HIERARCHICAL OBJECT LOCALIZATION FOR ROBOTIC BIN PICKING

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea "Politehnica" din Timişoara
în domeniul INGINERIE ELECTRONICĂ
ŞI TELECOMUNICAŢII
de către

Ing. Kay Erik Böhnke

Conducător ştiinţific:	prof.univ.dr.ing. Marius Otesteanu
Referenţi ştiinţifici:	prof.univ.dr. Victor Neagoe
	prof.dr.rer.nat. Achim Gottscheber
	prof.univ.dr.ing. Vasile Gui

Ziua susţinerii tezei: 23.01.2009

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Ingineria Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică şi Telecomunicaţii
8. Inginerie Industrială
9. Inginerie Mecanică
10. Ştiinţa Calculatoarelor
11. Ştiinţa şi Ingineria Materialelor

Universitatea „Politehnica" din Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.

România, 300159 Timişoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Abstract

Computer Vision is the scientific discipline of obtaining information from two- or three- dimensional images with the help of computers, theories and algorithms. It includes fields of Artificial Intelligence, Signal and Image processing and Robot Vision, and is a growing and active field of research. Businesses have invested a lot of money into intelligent machine vision, industrial robotics and automation technology. The proposed solution of this thesis deals with industrial applications of de-palletizing or robotic bin picking. Picking a known or unknown object out of a bin with an industrial robot is called the "bin picking problem". In detail, this thesis presents a novel solution for object localization, which is the most challenging part in the whole process of bin picking. Object recognition and object localization has a long history in two-dimensional image processing. The use of distance data provided by laser range sensors offers the possibility to find objects in three-dimensional (3D) scenes. Many of the known solutions of industrial automation processes are limited to simply shaped objects or objects with specific features. However, the handling of objects with complex shapes, without any specific features, is still an unsolved problem in the field of robotic automation.

During the last decade, range image sensors for industrial environments were commercialized. They can robustly obtain distance information of a scene and provide their data within milliseconds to allow real time range image processing. Therefore, 3D imaging algorithms and range data processing need efficient implementations.

In this thesis, a pose estimation approach is proposed to determine the coarse position and rotation of a known object. Therefore, an object model is put into a virtual scene and a range laser sensor simulation creates a virtual representation of this object. The virtual appearance of this object is compared to a scene acquired by a laser range sensor. This innovative idea for 2.5D object localization handles every kind of object without the need of preprocessing, expensive feature extraction or a learn process for new objects.

After the pose estimation, the position and rotation of the object candidates are refined with an improved registration algorithm. The registration of range images is an important topic in computer vision, with many applications such as reverse engineering, mobile robot navigation and industrial visual inspection, among others. A highly efficient variant of the most commonly used algorithm — the Iterative Closest Points algorithm — finds the exact position and rotation of a known object in a scene. This research will outline an enhanced concept for robotic bin picking, demonstrate the results and put them into action.

Acknowledgements

I would like to thank the thesis supervisor, Professor Marius Otesteanu, for his advice, encouragement and constructive criticism in this thesis. I had helpful discussions and best support from all members of the faculty of Electronics and Telecommunications at the University of Timisoara. Furthermore, I would like to thank Werner Neddermeyer and Wolfgang Winkler, for their support in the initial phase of this dissertation. I also wish to thank my colleague Philipp Roebrock for his friendship, time and feedback at all stages of this research. The work was particular supported by VMT Vision Machine Technic Bildverarbeitungssysteme GmbH. Many colleagues like Frank Grünewald, Klaus Lehmann, Jan Linnenkohl, Frank Höfts, Dirk Breitenreicher and Michael Kleinkes have contributed ideas, time, effort, and support to the project as well. I am especially grateful to Professor Kay Wilding. Without the collaboration and help of students of the University of Cooperative Education in Mannheim it would not be possible to learn and teach in this field of research. A special thank to Uwe Schäfer, Heinz Hollenberg and Marc Jones for their help in proof reading of this thesis. Additional thanks to all my friends, who helped me in this period of time.

Without the confidence and support of my family, I would never have gotten to graduate school. My parents and grandparents taught me to believe in perseverance and idealism to make my dreams come true. This work is dedicated to my wife Cynthia. Her love gives me the power to make everything possible in my life, come what may.

Böhnke, Kay Erik
Hierarchical Object Localization for Robotic Bin Picking

Teze de doctorat ale UPT, Seria 7, Nr. 10, Editura Politehnica, 2009,
140 pagini, 64 figuri, 2 tabele.

ISSN: 1842-7014

ISBN: 978-973-625-812-1

Cuvinte cheie:
Vedere artificială, achiziţia de imagini tridimensionale, senzori 3D, imagini virtuale 3D, modelare, timp de întârziere, triangulaţie, estimarea poziţiei, rafinarea poziţiei, algoritmul ICP, algoritmul *progressive mesh.*

Rezumat,
Lucrarea elaborează noi soluţii pentru aplicaţii industriale de tip de-paletizare şi extragere robotizată din recipiente, bazate pe vedere artificială, care au ca scop extragerea unui obiect cunoscut sau necunoscut, dintr-un pachet cu obiecte, cu ajutorul unui robot. Pentru identificarea scenelor tridimensionale, se analizează utilizarea senzorilor 3D, bazaţi pe timpul de întârziere, respectiv pe triangulaţie. Estimarea poziţiei se realizează prin compararea imaginii 3D virtuale, obţinută prin modelare, cu imaginea 3D reală, obţinută cu senzori 3D, în două faze ale modelării scenei: estimarea poziţiei şi, respectiv, rafinarea poziţei. Rafinarea poziţiei estimate se bazează pe algoritmul ICP *(Iterative Closest Points)* şi pe propunerile de modificare a acestuia, în scopul minimizării erorii. Apoi, este propusă şi analizată combinarea algoritmului cu reţele progresive *(progressive mesh)* cu algoritmul ICP. În final, sunt măsurate precizia şi performanţele sistemului, sunt prezentate metodele de testare a estimării poziţiei, respectiv a rafinării poziţiei, precum şi concluziile, bazate pe măsurări, tabele şi grafice.

Table of Contents

List of Figures

Chapter 1 Introduction

1.1 Motivation

Picking an object out of a bin has been a learning process for mankind from early on. Imagine a 2-year-old baby taking his favorite toy car out of a box full of toy cars of different shapes, sizes, and colors; it is easy for the baby. At first glance, everyone takes this for granted. An unsupervised recognition and localization of an unknown object in a cluttered environment is still an unresolved problem in the field of robot vision. Nowadays, robots get more and more involved in industrial processes because they are superior to man regarding the requirements of strength, speed and endurance. Robotic automation processes have become very successful in the last few years, and offer a huge field for research. This dissertation deals in part with the well known "bin picking problem" [1], [2]. It is also known as the de-palletizing problem, which occurs in nearly every industrial sector. Robotics has dealt with this task for a long time, but there are only a few solutions suitable for special applications. This dissertation focuses on object localization, which is the most challenging task in the whole process of bin picking and, especially, the use of range data vision systems that are able to find objects in a three-dimensional (3D) scene with high accuracy. With the availability of fast, non-contact 3D sensors and image processing with cheap standard personal computers, 3D vision is becoming more practical for industrial applications. Solutions for simple objects and non-complex environments have been developed and realized before [3], [4], [5], but systems with general usage are often needed in industrial applications.

1.2 The problem

The determination of the position and orientation in six degrees of freedom (DOF) of objects with different forms and shapes is still a challenging task. For this reason, a general solution for the bin picking problem is hard to find. The goal of object recognition and localization is to find instances of a given collection of objects in a scene, which are acquired by a sensor. Recognition means "what" needs to be found. Often in other research areas, most known solutions are limited to simple objects with rigid shapes; sometimes to a few simple kinds of objects [1], [2]. Robotic bin picking applications, e.g. depalletization of an object, are often uniform and rigid; nevertheless, the separation of known objects in the input data can be a complex step.

Furthermore, the determination of the exact position and rotation of an object is essential in the process of robotic bin picking. Moreover, object localization is most critical and very time consuming in the bin picking process. To meet the requirements of industrial applications, the calculation time factor is the most important and, as such, has become a major difficulty in robotic bin picking. Furthermore, 3D range sensor data results in a high computational complexity. Assuming that the processing time for one 3D point in the input data takes one millisecond, it therefore takes up to several minutes just to load and process the data in the computer. This is inadmissible in most industrial applications and, as a consequence, object localization is a very important part of this thesis.

Integrating distance sensors in industrial environments leads to many new problems including sensor inaccuracies or outliers in the input data. This also depends on the measurement methods of these sensors. Outliers lead to false detection in many of the already known algorithms for object localization, so less robustness is another problem for robotic bin picking.

