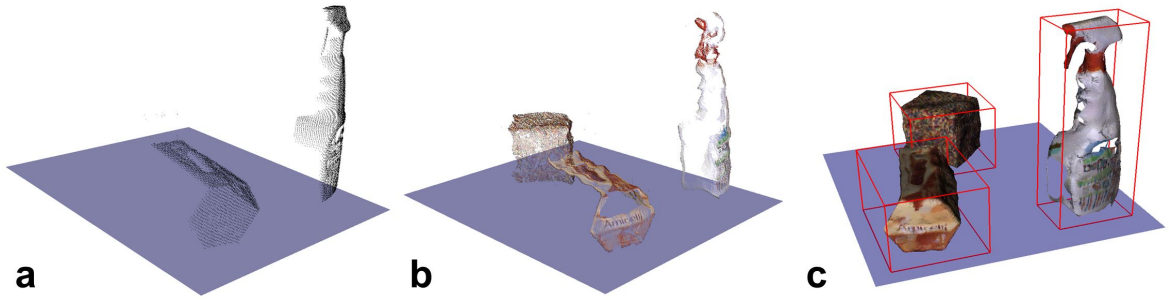


Figure 7.13: **a** Laser range data: The cube was not scanned, because of absorption. **b** Stereo point cloud: All objects could be detected, but the stereo data is very noisy. **c** Combination of laser range and stereo data. The red lines illustrate the boundaries of the different objects on the table.

7.6 Summary

Depth images directly deliver object shape. Laser and stereo data is used to obtain grasp points based on simple heuristics such as starting from the top surface opposing points in relation to the centre of gravity are used. The result is that a relatively large sample of objects can be grasped robustly if a good segmentation is available. More details can be found in the paper attached in Appendix A item B.

Remaining problems are typical for the sensing modalities, where absorbing or reflective materials are not good for laser sensing and textureless areas do not deliver depth in stereo images. The merging of both modalities improves performance. More details can be found in the paper attached in Appendix A item C.

The purpose of this work was to move towards grasping unknown objects. Although strong heuristics have been applied, the results indicate that shape is a good starting point. Certinaly, to fulfill the final goal of GRASP we need to relieve these assumptions.

The way proposed to relieve the constraints is to move from an object-based to a part-based description. This relieves the demand of a perfect segmentation, which is not reasnoable given present state of computer vision research. The work towards this goal is presented in the next Chapter.

Chapter 8

Detection of Graspable Object Parts using 2D Features

The object class recognition is one of the central topics in computer vision research field that can be applied also to the problem of identification of graspable object parts. In the ideal case we would like to know the type of analysed object which allows to associate it with an object specific tasks and consequently grasp allowances. Unfortunately current state of the art methods do not allow for efficient and reliable object detection and classification in general. The WP4 package will therefore combine an object detection and object part detection to increase the likelihood of successful grasp allowance identification.

This section presents an ongoing research on identification of graspable object parts in the still images which provides complementary information to 3D shape detection presented in Section 7 while it can also be used as a stand-alone sensing technique when a stereo data accuracy is low. The first working prototypes of methods described in this section are expected to appear in the second year of the project. The following sections contain detailed approach description accompanied by partial results.

8.1 Semi-local shape based image descriptor

Shape features applied to object recognition has been actively studied since the beginning of the field in 1950s and remain a viable alternative to appearance based methods e.g. local descriptors. This work address the problem of learning and detecting repeatable shape structures in images that may be incomplete, contain noise and/or clutter as well as vary in scale and orientation. A new approach is proposed where invariance to image transformations is obtained through invariant matching rather than typical invariant features. This philosophy is especially applicable to shape features such as open edges which do not have a specific scale or specific orientation until assembled into an object. Our primary contributions are: a new shape-based image descriptor that encodes a spatial configuration of edge parts, a technique for matching descriptors that is rotation and scale invariant and shape clustering that can extract frequently appearing image structures from training images without a supervision.

Edges are an intuitive way to represent shape information, but the problems associated with the edge detection such as edge fragmentation, missing edges due to occlusions or low contrast as well as changes in object scale and orientation affect the final result based on edge matching or classification [1]. To overcome these issues we introduce a novel semi-local shape descriptor which represents the shape of an image structure by means of edges and their configurations. Our *Radial Edge Configuration*-descriptor (REC) encodes edges found in a neighbourhood of an interest point (see Section 8.1.1) as a sequence of radial distances in a polar coordinate system (centered on the interest point). Thus, the similarity of shape is assessed by the comparison of local edge configurations. Here, our main contribution is the definition of a rotation and scale-invariant distance measure between edge configuration descriptors that is able to match multiple edges, preserving their spatial relationships, and reject outlier edge pairs at the same time. This allows for a comparison of image structures across different scales, with only partially

[1] Our method utilizes Canny edge detector.

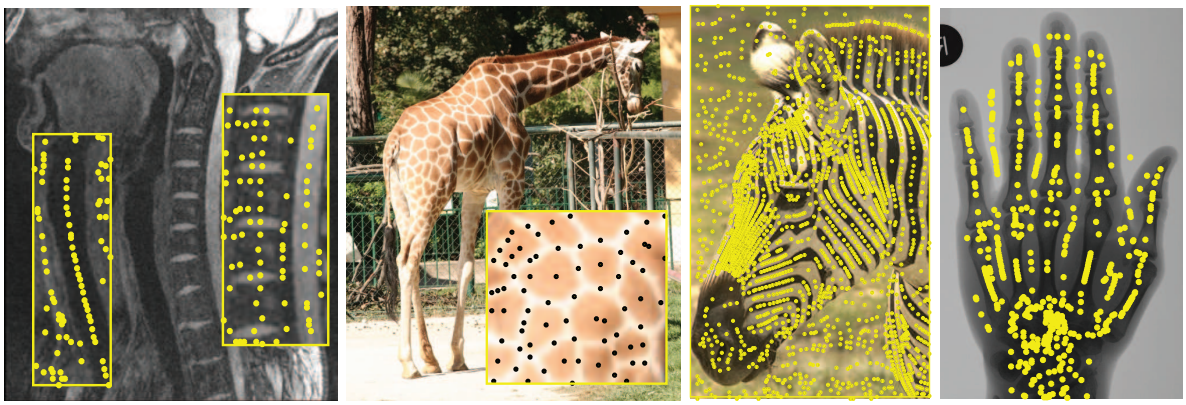


Figure 8.1: Examples of RST based interest points computed at a single scale ($r = \varsigma/50$, where ς is a lower value out of horizontal and vertical image size in pixels).

established correspondences. Another particularity of the chosen approach is that scale and orientation are not estimated during descriptor extraction. Instead they are established as relative entities between two REC descriptors during the distance calculation, which leads to more stable results.

We also introduce a method for weakly supervised learning of structure models that are represented by a set of REC descriptors with individual edges weighted accordingly to their repeatability and similarity within the same category of structures. The structure model learning is achieved through shape clustering presented in Section 8.3.

The shape clustering is related to agglomerative hierarchical clustering but operates on variable length feature vectors, specifically Radial Edge Configurations. The result of shape clustering are "mean" edge fragment configurations (represented by REC descriptors) that can be used to locate similar structures in the image.

8.1.1 Symmetry Based Interest Points

The *Radial Symmetry Transform* (RST) attempts to find locations in the image where the intensity distribution attains locally maximal radial symmetry. The method tends to locate interest points approximately at the centres of round/isotropic structures or along the symmetry axis of elongated shapes. The symmetry measure $S_r(x, y)$ is calculated for each pixel (x, y) of the image separately and the interest points are aligned with local symmetry maxima.

$$S_r(x,y) = -\sum_{i=-r}^{r}\sum_{j=0}^{r} g(\sqrt{i^2+j^2}, \sigma_r = 0.5r)\|\boldsymbol{I}(x+i, y+j) - \boldsymbol{I}(x-i, y-j)\| \tag{8.1}$$

where $\boldsymbol{I}(x+i, y+j)$ is an image pixel intensity or colour at coordinates $(x+i, y+j)$ and r defines the image window size used for the symmetry measure calculation to be a $(2r+1) \times (2r+1)$ rectangle. Each contribution of the pixel pair at $(x+i, y+j)$ and $(x-i, y-j)$ is weighted by the Gaussian $g(\sqrt{i^2+j^2}, r)$ which decreases the influence of pixel pairs at increasing distance from (x, y) and normalizes the transform with respect to the chosen scale R.

In the basic version, the interest point locations $(\vec{\hat{x}}, \vec{\hat{y}})$ correspond to the maxima of the S_r transform:

$$(\vec{\hat{x}}, \vec{\hat{y}}) = \underset{x,y}{\texttt{argmax}}\,(S_r) \tag{8.2}$$

It is also possible to obtain a scale adapted set of interest points using a similar iterative approach as for the scale adapted Harris detector [MS04]. In this case the interest point locations are detected using the symmetry transform and the related scale is detected using the Laplacian operator. Alternatively, an approximation of the scale adapted symmetry measure is a sum of S_r over a sparse set of radii R:

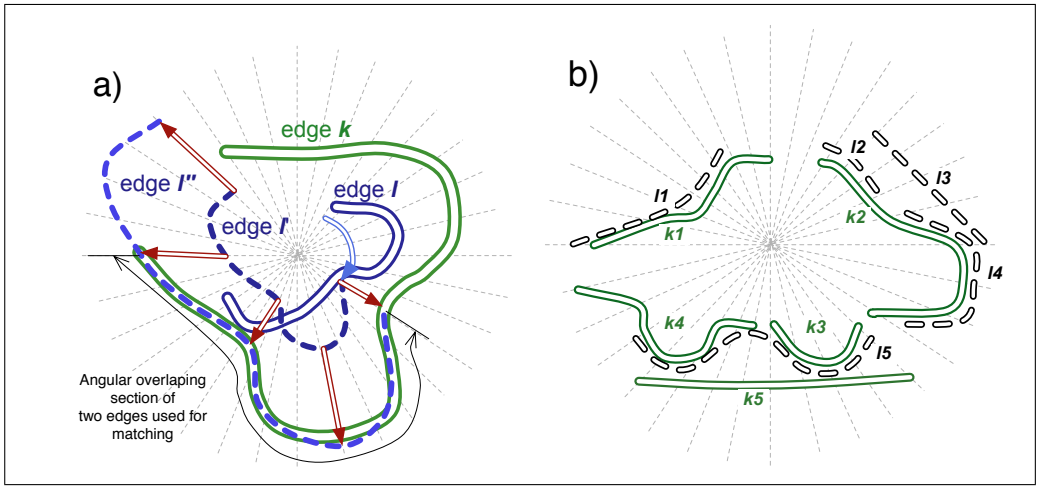


Figure 8.2: a) example of matching edge k and l in polar coordinates. Edge l' is a rotated version of l and l'' is scaled version of l' relative to the origin of the coordinate system. b) example of edge correspondences in two descriptors (edges k and l).

$$S = \sum_{r \in R} S_r \tag{8.3}$$

Examples of interest point detection are presented in Figure 8.1.

8.2 Matching of semi-local image descriptors

The complexity of edge matching is primarily associated with the difficulty in assigning a scale to the edge – a part of one edge may be matched to another edge or to itself at a larger scale (e.g. straight edges or fractal like structures). Polar coordinates allow the definition of an edge scale locally, based on the relative position to the origin of a coordinate system. However, the matching of a part of an edge to a part or whole of another edge is still admissible.

The origin of the coordinate system is associated with the interest point location.

The REC descriptor consists of a variable number of K continuous edges. The k-th edge Γ_k is encoded as an ordered list of radial boundary points, each representing the distance $r_{k,i}$ along the i-th ray from the origin of the polar coordinate system:

$$\Gamma_k = \{r_{k,i} : i \subset \mathbb{N}_0^+; i = (b_k ... b_k + n_k) \bmod N\} \tag{8.4}$$

where b_k denotes the index of the first ray and n_k is the number of rays the edge occupies. The modulo operation is used to ensure that index $i < N$, where N describes the total number of rays (polar resolution) and in all our experiments is set to 64, which we found to offer a good compromise between accuracy and computational cost.

Calculating the distance between two REC descriptors involves finding correspondences between multiple edges. We describe a method to find the best fit between two edges, assuming one of the edges can be rotated and scaled relative to the origin of the polar coordinate system associated with the interest point (as shown in Figure 8.2). This operation is a prerequisite for the estimation of distance between two REC descriptors.

Fitting one edge to another corresponds to finding a transformation (rotation and scaling) which globally minimizes the spatial distance between corresponding boundary points of the two edges. It is important to note that while the scaling of an edge is performed in the continuous domain, the relative rotation is quantized into N rays. The relative scale $\varsigma_{k,l}^{a,b}$ between edge k belonging to the descriptor a and edge l belonging to the descriptor b, rotated by α rays, is calculated as follows:

$$\varsigma_{k,l}^{a,b}(\alpha) = \left(\sum_{i=b_{kl}}^{b_{kl}+n_{kl}} r_{k,i}^{a} r_{l,\bar{i}}^{b} \right) / \left(\sum_{i=b_{kl}}^{b_{kl}+n_{kl}} (r_{l,\bar{i}}^{b})^2 \right) \tag{8.5}$$

where b_{kl} is the first ray containing boundary points of both edges, n_{kl} is the number of consecutive rays containing boundary points from both edges for a given rotation α and $\bar{i} = (i - \alpha) \bmod N$. It is important to note that this scheme allows for partial edge matching, which means that only the overlapping section of the two edges is matched (as shown in Figure 8.2). However, only combinations of α for which $n_{kl} \geqslant \tau$ (in our experiments τ=5) are used, due to the fact that extremely short sections of an edge usually carry less information, which is made worse by the quantization process. It can be easily proven that the spatial distance between corresponding boundary points of the edges k and l, for a given rotation α, is minimized when edge l is scaled (multiplied) by $\varsigma_{k,l}^{a,b}(\alpha)$.

One way of estimating how well two edges fit together is to calculate the variation of relative scale between the corresponding boundary points:

$$\epsilon_{k,l}^{a,b}(\alpha) = \frac{1}{n_{kl}} \sum_{i=b_{kl}}^{b_{kl}+n_{kl}} \left| \log^2 \left(\frac{r_{k,i}^{a}}{r_{l,\bar{i}}^{b}} \right) - \log^2 \left(\varsigma_{k,l}^{a,b}(\alpha) \right) \right| \tag{8.6}$$

This equation is a scale independent fitting distance between two edges for a given relative rotation α. The $\log^2()$ operation is used to avoid impairment associated with the $\frac{r_{k,i}^{a}}{r_{l,\bar{i}}^{b}}$ measure. The relative rotation giving the best fit of the two edges is the one which minimizes the distance $\epsilon_{k,l}^{a,b}$:

$$\epsilon_{k,l}^{a,b} = \min_{\alpha} \left(\epsilon_{k,l}^{a,b}(\alpha) : n_{kl} \geqslant \tau \right) \tag{8.7}$$

Finding the transformation resulting in the best fit between two edges requires $\epsilon_{k,l}^{a,b}(\alpha)$ to be evaluated for all α (for which $n_{kl} \geqslant \tau$ holds).

The REC descriptor contains a set of edges that are the result of edge detection around the corresponding interest point. In reality we should expect that some perceptible edges may be missing or fragmented due to weak gradients and noise. An additional problem is related to the fact that only a subset of edges in the two descriptors may correspond well, while others are related to non-similar image structures. For example we can find patches on a giraffe skin with a high shape similarity at a local scale, but the random distribution of the patches makes shape comparison irrelevant on a large scale. Thus we have to search for a subset of edges in both descriptors, which together give a low fitting error, while other edges are rejected as outliers.

The primary idea behind the matching of multiple edges in the descriptors a and b is summarized below:

1. Perform edge fitting for admissible edge pair combination k and l, resulting in P putative transformations.
2. Repeat multiple edge fitting for P transformations. Choose the one which gives the lowest overall fitting error for the descriptor.
   (a) Rotate and scale all edges in descriptor b according to the current transformation and find the edge correspondences between two descriptors.
   (b) Remove outliers and calculate the final distance from all corresponding edge pairs.

The most computationally demanding task is finding edge correspondences for a given relative scale and rotation. The difficulty is associated with the possibility that a single edge in one descriptor may correspond to more than one non-overlapping edges in the other descriptor. An example of such multi-correspondences is shown in the Figure 8.2-b – edge $k2$ corresponds to edges $l2$ and $l4$, while edges $k4$ and $k3$ correspond to edge $l5$. Note that edge $l3$ could be also matched to the edge $k2$, but it overlaps with edges $l2$ and $l4$, which produce a better fit with edge $k2$. The process of finding edge correspondences can be divided into several steps:

1. Find overlapping edge pairs in a: $\phi_{k1,k2}^{a} = \begin{cases} 1, & \text{if } k1 \text{ and } k2 \text{ overlap } \geqslant \tau \\ 0, & \text{otherwise} \end{cases}$

2. Find overlapping edge pairs in b: $\phi^b_{l1,l2} = \begin{cases} 1, & \text{if } l1 \text{ and } l2 \text{ overlap } \geqslant \tau \\ 0, & \text{otherwise} \end{cases}$

3. Find overlapping edge pairs between a and b: $\phi^{ab}_{k,l} = \begin{cases} 1, & \text{if } k \text{ and } l \text{ overlap } \geqslant \tau \\ 0, & \text{otherwise} \end{cases}$

4. Find edge correspondence. The edge l is correspondent to edge k if:

$$\epsilon^{a,b}_{k,l} = \min_{f,g}\Big(\epsilon^{a,b}_{f,g} : f \in \{\phi^{ab}_{f,l} = 1 \wedge \phi^{a}_{f,k} = 1\}; g \in \{\phi^{ab}_{k,g} = 1 \wedge \phi^{b}_{l,g} = 1\}\Big) \tag{8.8}$$

   which means that edges k and l correspond when the distance $\epsilon^{a,b}_{k,l}$ is the minimum among all combinations of edges f and g which overlap with k and l. This condition allows the association of multiple non-overlapping edges in one descriptor with a single edge in another descriptor.

The final distance between two descriptors a and b is a weighted sum of individual edge-pair (k, l) distances:

$$\epsilon^{a,b} = \frac{1}{\sum_{k,l} \upsilon^a_k \upsilon^b_l} \sum_{k,l} \upsilon^a_k \upsilon^b_l \epsilon^{a,b}_{k,l} \tag{8.9}$$

where the weights υ_k and υ_l describe the confidence of edge match:

$$\upsilon_k = \frac{\widehat{s}^a_k}{s^a_k} \tag{8.10}$$

where s^a_k is the total length of edge k in descriptor a and $\widehat{s}^a_k$ is the length of all edge fragments that were matched to edges in the descriptor b. The edge match confidence reaches 1 if it was completely matched to other edge or edges and is 0 if it was not matched to any edges.

During our matching tests we found that a simple outlier removal scheme helped to improve results when only a part of the structure in the two descriptors was found to correspond.

Examples of finding similar image structures through the edge matching are presented in Figures 8.3 and 8.4. Majority of descriptors are matched to similar structures despite differences in scale, orientation and shape deformations.

8.3 Clustering based extraction of repeatable image structures

Clustering of local image descriptors (e.g. SIFT) is the basis of object recognition techniques such as "bag of keypoints" [ZMLS07] as well as part based models [LLS04]. In these cases clustering allows for a compact (data reduction) representation of distinctive image structures. Among the most popular clustering methods are hierarchical, k-means and kd-tree clustering. The first difference between clustering of typical image descriptors and clustering of the REC descriptor is that the later produces a variable length feature vector (the number of edges can vary significantly). This prevents the use of k-means and kd-tree clustering which require constant dimensionality of the feature vectors. The second difference is that the clustering of REC descriptors assigns weights to edges and individual boundary points along the edges that depend on the edge repeatability across training instances of the same structure type and the amount of variability an edge exhibits across the training instances.