1.3 The solution

This dissertation proposes an object localization system, divided into two steps. The first step introduces a model-based scalable algorithm to find a coarse pose (position and orientation) of an object in a scene without the need of segmentation or feature extraction. This pose estimation includes a sensor simulation for coarse object localization. With the help of this sensor simulation, the appearance of virtually arranged objects is calculated. The resulting virtual 3D data are compared to the object's appearance in the real scene. The algorithm matches the geometry of a given model with the range image of a scene by minimizing the difference of the object's geometry. Because of this, the system is able to handle nearly all kinds of objects without any feature extraction. This universality is offset against high computational cost. Therefore, well known algorithms from the field of computer graphics are integrated to increase the performance. Depending on the position accuracy needed, the complexity of the object, the density of points in the data sets and the process time of the system can be adjusted to the requirements of the application. The pose estimation results in a number of pose candidates. In the following refinement step, these candidates are matched with an improved Iterative Closest Points (ICP) algorithm [6]. The refinement step in the hierarchical system finds the exact transformation of the model and scene data set using Progressive Meshes inside the iterative calculations of the ICP algorithms. This reduces its complexity by comparing only the reduced representation of an object model to a scene data set. The obvious advantage is an increasing performance, but another profound effect is the improved robustness against outliers. The position and orientation of the best candidate in the refinement step is transformed in the robot coordinate system, so an industrial robot can pick this object and transfer it to the target position.

1.4 The Gaps and Contributions of this Dissertation

The proposed system will be used in industrial robotic bin picking and includes — besides object localization — an adequate sensor selection, an application-invariant localization algorithm, a robot control interface, a grasp point definition and a collision avoidance strategy. The further proposed flexible coarse-to-fine object localization has the potential to meet different requirements and cover a high percentage of applications in robotic bin picking. It does not use any segmentation algorithms, but uses a mode-based alignment algorithm in a hierarchical system.

The algorithm is implemented independently from the data source. New sensor concepts with higher resolution can be modeled and integrated without any problem. All real range sensors deliver data with measurement errors, occlusions, and noise [7]. To increase the accuracy, this work will take these additional features into consideration.

The object localization is not limited to only one object because a proposed database can store as many objects as needed. Consequently, the object recognition process can be included in the localization step. The proposed solution includes a rudimentary solution for many kinds of objects. The pose estimation algorithm does not need any features extraction because of its universality. The system is also able to include feature-based approaches in the pose estimation.

New computational improvements in computer hardware, like parallel computing on Multi-Core processors and graphical processing units (GPU), enable the sensor simulation to overcome many performance problems in the pose estimation step. The fast implementation includes improvements in the refinement step. The complexity of the original ICP algorithm in the refinement step depends mainly on the number of points in the data set. With the proposed multi-resolution ICP, the computational complexity of the algorithm is reduced to a fraction of the process time compared to known ICP registration implementations.

The scalability of this system offers a great potential for the future. Starting with the coarse pose estimation, the required accuracy can be achieved by adjusting the step width in the position and orientation. Furthermore, the complexity of the object localization depends also on the chosen algorithm and the form and complexity of the object. The main problem of the algorithm is the high computational cost of using complex object models and sensors with high resolutions. Depending on the application and the available computational power used, the trade-off between process time and accuracy can be adjusted by the number of pre-calculated poses and pose candidates in the pose refinement. In the pose refinement, the use of Progressive Meshes leads to a better accuracy and robustness of the whole system.

This thesis does not include robot control and robot collision avoidance issues, even if, in some cases, these topics belong to a bin picking system. Most industrial robots provide interfaces, programming languages or tools, like working space monitoring, to handle these problems in robot controlling.

In some applications there exist much better algorithms for object recognition and object localization than the proposed algorithm of this thesis. This is especially true for objects with non-complex and easy-to-detect features. One example is introduced in the pose estimation (Section 3.6.2). In most applications, the surfaces of the objects are opaque and do not contain significantly large portions of high frequency with respect to the sensors. The proposed pose estimation algorithm will also succeed, but it will require additional process time. Nevertheless, the pose estimation can handle nearly every kind of object without further feature detection. The algorithm usually considers objects with three rotational and three translational degrees of freedom. But for featureless objects with arbitrary degrees of freedom (for example uniform cubes and cylinders), this algorithm cannot find the correct orientation in one or more degrees of freedom. Furthermore, the tests and results described in this thesis are based on rigid objects. This thesis also does not address issues related to deformable, non-rigid or generic objects. Nevertheless, the algorithm can be extended to non-rigid objects as shown in [8].

Despite the automatic adjustments for parameters in the algorithm, this system has a huge potential for optimization. Each new application will require another parameter setting, which then allows optimization of the system in its entirety.

1.5 Outline

The following chapters are organized as follows:

A general overview of the solution is presented in Chapter 2. The basic working principle is explained and the steps of the hierarchical system are defined. An introduction to range images, the data acquisition and their applications is given. Chapter 2 discusses several related approaches in the field of robotic bin picking and their relationship to one another.

Based on the overview, Chapter 3 then goes into the detailed description of the pose estimation process. Several methods are proposed for the sensor simulation in the pose estimation process. An example for a fast image-based pose estimation is given.

Chapter 4 proposes the combination of Progressive Meshes and the Iterative Closest Points algorithm of the pose refinement step. The ICP algorithm is presented in detail, along with the improvements of ICP. This chapter presents the main contributions of this thesis.

Chapter 5 gives a detailed description of the tests and results of each single step of the system. The core algorithm of the pose refinement step is compared to other state-of-the-art algorithms. The implementation details are proposed and experimental tests prove the potential for a robotic bin picking system.

The last chapter concludes this work with a summary. The contributions and limitations of this work are discussed and future capabilities and improvements are proposed.

Chapter 2 Overview and State of the Art

2.1 Solution Overview

An overview of the solution is shown in the Figure 1. The object localization is separated into pose estimation and refinement. Because the refinement process has a high computational cost, a pose estimation is introduced in order to reduce the number of possible poses. This hierarchical object localization is related to hypotheses and verification approaches [2], or a two step coarse-to-fine algorithm [9].

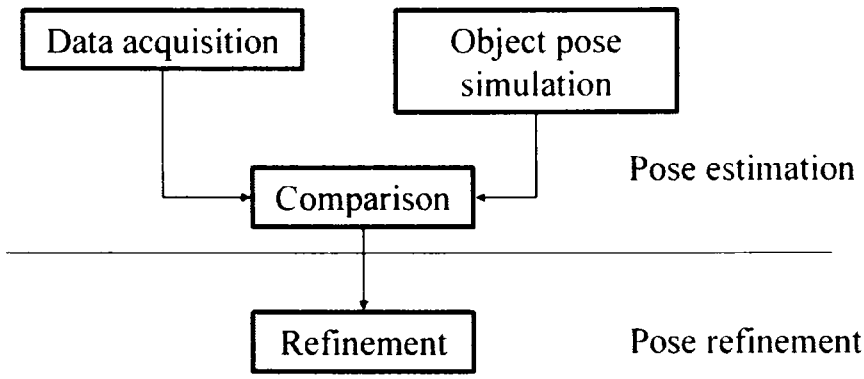

Figure 1 Hierarchical object localization and recognition
The overview shows the separation of the pose estimation and pose refinement. The acquired range data is compared to the result of the object pose simulation and the best pose candidates are refined afterwards.

Looking at the pose estimation, the input data from the range sensors is compared to the simulated range data. Therefore, a simulated sensor transfer cad aided design (CAD) models to a simulated 2.5D range image representation, taking the properties of the simulated laser range sensor into consideration. The pose estimation delivers a virtual range image of the pose of an object representation by comparing the shape appearance of every single object in the simulated scene with the real scene. Because of this, the object localization includes the object recognition and is able to handle nearly all kinds of objects without any feature extraction. The result of the pose estimation is used to define the start positions for the pose refinement. These two data sets are aligned with the Progressive Mesh-based Iterative Closest Points algorithm (PMICP) in the pose refinement step. The

determined transformation vector of each simulated object is transformed to the global coordinate system and transferred to an industrial robot to pick up the object. A short outline of the proposed system is given in the following sections.

2.1.1 Overview of the Pose Estimation

The purpose of the object pose estimation is to find adequate coarse positions of an object in 2.5D range data. This pre-selection is made to decrease the number of candidates for the refinement process. For this reason, the acquired data from the range sensor is compared to a simulated scene. One of the contributions of this thesis is the fact that features of real range sensors are taken into consideration to adapt the simulated range sensor to real range sensors.

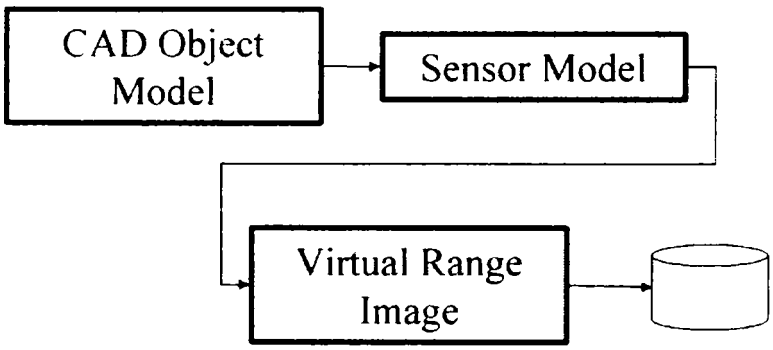

Figure 2 Sensor simulation
The CAD model is put into a virtual scene with a defined pose. The sensor model acquires a virtual range image. The virtual range image is stored in a database.

The components of the object pose estimation are shown in Figure 2. The object pose simulation creates a virtual range image (VRI) with the help of a simulated sensor and a virtual scene, therefore, placing a CAD object model in a virtual scene. In most industrial applications, CAD-models of the objects already exist. If not, the object model can be created manually. In any case, the object has to be known as a CAD-model. A common format to store CAD-models relies on triangulated points. This triangulated mesh is stored in the often used and very common STL (Structured Triangle List) file format. A big advantage of a triangulated mesh representation is the simplified calculation in the sensor simulation. The CAD-based object model is used to generate virtual range images with the help of the simulated range sensor. The sensor models adopt all properties of the real sensors. Most industrial range sensors deliver a contour so a linear movement is also necessary for the sensor simulation to get a full 2.5D range image. To compare the results from simulation to the real image acquisition process (Figure 3), the resolution of the data in the moving direction should be similar.

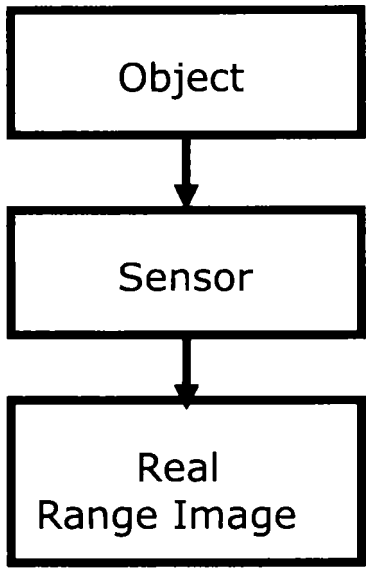

Figure 3 Sensor data acquisition
This figure shows the process of data acquisition for a real range image.

Therefore, the properties of the scanning process of the objects in the scene must be known. This includes the distance between the ground and the sensor and the direction of scanning. Moreover, all parameters and properties of the sensors must be known to form an adequate sensor model. In section 2.2.2, range sensors are introduced are introduced to highlight their characteristics and them to the sensor model. With its simulated geometric configuration, the sensor model determines the distance between the object and the laser source according to the measurement principle. If a CAD-model is put in the measurement range of this virtual sensor and the virtual sensor is moved over this model in parameterized steps, a virtual 2.5D range image is produced in this virtual scanning process. This virtual range image (VRI) contains one surface of the 3D-Model, and shows this surface of the model; the surface normals are orientated towards the sensor. For every possible position and orientation of the object model, a VRI is produced. The process of virtual scanning is very time consuming, therefore, it is necessary to generate all range images offline. This VRI is indexed with a known position and orientation of the model in the sensor measurement space and stored to a database, otherwise known as the VRI database. This contains scanned models in the form of range images for defined positions and orientations. Depending on the application, the positions and orientations, the system needs exorbitant space to store these range images. In most cases, the degrees-of-freedom are limited and only VRIs of defined step widths are stored in the database. In general, though, this process must be done for all required positions and orientations for every kind of object. A VRI is generated for every position of every object in the scene. These positions differ only by steps of a few millimeters, depending on the resolution of the real sensor and the size of the real object and the scene. For every position, there exist three possible rotation directions for the object. Due to the limit of

computer power and storage, the proposed simulation is limited to "coarse" or "rough" poses. The number of the coarse poses in the database mainly depends on the stipulated accuracy, the desired robustness, the sensor resolution and the resources of the system. The step of object recognition is integrated by comparing VRIs of different kinds of objects to the real scene in the same way. Every VRI is compared with the real scene with a defined error function returning an error value. If the error value is low, the VRI matches with the scene. Because of the fact that all VRIs are compared to the scene, each VRI receives an error value. All of these error values of VRI pose candidates must be lower than an error threshold, which depends on the complexity of the simulated object, the used sensor (with its sensor errors), outliers and invalid points. The simulated object poses with the lowest error values are selected for pose refinement.

2.1.2 Overview of Pose Refinement

In the previous section, the coarse pose estimation creates an error value for every pose. The best VRI candidates were chosen and used as inputs for the pose refinement to find the best matching candidate. A procedure for matching two range images (VRI and scene) to identify two sets of corresponding 3D points is presented by [10]; utilizing a quaternion-based non-linear optimization method for this purpose. If the corresponding 3D points are assigned to each other correctly in the data sets, this method produces the correct transformation matrix, including translation and rotation. If no transformation is known a priori, the Iterative Closest Point (ICP) algorithm [6] provides a method to find the correct correspondence of the 3D points in the data sets. The Iterative ICP algorithm is a common registration method used to transform the view of an object to another view of the same object or another. Important works for the classical and most commonly used the Iterative Closest Point (ICP) algorithm are [11], [12]. Due to the slow convergence speed, ICP was improved by many researchers [13]. As the name already implies, ICP is an iterative algorithm which determines its parameters with the help of a mathematical minimization. The algorithm searches for each point in one data set, the closest point in the other data set, and uses the corresponding point pairs to compute the relative transformation between the scans. The algorithm minimizes the mean square error of the point distance in several iteration steps. This process is repeated iteratively until the relative position of the scans converges. To abort the iterations, a threshold or a maximum number of iterations is implemented. One iteration of ICP is separated into three steps. At first, every point of one data set is assigned to the Euclidean closest point in the other data set. The elements of the normalized eigenvector of the largest positive eigenvalue of the constructed matrix (including the cross-correlation matrix) correspond to the elements of the unit quaternion representing the best rotation. With the help of the chosen energy function, the result is a rigid transformation in the used coordinate system. An iteration step finishes by applying the resulting transformation to one data set. The algorithm converts monotonously to a local minimum. Therefore, the knowledge of an approximate initial solution is important for the success of the method, which is done by the pose estimation. ICP is one of the most popular registration algorithms and real time implementations [13], [14] show the potential of the ICP algorithm. In this way the ICP algorithm is used for every VRI candidate. The resulting error value after applying ICP is calculated. The best VRI candidate is selected and the object coordinates in the sensor coordinate system are used as the final result in the object localization step. This pose refinement is introduced in detail in Chapter 4.

2.2 Range images

If you think about the history of pictures and photos, you would agree that pictures play an important role in our life. Especially when one considers motion pictures or videos, one is flooded with information almost daily. For human beings, it is easy to understand pictures — we easily separate and recognize objects in a scene, extract important information from it and create a context and a story for ourselves within milliseconds. Even with the incredible growth of computational capacity and power in the last years, computers are not able to understand images in every context. One of the main problems in the research field of image understanding — compared to the human data acquisition process — is the lack of three-dimensional (3D) data. We are able to estimate depth easily by using the information given by motion or the disparity between the two images seen by our eyes. The flexibility and capability of human beings in face detection is still a secret and challenging task in related fields of research like artificial intelligence and computer vision. Human beings are able to recognize a person from a picture taken from a view from which they have never seen this person before. In the same way, we have no problem in recognizing and picking up an unknown object from a cluttered environment. Consequently, the interest in range images for high-end research projects and applications has increased dramatically in the last decade. An image provided by a common camera depicts the intensity distribution of the scene without any 3D data. One way to capture 3D information is the ability to directly acquire range images with laser range sensors. These sensors deliver a discrete representation of the surface in the scene, which offers a greater chance for computers to increase the level in image understanding. This thesis focuses on close range laser sensors, but range imaging also deals with data acquired from radar, sonar or stereo vision. Many other fields of research deal with range images. This work compares three-dimensional data with 2.5D range images. 2.5D data is characterized by the fact that for every image point in the X-Y plane, only one distance value exists. Some authors use the term "3D" for range data. In this work, the terms "3D" and "2.5D" are different due to the fact that the proposed sensor simulation converts 3D models to 2.5D data.

2.2.1 Fields of research of range images

Systems that simultaneously acquire scene information, localize themselves in the scene and use this information to build maps have become popular in the last years under the keyword simultaneous localization and mapping (SLAM). This active field of research is driven by the mobile robotic community [15]. There exist many desirable aims for SLAM, up to applications that need free navigation in unknown environments like the current Mars expeditions or DARPA Grand Challenge [16]. In many cases, range sensors are needed to acquire range images of the environment [17], [18], [19].

Range images of architectural elements and sculptures [20], [8], paintings and other archaeological objects provide data for archival documentation. This can be used for a variety of research, conservation, archaeological and architectural applications, for fabricating accurate replicas as well as for interactive museum displays and Virtual 3D applications[21]. This can be extended to acquisition and modeling of archaeological artifacts and architectural historical buildings in the field

of photogrammetry.

In the same way certain buildings in a town are acquired and recognized for projects like[22], the task of producing detailed terrain maps [23] or 3D city models for city planning, visualization and landscaping has become important in recent years. Consequently, acquiring 3D representation of buildings (up to whole towns [24]) is an active field of range imaging [25]. Geographic information and 3D models of environments are interesting for military purposes as well as commercial maps [26], for navigation or other applications like Google Earth [27].