The REC descriptor is clustered using agglomerative hierarchical clustering [DHS00] based on the REC distance defined in Section 8.2. Clustering starts with finding the closest pairs between a set of descriptors extracted from the training data set labelled as clustering level $t = 0$. The closest pairs are merged into nodes at the next clustering level and the same procedure is repeated on these nodes. The closest descriptor pairs are merged only if the matching distance between them does not exceed the threshold τ. Therefore clustering is performed until no more pairs can be merged. Parameter $\tau = 0.4$ was experimentally chosen and used in all tests presented in this chapter. The merging of two descriptors is an operation which generates a single edge for each set of corresponding edges in two descriptors as described in Section 8.2. Recall that a single edge in one descriptor can correspond to several edges in another descriptor and that some edges do not have any correspondences and are down-weighted in the

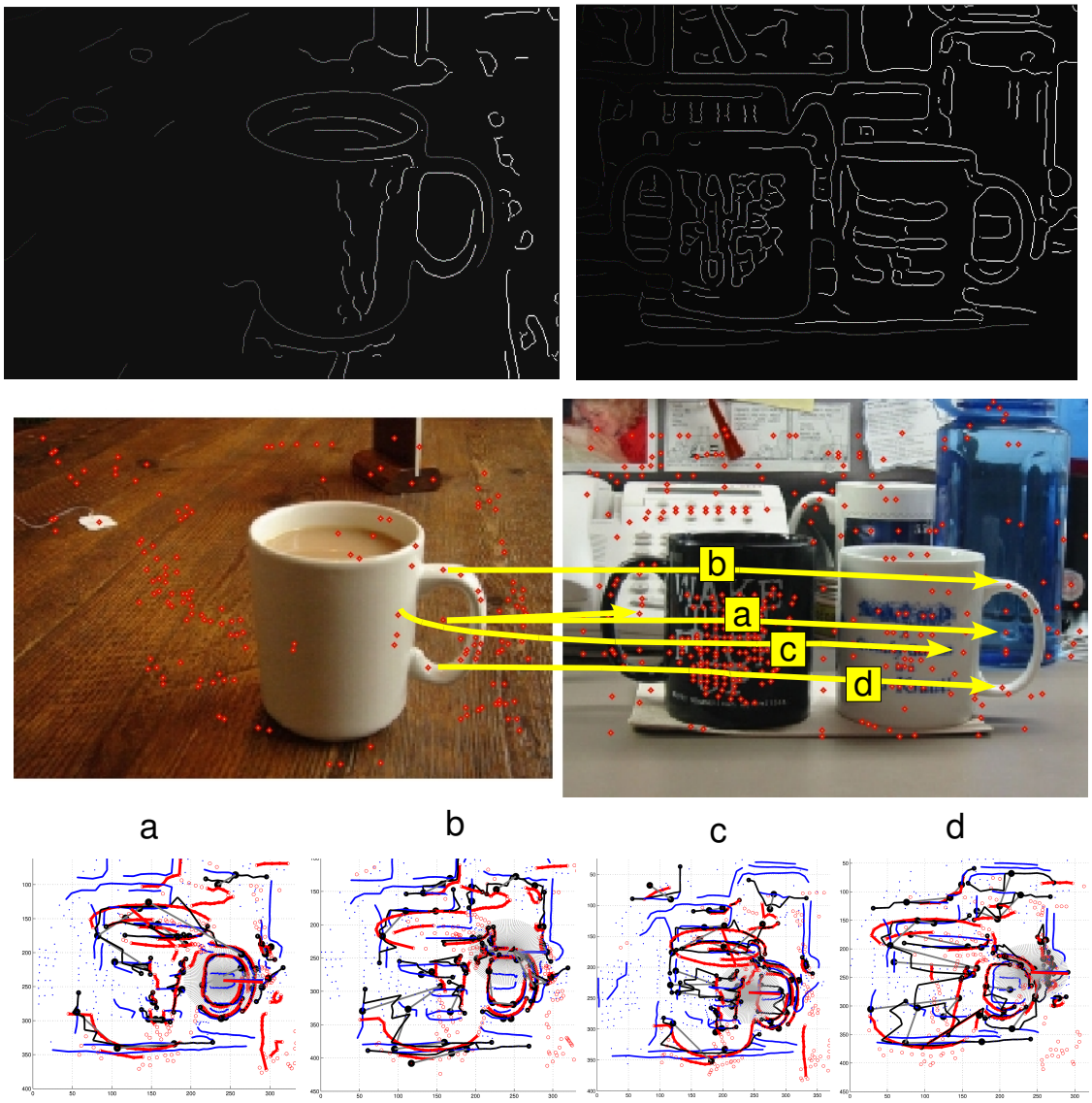


Figure 8.3: Top row: example of edge detection used for extraction of REC descriptors. Middle row: example of interest point correspondences based on descriptor matching. Only a representative subset of interest point matches is shown to avoid clutter. Bottom row: examples of edge correspondence in matched descriptors (red and blue) and the resulting mean edges after descriptor merging (black).

merged descriptor. The edge kl, which is a result of merging of edges k and l, is obtained by averaging the boundary point positions from both edges:

$$\Gamma_{kl} = \{0.5(r_{k,i} + r_{l,i-\alpha \bmod N_0^+}) : i \subset \mathbb{N}; i = (b_{kl} ... b_{kl} + n_{kl}) \bmod N\} \tag{8.11}$$

In addition, each boundary point is assigned the weight that is corresponding to the distance between two merged boundary points and includes the boundary point weights from the previous clustering level. This way edges are prioritized according to their similarity across the clustering levels.

$$w_{kl}^t(i) = \omega_p(w_k^{t-1} + w_l^{t-1}) + \omega_d \exp\left(- \left(1 - \frac{\max\left(r_{k,i}^a, \varsigma_{k,l}^{a,b} r_{l,\bar{i}}^b\right)}{\min\left(r_{k,i}^a, \varsigma_{k,l}^{a,b} r_{l,\bar{i}}^b\right)} \right)^2 / \sigma^2 \right) \tag{8.12}$$

where σ was set to 0.25 in all experiments and regulates the down-weighting depending on the local edge deformation – the difference between relative boundary point scale and the relative descriptor scale. The

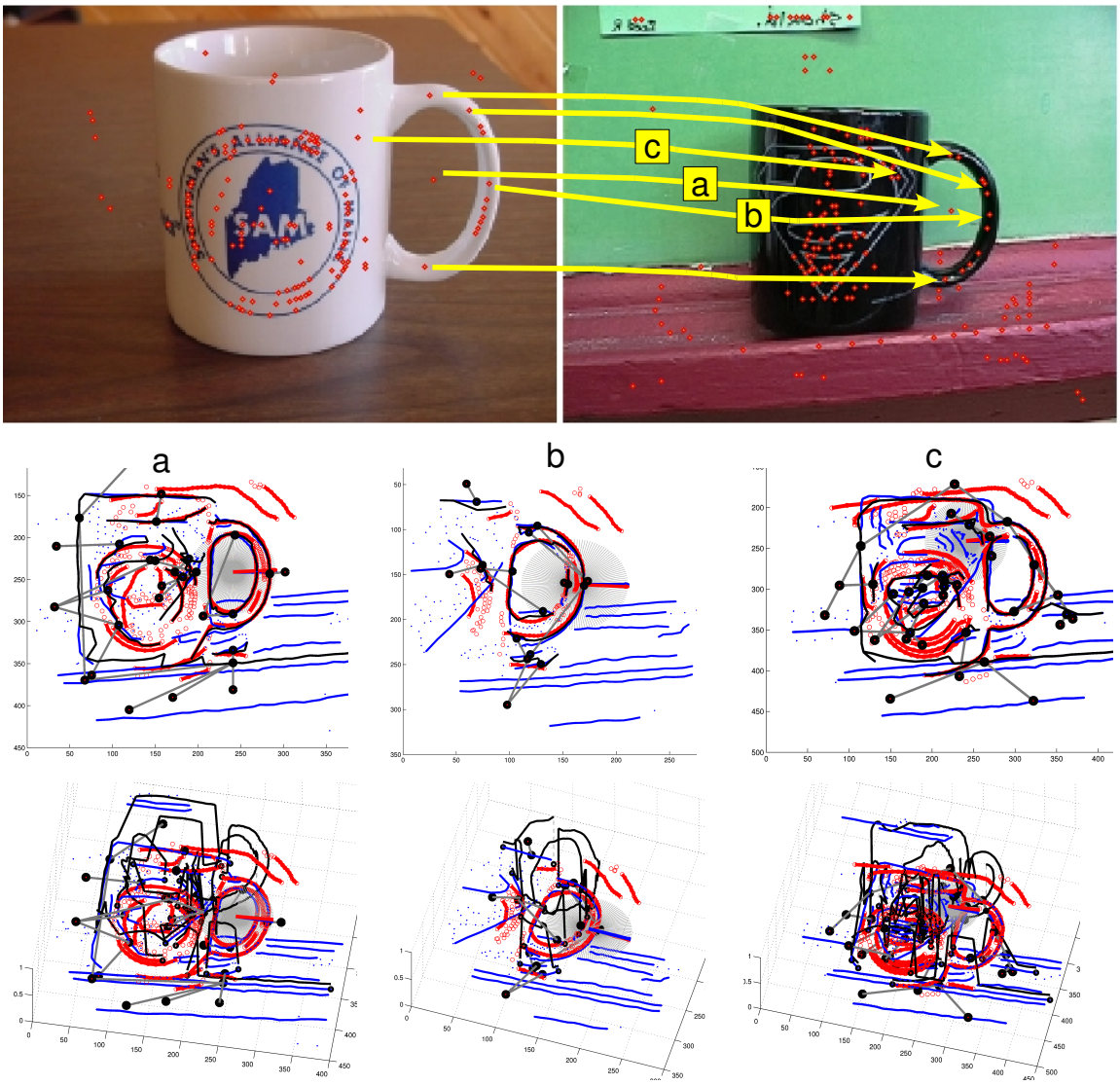


Figure 8.4: Top row: example of interest point correspondences based on descriptor matching. Only a representative subset of interest point matches is shown to avoid clutter. Middle row: examples of edge correspondence in matched descriptors (red and blue) and the resulting mean edges after descriptor merging (black). Note that not all edges have been matched. We strongly advise to view all images in colour. Bottom row: visualisation of mean edge weights (z axis) based on local edge similarity.

parameters ω_p and ω_d regulate the influence of edge weights from previous cluster level $t-1$ (history) and the differences between merged edges (deformation) respectively onto the final weight $w_{kl}^t(i)$. These were set to $\omega_p = 0.25$ and $\omega_d = 0.75$ in all experiments which prioritizes the influence of "deformation" over the "history". The edges without correspondences are copied into the merged descriptor and the corresponding weights are divided by two – if such an edge consequently has no correspondences at multiple clustering levels its weight is reduced to approximately 0.

At clustering level $t = 0$ all boundary point weights are set to 1 which means that all edges in every descriptor have identical priority.

The result of clustering is a set of REC descriptors, which contain edges resulting from edge merging across a number of clustering levels. The weights assigned to the edges are then used during matching cluster nodes (structure models) to descriptors in the test data set. The edge distance (8.6) is then replaced with:

$$\epsilon_{k,l}^{a,b}(\alpha) = \frac{\sum_{i=b_{kl}}^{b_{kl}+n_{kl}} w_{k,i}^{a} \left| \log^2\left(\frac{r_{k,i}^{a}}{r_{l,\bar{i}}^{b}}\right) - \log^2\left(\varsigma_{k,l}^{a,b}(\alpha)\right) \right|}{\sum_{i=b_{kl}}^{b_{kl}+n_{kl}} w_{k,i}^{a}} \tag{8.13}$$

where descriptor a corresponds to the cluster node and weights for descriptor b corresponding to the detected structure are set to 1.

This step will be used for building a weakly supervised feature codebook and used in conjuction with the classifier based on likelihood of particular image structure detection as well as spatial co-occurrence of the codebook features.

8.4 Conclusion and Next Steps

The work shows that local structure can be very well used to obtain shape information about objects. The method has been primarily developed during the first year of the project. First publications are planned for the next two months. The method shows promising results both in terms of detection rates as well as in terms of learnability.

We think the method is of particular interest towards the high goal of grasping any object, because it picks up local structure of objects. Only local structure will be able to convey information that may generalise to new objects. Hence we propose to continue the work on these methods. Consequently, the WP4 related research is currently focused on:

- 3D shape detection using 2.5D point clouds obtained from a stereo system.
- Shape based object class recognition that utilizes spatial configuration of shape features (edges) for weakly supervised learning and detection of multiple classes of objects at varying scales, orientations and positions in images (invariant to similarity transform).
- Part based shape detection based on the aforementioned principle.
- Co-occurrence based object detection with local image descriptors that is robust to image occlusions and is scale invariant.
- Improving efficiency of the REC descriptor (see Chapter sec:2Dfeatures) to enable faster comparison and grouping of local shape descriptors.

Chapter 9

Conclusion and Further Work

This deliverable presented the work in year one in WP4 towards Task 4.1. The main results are

- Hand and object tracking with handling occlusions (Chapter 3),
- Object recognition and pose estimation for initial robot grasping experiments (Chapter 4),
- Scene clustering (Chapter 5) and improved reconstruction (Chapter 6) for use in WP5 as well as pre computation for grasp point detection.
- Grasp point detection in depth images (Chapter 7), for more local part shapes (Chapter 8) and for rather complete part shapes (Chapter **??**).

With this a large sampling of cues is available for further work towards the goals of Task 4.1. Another next step is the partial integration of these methods. The reconstruction from stereo has been put to an on-line tool at TUW, such that other partners only need to send images and get the depth image back. This will be used for the demonstration at the first review.

9.1 Further Work

Our belief is that none of the currently available computer vision approaches is sufficient on its own as a reliable source of information for grasping in general. We therefore believe that a vision sensing should apply a broad range of complementary approaches to maximize the likelihood of identification of graspable object parts. For example, if the detection of a complete object is not possible due to occlusions or if an object is viewed that has never been seen before, a specific graspable object part may still be detected by the mean of local and semi-local structure analysis. We are therefore actively developing several types of complementary methods, e.g., as given in Chapters 7, 8 and **??**. The applicability and performance of each of these methods will be extensively evaluated in the future.

Regarding the completion of Task 4.1, we will further study the grouping of edge structure features to shapes and their relation to objects (TUW), surface reconstruction and tracking as basis for the figure/ground segmentation for WP5 (TUM, KTH), recognition and classification of objects (TUW, KTH, FORTH), the spatio-temproal relationships for learning from human examples (FORTH, KTH), establish first links to multi-modal grounding in WP3, and the integration in the ontology (WP2) including the combination with prediction (WP6). This is very ambitious. Additionally work on Task 4.2 will start, where the structural and shape features will be related to the affordance "graspable". This will be a large step forward towards grasping any object.

References

[AAD07] P. Azad, T. Asfour, and R. Dillmann. Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 919–924, San Diego, USA, 2007.

[AAV+08] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schr"oder, and R. Dillmann. Toward Humanoid Manipulation in Human-Centred Environments. *Robot. Auton. Syst.*, 56(1), 2008.

[ACK01] N. Amenta, S. Choi, and R. Kolluri. The power crust. *Sixth ACM Symposium on Solid Modeling and Applications*, pages 249–260, 2001.

[AL04] Antonis A. Argyros and Manolis I. A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *European Conference on Computer Vision (ECCV)*, pages 368–379, 2004.

[AMNS98] S. Arya, D. M. Mount, N. S. Netanyahu, and R. Silverman. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM*, 45(6):801–923, 1998.

[BA00] G.A. Borges and M.J. Aldon. A split-and-merge segmentation algorithm for line extraction in 2-d range images. In *ICPR '00: Proceedings of the International Conference on Pattern Recognition*, page 1441, Washington, DC, USA, 2000. IEEE Computer Society.

[BB07] W. Burger and M. Burge. *Digital Image Processing - An Algorithmic Introduction Using Java*. Springer, UK, London, 1st edition, 2007.

[BE99] Gabriel J. Brostow and Irfan A. Essa. Motion based decompositing of video. *Computer Vision, IEEE International Conference on*, 1:8, 1999.

[Ben75] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(19):509–517, 1975.

[BH02] D. Burschka and G. Hager. Stereo-Based Obstacle Avoidance in Indoor Environments with Active Sensor Re-Calibration. In *International Conference on Robotics and Automation*, pages 2066–2072, 2002.

[BH04] Darius Burschka and Gregory D. Hager. V-GPS(SLAM): – Vision-Based Inertial System for Mobile Robots. In *Proc. of ICRA*, pages 409–415, April 2004.

[BL95] G. Blais and M. Levine. Registering multiview range data to create 3d computer objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):820–824, 1995.

[BM92] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2), 1992.

[BS91] G. Bilbro and W. Snyder. Optimization of functions with many minima. *IEEE Trans. on Systems, Man, and Cybernetics*, 21(4):840–849, 1991.

[BSW85] Renee Baillargeon, Elizabeth S. Spelke, and Stanley Wasserman. Object permanence in five-month-old infants. *Cognition*, 20(3):191–208, 1985.

[BW98] D. Bulger and G. Wood. Hesitant adaptive search for global optimisation. *Math. Program.*, 81:89–102, 1998.

[CHC99] C.-S. Chen, Y.-P. Hung, and J.-B. Cheng. RANSAC-Based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11):1229–1234, 1999.

[CM91] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Robotics and Automation, Proceedings., IEEE International Conference on*, 3:2724–2729, 1991.

[DHS00] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[DWJ97] C. Dorai, J. Weng, and A. Jain. Optimal registration of object views using range data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(10):1131–1138, 1997.

[FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[HE05] Yan Huang and Irfan Essa. Tracking multiple objects through occlusions. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:1051–1058, 2005.

[HJBJ+] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher.

[HK08] K. Huebner and D. Kragic. Selection of Robot Pre-Grasps using Box-Based Shape Approximation. In *2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*, 2008.

[HP95] R. Horst and P. Pardalos, editors. *Handbook of Global Optimization*. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, 1995.

[IM01] M. Isard and J. Maccormick. Bramble: a bayesian multiple-blob tracker. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 34–41 vol.2, 2001.

[Ing89] L. Ingber. Very fast simulated reannealing (VFSR). *Mathematical and Computer Modeling.*, 12(8):967–973, 1989.

[JC] X. Jiang and D. C. Cheng.

[JF01] Nebojsa Jojic and Brendan J. Frey. Learning flexible sprites in video layers. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:199, 2001.

[JFEM03] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1296–1311, 2003.

[Kan90] K. Kanatani. *Group-Theoretical Methods in Image Understanding*, chapter 6. Springer Series in Information Sciences. Springer, 1990.

[KS00] Sohaib Khan and Mubarak Shah. Tracking people in presence of occlusion. In *In Asian Conference on Computer Vision*, pages 1132–1137, 2000.

[LLS04] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pages 17–32, Prague, Czech Republic, 2004.

[LSVG07] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.

[MGPG04] N. Mitra, N. Gelfand, H. Pottmann, and L. Guibas. Registration of point cloud data from a geometric optimization perspective. In *Symposium on Geometry Processing*, pages 23–32, 2004.

[MJAL03] Jorge S. Marques, Pedro M. Jorge, Arnaldo J. Abrantes, and J. M. Lemos. Tracking groups of pedestrians in video sequences. *Computer Vision and Pattern Recognition Workshop*, 9:101, 2003.

[MJD+00] Stephen J. Mckenna, Sumer Jabri, Zoran Duric, Harry Wechsler, and Azriel Rosenfeld. Tracking groups of people. *Computer Vision and Image Understanding*, 80:42–56, 2000.

[MMHM] K. Mhlmann, D. Maier, J. Hesser, and R. Mnner. Calculating dense disparity maps from color stereo images, an efficient implementation. *International Journal of Computer Vision*, 47(1).

[MS04] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[OA05] A. S. Ogale and Y. Aloimonos. Shape and the stereo correspondence problem. In *International Journal of Computer Vision*, volume 65(3), pages 147–162, 2005.

[O'R98] J. O'Rourke. *Computational Geometry in C.* Univ. Press, 1998.

[PH03] H. Pottmann and M. Hofer. *Geometry of the squared distance function to curves and surfaces*, pages 221–242. Visualization and Mathematics III. Springer, 2003.

[PHYH06] H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu. Geometry and convergence analysis of algorithms for registration of 3d shapes. *International Journal of Computer Vision*, 67(3):277–296, 2006.

[Pia54] J. Piaget. *The construction of reality in the child.* New York: Basic books, San Diego, CA, USA, 1937/1954.

[PR02] P. Pardalos and E. Romeijn, editors. *Handbook of Global Optimization II.* Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, 2002.

[RK95] James M. Rehg and Takeo Kanade. Model-based tracking of self-occluding articulated objects. In *International conference on Computer Vision (ICCV)*, pages 612–617, 1995.

[RL01] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3DIM*, pages 145–152. IEEE Computer Society, 2001.

[SS] D. Scharstein and R. Szeliski.

[TSK02] Hai Tao, H. S. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1):75–89, 2002.

[WYH03] Ying Wu, Ting Yu, and Gang Hua. Tracking appearances with occlusions. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:789, 2003.

[YMC07] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. pages 1–8, 2007.

[ZLN08] Li Zhang, Yuan Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[ZMLS07] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

[ZT03] Yue Zhou and Hai Tao. A background layer model for object tracking through occlusion. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1079, Washington, DC, USA, 2003. IEEE Computer Society.

[Z.Z04] Z.Zivkovic. Improved adaptive gausian mixture model for background subtraction. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2004.

Appendix A

Appendix A: Attached Papers

A Jeannette Bohg, Danica Kragic: Encoding Relative Object Shape for Robotic Grasping; Robotics and Autonomous Systems (submitted).

B Mario Richtsfeld, Markus Vincze: Robotic Grasping of Unknown Objects using 2 1/2D Point Clouds; 17th International Workshop on Robotics in AlpeAdriaDanube Region, 2008.

C Mario Richtsfeld, Markus Vincze: Grasping of Unknown Objects from a Table Top; Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments, adjunct to ECCV, Marseille, , 2008.

Encoding Relative Object Shape for Robotic Grasping

Jeannette Bohg, Danica Kragic

Computer Vision and Active Perception Lab
Centre for Autonomous Systems
School of Computer Science and Communication
Royal Institute of Technology, 10044 Stockholm, Sweden
{bohg, danik}@nada.kth.se

Abstract

This paper presents work on vision based robotic grasping. The proposed method relies on extracting and representing the global contour of an object in a monocular image. A suitable grasp is then generated using a learning framework where prototypical grasping points are learnt from several examples and then used on novel objects. For representation purposes, we apply the concept of shape context and for learning we use a supervised learning approach in which the classifier is trained with labelled synthetic images. We evaluate and compare the performance of a linear and a non-linear classifier. Our results show that a combination of a descriptor based on shape context with a non-linear classification algorithm leads to a stable detection of grasping points for a variety of objects.

Key words: Grasping, Shape Context, Affordances, SVM

1. Introduction

Robotic grasping of unknown objects remains an open problem in the robotic community. Given an object, the embodiment of the robot and the task itself, the amount of potential grasps that can be applied to that object is huge. How to choose a feasible grasp and, at the same time, deal with incomplete information about e.g. the object geometry is not a trivial task.

Although humans master this skill easily, no suitable representations of the whole process have yet been proposed in the neuroscientific literature, making it thus difficult to develop robotic systems that can mimic human grasping behaviour. However, there is some valuable insight. Goodale [1], Goodale et al. [2] propose that the human visual system is characterised by a division into the dorsal and ventral pathway. While the dorsal stream is mainly responsible for the spatial vision targeted towards extracting action relevant visual features, the ventral stream is engaged in the task of object identification. This dissociation also suggests two different grasp choice mechanisms dependent on whether a known or unknown object is to be picked up. Support for this thesis can be found

in behavioural studies by Borghi [3], Creem and Proffitt [4]. The authors claim that in the case of novel objects, our actions are purely guided by affordances as introduced by Gibson [5]. In case of known objects, semantic information (e.g., through grasp experience) is needed to grasp them appropriately according to their function. However as argued in [1, 6, 7] this division of labour is not absolute. In case of objects that are similar to previously encountered ones, the ventral system helps the dorsal stream in the action selection process by providing information about prehensile parts along with their afforded actions.