Even forensic analysis [28], crime scene investigations [29] or traffic accident reconstructions [30] belong to the field of range image applications. Many car manufacturers need detailed 3D information in car crash tests. For example, documentation of traffic accidents is getting more and more important in judiciary cases.

In medicine, image processing systems have been more accepted in the last few years. These systems are often used for evaluation as well as digital images. The data are produced by Magnet Resonance Tomography (MRT), Computer Tomography (CT), Positron Emission Tomography (PET) or from classic radiographs, ultrasonograms and range images acquired by stereo vision or range sensors [31]. In telesurgery or endoscopic surgery, the determination of the body position and surgery tools are implemented using range measurement systems [32], [33].

This field is highly related to many other applications in virtual reality [34]. Extraction of Computer-aided Design Models (CAD-Models) from range images is often used in reverse engineering [35]. Industrial systems measure tolerances of manufactured parts [36], [37] or industrial robots find the positions and sizes of objects to guide them to target positions, e.g. for manipulation purposes. An almost classic application in the industry for laser range sensors is the measurement of the volumes and geometries of products of (endless) production lines [38]. The geometry and volume of the sealing mass of car windows in the automobile industry is checked with the help of laser range sensors in 2D[39], directly after being applied by the corresponding device.

2.2.2 Range data acquisition

One important contribution to the proposed system is the fact that the proposed sensor simulation takes properties of commonly available laser range sensors in consideration. Therefore, it is necessary to introduce different types of range data measurement principles in this section. This section focuses on measurement principles of laser sensors available for industrial environments. The properties, measurement principles, the setup, operation modes of available laser range sensors are introduced in [40].

The evaluation of depth information from different sources is a very popular field of research in data processing [41]. Many fields of business and science, such as the automobile industry, medicine and physics, are interested and involved in it. Depth information results from a number of different sources — they vary from a simple distance measuring system to complex 3D scanners for any size of objects. The non-contact approach is the most important aspect of visual range

measurement methods. This allows for the measurements of substances which may be hot, chemically aggressive, sticky or sensitive, provided that sufficient light is reflected back from the surface. There is no possibility of any damage or wastage to the object. In addition, they are relatively fast and economical. On the other hand, visual non-contact methods are vulnerable against transparency and multiple reflections. Different methods exist for the visual data acquisition and even range data is obtained in many different ways. In general, the range data acquisition is separated into two categories — active and passive range imaging, respectively [42]. In the passive method, no special light is required except for the ambient light for illumination. The most common data sources for industrial applications are still passive camera systems. Cameras provide a two-dimensional projection of a scene, so no depth information can be obtained without any further processing [43]. The most common passive range imaging technique is stereovision. Sensors using other techniques like structure from motion, focus/defocus, shading and photometric stereo [41] are either not commonly available for industrial applications or not suitable for real time range acquisition. The basic concept of stereovision is triangulation: when knowing the baseline and the angle between the two cameras, the distance to the object can be determined with the displacement of objects in the two images. These systems are required to solve the so-called correspondence problem. The solutions of the correspondence problem try to determine which pairs of points in two images are projections of the same point in the real world. Usually this is a very complex problem and the solution is computationally expensive[41]. Most common depth measurement systems use two or more cameras (i.e. *multiview vision*) to acquire distance information. Unfortunately, many camera-based solutions for range data acquisition have problems with their robustness [41] and sensitivity to lightning conditions. More information can be found in [44], [4].

In active range imaging, a dedicated and well defined light source (e.g. laser light source) is used in cooperation with a visual capture device. In [40] non-contact industrial laser range sensors are introduced. At the moment, these active sensors are superior to other industrial measurement methods regarding their accuracy, costs and robustness compared to stereo camera systems [2]. The well known methods "time-of-flight" (TOF) and "triangulation" are part of the active methods. In the active triangulation scheme, the scene is illuminated by a laser source from one direction and viewed by a sensor from the other direction. TOF measures the time of a reflected laser pulse to determine the distance to an object. The advantages of the active methods are the production of dense sampling points and the high robustness and precision compared to the passive methods. However, additional light sources must be added in the scene and the methodology does not correspond to human stereo vision. Figure 4 shows the variety of different measurement technologies. TOF and phase measurement methods are long range technologies (over 1 meter) and triangulation-based methods belong to close range methods. Most long range measurement sensors are used for surveying and mapping in architectural and cultural heritage, geodesic laser scanning, archaeological heritage conservation, and the 3D Scanning of buildings.

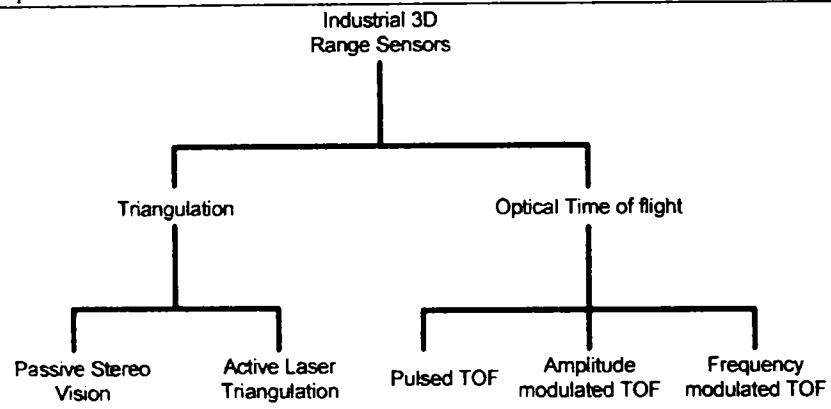

Figure 4 Industrial 3D Range sensors
Industrial range sensors are separated into two measurement principles. The triangulation can be realized as passive stereo vision and laser triangulation methods. Time of flight methods send out a pulsed or modulated signal to measure distance information.

Active close range 3D sensors are often used in quality management, reverse engineering, visualization and 3D modeling, and have become one of the major aspects of computer vision and robotics. This work focuses mainly on laser range sensors, which deliver two-dimensional contour distance data. Depending on an industrial application of the sensors, different measurement ranges have to be selected. Therefore, a direct comparison of the properties is hardly possible. The choice of the measurement range, rather, plays a subordinate role for the data representation in this thesis. Distance sensors provide simple distance values. But besides distance values, the intensity values are provide by many sensors and could possibly help to identify the properties of the surface [45]. The intensity values of the points are often used for the filtration of the point clouds and for the detection of sensor errors [46]. To increase the accuracy of the scans, the measurement rates often have to be decreased and, if possible, the distance between the object and sensor has to be reduced. A short overview of possible data acquisition devices available in the market is presented in [40]. The next sections briefly describe the different principles of depth measurement and laser range sensors that are already available in the market for industrial environments.

2.2.2.1 Active time-of-flight laser range sensor

Time-of-flight (TOF) laser distance sensors measure the distance between the object and the light source along a light beam. Time-of-flight (TOF) systems send out a light beam towards an object. The light is diffusely reflected by the surface and a part of the light returns to the receiver. The time that light needs to travel from the laser diode to the object surface and back is measured. When the light pulse is emitted a high accuracy stopwatch is started. The light pulse travels to the target and back to the receiver. When the light pulse arrives, the stopwatch is stopped and the time of the flight is calculated. With the known speed of light the distance to the object is determined.

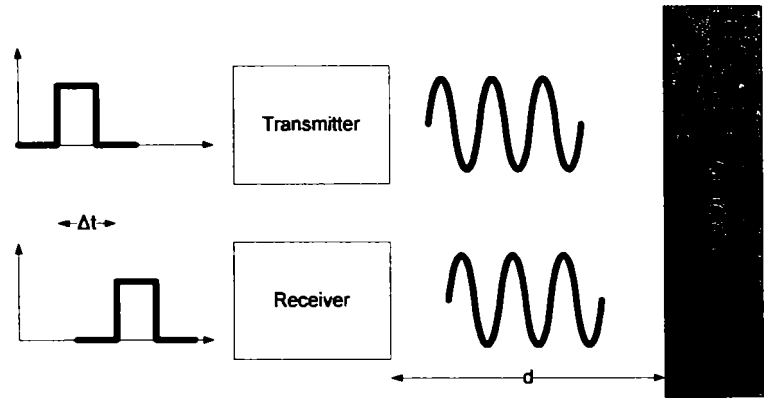

Figure 5 Pulsed TOF measurement principle
The transmitted signal is reflected by the object. The delay between transmitted and the reflected signal results in the distance.

Figure 5 shows the TOF configuration. In practice, the active light source and the receiver are located very closely to each other. Illumination and observation directions are approximately collinear, so this avoids shadowing effects.

The existing methods relying on the principle of TOF can be separated in the following categories [47]:

- Pulsed time-of flight
- Amplitude Modulated Continuous Wave
- Frequency Modulated Continuous Wave

In the case of pulsed TOF, the travel time is directly proportional to the distance traveled, taking into account the velocity of light in the involved medium. This velocity may be easily derived from the velocity of light in a vacuum. This distance measurement is possible since the speed of light is known very precisely: c = 2.99792458 * 10^8 m/s.