In this paper, we review different approaches towards solving the object grasping problem in the robotic community and propose a vision based system that models several important steps in object grasping. We start by proposing three ways for approaching the problem, namely grasping of:

- *Known Objects*: These approaches consider grasping of *a-priori* known objects. The goal is then to estimate object's pose and retrieve a suitable grasp, e.g., from an experience database, [8, 9, 10].
- *Unknown Objects*: Approaches that fall into this category commonly represent the shape of an *unknown* object and apply rules or heuristics to reduce the number of potential grasps [11, 12, 13, 14, 15].
- *Familiar Objects*: These approaches try to re-use grasp experience that was gathered beforehand on specific objects to pick up objects that look similar to them. Objects can be *familiar* in different ways, e.g, in terms of shape, colour or texture. A common assumption is that new objects similar to the old ones can be grasped in a similar way [16, 17, 18].

A general observation considering the related work is that there is a trade-off between the quality of an inferred grasp and the applicability of the method in a real world scenario. The more precise, accurate and detailed an object model is, the more suitable it is for doing grasp planning based on criteria such as, for example, stability. It is however difficult to provide a representation that takes into account all different aspects of real world scenarios. If a representation incorporates different types of errors and noise, more assumptions have to be introduced regarding object geometry and generated grasps. Thus, although applicable in a real world scenario, there may be a tendency of decreased quality of inferred grasps.

In our approach, we formulate the basic requirements for an object representation. First, it has to be suitable to be extracted from sensory data such as stereo cameras. Second, it has to be rich enough to allow for the inference of the most important grasp parameters. In our case that is the *approach vector* [8] and the wrist orientation of the robotic hand. We see precise shape, texture and weight to be handled by a subsequent fine controller based on tactile-feedback and corrective movements as presented in our previous work, Tegin et al. [19]. Thus, we introduce a method that applies an object representation fulfilling these requirements. We detect a grasping point based on the global shape of an arbitrary object in a monocular image [20]. This results in relating the 2D

form of an object to a single point in left and right images. After inferring the grasping point's 3D representation from stereo geometry, the approach vector can be defined. The advantage of using global shape over e.g. local appearance lies in the fact that it infers not only a grasping point but the whole approach vector. Research in the area of neuropsychology also emphasises the influence of global shape when humans choose a grasp [2, 21, 22]. We further apply a supervised learning algorithm, thus providing a methodology for grasping objects of *similar* shape. The contributions of our approach are:
i) We apply the concept of shape context to the task of robotic grasping which to the best of our knowledge has not yet been applied for that purpose. The approach is different from the one taken in [16, 18] where only local appearance is used instead of global shape.
ii) We are inferring full grasp configurations for arbitrarily shaped objects from a stereo image pair. These are the main difference to the work presented in [17, 23] where either only planar objects are considered or three views from an object have to be obtained by moving the camera.
iii) We analyse how stable our algorithm is for a general tabletop scenario in the presence of background clutter without having trained with examples of that specific scenario as for example done in [16].
iv) We apply a supervised learning algorithm trained using synthetic labelled images from the database provided by [16]. We compare the classification performance when using a linear classifier such as logistic regressions and a non-linear classifier such as *Support Vector Machines* (SVMs).

The remainder of this paper is organised as follows: In the next section, we present related work. In Section 3, the method of applying shape context to grasping is introduced. We also describe and comment on the database that we used for training and give some background knowledge on the two different classification methods. The section concludes with a presentation on how a whole grasp configuration can be derived. In Section 4 we evaluate our method both on simulated and real data. The last section concludes the paper and gives an outlook on future work.

2. Related Work

There is a significant body of work dealing with grasp modelling. We use the division proposed in the previous section to review the related work.

2.1. Grasping Known Objects

The main problem in the area of grasp planning is the huge search space from which a *good* grasp has to be retrieved. Its size is due to the large number of hand configurations that can be applied to a given object. In the theory of contact-level grasping [24, 25] a good grasp is defined from the perspective of forces, friction and wrenches. Based on this different criteria are defined to rate grasp configurations, e.g., force closure, dexterity, equilibrium, stability and dynamic behaviour.

Several approaches in the area of grasp planning exists that apply these criteria to find a good grasp for an object with a given 3D model. Some of them approximate the object's shape with a number of primitives such as spheres, cones, cylinders and boxes [26] or superquadrics (SQ) [27]. These shape primitives are then used to limit the amount of candidate grasps and thus prune the search tree for finding the most stable grasp. Ciorcarlie et al. [28] exploited results from neuroscience that showed that human hand control takes place in a much lower dimension than the actual number of its degrees of freedom. This finding was applied to directly reduce the configuration space of a robotic hand to find pre-grasp postures. From these so called eigengrasps the system searches for stable grasps. Borst et al. [29] reduce the number of candidate grasps by randomly generating a number of them dependent on the object surface. The authors show that this approach works well if the goal is not to find an optimal grasp but instead a fairly good grasp that works well for " everyday tasks". Quite a different approach is taken by Li and Pollard [30]. Although, the method is independent of the ideas of contact-level grasping it still relies on the availability of a 3D object model. The authors treat the problem of finding a suitable grasp as a shape matching problem between the hand and the object. The approach starts off with a database of human grasp examples. From this database a suitable grasp is retrieved when queried with a new object. Shape features of this object are matched against the shape of the inside of the available hand postures.

All these approaches are developed and evaluated in simulation. However, Ekvall and Kragic [8] and Morales et al. [9] combine real and simulated data for the purpose of grasping *known* objects, i.e. their 3D model is available. In a monocular image a known object is recognised and its pose within the scene is estimated. Given that information, an appropriate grasp configuration can be selected from a grasp experience database. This database was acquired offline through simulations of grasps on 3D models of a set of these known objects. While Ekvall and Kragic [8] still apply the selected grasp in simulation, Morales et al. [9] ported this approach to the robotic platform described in Asfour et al. [31]. Glover et al. [10] consider known deformable objects. For representing them probabilistic models of their 2D shape are learnt. The objects can then be detected in monocular images of cluttered scenes even when they are partially occluded. The visible object parts serve as a basis for planning a stable grasp under consideration of the global object shape. However, all these approaches are dependent on an a-priori known dense or detailed object model either in 2D or in 3D.

2.2. Grasping Unknown Objects

If the goal is to grasp an *unknown* object these approaches are not applicable since in practise it is very difficult to infer its geometry fully and accurately from measurements taken from sensor devices such as cameras and laser range finders. There are various ways to deal with this sparse, incomplete and noisy data. Hübner and Kragic [11], Dunes et al. [12] for example approximate an object with shape primitives that provide cues for potential grasps. Hübner

and Kragic [11] decompose a point cloud derived from a stereo camera into a constellation of boxes. The simple geometry of a box reduces the number of potential grasps significantly. Dunes et al. [12] approximate the rough object shape with a quadric whose minor axis is used to infer the wrist orientation, the object centroid serves as the approach target and the rough object size helps to determine the hand pre-shape. The quadric is estimated from multi-view measurements of the rough object shape in monocular images. Opposed to the above mentioned techniques Bone et al. [15] made no prior assumption about the rough shape of the object. They applied shape carving for the purpose of grasping with a parallel-jaw gripper. After obtaining a model of the object, they search for a pair of reasonably flat and parallel surfaces that are best suited for this kind of manipulator. Richtsfeld and Vincze [13] use a point cloud of an object that is obtained from a stereo camera at a fixed viewpoint. They are searching for a suitable grasp with a simple gripper based on the shift of the top plane of an object into its centre of mass. Kraft et al. [14] also use a stereo camera to extract an object model. Instead of a raw point cloud, they are processing it further to obtain a sparser model consisting of local multi-modal contour descriptors. Four elementary grasping actions are associated to specific constellations of these features. With the help of heuristics the huge number of resulting grasp hypothesis is reduced to only a few of them.

2.3. Grasping Familiar Objects

A promising direction in the area of grasp planning is to re-use experience to grasp *familiar* objects. Many of the objects surrounding us can be grouped together into categories of common characteristics. There are different possibilities what these commonalities can be. In the computer vision community for example, objects within one category usually share characteristic visual properties. These can be, e.g., a common texture [32] or shape [33, 20], the occurrence of specific local features [34, 35] or their specific spatial constellation [36, 37]. These categories are usually referred to as *basic level categories* and emerged from the area of cognitive psychology [38].

In robotics however, and specifically in the area of manipulation, the goal is to enable an embodied, cognitive agent to interact with these objects. In this case, objects in one category should share common affordances [18]. More specifically, this means that they should also be graspable in a similar way. The difficulty then is to find a representation that can encode this common affordance and is grounded in the embodiment and cognitive capabilities of the agent.

Our approach, and also the methods that are going to be mentioned in the following, try to learn from experience how different objects can be grasped given different representations. This is different from the above mentioned systems in which unknown objects are grasped. There the difficulty lies in finding appropriate rules and heuristics. In the following, we will present related work that tackle the grasping of familiar objects and specifically focus on the applied representations.

2.3.1. Based on 3D Data

First of all, there are approaches that rely on 3D data only. El-Khoury and Sahbani [39] for example segment a given point cloud into parts and approximate each part by a superquadric. Their parameters are then fed into an artificial neural net (ANN) in order to classify it as prehensile or not. The ANN has been trained beforehand on labelled SQs. If one of the object parts is chosen as the *handle* a method for determining an n-fingered force-closure grasp is applied on the 3D mesh model of this object part. Pelossof et al. [40] instead directly use a single SQ to find a suitable grasp configuration for a Barrett hand consisting of the approach vector, wrist orientation and finger spread. The experience for doing that is provided by an SVM. The training data consisted of feature vectors containing the parameters of the SQ and of the grasp configuration. They were labelled with a scalar estimating the grasp quality. When feeding the SVM only with the shape parameters of the SQ, their algorithm searches efficiently through the grasp configuration space for parameters that maximise the grasp quality. Curtis and Xiao [41] build upon a database of 3D object annotated with the best grasps that can be applied to them. To infer a good grasp for a new object, very basic shape features, e.g., the aspect ration of the object's bounding box, are extracted to classify it as similar to an object in the database. The assumption made in this approach is that similarly shaped objects can be grasped in a similar way.

2.3.2. Based on 2D Data

A commonality of all these approaches is that they are all done only in simulation where accurate and detailed 3D models are available. As mentioned earlier, this assumption is arguable since sensors like laser range finders or stereo cameras will produce noisy, sparse or incomplete models. However, like the method presented in this paper, there are experience based approaches that avoid this difficulty by relying mainly on 2D data. Saxena et al. [16] proposed a system that infers a point at where to grasp an object directly as a function of its image. They apply machine learning to train a grasping point model on labelled synthetic images of a number of different objects. The classification is based on a feature vector containing local appearance cues regarding colour, texture and edges of an image patch in several scales and of its 24 neighbouring patches in the lowest scale. The authors used their system specifically trained for a dishwasher scenario to pick up objects from it and achieved impressive results. However, if more complex goals are considered that require subsequent actions, e.g., pouring something from one container into another, semantic knowledge about the object and about suitable grasps regarding their functionality becomes necessary [3, 4, 42]. Then, to only represent graspable points without the conception of *objectness* [14, 43] is not sufficient.

Another example of a system involving 2D data and grasp experience is presented by [18]. Here, an object is represented by a composition of prehensile parts. These so called *affordance cues* are obtained by observing the interaction of a person with a specific object. Grasp hypotheses for new stimuli are inferred by matching features of that object against a codebook of learnt *affordance cues*

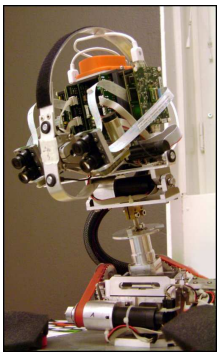
that are stored along with relative object position and scale. However, how exactly to grasp these detected prehensile parts is not yet solved since hand orientation and finger configuration are not inferred from the affordance cues. More successful in terms of the inference of full grasp configurations are Morales et al. [17] who use visual feedback to even predict fingertip positions. The authors also take the hand kinematics into consideration when selecting a number of planar grasp hypothesis directly from 2D object contours. To predict which of these grasps is the most stable one, a KNN-approach is applied in connection with a grasp experience database. However, the approach is restricted to planar objects.

There are also approaches that integrate 2D and 3D information. In [44], two depth sensors are applied to obtain a point cloud of a tabletop scene with several objects. The authors extend their previous work to infer initial grasping point hypothesis. Then, the shape of the point cloud within a sphere centred around an hypothesis is analysed with respect to hand kinematics. This enhances the prediction of a stable grasp and also allows for the inference of grasp parameters like approach vector and finger spread. In their earlier work [16], only downward or outward grasp where possible with the manipulators in a fixed pinch grasp configuration. Speth et al. [23] showed that their earlier 2D based approach [17] is also applicable when considering 3D object. The camera is used to explore the object to retrieve crucial information like height, 3D position and pose. However, all this additional information is not applied in the inference and final selection of a suitable grasp configuration. Each of the above mentioned approaches [16, 23] is actually still mainly relying on 2D information.

3. Grasping an Object Based on its Two-Dimensional Shape

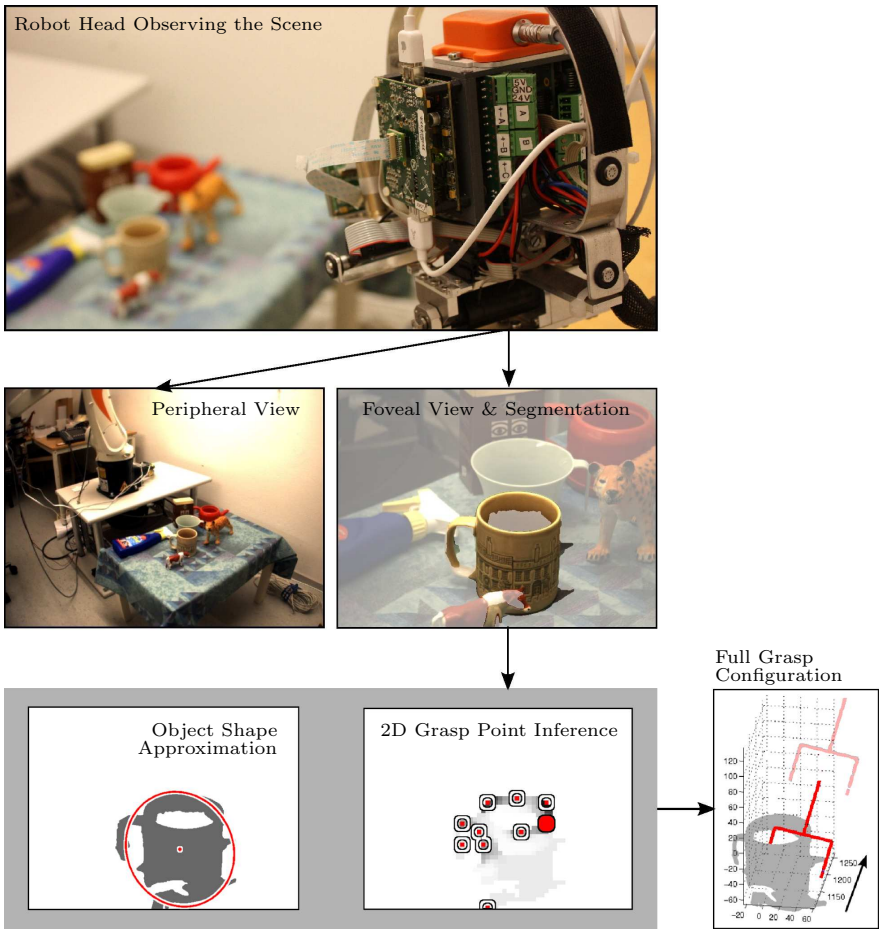
Our approach is based on the hypothesis that visual attributes of objects afford specific actions. The action considered here is a stable grasp of an object. The visual attribute is the object's shape context calculated based on its contour in a monocular image. We assume that we can apply grasping experience gathered from a set of known objects to grasp yet unknown objects that have similar shaped prehensile parts. Being able to grasp cup handles, the robot will also be able to grasp similar shaped but novel objects, e.g., teapots or even briefcases at their handles. To that end, we use a supervised learning technique that provides the robot with that sort of experience from a database of synthetic images.

In our approach, we use a stereo image pair to perform scene segmentation resulting in hypotheses of several objects. Shape context is then computed on each of the hypotheses. Further, 2D points are determined at which each of the hypotheses can be grasped. The model for this is computed beforehand through offline training on an image database. The points in the left and in the right image are associated to each other to infer a 3D grasping point via triangulation. In parallel to the grasping point detection, the segments are analysed in terms of rough object pose. By integrating the 3D grasping point with this pose, a full grasp configuration can be determined and then executed. In the following



(a) Armar Stereo Head [31]

(b) Kuka Arm [45] and Barrett Hand [46]



(c) Flow Chart

Figure 1: Components of the Stereo Vision based Grasp Inference System

(a) Segmentation based on zero disparities only

(b) Segmentation based additionally on a table plane assumption

(c) Segmentation based additionally on table plane and hue assumption

Figure 2: Segmentation results for different segmentation techniques on scenes with different levels of difficulty. 1st column) One textured object. 2nd column) Cluttered table plane. 3rd column) Non-textured Object. 4th column) Two similarly coloured objects. 5th column) Occlusion.

sections, the individual steps of the system are explained in more detail. A detailed flow chart of the whole system is given in Figure 1 along with the used hardware.

3.1. Segmentation of the Object

The system starts by performing the *figure-ground segmentation*. It general, this is a very challenging task and is not considered solved yet in the computer vision area. Recently, a lot of successful approaches developed in this community achieved remarkable results by interleaving object recognition and segmentation [36, 47, 32]. Here, the recognition of a specific object helps the segmentation and vice versa. However, an appearance model of the object has to be learnt beforehand.

In our case, we have no knowledge about *what* the object actually is. Our task here is to model the grasping process, that is, model and represent *how* each of the generated object hypotheses should be grasped. We approach this problem through reasoning on what constitutes an object in a scene.

3.1.1. Zero-Disparity

The advantage of using an active stereo head lies in its capability to fixate on certain objects of interest. A system that implements this attentional mechanism has been presented by Rasolzadeh et al. [48]. Once the system is in fixation, zero-disparities can be employed as a cue for figure-ground segmentation through different segmentation techniques such as for example watersheding

as it has been shown by Björkman and Eklundh [49]. The assumption made is that continuity in depth points towards a coherent object. However, in Figure 2 it can be observed that the ground on which the object in fixation stands is usually also classified as foreground.

3.1.2. Planar Surfaces

The environment in which we expect service robots to perform are dominated by surfaces that are parallel to the ground. In order to overcome the previously mentioned segmentation problem, we can include the assumption that a dominant plane is present in the scene. In our examples, this plane represents the table plane objects are placed on. For that purpose, we fit a planar surface to the disparity image. The probability for each pixel in the disparity image to belong to that plane or not depends on its distance to the most likely plane. In that way, objects standing out of a plane are well segmented. Problems can arise with non-textured objects when the disparity image has large hollow regions. When the table plane assumption is violated or seriously challenged through, e.g., clutter, the segmentation of the fixated object becomes less stable. Examples for these cases are depicted in Figure 2.

3.1.3. Uniform Texture and Colour

An additional assumption that can be introduced into the system is that objects are usually either uniformly coloured or textured. By introducing this cue in conjunction with the table plane assumption, we can stabilise the figure-ground segmentation. The probability that a specific hue indicates a foreground object depends on the foreground probability (including the table plane assumption) of pixels in which it occurs. This holds equivalently for the background probability of this hue. The colour cue contributes to the overall estimate with the likelihood ratio between foreground and background probability of the hue. In this way, we can overcome disparity holes and instabilities due to an uncertain table plane detection. This can be observed in Figure 2. However, problems can arise when a background object has a similar texture or colour as the foreground object.

From looking at the example results of the different segmentation methods, we can conclude that in a real world scenario, we will be able to obtain a reasonable hypothesis of where an object of interest in the image is. However, our representation has to be able to cope with some amount of clutter. In Section 4, we will therefore evaluate our approach assuming that the assumption of perfect segmentation holds. In Section 4.3, we will analyse how well our method performs given different qualities of segmentation.

3.2. Representing Relative Shape

In this section, we propose a representation that fulfils the mentioned requirements for an object representation: it is rich enough to infer necessary grasp parameters and can be extracted from real world sensors. We assume that the object is segmented in the image. Intuitively seen, how to grasp an

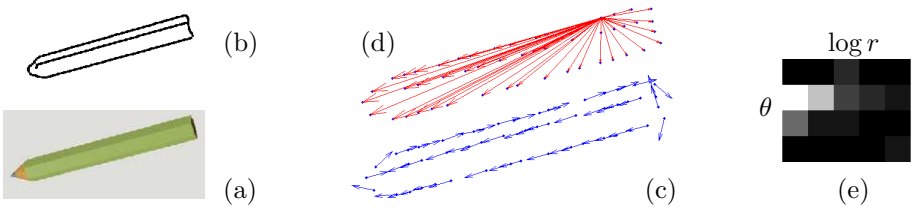


Figure 3: Example of deriving the shape context descriptor for the image of a pencil. (a) Input image of the pencil. (b) Contour of the pencil derived with the Canny operator. (c) Sampled points of the contour with gradients. (d) All vectors from one point to all other sample points. (e) Histogram with four angle and five log-radius bins comprising the vectors depicted in (d).

object depends to a large extent on its global shape. However, we need a local descriptor that relates this global property to each single point on the object. Consider for example elongated objects such as pens or screwdrivers. Most people would grasp them in their middle, roughly at the centre of mass. The shape *seen* from this point is approximately symmetric. In contrast to that, the shape *seen* from a point at one of the ends of the object is highly asymmetric. This particular example shows that relative shape can differentiate between good and bad grasping points. Our idea on how to exploit this object attribute is to apply the concept of shape context that was up till now mainly used for shape matching and object recognition. In the following, we will briefly summarise the main ideas of shape context. For a more elaborate description, we refer to [20].