The total time needed by the signal to travel from the source to the scene and back is calculated using the following equation:

$$d = \frac{c\,\Delta t}{2\,n} \qquad 2.1$$

It applies here: c is the velocity of light and Δt is the time taken by the signal to travel from the source to the object and back. The involved medium is integrated as the refraction index n. The equation contains a factor of ½ because of the way to the object and back. Theoretically, the accuracy of the depth measuring is independent of the distance of the object to the camera and only depends on the precision of measuring the travel time. But precision in the millimeter and sub-millimeter range requires pulse lengths of a few picoseconds and the associated electronics. Mainly, the pulse rate influences the maximum range for TOF sensors.

To send out a new pulse, the receiving unit has to wait for the last echo arriving from the object. Some long range sensors use the pulsed TOF method to measure distance up to a few kilometers for cartographic mapping [48]. At ranges of a few kilometers and above, a different problem arises: at such distances the amount of reflected photons that reach the detector is very small. The sensitivity of the receiving unit and the power of the emitted light pulse are limited in all real range sensors. This leads to a limitation of the range of the sensors.

A variation of the time-of-flight distance measuring is the measuring of the phase shift. This method effectively measures the difference between emitted and received signals. A continuous wave (CW) laser emits light continuously and, therefore, is called a CW-laser.

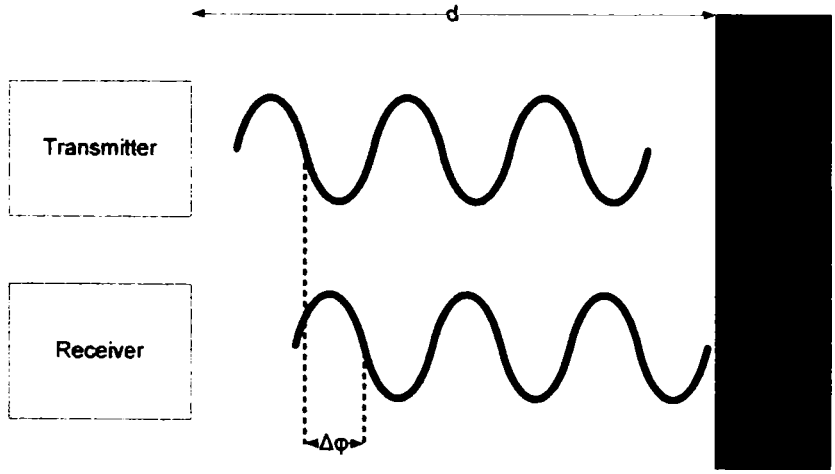

Figure 6 Phase shift TOF measurement principle
According to the difference in the phase between the transmitted and received signal the distance to the object can be determined.

The distance information is extracted from the received signal by comparing its modulation phase to that of the emitted signal.

With

$$\Delta t = \frac{\Delta \varphi}{2\Pi} T_m \qquad 2.2$$

the distance can be calculated in consideration of equation 2.1:

$$d = \frac{c\Delta \varphi}{4\Pi n} T_m = \frac{\lambda_m}{4\Pi n} \Delta \varphi \qquad 2.3$$

The range of phase measurement TOF sensors depends on the wavelength of the modulated signal so the resolution of these sensors can be improved if signals with short wavelength are used. That said, this leads to a reduced maximum range of phase shift measurement. The maximum unambiguous detectable phase delay is

a full cycle of the modulation period. For phase shifts over 360°, however, an unequivocal determination of the distance is not trivial, which means that the maximum useful measurable distance is half of the distance traveled by light during one period. This continuous wave can be modulated in the amplitude or the frequency. An amplitude modulated continuous wave (AMCW) is often a sinusoid wave and this wave is modulated in amplitude by varying the power. Frequency modulated continuous wave (FMCW) distance measurement is achieved by measuring the phase of the modulation of the transmitted light.

The reflected wave must arrive with adequate quality to calculate the right distance. Sometimes this problem leads to non-valid measurements in realized industrial sensors. Phase shift measurement has a higher precision than that of conventional TOF measuring. In practice, a combination of these two procedures is often used. This method is typically used for measurement distances of a few tens of meters. The TOF principle implies a distance measurement in one dimension along a laser beam. To acquire a full surface most industrial laser range sensors "scan" the surface by a defined displacement of the laser beam. Most of the available TOF laser sensors provide a two-dimensional contour scan. Therefore, the beam is moved incrementally with a parameterized angle step width and that results in a radial distance contour. For each angle, a distance value is measured. If the scanning sensor itself is moved perpendicular to the laser beam fan, a full surface can be acquired. Other sensors consist of many laser emitting and receiving units in one sensor system [49], [50] in order to acquire a full range image. TOF sensors provide big advantages regarding the accuracy and resolution of measurement ranges up to 100m. One of the major advantages of TOF is that it is free from the corresponding problems of passive triangulation and the range ambiguities of the passive triangulation. They are, however, less accurate, especially for close range measurements. The accuracy is between a few millimeters and two or three centimeters, depending on time measurement and on the distance between the object and the scanner (object distance). The basic problem of the TOF system is the realization of a high accuracy time measurement at the current state of realization.

2.2.2.2 Triangulation-based sensors

The principle of triangulation is based on simple geometrical constraints. An active triangulation system consists of a light source and a receiving unit. In the case of laser triangulation a laser diode emits a laser beam with a defined angle toward the object. The surface of this object reflects this beam to the receiver. The distance from the sensor to the object is geometrically determined from the recorded angle and base length. The base length between the laser source and the receiving unit is known from calibration. Depending on the resulting dimension, the active triangulation methods can be separated as follows:

- Single Spot Triangulation
- Sheet of Light Triangulation
- Coded Light Triangulation

There are triangulation-based sensors existing that deliver one-dimensional (1D), two-dimensional (2D) and range image (2.5D) data.

Single Spot Laser Triangulation is based on simple trigonometric equations. A laser spot is projected onto the object. The scene is recorded with a CCD array. If the distance changes to the laser, the position of the reflection in the CCD array also changes. Due to geometric relations, the changed distance can be calculated the other way round.

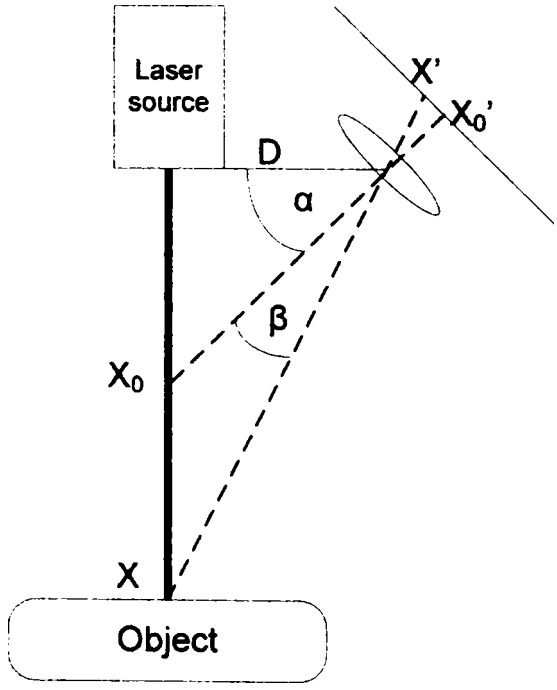

Figure 7 Triangulation principle
The distance x to object is calculated with the help of geometrical constraints and the known distance between camera and light source.

Figure 7 shows the tangent law:

$$\tan(\alpha + \beta) = \frac{x}{D} \qquad 2.4$$

After applying the addition theorem of tangent and dissolving the equation to X this leads to

$$x = D\frac{\tan(\alpha) + \tan(\beta)}{1 - \tan(\alpha)\tan(\beta)} \qquad 2.5$$

with $\tan(\alpha) = \frac{x_0}{D}$ and $\tan(\beta) = \frac{x' - x_0}{f}$ (see Figure 7).

The equation can be written as:

$$x = D\frac{\frac{x_0}{D} + \frac{x' - x_0}{f}}{1 - \frac{x_0}{D}\frac{x' - x_0}{f}} \qquad 2.6$$

Figure 7 shows the configuration for a reflected laser spot and a CCD-array, which can be used for determining a one-dimensional distance value. The accuracy (usually ~1:1000) depends on the distance between the laser and receiving unit and the object distance. Active triangulation is usually used in measuring a range of 0.1-5m [40]. For larger distances, the distance between the laser and receiving unit must be increased. This leads to a vulnerability of occlusions. Measurement times of less than 10ms are common, allowing real-time study of moving or vibrating objects. The angle range is important too — smaller angles lead to bad sampling and large angles lead to occlusions, so the angle of emitted beams ranges usually between 15°-30°. To get the range image data (2.5D) of an object, a single beam is scanned over the scene. Therefore, only one range value is acquired for each position of the arrangement. To acquire an image of size $m \times n$, just as much integrations and measurements are necessary. Single spot triangulation is a triangulation technique, but the use of only one sensing element often reduces the speed for 3D scanning. Active triangulation can also be extended to a laser line and CCD-matrix, resulting in a two-dimensional distance array.

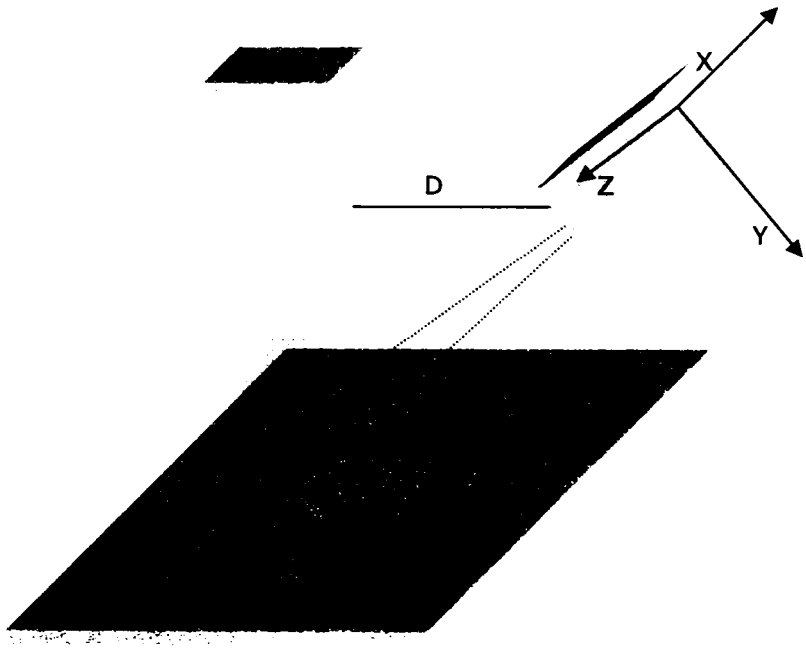

Figure 8 2D Triangulation principle
The 2D extension is characterized by the fact of the camera and the sheet of light which is sent out to the scene.

Figure 8 shows a triangulation system to acquire a fully two-dimensional profile. A camera captures the projected line. With the help of the geometric configuration the distance can be acquired. For each column X_i in the camera matrix, the geometrical considerations (Equation 2.6) of single spot triangulation are applied. The reflection of a stripe of light, which is generated, for example, by passing a laser beam through a lens, is analyzed by a CCD matrix camera. For a full 2.5D image, many measurements are needed, while the system is moved relative to the object. This system needs no internal moving parts like scanning TOF sensors.

A further method of triangulation sensors belongs to structured or coded light techniques. A coded pattern — such as a gray coded or phase-coded pattern — is used to illuminate the scene for acquisition. In some industrial applications, structured light approaches are realized [51]. For the acquisition of 3D scenes, no scanning or moving profile sensors are required, so this method is usually faster than other 3D scanning techniques.

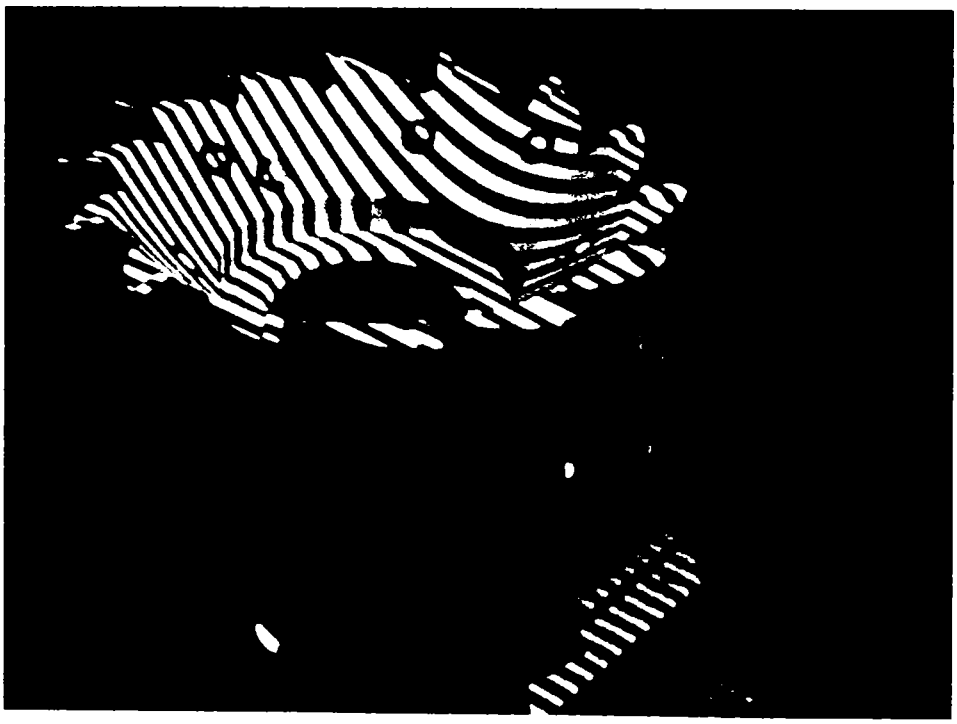

Figure 9 Structured light data [52]
By changing the light pattern in time the scene can be fully reconstructed by many camera images. The position of the camera and the light source is known before.

An example of structured light acquisition is shown in Figure 9. With the help of the curvature of the projected lines, 2.5D information is calculated. One of the most important disadvantages of structured light approaches is the sensitivity to ambient light and the limitation of the measurement range. A detailed description of structured light can be found in [53]. With the help of stereo cameras or the solution of structured light sensors, distance values can be determined [44]. In the last few years, the accuracy of structured light range data acquisition has increased

up to 1µm [52]. More and more companies offer some promising solutions. Unfortunately, this measurement technique still suffers from ambient light influences, complex calibration and the lack of a ready-to-use solution for industrial environments. Therefore, the focus is on laser triangulation range sensors.

Several 2D triangulation-based laser range sensors are available for industrial applications which are reviewed in[40]. Most of these are close range sensors with a laser stripe source and a camera inside a fixed frame without the need of calibration. Furthermore, a special high speed camera can be used in combination with a laser source. The optimal setup for the desired application must be found and calibrated to create such a 3D laser system.

2.2.2.3 Geometrical triangulation setup

The principle of the triangulation is extended in common laser range measurements using a high-resolution CCD/CMOS camera and a line projecting laser. With this extension, a complete height profile of an object can be determined with the help of a camera from the displacement of a laser line. The height information of the object can be calculated from this laser displacement. To get the correct distance values of such a camera system, the exact geometric construction, the environment and parameters should be known. This chapter introduces the commonly used camera/laser source setups and their influence to measurement accuracy and calibration. In general, the closer the camera is to the object, the more precise are the measurement accuracy and the resolution, respectively. With a closer distance to the object, a high-quality suitable camera lens is needed. This is important for small field-of-views for achieving a good vision measurement with sharp images and low distortion. Considering the geometric setup of the camera and the laser, the camera and the laser should be mounted so that the laser illuminates the object from one direction, and the camera views the object from another direction. In addition, if the camera looks at the object in a small angle (see Figure 10), the measurement pixel range is too small. This leads to a lesser height resolution in general. The larger the angle of the camera, the larger is the measured range in pixels. Smaller angles lead to bad sampling, and large angles lead to occlusions. In most cases, the angle ranges from 15° to 65°, referring to Figure 12 (reversed ordinary setup).