The basis for the computation of shape context is an edge image of the object. N samples are taken with a uniform distribution from the contour. Whether these points lie on the inner or outer contour is of no importance. For each point we consider the vectors that lead to all the other sample points. These vectors relate the global shape of the object to the considered reference point. To comprise this information into a compact descriptor for each point, we create a two dimensional histogram with angle and radius bins. In [20] it is proposed to use a log-polar coordinate system in order to emphasise the influence of nearby samples. An example for this whole process is shown in Figure 3.

A big advantage of shape context is that it is invariant to a number of transformations. Invariance to translation is intrinsic since both the angle and the radius values are determined relative to points on the object. To achieve scale invariance, [20] proposed to normalise all radial distances by the median distance between all N^2 point pairs in the shape. Also rotation invariance can be easily achieved by measuring the angles relative to the gradient of the sample points. In the following, we will describe how to apply the *relative shape* representation to form a feature vector that can later be classified as either graspable or not.

3.2.1. Contour Detection

In the segmented image, we compute the contour of the object by applying the Canny edge detector. This raw output is then filtered to remove spurious

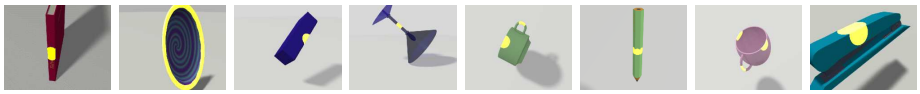


Figure 4: One example picture for each of the eight object classes used for training along with their grasp labels (in yellow). Depicted are a book, a cereal bowl, a white board eraser, a martini glass, a cup, a pencil, a mug and a stapler. The database is adopted from Saxena et al. [16].

edge segments that are either too short or have a very high curvature.

3.2.2. Feature Vector

The result serves as the input for computing shape context as described above. Our goal is to infer a grasping point in the image. We do not consider single contour points but instead subdivide the image into rectangular patches (in our case 10×10 pixels). A descriptor for each patch serves as the basis to decide whether it is a grasping point or not. This descriptor is simply composed of the accumulated histograms of all sample points on the object's contour that lie in that patch. Typically only few sample points will be in a 10×10 pixel wide window. This turned out not to be sufficient for the classification task. We therefore calculated the accumulated histograms in three different spacial scales centred at the current patch and concatenated them to form the final feature descriptor of dimension 120.

3.3. Training Database

Saxena et al. [16] developed a database containing synthetic images of eight different object classes that are depicted along with their grasp labels in Figure 4. Synthetic in this case means that a ray tracer was used to render images of different object models along with the correct grasp labels. Additionally, lighting conditions, object attributes (like colour, texture and scale), camera positions and orientations can be easily varied automatically. Compared to the collection and manual labelling of real images, this method is very elegant considering aspects like time consumption, false labelling and diversity. The database contains almost 12000 images. Due to the mentioned advantages of the database and in order to be able to compare our system with the one of Saxena et al. [16], we train our classifier with this database.

However, the database also has a disadvantage. The way the labels are chosen for the different objects is not always consistent. The following examples can be observed in Figure 4. A cup for example is labelled at two specific point on its rim, in practise however you can grasp a cup on the whole rim. The white board eraser is quite a symmetric objects. Neither the local appearance nor the relative shape of its ground truth grasping point are discriminative descriptors. This ambiguity in the labelling will affect both classifiers, although the one based on global shape will due to its enhanced discriminativity be able to cope with it better.

3.4. Classification of 2D Grasping Points

The goal of the presented approach is to identify a point of an object at which it can be grasped. An image of that object serves as input data. In our case, we consider input patches and classify them based on the object's shape as either good grasping points or not. This decision is made based on experience obtained during the training of the classifier. We use a supervised approach, meaning that a labelled database is applied (see previous section). In this paper, we examine two different classification methods: a linear one (logistic regression) and a non-linear one (SVMs). In the following we will briefly describe their concepts. For a more in depth theory we refer to [50].

Let g_i denote the binary variable for the ith image patch in the image. It can either carry the value 1 or 0 for being a grasping point or not. The posterior probability for the former case will be denoted as $P(g_i = 1|D_i)$ where D_i is the feature descriptor of the ith image patch. For logistic regression, this probability is modelled as the sigmoid of a linear function of the feature descriptor:

$$P(g_i = 1|D_i) = \frac{1}{1 + e^{-wD_i}} \tag{1}$$

where w is the weight vector of the linear model. These weights are estimated by maximum likelihood:

$$w = \arg\max_{w'} \prod_i P(g_i = 1|D_i, w') \tag{2}$$

where here g_i and D_i are the labels and feature descriptors of our training data, respectively. Logistic regression produces a linear decision function in feature space. In case our data is not linearly separable, we will have to deal with a classification error whose magnitude depends on the actual distribution of the sample points. In order to minimise this error, we would like to have a non-linear decision function instead.

SVMs can produce arbitrary decision functions in feature space by still doing a linear separation but in a higher dimensional space. The mapping of the input data into that space is accomplished by a non-linear kernel function K. In order to obtain the model for the decision function when applying SVMs, we need to solve the following optimisation problem:

$$\max \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j g_i g_j K(D_i, D_j) \tag{3}$$

subject to $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i g_i = 0$ with the solution $w = \sum_i^{N_s} \alpha_i g_i D_i$. As a kernel we have chosen a *Radial Basis Function* (RBF):

$$K(D_i, D_j) = e^{-\gamma ||D_i - D_j||^2}, \gamma > 0 \text{ and } \gamma = \frac{1}{2\sigma^2} \tag{4}$$

The two parameters C and σ are determined by a grid search over parameter space.

3.5. Approximating the Object's Pose

As a manipulator we are considering a three-fingered Barrett hand [46] in a pinch grasp configuration where the two fingers are in parallel and opposing the thumb. Our goal is to approach a 3D grasping point with the palm of the hand in a specific wrist orientation. Given a 2D grasping point in the left image of the stereo camera, we can determine its 3D position if we also know its position in the right image. This will be described in more detail in the next paragraph.

In order to infer the orientation of the Barrett hand we have to at least roughly estimate the pose of the given unknown object. The question remains how we can derive this from stereo images without relying on 3D reconstruction. According to Cuijpers et al. [21], humans grasp a cylindric object highly dependent on the position of the major and minor axes of its cross section provided that a pinch grasp (grasp with index finger and thumb) is applied.

Here, we generalise this approach to arbitrarily shaped objects by fitting an ellipse to the segmented object in the image plane. The major and minor axis of this ellipse in 2D serve as the basis to obtain a rough approximation of the three dimensional object pose. For this purpose, we detect three points in the left image: the centroid of the segment, an object point on the major axis and an object point on the minor axis. Via stereo matching we can find the corresponding points in the right image and thus obtain three 3D points that define a tilted plane. The objects pose is then associated with the three dimensional position of its segment centroid and the orientation of the plane.

The assumption we make is that a single plane can in general roughly approximate the orientation of an object. This indeed holds for a lot of objects like boxes, various elongated objects or even rather irregularly shaped toys as shown in Figure 4. It does however not hold for cylindrical objects like bottles, cups or cans. As we will see later on, this is not crucial for the way we select a grasp configuration. Consider for example the grasp of a can from the top. Due to the symmetry of the object any wrist orientation of the Barrett hand will result in a valid grasp.

3.6. Generation of Grasp Hypotheses

In the following, we describe the integration of the 2D grasping points with the objects pose approximation for inferring the full grasp configuration.

3.6.1. 3D Grasping Point

After we run the classifier on each image of the stereo image pair, we have to associate the resulting 2D grasping hypotheses to each other in order to obtain a 3D point via triangulation. For this purpose we create a set $B_l = \{b_{(i,l)} | i = 1 \cdots m\}$ of m image patches i in the left image that are local maxima regarding the classifier response $P(g_i = 1|D_i)$ and whose adjacent patches in the 8-neighbourhood carry values close to that of the centre patch. We apply stereo matching to obtain the corresponding patches $B_r = \{b_{(i,r)} | i = 1 \cdots m\}$ in the right image. Let $P(b_{(i,l)}|D_{(i,l)})$ and $P(b_{(i,r)}|D_{(i,r)})$ be the probability

for each image patch in set B_l and B_r to be a grasping point given the respective feature descriptors $D_{(i,l)}$ or $D_{(i,r)}$. Assuming naive Bayesian independence between corresponding patches in the left and right image, the probability $P(b_i|D_{(i,l)}, D_{(i,r)})$ for a 3D point b_i to be a grasping point is determined by

$$P(b_{(i,l)}|D_{(i,l)}, D_{(i,r)}) = P(b_{(i,l)}|D_{(i,l)}) * P(b_{(i,r)}|D_{(i,r)}). \tag{5}$$

According to this measure, we can rank the 3D grasping points. The best patch is then

$$b = \arg\max_i P(b_{(i,l)}|D_{(i,l)}, D_{(i,r)}). \tag{6}$$

3.6.2. Orientation of the Barrett Hand

Given a 3D grasping point and an object pose, we define three possibilities to choose the approach vector:

(i) vector a_{ma} defined by the major axis of the ellipse in 3D,
(ii) vector a_{mi} defined by the minor axis in 3D or
(iii) normal vector n_p of the plane p spanned by these two vectors.

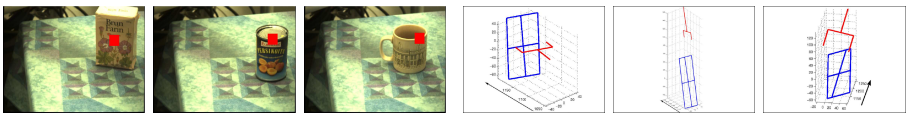
Which of them is chosen depends on the position of the 2D grasping point within the 2D ellipse. Let $x_{b_{(i,l)}}$ be the vector defined from the grasping point $b_{(i,l)}$ to the centre of mass c_l of the segment in the left image. Let x_{mi} be the vector from c_l to the point on the minor axis of the 2D ellipse lying on the segment boundary. Let x_{ma} be the vector defined equivalently for the major axis. Let ϕ be a given threshold for the distance between $b_{(i,l)}$ and c_l. If $|x_{b_{(i,l)}}| < \phi$ then the hand will approach the grasping point b_i with a vector n_p. The wrist orientation will be determined by aligning the vector between the thumb and two fingers with a_{mi}. If

$$\frac{|x_{ma} \cdot x_{b_{(i,l)}}|}{|x_{ma}|} > \frac{|x_{mi} \cdot x_{b_{(i,l)}}|}{|x_{mi}|}, \tag{7}$$

i.e. $x_{b_{(i,l)}}$ is better aligned with x_{ma} than with x_{mi}, a_{ma} will be chosen as approach direction towards b_i. The wrist orientation will be fixed by aligning the vector between the thumb and two fingers with n_p. In case $x_{b_{(i,l)}}$ is better aligned with x_{mi} than with x_{ma}, a_{mi} will be chosen as approach vector. The wrist orientation will be fixed in the same way as for the previous case. Examples of grasp configurations are given in Figure 5.

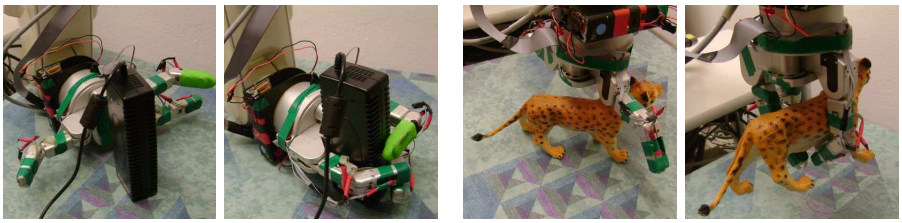
Although these examples show promising results, this method turned out to be not suitable for our specific hardware setting in which the manipulator is situated opposite to the stereo camera system with the objects in between them.

Therefore, for the demonstration on our hardware we used another method with a fixed approach vector either vertically or horizontally towards the grasping point (top or side grasp). Which of them is chosen depends on the orientation of the major axis. If this axis is closer to a horizontal orientation than to a vertical one, a top grasp will be performed. Otherwise a side gasp is applied.



(a) Objects with grasping points (b) Grasp Configurations

Figure 5: Examples for generated grasp configurations. (a) Right image of the stereo camera with grasp point labelled. (b) Related grasp configuration with a schematic gripper and the plane with the axes approximating the object pose. the viewing direction is indicated by the arrow.



(a) Side Grasp (b) Top Grasp

Figure 6: An example for the execution of a top grasp and a side grasp on our robotic platform.

The wrist orientation is determined in the same way as for the first method. Figure 6 shows an example for the top and for the side grasp.

For the demonstration of the whole system we used the hardware setup as depicted in Figure 1(a) and 1(b): a 6 DoF KUKA robotic arm [45], a three-fingered 4 DoF Barrett Hand [46] and the 7 DoF Armar Head [31].

4. Experimental Evaluation

In this section, we present the results of several experiments. In the first part, we use synthetic images and compare our method with the state of the art method by [16] that applies local appearance cues such as colour, texture and edges. In the second part, we are conducting a more thorough analysis of the different grasping models to understand what they exactly encode. Finally, we consider real data to investigate the applicability of our methods in real settings. Additionally, we evaluate the application of logistic regression and SVMs.

4.1. Evaluation on Synthetic Images

In this section, we focus on the question how our descriptor and the one by [16] perform when being trained on different sets of synthetic objects. We are especially interested in how well the classifiers generalise over different types of grasps facing varying global shape or local appearance of the grasping points. For this purpose we applied four different sets of objects to train the classifiers.

- *Pencils* are grasped at their centre of mass.

- *Mugs & cups* have a handle at which to pick them up. They only differ slightly in global shape and local grasping point appearance.
- *Pencils, white board erasers & martini glasses* are all grasped approximately at their centre of mass at two parallel straight edges. However, their global shape and local appearance differ very much.
- *Pencils & mugs* are grasped in very different ways and also look very different.

We divided each set in a training and test set. On the training sets we trained four different classifiers.

- *Shape context & SVM* (SCSVM). We employed twelve angle and five log radius bins for the shape context histogram. We sample the contour with 300 points. The same parameters were applied by [20] and have proven to perform well for grasping point detection.
- *Local appearance features & logistic regression* (OrigLog) is the classifier by [16].
- *Local appearance features & SVM* (OrigSVM) applies an SVM instead of logistic regression.
- *Shape context, local appearance features & SVM* (SCOrigSVM) integrates shape context features with local appearance cues. The resulting feature vector is used to train an SVM.

4.1.1. Accuracy

Each model was evaluated on the respective test sets. The results can be observed in form of ROC curves in Figure 7 and as accuracy values[1] in Table 1. The first general observation is that the usage of an SVM is, as expected, advantageous over logistic regression. On average, the classification performance for each set of objects rose about 7.65% when comparing OrigSVM with OrigLog. A second general observation is that classifiers that employ global shape (either integrated or not integrated with appearance cues) have the best classification performance for all training sets. In the following we will discuss the results for each set.

- *Pencils*. The local appearance of a pencil does not vary a lot when looking at different positions on its surface whereas relative shape does. Therefore, local appearance based features cannot be very discriminative. This is confirmed for the models, that are only trained on images of pencils. SCSVM performs slightly better than OrigSVM. The classification performance gets enhanced when applying an integrated feature vector.

[1] Accuracy is defined as the sum of true positives and true negatives over the total number of examples. Table 1 presents the maximum accuracy for a varying threshold.

- *Mugs & Cups*. These objects are grasped at their handle which is characterised by a local structure that is rather constant even when the global shape changes. Thus, OrigSVM outperforms slightly the classifier that applies shape context only. However, an integration of both features leads to an even better performance.

- *Pencils, white board erasers & martini glasses*. For this set of objects the position of the grasp is very similar when considering their global shape whereas the local appearance of the grasping points differs a lot. Also here, the models based on shape context performs best. An integration of the different kinds of features degrades the performance.

- *Pencils & mugs*. The performance of the different classifiers for the previous set of objects is a first indication for a weaker generalisation capability of OrigSVM and OrigLog over varying local appearance compared to SCSVM and SCOrigSVM. This is further confirmed for the last set where not just local appearance but also global shape changes a lot. SCSVM improves the performance of OrigSVM about 6.75% even though the grasping points are very different when related to global object shape. Integrating both kinds of features increases the performance only slightly.

4.1.2. Stability

Our goal is to make a robot grasp arbitrary and novel objects in a stable way. This means that we are for practical purposes interested in whether the *best* grasping point hypotheses correspond to object points that in reality afford a stable grasp. Thus, our second experiment evaluates whether the best hypothesis is located on grasping point labels in our image database or at least close to them. We constructed a set of 80 pictures from the synthetic image database with ten randomly selected pictures of each of the eight object classes. Thus, also novel objects that were not used for training are considered. On every image we run all the aforementioned models and for each one picked out the best ten grasping points b_i. In the database, a label is not a single point, but actually covers a certain area. We evaluated the Euclidean distance d_i of each of the ten grasping points measured from the border of this ground truth label at position p_j and normalised with respect to the length l_j of its major axis. In that way, the distance is dependent on the scale of the object in the image. In case there is more than one label in the image, we choose the one with the minimum distance. If a point b_i lies directly on the label, the distance $d_i = 0$. If a point lies outside of the label, the distance d_i gets weighted with a Gaussian function ($\sigma = 1, \mu = 0$) multiplied with $\sqrt{2\pi}$. The number of hits h_m of each model m on the picture set is counted as follow:

$$\begin{aligned} h_m &= \sum_{k=1}^{K} \sum_{i=1}^{N_k} e^{-\frac{d_{(i,k)}^2}{2}} \\ \text{with } d_{(i,k)} &= \min_{j=1}^{M_k} \frac{dist(b_{(i,k)}, p_{(j,k)})}{2l_{(j,k)}} \end{aligned}$$

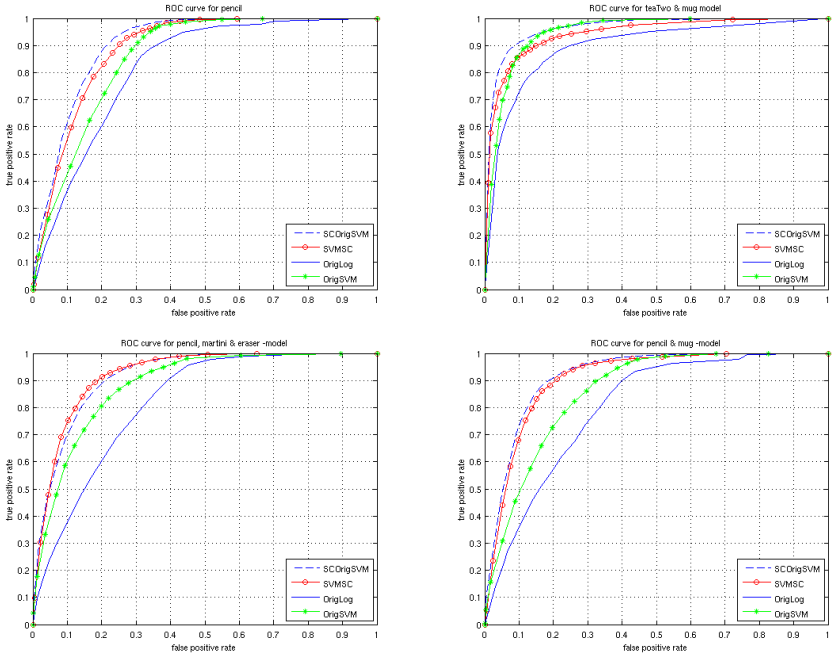


Figure 7: ROC curves for models trained on different objects.

where K is the number of image in the set, M_k is the number of grasp labels in that picture and N_k is the number of detected grasping points. Grasping points whose distance d_i exceed a value of $3 * \sigma$ are considered as outliers. In Figure 8 the number of hits, that is, the amount of good grasps for each model are depicted.

Apart from the model trained on cups and mugs, the SVM trained only on shape context features performs always best. The performance drop for the second object set can be explained in the same way as in the previous chapter: handles have a very distinctive local appearance and are therefore easily detected with features that capture this. In general, this result indicates that classifiers based on shape context detect grasping points in a more stable manner. This is particularly important for the inference of 3D grasping point that rely on the assumption that 2D grasping in different viewpoint of the same object correspond to each other.

4.1.3. Summary of Results

To recapitulate this experiment section, we can draw several conclusion for the case of synthetic images. First of all, independent of which feature representation is chosen, SVM outperforms logistic regression. Secondly, our simple and compact feature descriptor that encodes relative object shape improves the detection of grasping points both in accuracy and stability in most cases. In case of

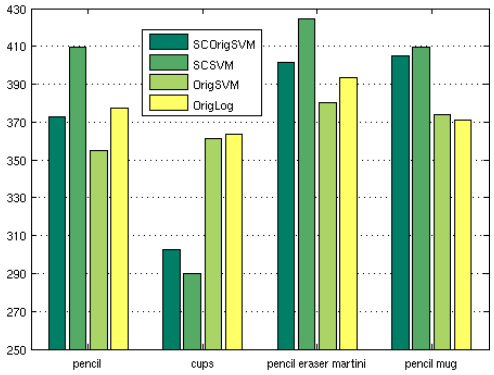


Figure 8: Evaluation of the best ten grasping points of each model on a picture test set containing in total 80 pictures of *familiar* and novel objects (see Figure 4).

Table 1: Accuracy of the models trained on different objects.

	SCOrigSVM	SCSVM	OrigSVM	OrigLog
Pencil	**84.45%**	82.55%	70,70%	77.07%
Cup & Mug	**90.71%**	88.01%	88.67%	83.85%
Pencil, Martini & Eraser	84.38%	**85.65%**	80.79%	74.92%
Pencil & Mug	**85.71%**	84.64%	77.80%	74.32%

very distinct local features, both representations are comparable. And last but not least, integrating the two representations into a single feature vector leads only to slight improvements or even decreases the classification performance.

4.2. Opening the Black Box

In the previous section, we presented evidence that the shape context based models detect grasping points more accurately than the models trained on local appearance features. As argued in Section 3, we see relative shape as a better cue for graspability than local appearance. In this section, we would like to confirm this intuition by analysing what it actually is that the different grasping point models encode. We conduct this analysis by applying the Trepan Algorithm by Craven and Shavlik [51] to the learnt classifiers. This algorithm builds a decision tree that approximates a concept represented by a given *black box* classifier. Although originally proposed for neural networks, Martens et al. [52] showed that it is also applicable for SVMs.

We apply the same sets as mentioned in the previous section. The extracted trees are binary with leafs that are classifying feature vectors as either graspable or non-graspable. The decisions at the non-leaf nodes are made based on either

one or more components of the feature vector. We consider each *positive* leaf node as encoding a prototypical visual feature that indicates graspability.

As previously mentioned, the extracted trees are only approximations of the actual learnt models. Thus, the feature vectors that end up at a specific leaf of the tree will be of three different kinds.

- *Ground truth.* Features that are graspable according to the ground truth labels in the database.
- *False positives by model.* Features that are not graspable according to the labels but are so according to the classifier.
- *False positives by tree.* Features that are neither labelled in the database nor classified by the model to be graspable, but are considered to be so by the tree.

We will analyse these samples separately and also rate the trees by stating their *fidelity* and *accuracy*. *Fidelity* is a measure of how well the extracted trees approximate the considered models. It states the amount of features vectors whose classification is compliant with the classification of the approximated model. *Accuracy* measures how good the classification rate for either the tree or the model is when run on a test set.

The analysis of these samples is conducted through PCA. The resulting eigenvectors form an orthonormal basis with the first eigenvector representing the direction of the highest variance, the second one the direction with the second largest variance, etc. In the following sections we will only visualise those eigenvectors whose *energy* is above a certain threshold and at maximum ten of these. The *energy* e_i of an eigenvector i is defined as

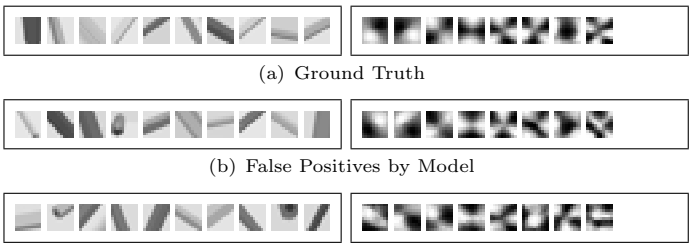
$$e_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{j=1}^{k} \lambda_j} \tag{8}$$

where λ_j is the eigenvalues of eigenvector j with k eigenvectors in total. As a threshold we use $\theta = 0.9$.

The remainder of this section is structured as follows. In Section 4.2.1, we will visualise the prototypical features for the local appearance method by applying PCA to the samples at positive nodes. In Section 4.2.1 we will do the same for the relative shape based representation.

4.2.1. Local Appearance Features

Saxena et al. [16] applied a filter bank to 10×10 pixel patches in three spatial scales. The filter bank contains edge, texture (Law's masks) and colour filters. In this section, we will depict samples of these 10×10 pixel patches in the largest scale. They will be taken from every positive node of each tree that is trained for a specific object set. All feature vectors that ended up at one of these positive nodes are used as an input to PCA in order to visualise prototypical visual features that indicate graspability.



(a) Ground Truth

(b) False Positives by Model

(c) False Positives by Tree

Figure 9: Pencils: Ten samples and PCA components of the positive node of the decision tree.

The first set we are looking at is the set that only consists of pictures from a pencil (see Figure 4) that is labelled in its centre of mass. The tree that is built by using the according grasping point model is surprisingly shallow. It has only four leaf nodes of which one is positive. The decisions on the non-leaf nodes are made based on the output of the texture filters only. Neither colour nor edge information are considered. This means that this part of the feature vector is not necessary to achieve a classification performance of 75.41% (see Table 2). Ten random samples from the positive node are shown in Figure 9 subdivided dependent on whether they are graspable according to the ground truth labels from the database or only according to the model and tree, respectively.

In order to visualise to which visual cues this grasping point models actually respond, we run PCA on the set of feature vectors that ended up at that node. The resulting principal components selected according to Equation 8 are also depicted in Figure 9. Encoded are close-ups of the body of the pencil and perspective distortions.

However, the problem with this is that basically the almost the whole length of the pencil complies with these components. Due to that, the samples from the set of false positives by the model are very similar to the ground truth samples. The appearance of the centre of mass of the pencil is not that different from the rest of the pencil. This is further clarified by Figure 10 where the false positives by the model and tree are projected into the space spanned by the first three principal components from the ground truth grasping points. They are basically strongly overlapping. We will show later that given our relative shape based representation these three first principal components are already enough to define a space in which graspable points can be better separated from non-graspable points.

For the other sets of objects we applied the same procedure. The principal components of the samples at each positive node are shown in Figure 11, 12 and 13.

In Table 2, the fidelity of the respective trees in relation to the model and their accuracies are given.

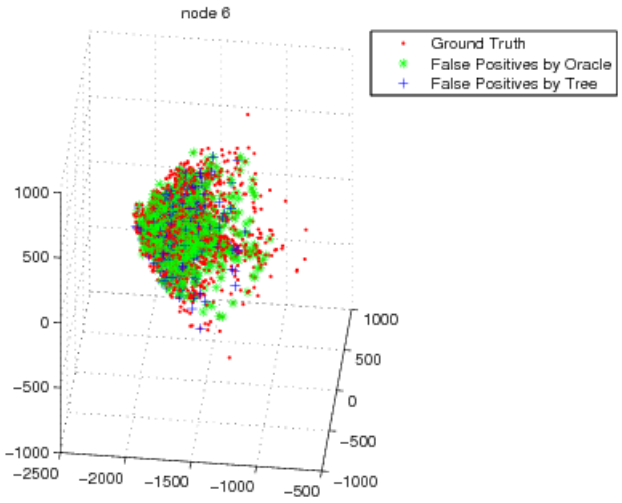
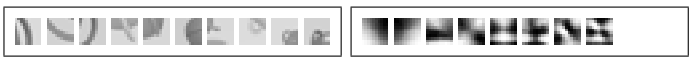


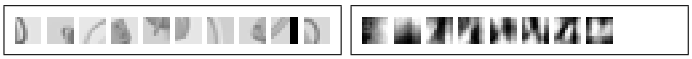
Figure 10: Pencils: Feature vectors projected into the three dimensional space spanned by the three eigenvectors of the sample set of true grasping points with the highest variance.

Table 2: Accuracy of the models trained on different objects given the local appearance representation.

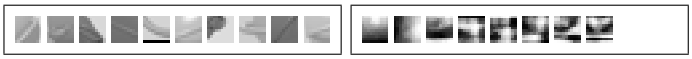
	Pencil	Cups	Elongated	Pencil & Mug
Fidelity	86.78%	83.97%	87.55%	89.29%
Accuracy Tree	75.41%	82.61%	72.12%	73.30%
Accuracy Model	77.07%	83.85%	74.92%	74.32%



(a) First Node

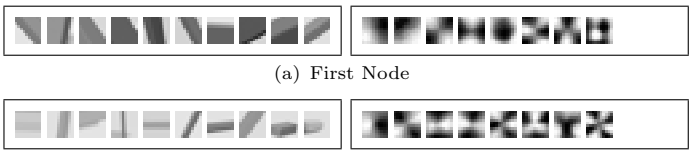


(b) Second Node



(c) Third Node

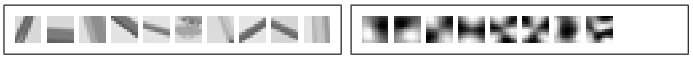
Figure 11: Cups: Ten samples and PCA components for each of the positive nodes of the decision tree.



(a) First Node

(b) Second Node

Figure 12: Elongated Objects: Ten samples and PCA components for each of the positive nodes of the decision tree.



(a) First Node

Figure 13: Pencils and Mugs: Ten samples and PCA components for the positive node of the decision tree.

4.2.2. Relative Shape

In this section we will evaluate the models that are also trained on the different objects sets but are applying a different representation of the visual data. The cue that we are using is *relative shape*, i.e., how does the global shape of an object look like relative to a graspable point.

We use shape context for this approach that is invariant to translation, rotation and scale. In order to account for that before applying PCA to the feature vectors at the different positive nodes of the induced decision trees, we pre-process each picture in the test sets. In detail this means that we are

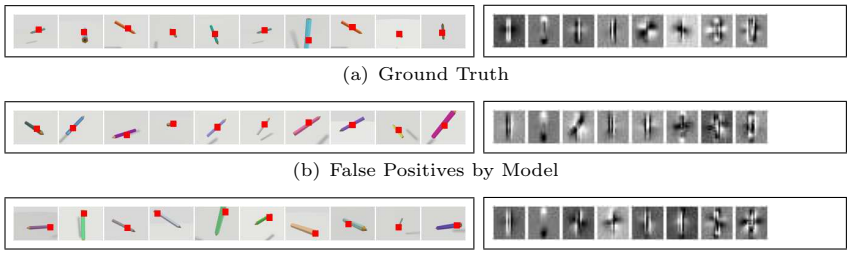
(i) extracting the contour with the Canny edge detector,
(ii) filtering out spurious edge segments,
(iii) subsampling the contour,
(iv) normalising the sampled contour with the median distance between contour points,
(v) rotating the whole contour according to the average tangent directions of all the contour points falling into the patch that is currently considered by the classifier
(vi) and finally plotting the resulting contour on a $20x20$ pixels patch with the grasping point in the centre.

The output of this procedure forms the input for PCA. The sample feature vectors for each node are depicted not as patches but as red squared labels located at the grasping point on the object.

Each of the induced trees in this section is of a slightly worse quality in terms of fidelity when compared with the trees obtained from the logistic regression method (see Table 2). This is probably due to worse performance of the Trepan algorithm when approximating SVMs. Nevertheless, for the purpose of this section that is targeted at the visualisation of prototypical grasping point features rather than impeccable classification, this performance is still acceptable. The results for the trees induced in the following are given in Table 3.

Table 3: Accuracy of the models trained on different objects given the relative shape representation.

	Pencil	Cups	Elongated	Pencil & Mug
Fidelity	78.97%	79.66%	78.79%	80.82%
Accuracy Tree	71.38%	76.89%	73.40%	73.41%
Accuracy Model	82.55%	88.01%	85.56%	84.64%



(a) Ground Truth

(b) False Positives by Model

(c) False Positives by Tree

Figure 14: Pencil: Ten samples and PCA components of the positive node of the decision tree.

We will again start by analysing the model trained on the set of pencils. The induced decision tree has one positive node. The samples from this node are depicted in Figure 14 along with the most relevant PCA components to which we will refer in the remainder of this paper as *eigencontours*. These components do not encode the local appearance but clearly the symmetric relative shape of the grasping point.

What is even more remarkable is that the feature vectors projected into the space spanned by the three best principal components of the ground truth samples are quite well separable, even with a linear decision boundary. There is almost no overlap between false positives produced by the tree and the ground truth features and quite few overlap between false positives produced by the models and the true graspable features. This result is depicted in Figure 14.

We applied the same procedure to the models trained on the other sets of objects. The eigencontours for these are depicted in 17, 18 and 19. For the sets consisting of different objects, each positive node in the decision tree is mainly associated with one of the objects and encodes where they are graspable.

Furthermore, we can observe a better separability compared to the models trained on local appearance. In order to quantify this observation, we analysed the distribution of the samples in the three-dimensional PCA space in terms of linear separability. As measures for that we employed Fisher's discriminant ratio and the volume of the overlap regions. In Figure 16 a comparative plot of these two measures for all the models considered in this section is depicted.

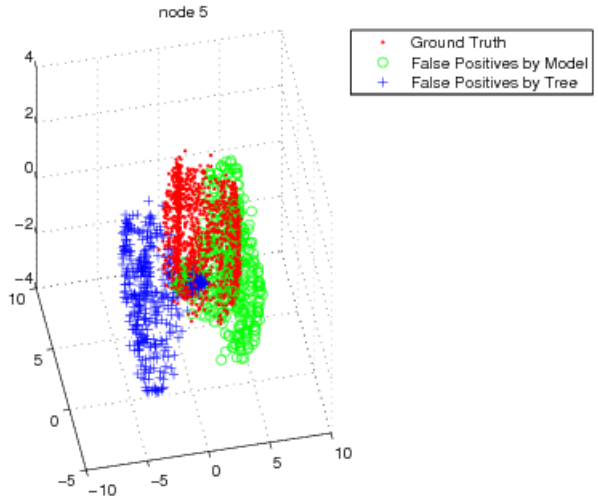
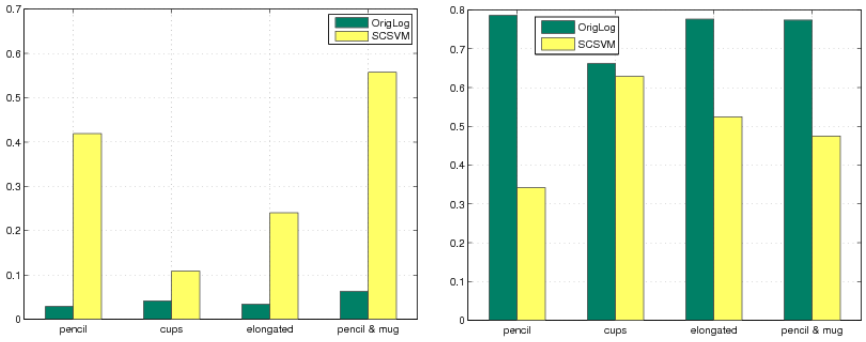


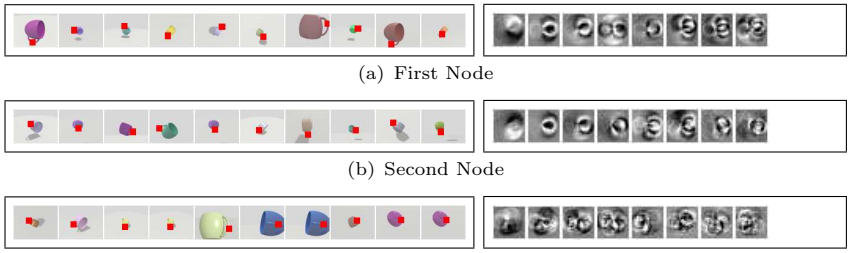
Figure 15: Pencil: Feature vectors projected into the three dimensional space spanned by the three eigenvectors of the sample set of true grasping points with the highest variance.



(a) Fisher's Discriminant Ratio

(b) Volume of Overlap Region

Figure 16: Measures of linear separability for models trained on different training sets and with different classification methods.

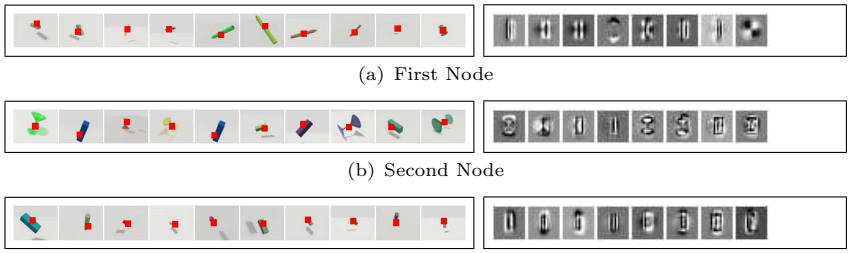


(a) First Node

(b) Second Node

(c) Third Node

Figure 17: Cups: Ten samples and PCA components for each of the positive nodes of the decision tree.

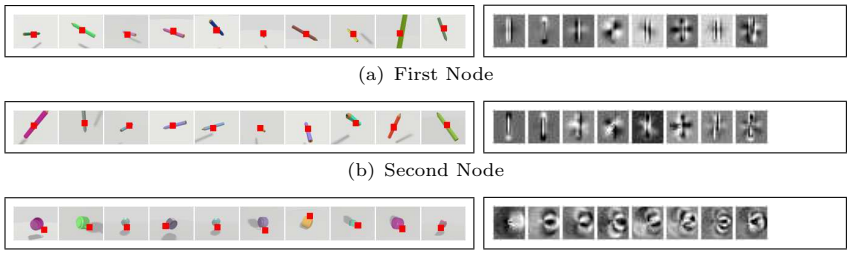


(a) First Node

(b) Second Node

(c) Third Node

Figure 18: Elongated: Ten samples and PCA components for each of the positive nodes of the decision tree.



(a) First Node

(b) Second Node

(c) Third Node

Figure 19: Pencils and Mugs: Ten samples and PCA components of the first positive node of the decision tree.

4.2.3. Summary of Results

In this section, we gained some insights into what a grasping point model that is based on different representations actually encodes. We could observe that our compact feature descriptor based on relative shape is more discriminative in terms of grasping points than a state of the art feature descriptor that combines the output of a filter bank. The dimension of our descriptor is almost four times smaller which also has implications for the time needed to train an SVM. The classification performance achieved with an SVM could even be improved by finding a decision boundary in the space spanned by the first three principal components of a set of ground truth prototypical features. This is considered as future work.

4.3. Evaluation on Real Images

In the previous section, we showed that the performance in grasp point detection of the relative shape based classifier is increased when compared to a method that applies local appearance. In these synthetic images no background clutter was present. However, in a real world scenario this will be the case. Several objects will be visible in the scene; occlusions will occur; tables might carry a lot of texture. How can we make our representation robust against these distractions? One solution to that problem is presented in [16]. The authors demonstrated a system for the scenario of emptying a dishwasher. In order to cope with the visual clutter occurring in such a scenario, the grasping point model was trained on hand labelled images of the dishwasher. Although, this dishwasher was unloaded successfully, for a new scenario the model has to be re-trained to cope with new kind of backgrounds.

We argue that we need a way to cope with backgrounds based on more general assumptions. As described earlier in Section 3.1, our method relies on scene segmentation. The quality of the segmentation is affected by the cues that are integrated and on how the considered environment complies to the assumptions, e.g, dominant plane, uniformity in colour or texture. In this section, we evaluate how the relative shape based representation is affected by different segmentation qualities. For that purpose, we collected images of differently textured and texture-less objects, e.g., boxes, cans, cups, elongated objects, or toys, composed in scenes of different levels of complexity. This ranges from single objects on a table to several objects occluding each other. These scenes were segmented with the three different techniques described in Section 3.1.

Ideally, we would like to achieve two things. First of all, the grasping points that are inferred in two images of the same object given different qualities of segmentation have to correspond to each other. This is important because later on we would like to match grasping points in a stereo image pair to obtain a 3D point.

Secondly, the quality of the inferred grasping points should only be minimally affected by the amount of clutter in the segment. Regarding the latter point, a quantitative evaluation can only be performed by applying the inferred grasps

in practise. Although we showed some real grasps in Section 3.6.2, this is out of the scope of this paper. Instead, we present some representative examples of the grasping point inference methods when applied to different sets of objects.

4.3.1. Examples for Grasping Point Detection

For example in Figure 20, we show the results of the grasping point classification for a red texture-less teapot. The left column shows the segmented input of which the first one is always the ground truth segment. The middle column shows the result of the grasping point classification when applying the local appearance based descriptor by [16] and the right one the results of the classification when using the relative shape based descriptor. The red dots label the detected grasping points. They are the local maxima in the resulting probability distribution. Maximally the ten highest valued local maxima are selected.

In Figure 20, the grasping point classification for the teapot can be observed, first, when it is the only object in the scene and second, when it is partially occluded. Note that the segmentation in the case of local appearance based features is only influencing which patches are considered for the selection of grasping points. In case of the relative shape based descriptor, the segmentation also influences the classification by determining which edge points are included in the shape context representation. Nevertheless, what can be observed is that the detection of grasping points for the representation proposed in this paper is quite stable when facing decreasing quality of segmentation and occlusion. For example in Figure 20(b) (last row), even though there is a second handle now in the segmented region, the rim of the teapot is still detected as graspable and the general resulting grasping point distribution looks similar to the cases in which the handle was not yet in the segment. This means, that the object that is currently in fixation by the vision system, the one that dominates the scene, produces the strongest responses of the grasping point model even in the presence of other graspable objects.

In Figure 21, we applied the models trained on mugs and cups to images of a can and a cup. The descriptor based on local appearance responds very strongly to textured areas whereas the relative shape based descriptor gets not distracted by that since the whole object shape is included in the grasping point inference. Finally in Figure 22, we applied some models to objects that are not very similar to any object that the grasping point models were trained on. In case of the local appearance based descriptor, the grasping point probability is almost uniform and very high valued for both objects. In the case of shape context there are some peaks in the distribution. This suggest that the ability of these models to generalise over different shapes is higher than for local appearance based models.

4.3.2. Stability of the Detection

Earlier we mentioned, that the detected grasping points in images of the same object given different segmentation should ideally correspond to each other. In order to evaluate this, we measured the difference of the detected grasping points

in the differently segmented images. For real images, we do not have any ground truth labels available as in the case of synthetic data. Thus, we cannot really evaluate the grasp quality as done in Section 4.1. Instead, we use the detected grasping points in a manually segmented image as a reference to quantify the stability of the grasping point detection.

We have a set $B = \{b_i \| i = 1 \dots N\}$ of pictures and three different cues based on which they are segmented: zero disparity, a dominant plane and hue. If we want to measure the difference d_{b_i} between the set of grasping points $G_{b_i} = \{g_{(b_i,j)} \| j = 1 \dots M\}$ and the set of reference points $G_{b_i} = \{g_{(b_i,r)} \| k = 1 \dots R\}$ for a specific kind of segmentation of the image b_i, then

$$d_{b_i} = \frac{1}{K} \sum_{j=1}^{M} e^{\frac{-d_j^2}{2}} \text{ where} \tag{9}$$

$$d_j = \min_{r=1}^{R} dist(g_{(b_i,r)}, g_{(b_i,j)}) \tag{10}$$

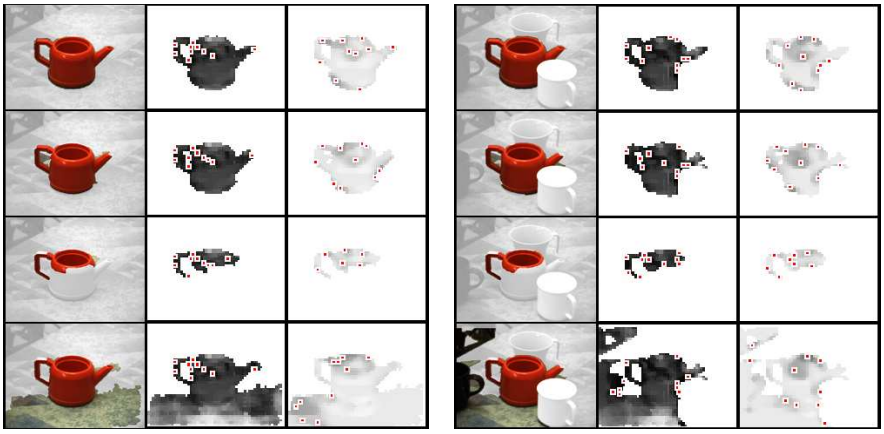
where $dist$ is the Euclidean distance and K the length of the image diagonal[2]. The mean and standard deviation of d_{b_i} for all images in the set B that are segmented with a specific cue is then our measure of deviation of the detected from the reference grasping points.