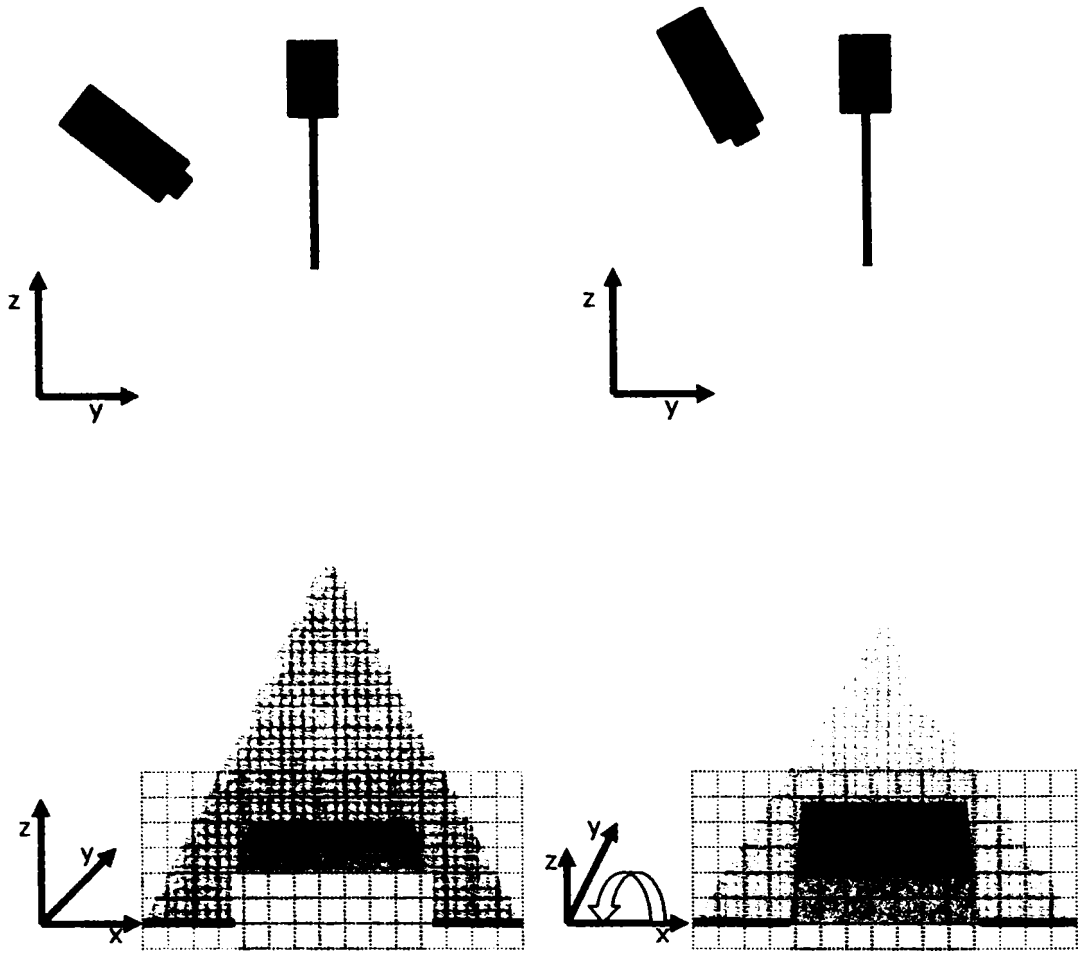

Figure 10 Angle of Triangulation setup (Reversed Ordinary)
The setup can be changed by changing the angle between the camera and the light source in a 2D triangulation.

The geometric setup between the laser and camera is probably the most important influence factor on the measuring precision and height resolution of the camera system. Mounting the camera and the laser in the right way depends on different requirements and environmental characteristics. At first, the object itself influences the setup by its surface features. The types of objects range from transparent, glossy or matt in every color and with different shapes. Additionally, the type of measurement plays an important role because common distance measurements can be combined with gray-scale and scatter measurements. Each measurement requires defined resolutions and region of interests. According to [54], in order to counter these different types of requirements, there are at least four main principles for mounting the camera and the laser.

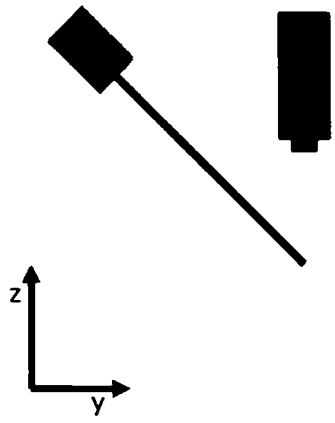

Figure 11 Ordinary setup
The camera is placed on top of the object and the light source emits the laser line in a defined angle rotated around the orthogonal axis of the transportation direction.

The *ordinary setup* shown in Figure 11 achieves the maximum height resolution. In this setup the camera is placed over the object, perpendicular to the direction of movement, and the laser projects a laser line from the side. This geometry gives the highest resolution but suffers from miss-register, i.e. the laser line in the camera image moves to a different Y coordinate if the distance value changes. This requires a higher computational complexity to reconstruct the distance measurement information.

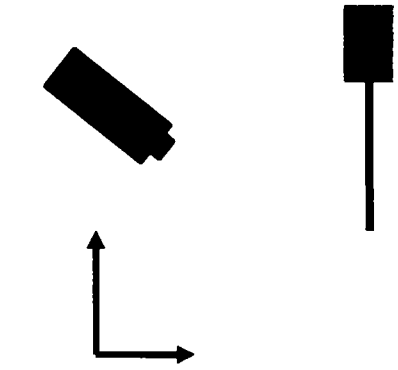

Figure 12 Reversed Ordinary setup
The laser sends out the laser beam orthogonal to the transportation plane.

In the *reversed ordinary setup* (Figure 12) the laser is mounted over the object. The camera is located in the same direction, but is moved with a certain angle while looking at the projected laser line. This measuring setup has a good

height resolution and does not suffer from miss-register, so this setup is the most common one used. This setup is similar to the ordinary setup, but the placement of the laser and the Ranger has been switched.

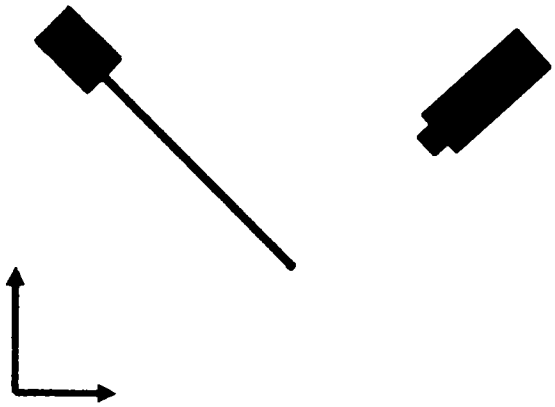

Figure 13 Specular setup
The laser beam and the camera have full reflection.

The camera and the laser are mounted opposite to each other. Through the high intensity of light emitted by the laser, the *specular setup* (Figure 13) is useful for dark or matt object surfaces. As a disadvantage, occlusion areas arise in front of and behind the object. The setup offers a lesser height resolution and is often used for surface analysis applications.

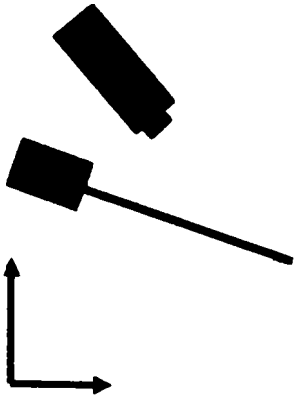

Figure 14 Look away setup
The camera and laser are looking from the same side.

As with *specular*, *the look away setup* is not suitable for range measurement applications. The camera and the lighting are mounted on the same side (Figure 14). A strong light source is needed and occlusions also occur with this setup, but

this setup avoids reflections and can be used for surface analysis and inspections.

The most important factor is the geometric setup, i.e. the distance between the laser and receiving unit and the object distance for the measurement accuracy. For large distances to the object, the distance between the laser and receiving unit must be increased. In turn, this leads to a vulnerability of occlusions. Therefore, triangulation-based sensors are commonly used in close range environments of up to 1-2m.

2.2.2.4 Extrinsic calibration for triangulation-based laser range sensors

The task of the extrinsic calibration process is to find the transformation between these two coordinate systems, and is described in this section. The required intrinsic camera calibration is supposed to be known before [55]. To obtain calibrated measurements (e.g. coordinates and heights in millimeters), the calibration algorithm has to transform the point information from the sensor coordinate system (row, column, profile number) to world coordinates (X,Y,Z)[56]. This transformation depends on a few factors, for example, the distance between the camera and the object, the angle between the camera and the laser, and the properties of the lens. The distance and the angle are variable to ensure flexibility for many application cases. The calibration is done exemplarily for a reverse ordinary setup (see Figure 12) and the position of the camera is not fixed for the measurements. The angle of the camera is measured from the normal transport direction to the axis through the center of the lens. The previous sections show how the height profile information of the laser line can be acquired from the camera. These values do not contain the actual height of the object but the height of the laser line as a number of pixels. The points along the laser line are given as X coordinate according to Figure 10. The X axis must have the same direction for the sensor image coordinate systems and the world coordinate system. The location of the point in Z (distance to the camera) must be calculated by the brightest pixel in one column of the picture. The location of a point along the transport direction (Y coordinate) is represented by the sequence number of the measurements.

The calibration process connects the height profile acquired from the camera with the real size of a reference object. The used reference object is shown in Figure 18 laying on a conveyor belt. One measurement results in a profile. This original profile is an idealized rectangle (left in Figure 15) of the rectangular reference object. Depending on the geometric setup, the resulting measurement consists of a deformed profile.

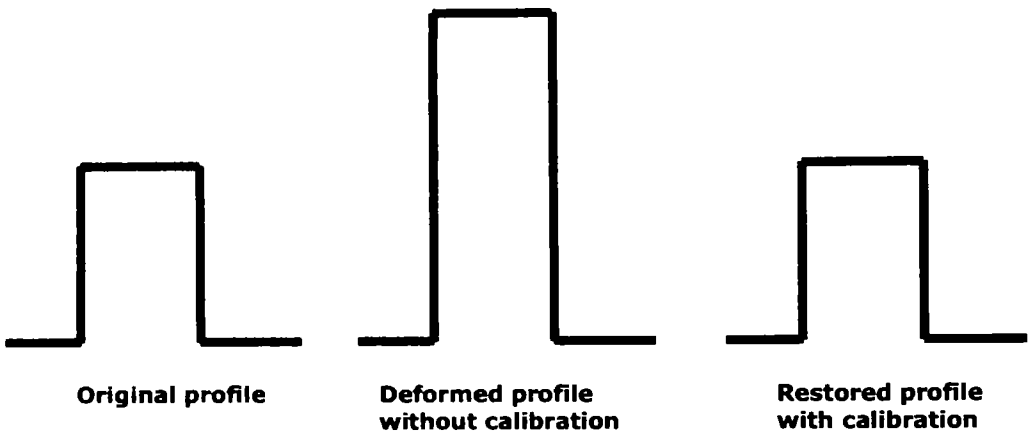

Figure 15 Reference object profiles
The deformed profile (in the middle) is restored (right) to the same shape of the original profile (left).