In Figure 23 we plotted this measure for a representative selection of objects and models. As already mentioned, ideally we would like to see no difference between detected grasping points when facing different qualities of segmentation. In practise, we can observe a flat slope. As expected for both methods, the grasping points detected in the image segmented with zero-disparity cues are the ones that are deviating most from the reference points. Although, the selection of points that are included in our representation is directly influenced by the segmentation, the difference between detected and reference grasping points is not always bigger than for the appearance based method. In fact, sometimes it performs even better. This holds for example for the models trained on mugs and cups for which both methods show a similar accuracy on synthetic data (Figure 23(a) and 23(b)). Also, if the models are applied to novel objects, as can be observed in Figure 23(c), our descriptors shows a better repeatability. This suggests again a better capability of the models to generalise across different relative shapes. In general, we can say that both methods are comparable in terms of repeatability.

4.3.3. Summary of Results

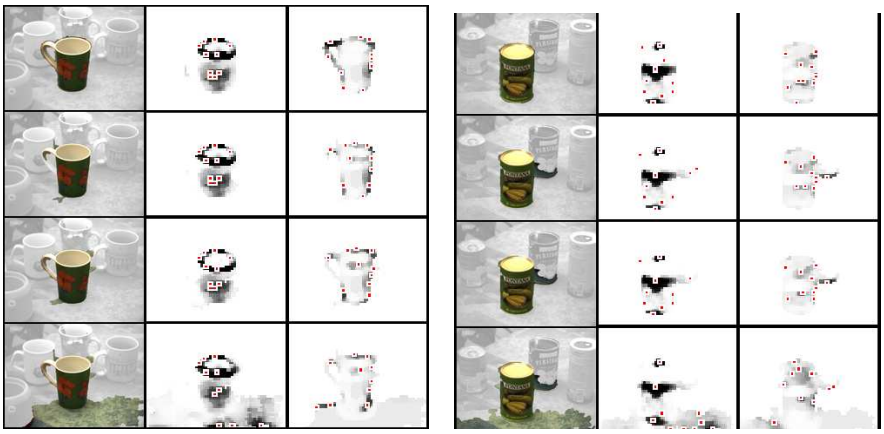
In this section, we evaluated the performance of our approach on real images. Due to the encoding of global shape, the method is robust against occlusions and strong texture. Although our representation is strongly dependent on the

[2]In our case $K = 80$ since we are evaluating 10×10 pixel patches in images of size 640×480 pixels



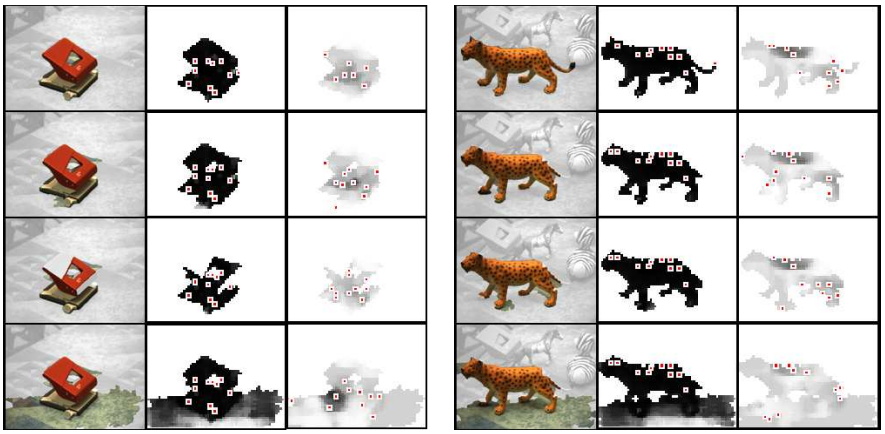
(a) Single Teapot (b) Partly Occluded Teapot

Figure 20: Grasping point model trained on mugs and pencils applied to a textureless teapot. The darker a pixel, the higher is the probability that it is a grasping point.



(a) Textured Cup. (b) Textured Can

Figure 21: Grasping point model trained on mugs and cups applied to a textured can or cup. The darker a pixel, the higher is the probability that it is a grasping point.



(a) Hole Puncher (b) Toy Leopard

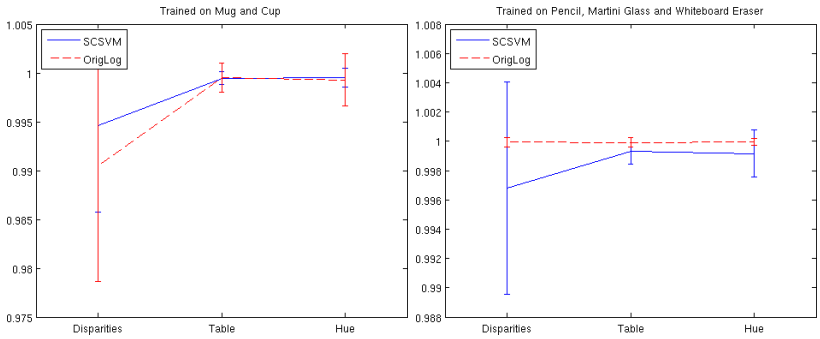
Figure 22: Grasping point model trained on pencils, martini glasses and whiteboard eraser applied to novel objects. The darker a pixel, the higher is the probability that it is a grasping point.

segmentation, we could observe that the repeatability of grasping points is comparable to the local appearance based method. The analysis included images of varying qualities of segmentation as well occlusion.

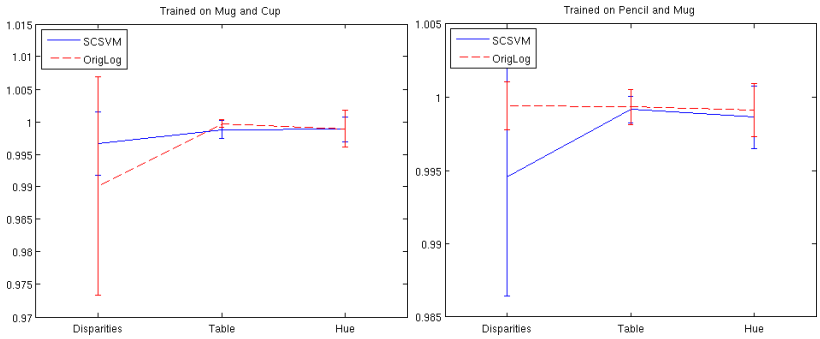
5. Conclusions

Grasping of unknown objects in natural environments is an important and unsolved problem in the robotic community. In this paper, we have developed a method for detecting a grasping point on an object by analysing it in a monocular image and reconstructing the suitable 3D grasping representation based on a stereo view . Referring to neuropsychological research mentioned in Section 2, we argued that for the purpose of grasping a yet unseen object, its global shape has to be taken into account. Therefore, we applied shape context as a visual feature descriptor that relates the object's global shape to a single point.

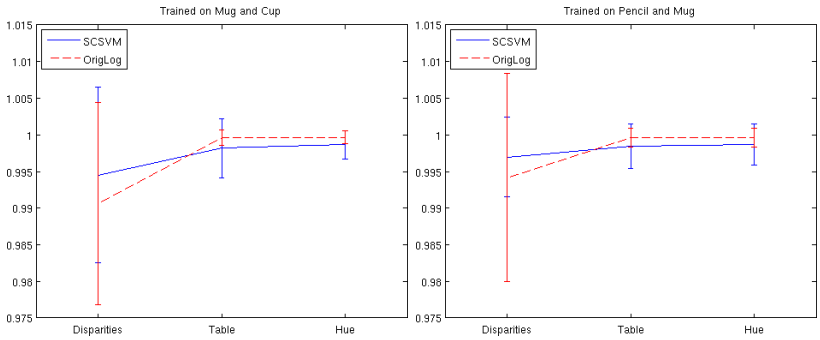
The experimental evaluation was performed both in simulation and in real scenes. The motivation for the simulated experiments was both to compare our approach with some other state of the art approaches as well as to provide a more insight into the complexity of the whole modelling process. We showed that a combination of a relative shape based representation and a non-linear classifier leads to an improved performance of the grasping point classification due to better discriminativity. Evaluation in the real scene has proven the applicability of our approach in the presence of clutter and provides further insight into the difficulty of the object grasping process. We see several aspects to be evaluated in the future work. We will continue to further develop the method but integrate it more on the stereo level for generating the grasping



(a) Two Grasping Point Models Applied to a Set of Cans



(b) Two Grasping Point Models Applied to a Set of Textureless Tea Pots



(c) Two Grasping Point Models Applied to a Set of Novel Objects

Figure 23: Comparing the stability of grasp point detection of SCSVM and OrigLog for different sets of objects and different grasping point models when facing imperfect segmentation.

point hypotheses. In addition, we will consider other type of representations that take into account several aspects of 2D-3D information.

Acknowledgement

The authors would like to thank Mårten Björkman for providing the segmentation and Xavi Gratal Martínez for his great help with the hardware system. This project has been supported by the EU IST-FP7-IP GRASP (2008-2012).

References

[1] M. Goodale, Separate Visual Pathways for Perception and Action, Trends in Neurosciences 15 (1) (1992) 20–25.

[2] M. A. Goodale, J. P. Meenan, H. H. Bülthoff, D. A. Nicolle, K. J. Murphy, C. I. Racicot, Separate Neural Pathways for the Visual Analysis of Object Shape in Perception and Prehension, Current Biology 4 (7) (1994) 604–610.

[3] A. Borghi, Grounding Cognition: The Role of Perception and Action in Memory, Language, and Thinking, chap. Object Concepts and Action, Cambridge University Press, 2005.

[4] S. H. Creem, D. R. Proffitt, Grasping Objects by Their Handles: A Necessary Interaction between Cognition and Action, Journal of Experimental Psychology: Human Perception and Performance 27 (1) (2001) 218–228.

[5] J. Gibson, The Ecological Approach to Visual Perception, Lawrence Erlbaum Associates, 1979.

[6] U. Castiello, M. Jeannerod, Measuring Time to Awareness, Neuroreport 2 (12) (1991) 797–800.

[7] M. J. Webster, J. Bachevalier, L. G. Ungerleider, Connections of Inferior Temporal Areas TEO and TE with Parietal and Frontal Cortex in Macaque Monkeys, Cerebral cortex 4 (5) (1994) 470–483.

[8] S. Ekvall, D. Kragic, Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning, in: IEEE International Conference on Robotics and Automation, 4715–4720, 2007.

[9] A. Morales, P. Azad, T. Asfour, D. Kraft, S. Knoop, R. Dillmann, A. Kargov, C. Pylatiuk, S. Schulz, An Anthropomorphic Grasping Approach for an Assistant Humanoid Robot, in: 37th International Symposium on Robotics, 149–152, 2006.

[10] J. Glover, D. Rus, N. Roy, Probabilistic Models of Object Geometry for Grasp Planning, in: IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 2008.

[11] K. Hübner, D. Kragic, Selection of Robot Pre-Grasps using Box-Based Shape Approximation, in: IEEE Int. Conference on Intelligent Robots and Systems, 1765–1770, 2008.

[12] C. Dunes, E. Marchand, C. Collowet, C. Leroux, Active Rough Shape Estimation of Unknown Objects, in: IEEE International Conference on Robotics and Automation, 3622–3627, 2008.

[13] M. Richtsfeld, M. Vincze, Grasping of Unknown Objects from a Table Top, in: ECCV Workshop on 'Vision in Action: Efficient strategies for cognitive agents in complex environments', Marseille, France, 2008.

[14] D. Kraft, N. Pugeault, E. Baseski, M. Popovic, D. Kragic, S. Kalkan, F. Wörgötter, N. Krueger, Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes, International Journal of Humanoid Robotics .

[15] G. M. Bone, A. Lambert, M. Edwards, Automated Modelling and Robotic Grasping of Unknown Three-Dimensional Objects, in: IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 292–298, 2008.

[16] A. Saxena, J. Driemeyer, J. Kearns, A. Y. Ng, Robotic Grasping of Novel Objects, Neural Information Processing Systems 19 (2007) 1209–1216.

[17] A. Morales, E. Chinellato, A. Fagg, A. del Pobil, Using Experience for Assessing Grasp Reliability, International Journal of Humanoid Robotics 1 (4) (2004) 671–691.

[18] M. Stark, P. Lies, M. Zillich, J. Wyatt, B. Schiele, Functional Object Class Detection Based on Learned Affordance Cues, in: 6th International Conference on Computer Vision Systems, vol. 5008 of *LNAI*, Springer-Verlag, 435–444, 2008.

[19] J. Tegin, S. Ekvall, D. Kragic, B. Iliev, J. Wikander, Demonstration based Learning and Control for Automatic Grasping, Journal of Intelligent Service Robotics To appear.

[20] S. Belongie, J. Malik, J. Puzicha, Shape Matching and Object Recognition Using Shape Contexts, IEEE Trans. on Pattern Analysis and Machine Intelligence 24 (4) (2002) 509–522.

[21] R. H. Cuijpers, J. B. J. Smeets, E. Brenner, On the Relation Between Object Shape and Grasping Kinematics, Journal of Neurophysiology 91 (2004) 2598–2606.

[22] M. Gentilucci, Object Motor Representation and Reaching-Grasping Control, Neuropsychologia 40 (8) (2002) 1139–1153.

[23] J. Speth, A. Morales, P. J. Sanz, Vision-Based Grasp Planning of 3D Objects by Extending 2D Contour Based Algorithms, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2240–2245, 2008.

[24] V.-D. Nguyen, Constructing stable grasps, International Journal on Robotics Research 8 (1) (1989) 26–37.

[25] K. Shimoga, Robot Grasp Synthesis Algorithms: A Survey, International Journal of Robotic Research 15 (3) (1996) 230–266.

[26] A. T. Miller, S. Knoop, H. I. Christensen, P. K. Allen, Automatic Grasp Planning Using Shape Primitives, in: IEEE Int. Conf. on Robotics and Automation, 1824–1829, 2003.

[27] C. Goldfeder, P. K. Allen, C. Lackner, R. Pelossof, Grasp Planning Via Decomposition Trees, in: IEEE International Conference on Robotics and Automation, 4679–4684, 2007.

[28] M. Ciorcarlie, C. Goldfeder, P. Allen, Dexterous Grasping via Eigengrasps: A Low-Dimensional Approach to a High-Complexity Problem, Robotics: Science and Systems Manipulation Workshop .

[29] C. Borst, M. Fischer, G. Hirzinger, Grasping the Dice by Dicing the Grasp, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 3692–3697, 2003.

[30] Y. Li, N. Pollard, A Shape Matching Algorithm for Synthesizing Humanlike Enveloping Grasps, Humanoid Robots, 2005 5th IEEE-RAS International Conference on (2005) 442–449.

[31] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, R. Dillmann, ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control, in: 6th IEEE-RAS International Conference on Humanoid Robots, 169–175, 2006.

[32] J. Shotton, J. Winn, C. Rother, A. Criminisi, TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation, in: Proceedings of European Conference Computer Vision (ECCV), 2006.

[33] V. Ferrari, L. Fevrier, F. Jurie, C. Schmid, Groups of Adjacent Contour Segments for Object Detection, IEEE Trans. Pattern Anal. Mach. Intell. 30 (1) (2008) 36–51.

[34] C. Dance, J. Willamowski, L. Fan, C. Bray, G. Csurka, Visual categorization with bags of keypoints, in: ECCV International Workshop on Statistical Learning in Computer Vision, 2004.

[35] F.-F. L., P. Perona, A Bayesian Hierarchical Model for Learning Natural Scene Categories, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 2 (2005) 524–531.

[36] B. Leibe, A. Leonardis, B. Schiele, An Implicit Shape Model for Combined Object Categorization and Segmentation, in: Toward Category-Level Object Recognition, 508–524, 2006.

[37] S. Lazebnik, C. Schmid, J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, 2169–2178, 2006.

[38] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, P. Boyes-Braem, Basic objects in natural categories, Cognitive Psychology 8 (3) (1976) 382–439.

[39] S. El-Khoury, A. Sahbani, Handling Objects By Their Handles, in: IROS-2008 Workshop on Grasp and Task Learning by Imitation, 2008.

[40] R. Pelossof, A. Miller, P. Allen, T. Jebera, An SVM Learning Approach to Robotic Grasping, in: IEEE International Conference on Robotics and Automation, 3512–3518, 2004.

[41] N. Curtis, J. Xiao, Efficient and Effective Grasping of Novel Objects through Learning and Adapting a Knowledge Base, in: IEEE International Conference on Robotics and Automation, 2252–2257, 2008.

[42] J. Grezes, J. Decety, Does Visual Perception of Object Afford Action? Evidence from a Neuroimaging Study., Neuropsychologia 40 (2) (2002) 212–222.

[43] J. Bohg, C.Barck-Holst, K. Hübner, M. Ralph, D. Song, D. Kragic, Towards Grasp-Oriented Visual Perception in Humanoid Robotics, International Journal of Humanoid Robotics Submitted.

[44] A. Saxena, L. Wong, A. Y. Ng, Learning Grasp Strategies with Partial Shape Information, in: 23rd AAAI Conference on Artificial Intelligence, 1491–1494, 2008.

[45] KUKA, KR 5 sixx R650, `www.kuka-robotics.com`, ????

[46] W. T. Townsend, The BarrettHand Grasper – Programmably Flexible Part Handling and Assembly, Industrial Robot: An International Journal 27 (3) (2000) 181–188.

[47] D. Hoiem, C. Rother, J. Winn, 3D LayoutCRF for Multi-View Object Class Recognition and Segmentation, in: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, 1–8, 2007.

[48] B. Rasolzadeh, A. T. Targhi, J.-O. Eklundh, An Attentional System Combining Top-Down and Bottom-Up Influences, in: Workshop on Attention and Performance in Computational Vision, 123–140, 2007.

[49] M. Björkman, J. Eklundh, Foveated Figure-Ground Segmentation and Its Role in Recognition, in: British Machine Vision Conference, 2005.

[50] C. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, 2006.

[51] M. Craven, J. Shavlik, Extracting Tree-Structured Representations of Trained Networks, in: Advances in Neural Information Processing Systems (NIPS-8), MIT Press, 24–30, 1995.

[52] D. Martens, B. Baesens, T. V. Gestel, J. Vanthienen, Comprehensible Credit Scoring Models using Rule Extraction from Support Vector Machines, European Journal of Operational Research 183 (3) (2007) 1466–1476.

Proceedings of the RAAD 2008
17th International Workshop on Robotics in Alpe–Adria–Danube Region
September 15-17, 2008, Ancona, Italy

Robotic Grasping of Unknown Objects using $2^1/_2$D Point Clouds

Mario Richtsfeld and Markus Vincze

Institute of Automation and Control
Vienna University of Technology
Gusshausstr. 27-29, Vienna, Austria
[*rm*, *vm*]@acin.tuwien.ac.at

Abstract. In this paper, we deal with the problem of robotic grasping of unknown objects in a fully automatic way. Based on $2^1/_2$D point clouds obtained with a laser range scanner we segment objects and find suitable grasping points based on the convex hulls of objects. We calculate collision free paths and perform grasping using a seven degrees of freedom arm manipulator and a hand prosthesis as gripper. We also present a simple method for removing noise and outliers based on point density. Our results show that the presented method leads to a very stable grasp point detection for a variety of different objects.

Keywords. Grasping, Motion Planning, Laser Range Scanning.

1. Introduction

This paper shows how to detect, grasp and manipulate a variety of different objects in a fully autonomous way[1]. We present an algorithm that automatically filters and segments a $2^1/_2$D point cloud[2] and calculates grasping points. In this work, we only consider objects with flat top surfaces and these objects can be open or closed. The challenge is analyzed by a fixed setup, which consists of a laser range scanner and an AMTEC[3] robot arm with seven degrees of freedom. The robot arm is equipped with a hand prosthesis from the company Otto Bock[4] as gripper.

1.1. Problem Statement and Contribution

We start with a $2^1/_2$D point cloud of a typical table scene, i.e. the scene is viewed from just one range scanner position and the backsides of objects are not visible. Fig. 1 shows five different objects. The goal of our work is a robust way to find the grasping points of every object in the point cloud, see the lower figure of Fig. 1. The main problems to overcome are missing sensor data from shadows or poor surface reflectance, segmentation of the point cloud into objects and dealing with noise and outliers. To reduce complexity we only consider objects with flat top surfaces. Fig. 2 gives an overview of our multi-step solution procedure.

The main contribution of the presented paper is to use $2^1/_2$D point clouds for shape reconstruction based on the convex hull of objects to calculate grasping points. We also present an advanced filter to reduce noise and outliers.

The algorithm consists of five main steps:

- Raw Data Preprocessing: The raw data points are preprocessed with a geometrical filter to reduce noise and outliers.
- Mesh Generation: This step is used to reduce the number of points and to realize faster computations.
- Range Image Segmentation: This step identifies different objects.
- Grasping Point Detection: Calculation of possible grasping points based on the top surfaces.
- Path Planning Tool: Transmission of the Calculated Object Pose to the Path Planning Tool.

[1]This work was supported by the EU-Project "GRASP" with the grant agreement number 215821.
[2]All objects are scanned from the same range scanner position.
[3]http://www.amtec-robotics.com
[4]http://www.ottobock.de/

Fig. 1. Table scene with five different objects (from left: 1. package manner, 2. package blue, 3. eraser, 4. package uhu, 5. package brackets). The lower figure shows the generated mesh with 6.429 object points and 111.628 plane points from originally 124.486 points. The two shadows from laser and camera and the grasping points (green colored) and the rim points (blue colored) are clearly visible. Best viewed in color.

To realize an unbiased evaluation of our multi-step solution procedure, we defined fifteen different objects, see Fig. 3. The blue lines represent the optimal positions for grasping points. Optimally grasping points are required to be placed on parallel surfaces. If that is not possible the second grasping point will be placed on a corner point on the opposite side.

The outline of the paper is as follows: The next Section 2 introduces our robotic system and its components. Section 3 describes the geometric filter, mesh generation and object segmentation. Section 4 details the analysis of the objects and describes the efficient use of the gained information to calculate grasping points. Section 5 describes the grasping and manipulation, Section 6 shows our results and Section 7 finally concludes the paper.

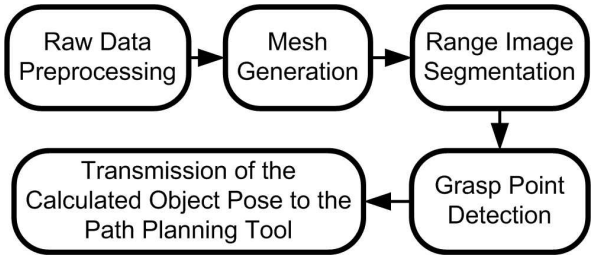


Fig. 2. Overview of our grasp point detection algorithm

1.2. State of the Art

From 1983 to 1988 the mobile manipulator MoVAR (8) was developed. This PUMA-250 robot was instrumented with a camera for remote sensing, a six-axis force sensor and a gripper with finger pad-mounted proximity sensors. A nice overview of different systems, such as the Wolfson-Robot and the Wessex-Robot is given by Hagen and Hillmann (6). Up to now a number of scientists have been working on the same idea to develop a wheelchair mounted robot or a mobile robot system with arms to handle objects and assist elderly and handicapped persons. Martens et al. (10), (7) developed the FRIEND-I and FRIEND-II systems, where the the robot arm is mounted on a wheelchair. For object detection they use a stereo camera system and the user interaction is based on a LC-display. The objects must be placed on a predefined area and a successful execution of the grasping task in these systems is only possible for a limited number of objects.

Miller et al. (9) presented an automatic grasp planning system “GraspIt!” for hand configurations. They use shape primitives such as boxes, spheres, cones and cylinders. These shape primitives are used to limit the number of potential grasps. In our case the vision task is to detect edges and surfaces of objects that are analyzed to calculate grasp points. Wang et al. (13) presented a framework of automatic grasping of unknown objects by using a laser range scanner and a simulation environment. Wheeler (14) formulated a robotic pick and place operation as prospective behavior. They analyze the traditional planning methods in nature and require geometric models of parts, fixtures, and motions to identify and avoid the constraints, but these methods can easily become computationally expensive. Balch and Arkin (2) also applied primitive behaviors to mobile robotics. Borst et al. (3) show that is not necessary in every case to generate optimal grasps, 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

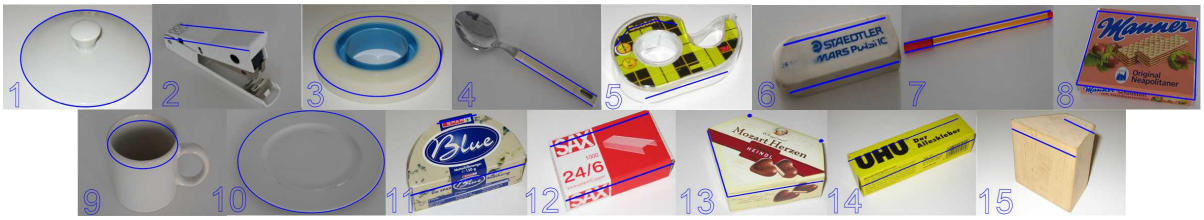


Fig. 3. Specified fifteen different objects. The blue lines represent the optimal positions for grasping points. Best viewed in color.

good grasp fast and suitable. Aarno et al. (1) 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. (12) developed a learning algorithm that predicts the grasp position of an object directly as a function of its image. Their work focuses on the task of identifying grasping points that are trained with labeled synthetic images of a different number of objects. The classification is based on feature vectors based on color, texture and edges in several scales. In our work we do not use a supervised learning approach, we find grasping points according to predefined rules. Moreover we find two grasp points, opposed to only one as in Saxena's work, which results in a more stable grasp.

2. System Approach

Our approach is based on scanning the objects with a rotating laser range scanner and execution of subsequent path planning and grasping motion (see Fig. 4). The laser range scanner consists of a red-light LASIRIS laser from StockerYale[5] and a MAPP2500 CCD-camera from SICK-IVP[6] mounted on a pan/tilt-unit. This time we use the "Light Weight Arm 7 DOF" from AMTEC robotics and a hand prosthesis from Otto Bock as gripper. The seventh degree of freedom is important to enable complex object grasping and manipulation and also to realize some flexibility. The prosthesis has three active fingers, the thumb, the index finger and the middle finger. A commercial path planning tool from AMROSE[7] calculates the trajectory to grasp the object. The algorithm is implemented in C++ using the Visualization Tool Kit (VTK)[8].

[5]http://www.stockeryale.com/index.htm
[6]http://www.sickivp.se/sickivp/de.html
[7]http://www.amrose.dk/
[8]open source software: http://public.kitware.com/vtk.

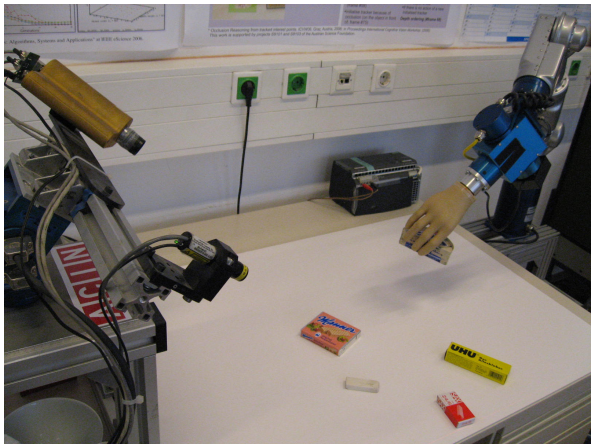


Fig. 4. Overview of the system components and their interrelations.

3. Range Data Segmentation

At the beginning the recorded point cloud from the laser range scanner should be filtered to reduce noise and outliers. Ferrari et al. (5) presented a general method to reduce the number of points in a point cloud. We use a modified version of this interesting method to remove outliers and the threshold parameters are calculated with the help of the compression rate. This filter calculates for each point the distance to the nearest neighbor and then the minimum d_{min}, maximum d_{max} and average d_a of these distances. This distances are used to calculate the compression rate τ.

$$d_k = \frac{d_{min} + d_{max}}{2} \tag{1}$$

$$\tau = \frac{d_a}{d_k} \tag{2}$$

Then all N_a points inside the sphere with radius d_a and all N_k points inside the sphere with radius d_k around a regarded point are used to decide with the compression rate τ if the regarded point is an outlier or not.

$$\alpha = (d_a/N_a)^{\tau} \tag{3}$$

$$\beta = (d_k/N_k) \tag{4}$$

If $\alpha > \beta$ the regarded point is an outlier and will be removed. Fig. 5 shows an example after our filtering procedure.

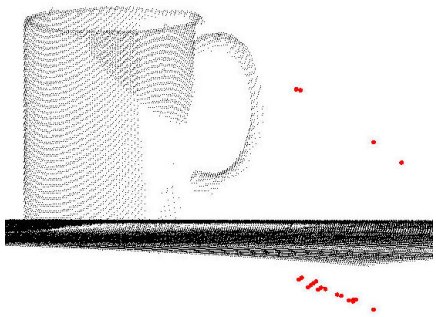


Fig. 5. Filtered point cloud. The red points represent the outliers.

The range data segmentation starts by detecting the surface of the table with a RANSAC (4) based plane fit. After filtering and removing the table plane points only $n = 12.858$ points remain of the original $n = 124.486$ points, see Fig. 1. The segmentation of the remaining points will be achieved with the help of a 3D mesh generation, based on the triangles calculated by a DeLaunay triangulation (11). The necessary settings for the mesh generation are already determined with d_{min} and d_{max} of the filter computation. After the mesh generation step only $n = 6.429$ points remain.

4. Grasp Point Detection

The algorithm for grasp point detection finds the top surface of all objects in the table scene, see Fig. 1 and Fig. 6 (blue colored planes). The grasping points are at a defined value of 5*mm* under the top surface. Naturally we check if there is a plane or not and also the height of the top surface. In case there is none, see Fig. 8 counting the disk, the grasping point will not be moved under the top surface.

Next we find the convex hull of the top surface points (see Fig. 6, blue points). We now detect the longest side of the convex hull c (between red and green colored points), see Fig. 6.

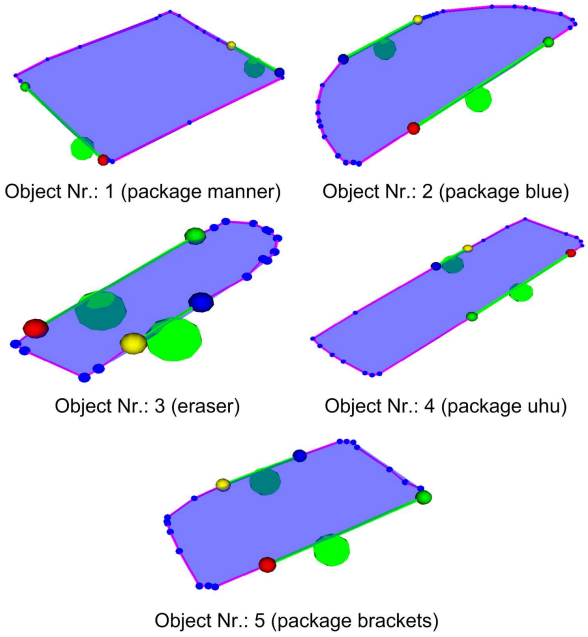


Fig. 6. Top surfaces of the five objects from Fig. 1. The red and green colored points represent the longest distance between neighboring rim points. The yellow and blue points represent the longest distance on the opposite side. Best viewed in color.

We look then for a parallel line on the opposite side or if we can not find a parallel line, then we look for a corner point on the opposite side. With the distances a and b (distances from the red and green colored points) to the opposite point we calculate the altitude h of the triangle abc, where β is the angle between a and b. We check the lines left and right of the furthest point for parallelism with an angle tolerance of 5° (see Fig. 6, green lines (optimal grasping surfaces)). If the angle difference is larger than 5° and there are several remaining points we analyze the next largest distances. If no suitable line can be found, we just take the furthest point. If the distance of this corner point is bigger than the maximum opening angle of the hand prosthesis (110mm) no suitable grasp point can be detected.

$$\beta = \arccos\left(\frac{a^2 - b^2 - c^2}{-2 \cdot b \cdot c}\right) \tag{5}$$

$$h = b \cdot \sin(\beta) \tag{6}$$

5. Object Grasping and Manipulation

Here we calculate a collision free robot trajectory and execute the grasping activity safely. In the last step we

calculate the object position in the actual environment model and transmit it to the path planning tool. For that the generated 3D mesh will be used. The robot path is calculated by a path planning tool from AMROSE[9]. The input is the detected object pose, the environment model, the grasping points and a transformation between the robot coordinate system and the laser range scanner coordinate system, see Fig. 7. The output is a collision free robot trajectory to the desired object.

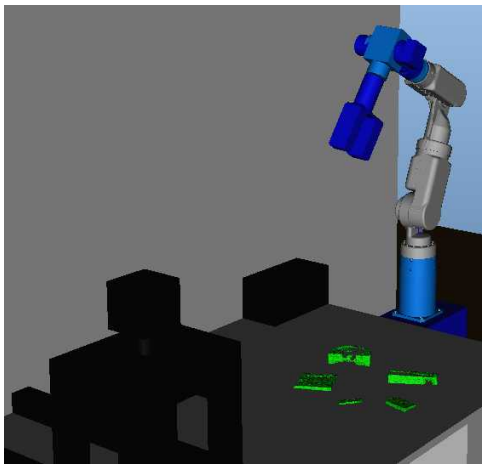


Fig. 7. Visualization of the experimental setup by a simulation tool, which is suitable to calculate the trajectory of the robot arm. Best viewed in color.

Before the robot executes the trajectory, the user can check a simulation of the calculated trajectory and decide whether it is safe enough to handle the object or not. Finally the desired object can be placed at a defined position or directly handed over to the user.

6. Experiments and Results

In our work, we demonstrate that our grasping point detection algorithm for objects with flat top surfaces shows very good results, see Tab. 2. We evaluated the detected grasping points by comparing them to the optimal grasping points as defined in Fig. 3.

A remaining problem is, that in some cases for shiny objects interesting parts of the objects are not visible for the laser range scanner and thus our algorithm is not able to calculate the correct grasping points of the object (see Fig. 8, object number 1 and 5). The quality of the point cloud of the first and the last object (the polygon on the left and the cap on the right side (see Tab. 2 object number 1 and 15)) is not good enough to guarantee a successful grasp. So the success of our grasping point algorithm depends on the ambient light, object surface properties, laser beam reflectance, absorption of the objects and vibrations. Therefore, the laser range scanner must be configured to the respective environment. By using an additional red-light filter the impact of light or reflections can be minimized. The poor grasp-rate of the object number 13 “package mozart” in Tab. 2 of 55% can be explained by the leg of parallel surfaces.

Fig. 8. The success of our grasping point algorithm depends on the reflectance, absorption and form of the objects to grasp. By the 1. object (the polygon) absorption appearance, by the 5. object (the cap) one part of the object hides the rest of the object. Best viewed in color.

The experience were performed on a PC with 3.2GHz Pentium dual-core processor and average run time is about 18sec, see Tab. 1.

Tab. 1. Duration of every calculation step.

Calculation Steps	Time [sec]
Geometric Filter	7.570sec
Plane Fit	1.391sec
Mesh	7.493sec
Object Segmentation	0.508sec
Grasp-Point Detection	0.820sec
Sum	17.782sec

The results indicate this strategy is feasible to complete a grasping task automatically under this framework.

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

Tab. 2. Grasp-rate of different objects.

Obj. Nr.	Objects	Grasp-Rate [%]
1	Cap	25%
2	Stapler	97.5%
3	Adhesive Foil	100%
4	Spoon	100%
5	Double-Sided Adhesive Foil	75%
6	Eraser	97.5%
7	Pen	97.5%
8	Package Manner	100%
9	Mugs (2 x handle)	97.5% (85%)
10	Disk	100%
11	Package Blue	100%
12	Package Brackets	100%
13	Package Mozart	55%
14	Package UHU	100%
15	Polygon	47.5%
Overall		86.2%

7. Conclusion and Future Work

In this paper we present a framework for automatic grasping of unknown objects with a hand prosthesis as gripper, by incorporating a laser range scanner. We present a method to get accurate grasping points of unknown objects in point clouds. This includes a simple modified geometric filter for outlier and noise removal, which shows high adaptability to the laser resolution. We calculate the grasping points with the help of the top surfaces. The presented method shows high reliability and the grasping approach can be applied to a reasonable set of objects.

In the near future we will add a step to reassemble objects which are broken into parts because of shadows or occlusions. The hand configuration will also be included in our future research.

8. References

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

[2] Balch, T., Arkin, R.C.: Behavior-based formation control for multirobot teams. IEEE Transactions on Robotics and Automation, Vol. 14, No. 6, pp. 926-939, 1998.

[3] Borst, C., Fischer, M., Hirzinger, G.: Grasping the Dice by Dicing the Grasp. Proceedings of the IEEE/RSJ International Conference on Robotics and Systems (IROS 2003), Vol. 4, pp. 3692-3697, 2003.

[4] Fischler, M.A., Bolles, R.C.: 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, 1981.

[5] Ferrari, S., Ferrigno, G., Piuri, V., Borghese, N.A.: Reducing and Filtering Point Clouds With Enhanced Vector Quantization. IEEE Transactions on Neural Networks, Vol. 18, No. 1, pp. 161-177, 2007.

[6] Hagan, K., Hillman, M.: The design of a wheelchair mounted robot. Colloquium on Computers in the Service of Mankind: Helping the Disabled, pp. 1-6, 1997.

[7] Ivlev, O., Martens, C.: Rehabilitation Robots FRIEND-I and FRIEND-I with the dexterous lightweight manipulator. IOS Press, Vol. 17, No. 2, pp. 111-123, 2005.

[8] Van der Loos, H.: VA/Stanford Rehabilitation Robotics Research and Development Program: Lessons Learned in the Application of Robotics Technology to Field of Rehabilitation. IEEE Press, Transactions on Rehabilitation Engineering, Vol. 3, No. 1, pp. 46-55, 1995.

[9] Miller, A.T., Knoop, S., Christensen, H.I., Allen, P.K.: Automatic grasp planning using shape primitives. Proceedings of the ICRA International Conference on Robotics and Automation (ICRA 2003), IEEE, Vol. 2, pp. 1824-1829, 2003.

[10] Martens, C., Ruchel, N.: A FRIEND for Assisting Handicapped People. IEEE Press, Robotics and Automation Magazine, Vol. 8, No. 1, pp. 57-65, 2001.

[11] O'Rourke, J.: Computational Geometry in C. Univ. Press, Cambridge, 2nd edition, 1998.

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

[13] Wang, B., Jiang, L., LI, J.W., Cai, H.G.: Grasping Unknown Objects Based on 3d Model Reconstruction. Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (ASME 2005), IEEE, pp. 461-466, 2005.

[14] Wheeler, D.S., Fagg, A.H. Grupen, R.A.: Learning prospective pick and place behavior. Proceedings of the 2nd International Conference on Development and Learning, pp. 197-202, 2002.

Author manuscript, published in "Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments, Marseille : France (2008)"

inria-00325794, version 1 - 30 Sep 2008

Grasping of Unknown Objects from a Table Top*

Mario Richtsfeld and Markus Vincze
Institute of Automation and Control
Vienna University of Technology
Gusshausstr. 27-29, Vienna, Austria
[rm, vm]@acin.tuwien.ac.at

Paper ID 11

Abstract. This paper describes the development of a novel vision-based grasping system for unknown objects based on range images. We realize a synthesis of the calculated grasp points with a 3D model of a hand prosthesis, which we are using as gripper. We locally find grasp point candidates based on the shape of the object and validate the globally by checking collisions between the gripper and surrounding objects and the table top. Our approach integrates a robust object segmentation and grasp point detection for every object on a table in front of a 7-DOF robot arm. The algorithm analyzes the top surface of every object and outputs the generated grasp points and the required gripper pose to grasp the desired object. Additionally we can calculate the optimal opening angle of the gripper. The first experimental results show that the presented automated grasping system is able to generate successful grasp points for a wide range of different objects.

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 [1]." The grasping task was studied from a psychological, biological and engineering focus but still remains unresolved. There exist partial solutions for certain cases, however there is still no general valid solution. This paper presents an approach that detects potential grasp points to realize the task of grasping arbitrary objects in arbitrary poses. Our vision is to find a fully autonomous way to detect, grasp and manipulate any kind of object. The human has a sophisticated system, which allows him to grasp a wide range of different objects in different cases. Human hands are characterized by five soft fingers with high dexterity and the humans know the shape, dimension and properties of their hands. Additionally humans have as yet unmatched visual capabilities. In this work we try to realize this combination of the human abilities with a laser range scanner and a 3D model of the used

* This work was supported by the EU-Project "GRASP" with the grant agreement number 215821.

inria-00325794, version 1 - 30 Sep 2008

gripper, which is a prosthetic hand from the company Otto Bock[1]. Humans are also able to grasp unknown objects. They learn different shapes from early on and people are able to generalize to new objects. We present an algorithm that automatically segments a 2.5D point cloud, calculates practical grasp points and checks the validity of the grasp points with a 3D model of the hand prosthesis. Thereby the algorithm finds the best gripper pose for the used hand prosthesis to grasp the desired object without any collision.

1.1 Problem Statement and Contribution

We operate on a 2.5D point cloud of a typical table scene, where every object is scanned from the same laser range scanner position. All considered objects have mostly horizontal planar top surfaces. Fig. 1[2] shows seven different objects, where object no. 1 to 6 have a convex shape and object no. 7 has a concave shape. We define what we consider as grasp points, the blue lines represent the optimal positions for grasp points. The first goal of this work is a robust detection of the grasp points of any kind of object in the point cloud, see fig. 3. This including robustness despite to noise, outliers and shadows, which can be caused by specular or reflective surfaces. Fig. 2 gives an overview of our proposed method.

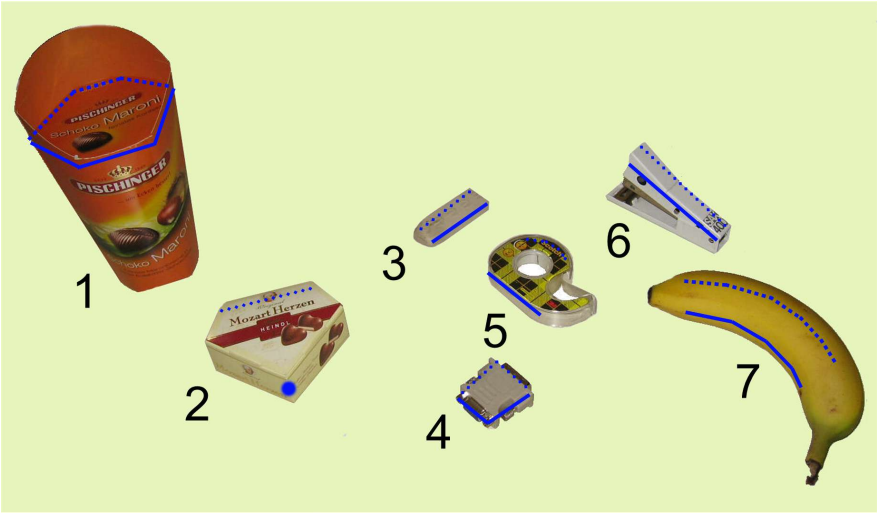


Fig. 1. Table scene with seven different objects (from left: 1. chocolate package, 2. package Mozart, 3. eraser, 4. plug, 5. adhesive foil, 6. stapler, 7. Banana). The blue lines represent the optimal positions for grasp points.

[1] http://www.ottobock.de/
[2] All images are best viewed in color.

inria-00325794, version 1 - 30 Sep 2008

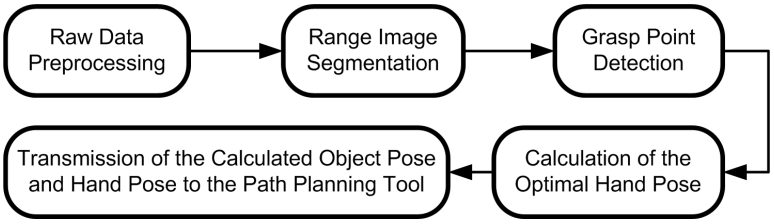


Fig. 2. Overview of our grasp point detection and synthesis algorithm.

The algorithm consists of five main steps:

- Raw Data Preprocessing: The raw data points are preprocessed with a geometrical filter to reduce noise and outliers.
- Range Image Segmentation: This step identifies different objects based on a 3D DeLaunay triangulation.
- Grasp Point Detection: Calculation of possible grasp points based on the top surfaces of the objects.
- Validity Check of the Grasp Points: Considering surrounding objects and the table surface as obstacles, find optimal gripper pose, which maximizes distances to obstacles.
- Path Planning Tool: Transmission of the calculated object pose and hand pose to the path planning tool.

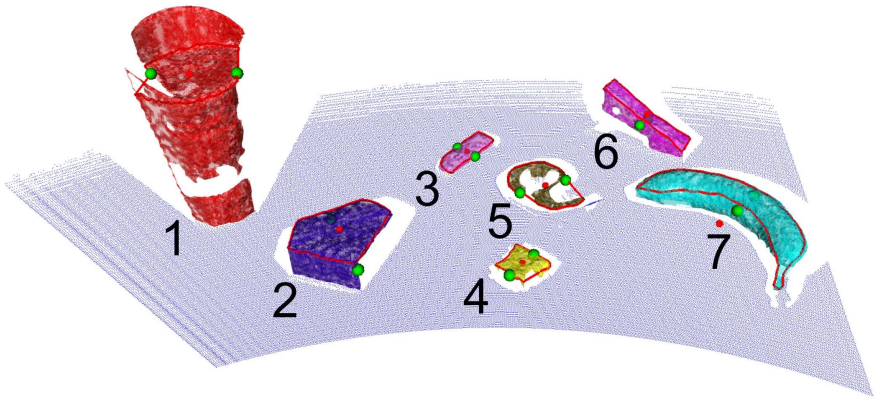


Fig. 3. The figure shows the generated meshes with 12.437 object points and 81.691 plane points from originally 100.843 points. The two shadows from laser and camera and the grasp points (green colored) and the rim line (red colored) are clearly visible. The red points represent the calculated center of mass of the different top surfaces.

The second goal of our work is to analyze the calculated grasp points with the help of a 3D model of the hand prosthesis, which we are using as gripper,

see fig. 4. It has three active fingers, the thumb, the index finger and the middle finger. The last two fingers are just for cosmetic reasons. So the proposed algorithm is based on two grasping points between the thumb and index finger. The 3D model of the gripper is realized with a Minolta VIVID 700 range scanner[3]. This 3D model enables it to calculate the optimal position and orientation of the gripper to successfully grasp the desired object. Furthermore it affords to consider all surrounding objects to identify potential obstacles. As well the opening angle can be observed to detect a possible collision with the table. All these information is important for the path planner to calculated a successfully path to grasp the desired object.

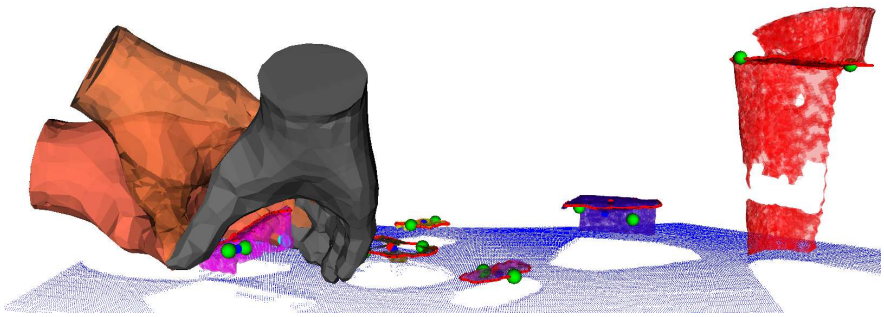


Fig. 4. This figure shows on the left side three different hand configurations to grasp the stapler. The left 3D model of the hand (red colored) shows the maximum positive hand orientation by 90°, the right hand (black colored) shows the maximum negative hand orientation by −30° and the hand model in the middle (orange colored) shows the optimum orientation by 60°.

We simulate the complete grasping process with a commercial path planning tool from AMROSE[4]. The input is the detected object pose, the gripper pose, the environment model, the grasp points and a transformation between the robot coordinate system and the laser range scanner coordinate system. The output is a collision free robot trajectory to the desired object. Before the robot executes the trajectory, the user can check a simulation of the calculated trajectory.

The outline of the paper is as follows: The next Section 2 details the analysis of the objects and describes the calculation of possible grasp points. Section 3 describes the implementation of a 3D model of the gripper. Section 4 shows our results and Section 5 finally concludes the paper.

[3] http://www.konicaminolta.com/sensingusa/products/3d
[4] http://www.amrose.dk/

inria-00325794, version 1 - 30 Sep 2008

inria-00325794, version 1 - 30 Sep 2008

1.2 Related Work

Fagg and Arbib [2] developed the FARS model, which focuses especially on the action-execution step. Nevertheless, no robotic application has been yet developed following this path. Saxena et al. [3] developed a supervised learning algorithm that is able to predict the grasp position of novel objects as a function of 2D images. The work focuses on the task of identifying grasp positions. In our work we do not use learning, but we believe a priori that we consider possible grasp points. Saxena also defines for every object only one grasp point, in some cases objects can be grasped slanted. In our approach we calculate two grasp points to realize a more stable grasp. Stansfield [4] developed a system for grasping objects with unknown geometry. At the beginning every object was placed on a rotary disc. Then the object was rotated and translated under a laser range scanner to generate a 3D model of the object. The scanned 3D model formed the input to an expert system that planned the grasping process. This system was tested for several objects. Miller et al. [5] specify an automatic grasp planning system "GraspIt!" for hand configurations using shape primitives, by modeling an object as a sphere, cylinder, cone or box. They use a set of rules to generate possible grasp positions. In our case the vision task is to detect edges and surfaces of objects that are analyzed to calculate grasping points. We use a 3D model of the hand prosthesis, which we are using as gripper to find an optimal grasping angle to grasp the object. Wang et al. [6] developed a framework of automatic grasping of unknown objects by using a laser scanner and a simulation environment. Boughorbel et al. [7] 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 modeling approach can only be used for rotationally symmetric objects. Bone et al. [8] presented an interesting approach, which combines online silhouette and structured-light 3D object modeling 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 analyze the validity of the calculated grasping points with a 3D model of the hand, thereby our algorithm also outputs the required gripper pose to grasp the object. Borst et al. [9] 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. Kragic and Bjrkman [10] developed another 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. Recatalà et. al. [11] developed a framework for the development of robotic applications on the synthesis and execution of grasps. Li et al. [12] presented a data-driven approach to grasp synthesis. Their algorithm uses a database of

inria-00325794, version 1 - 30 Sep 2008

captured human grasps to find the best grasp by matching hand shape to object shape.

2 Grasp Point Detection

At the beginning the recorded point cloud from the laser range scanner should be filtered with a low pass filter to reduce any noise and outliers. The range data segmentation starts by detecting the surface of the table with a RANSAC [13] based plane fit. The segmentation of the remaining points is achieved with a 3D mesh generation, based on the triangles calculated by a 3D DeLaunay triangulation [14].

The algorithm for grasp point detection finds the top surface of all objects with a defined threshold of $3mm$ and generates a 2D DeLaunay triangulation, with this 2D surface information the rim points and feature edges of every object can be detected, see fig. 5. Then we calculate the center of mass for every objects top surface (red colored points in fig. 5). For convex shapes the center of mass is inside the surface, but for concave shapes the center of mass may be outside as illustrated in fig. 5 by object no. 7.

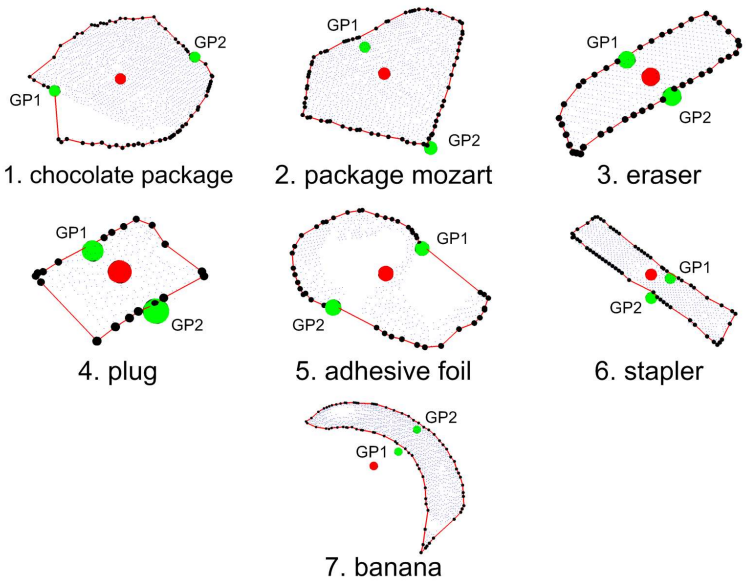


Fig. 5. Top surfaces of the seven objects from fig. 1. The red lines represent the form of the top surfaces, the red point represents the center of mass. The green points are the calculated grasp points, GP1 is the first grasp point with the shortest distance to the center of mass and GP2 is the second grasp point.

The first grasp point (GP1) is that point along the rim line (red line), which has the shortest euclidian distance to the center of mass (red point). The second grasp point (GP2) is on the opposite rim line. Thereby, the first grasp point (GP1) should have with the second grasp (GP2) and the center of mass should lie on a line. To grasp an object on the top rim line can create a possible slipping through the fingers of the hand prosthesis. To avoid that, the height of the top surface is calculated and both grasp points are shifted down. The shifting distance in our case is maximal $30mm$, this distance is pretended through the gripper. Additionally we check that at least one of the shifted grasp points lies on a visible surface, i.e. is not shifted into thin air.

3 Feasibility of Objects Grasp Points

In order to successfully grasp an object it is not sufficient to find locally the best grasp points, the algorithm must also decide like humans from which angle it is possible to grasp it. Moreover the algorithm checks the validity of the grasp points. For that approach we rotate the 3D model of the hand prosthesis around the rotation axis, which is defined through the grasp points. The rotation axis of the hand is defined by the thumb and the index finger of the hand as illustrated in fig. 6 with the cyan colored points. At the beginning the hand is placed accurately over the grasping object. This start position is defined with a grasping angle of $0°$. Furthermore the opening angle of the hand is set to its maximum. The algorithm checks for a collision of the hand with the table or other objects. If there is no collision our approach calculates the maximum and minimum possible rotation angles. We find the best gripper position and orientation by an averaging of the maximum and minimum possible rotation angles. Through that, the algorithm calculates the best gripper pose to grasp the desired object for the path planning tool. If there is a collision the grasp point detection algorithm calculates new grasp points for the desired object. Then the algorithm takes for the first grasp point (GP1) the second shortest euclidian distance between the center of mass and the rim line and all other calculations are repeated.

We decide to use the power crust algorithm for the surface reconstruction [15] of the 3D model of the hand prosthesis, because this algorithm delivers very good results and is quite fast. It realizes a construction which takes a sample of points from the surface of a 3D object and produces a surface mesh and an approximate medial surface axis. The approach approximates the medial axis transform (MAT) of the object. Then it uses an inverse transform to produce the surface representation from the MAT.

This approach allows it to change the start position and orientation of the gripper online depending on the grasping object. The grasping pose depends on the grasping object itself, surrounding objects and the calculated grasp points. The advantage of this novel implementation is that it realizes a alleviation for the path planner to grasp an object fast and successfully, as illustrated in fig. 6.

inria-00325794, version 1 - 30 Sep 2008

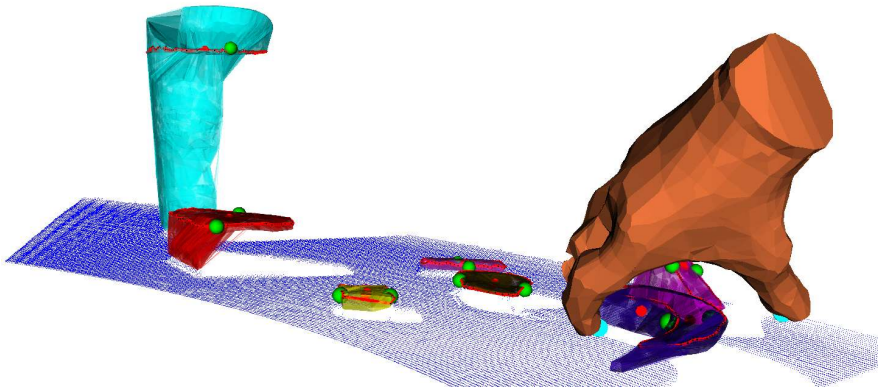


Fig. 6. The rotation axis of the hand is defined through the thumb and the index finger of the hand with the cyan colored points. This rotation axis must be aligned with the axis defined by the grasp points.

4 Experiments and Results

In our work, we demonstrate that our grasp point detection algorithm for objects with flat top surfaces shows promising results, see tab. 1. We evaluated the detected grasp points by comparing them to the optimal grasp points as defined in fig. 1. The object segmentation and grasp point detection is performed by a PC with 3.2GHz dual-core processor and it takes about 30sec. to compute the grasp points and to syntheses the calculated grasp points takes about 51sec., this calculation depends on the number of the surrounding objects and their shape. The algorithm is implemented in C++ using the Visualization Tool Kit (VTK)[5]. In testing of 10 different point clouds with the seven objects, the algorithm shows very good results, see tab. 1. For the objects no. 1, 2 and 7 in some cases the algorithm can not detect the pre-defined grasp points, because of shadows of the laser range scanner. The difference of object no. 7 to all other considered objects is that it has a concave and not a convex shape, which represents a problem for many published methods.

Tab. 2 shows the maximum positive grasping angle of every object and tab. 3 shows the maximum negative grasping angle. These tables show also the reason of the collision, which can be caused by the table, other surrounding objects or the grasping object itself. Using these values we calculate the optimal 3D hand pose to grasp the desired object, see tab. 4. The final grasping angle results as average from the maximum positive and negative grasps, where minimum and maximum angles are +/-90°. The first object illustrates that it is not ideal in every case use a vertical gripper orientation as starting position to grasp an object. The second object can be grasped with a reduced opening angle. In this

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

inria-00325794, version 1 - 30 Sep 2008

Table 1. Grasping rate of different objects.

No.	Objects	Grasp-Rate [%]
1	chocolate package	70%
2	package mozart	70%
3	eraser	100%
4	plug	100%
5	adhesive foil	80%
6	stapler	100%
7	banana	80%
Overall		85.71%

case the distance between the grasp points is about $75mm$, so the opening angle must reduced to $75mm$ with a safety distance of $5mm$ to avoid a possible collision with the object itself at the beginning. After that step the optimal hand pose can be calculated again.

Table 2. Maximal positive grasping angle.

No.	Objects	Maximal positive Grasping Angle [°]
1	chocolate package	0° (object collision itself)
2	package mozart	23° (object collision itself)
3	eraser	35° (object collision with obj. no. 2)
4	plug	90°
5	adhesive foil	90° (table collision)
6	stapler	90°
7	banana	80° (object collision with obj. no. 6)

Fig. 7 shows the positive influence of the angle adjustment. Through the calculation of the optimized grasping angle we realize a safer grasp. There is a higher distance to the all surrounding objects. Thereby it realizes a faster and safer calculation of the needed robot path with the path planning tool to grasp the desired object.

5 Conclusion and Future Work

In this paper we present a framework to successfully calculate grasp points of unknown objects in 2.5D point clouds from laser range data. The presented method shows high reliability. We calculate the grasp points from the top surfaces. The grasp point detection approach can be applied to a reasonable set of objects. This idea can be applied to every gripper type with a suitable 3D model of the used gripper.

inria-00325794, version 1 - 30 Sep 2008

Table 3. Maximal negative grasping angle.

No.	Objects	Maximal negative Grasping Angle [°]
1	chocolate package	$-2°$ (object collision itself)
2	package mozart	$-10°$ (table collision)
3	eraser	$-10°$ (table collision)
4	plug	$-5°$ (table collision)
5	adhesive foil	$-15°$ (table collision)
6	stapler	$-30°$ (table collision and object collision itself)
7	banana	$-5°$ (object collision itself)

Table 4. Optimized grasping angle.

No.	Objects	Optimized Grasping Angle [°]
1	chocolate package	$-1°$
2	package mozart	16.5°
3	eraser	22.5°
4	plug	47.5°
5	adhesive foil	52.5°
6	stapler	60°
7	banana	42.5°

In the near future we will check the quality of the calculated grasp points directly on the robot, this time we simulate the total grasping process with a commercial path planning tool from AMROSE. 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. Also the rest of the robot arm will be used in the collision detection step. Furthermore most experiments are in simulation and will be carried out to the real robot later.

References

1. Castiello, U.: The neurosience of grasping. Nature Reviews Neurosience **6** (2005) 726–736
2. Fagg, A.H., Arbib, M.A.: Modeling parietal-premotor interactions in primate control of grasping. Neural Networks **11** (1998) 1277–1303
3. Saxena, A., Driemeyer, J., Kearns, J., Osondu, C., Ng, A.Y.: Learning to grasp novel objects using vision. RSS Workshop on Manipulation for Human Environments (2006)
4. Stansfield, S.A.: Robotic grasping of unknown objects: a knowledge-based approach. The International Journal of Robotics Research **10** (1991) 314–326
5. Miller, A.T., Knoop, S.: Automatic grasp planning using shape primitives. International Conference on Robotics and Automation / ICRA **2** (2003) 1824–1829
6. Wang, B., Jiang, L.: Grasping unknown objects based on 3d model reconstruction. Proceedings of International Conference on Advanced Intelligent Mechatronics / ASME (2005) 461–466

inria-00325794, version 1 - 30 Sep 2008

inria-00325794, version 1 - 30 Sep 2008

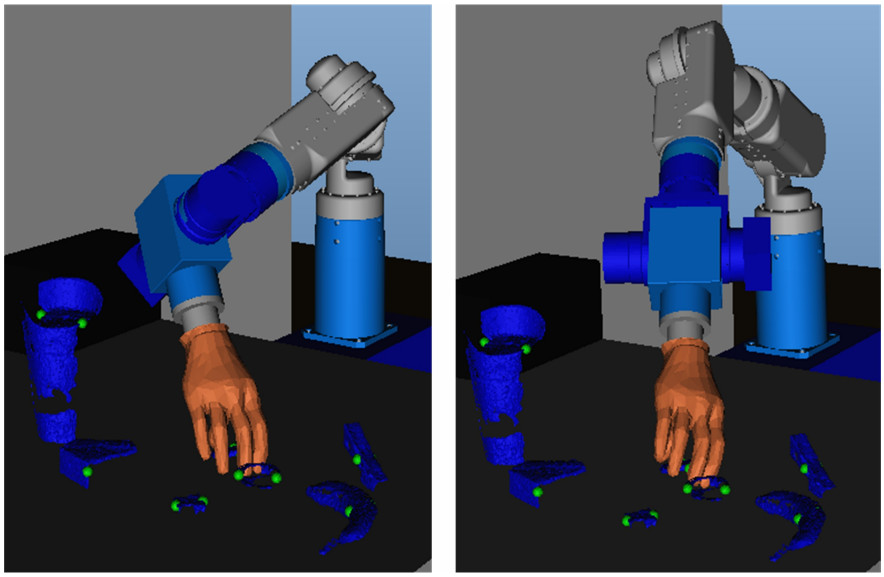


Fig. 7. The left figure shows the calculated grasping angle without an angle adjustment. It shows a higher collision risk with object no. 1 as the right figure with an angle adjustment of 22.5°, as illustrated in tab. 4.

7. Boughorbel, F., Zhang, Y.: Laser ranging and video imaging for bin picking. Assembly Automation **23** (2007) 53–59
8. Bone, G.M., Lambert, A., Edwards, M.: Automated modelling and robotic grasping of unknown three-dimensional objects. International Conference on Robotics and Automation / ICRA (2008) 292–298
9. Borst, C., Fischer, M., Hirzinger, G.: Grasping the dice by dicing the grasp. Proceedings of the IEEE/RSJ International Conference on Robotics and Systems / IROS **4** (2003) 3692–3697
10. Kragic, D., Bjorkman, M.: Strategies for object manipulation using foveal and peripheral vision. International Conference on Computer Vision Systems (2006) 50–55
11. Recatalà, G., Chinellato, E., *Á.* P. Del Pobil, Mezouar, Y., Martinet, P.: Biologically-inspired 3d grasp synthesis based on visual exploration. Autonomous Robots **25** (2008) 59–70
12. Li, Y., Fu, J.L., Pollard, N.S.: Data-driven grasp synthesis using shape matching and task-based pruning. IEEE Transactions on Visulaization and Computer Grasphics **13** (2007) 732–747
13. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395
14. O'Rourke, J.: Computational geometry in c. Univ. Press (1998)
15. Amenta, N., Choi, S., Kolluri, R.: The power crust. Sixth ACM Symposium on Solid Modeling and Applications (2001) 249–260