The calibration corrects the distortion between the heights and width of the measured profile to the object in its real height and width. The calibration algorithm consists of different steps and has to be done only once for the setup. If the camera has to be moved for any reason, the calibration measurements must be made again. If an unknown object is scanned with the calibrated setup, the measurement results are multiplied with the scaling factors and the object is represented correctly.

To correct the distortion, two scaling factors are calculated. The calibration process [57] consists of the following steps:

- The reference object is put in the camera's field of view and the image is acquired
- The profile of the laser line is extracted
- The scaling factors are calculated

The calibration method bases itself on a rectangular calibrating object with known height and width.

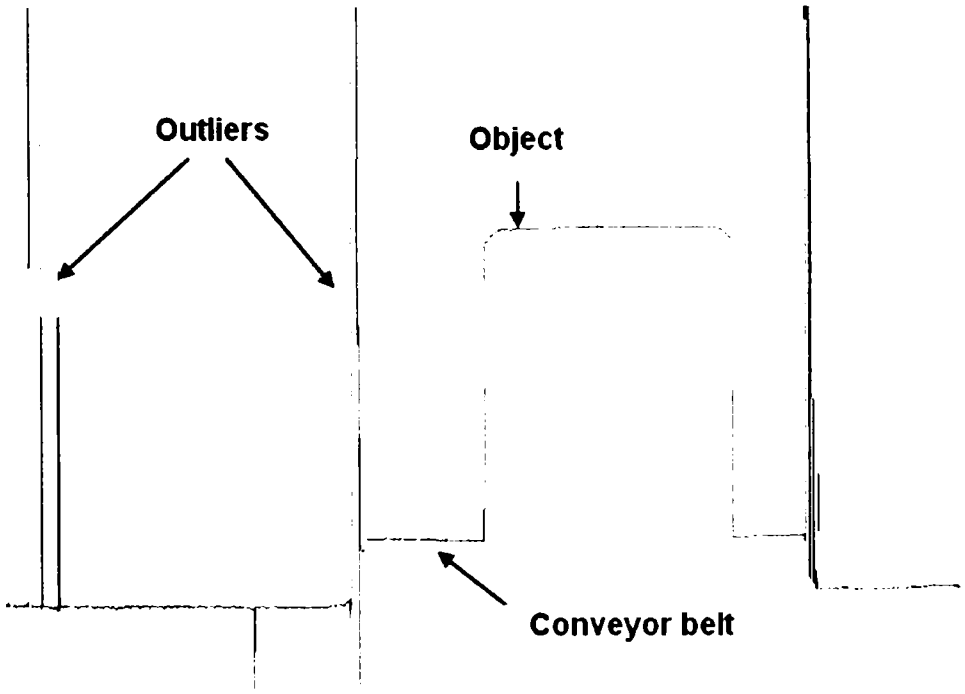

Figure 16 Outliers in reference object profile
Outliers occur from occlusions as well as non reflection on the object surface.

After the image is acquired, the height profile is extracted. Therefore, the maximal brightness value for each column in the image is selected. The profile is the array of the X position with the maximum gray-scale level in every column. A signal filter must be applied to this signal to be able to determine the object size robustly. The original measurement signal also contains some outliers. These outliers lead to wrong edge detections because of their high gradients, and occur in the recognition process of the laser line. In some cases, the algorithm cannot determine the right value in the column because the laser line is hidden from the camera view, so a completely incorrect height value will be determined because of other reflections. If the minimum brightness threshold of the laser line is not reached, the X value in this Y column is set to the maximum value, which is shown in Figure 16. The derivation values of these outliers are possibly higher than the derivation values of the edges. For this reason, a measurement filter that filters the outliers from the signal is necessary. A median filter is a very good non-linear digital filter for extreme outliers and is often used in signal and image processing. A window is used to select n values from the signal. The values in the window are sorted into numerical order. The number of window values is always odd, so the center value (median value) of the window is selected as the resulting output value. If the edge of the rectangular signal is not smoothed, the median is an edge-preserving filter. Another advantage of the median filter over averaging filters lies in its non-linearity. The height of the outlier does not have any influence on the result value. A ten-times greater value of the outlier provides the same result as one with a thousand-times greater value. A standard median filter is implemented in the calibration with a window size of 17 values. With this window size, an outlier width of up to eight pixels can appear without any consequences for the measuring. This leads to a

smoothed measurement signal of the reference rectangle, as shown in Figure 15.

To be able to determine the object width, the edges of the rectangle must be detected and then their distance must be calculated in the first step. The signal is treated as a mathematical function. The high height difference at the edge of the rectangle can be determined with a derivation. The gradients at the edges are very high, so the edge can be determined by the derivation of the signal and by applying a threshold for the minimum and maximum gradients.

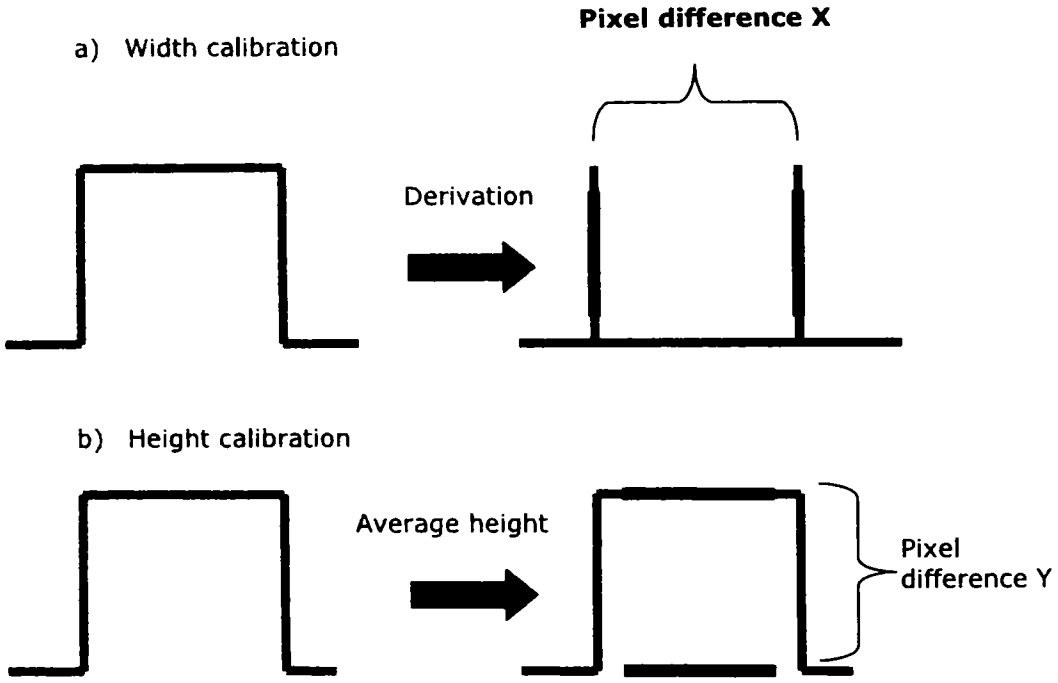

Figure 17 Edge detection in height profiles
The calibration object provides the pixel difference in a) height and b) width.

After the edges of the measurement signal are known, the object height is determined. The calibration algorithm averages the signal between the two detected object edges. To avoid possible rounded corners or signal delays, the algorithm rejects a fixed number of pixels at the boundaries of the mean average value section between the edges. The result is the absolute height of the laser line between the two object boundaries. For the determination of the object height, the lower object boundary still must be detected. The reference object is removed from the view of the camera so that the laser beam is projected on the bottom of the conveyor belt. The height value of the same measurement section is determined in the same way and the height of the bottom of the conveyor belt is measured. After this, the calibration factors can be determined as follows:

$$X_{Calcib} = \frac{object\ width\ (in\ mm)}{number\ of\ pixels\ (X)} \qquad 2.7$$

$$Y_{Calcib} = \frac{object\ height\ (in\ mm)}{number\ of\ pixels\ (Y)} \qquad 2.8$$

The calibration factor defines the width and height of a pixel to the corresponding millimeter units. By multiplication with the calibration factors, the camera distortion disappears and the object profile can be reprocessed true to scale and independently of the camera position.

2.2.2.5 Range data acquisition with triangulation-based range sensors

A laser range sensor acquires range data by using triangulation, which means that the object is illuminated with a laser line from one direction. The object is captured by the camera from another direction. The result of one measurement is a height profile, containing one value for each measured point along an object surface. Each time the camera makes a measurement, one profile of the object is recorded and stored. If the object is moved from measurement to measurement, the entire object can be scanned. In most of the applications, the object is placed on a conveyor belt and is moved in a linear way. The height profiles of the object are recorded piecewise (Figure 18). After that, the profiles are merged together into a complete 3D model. This process is shown in Figure 18, with a reversed ordinary setup.

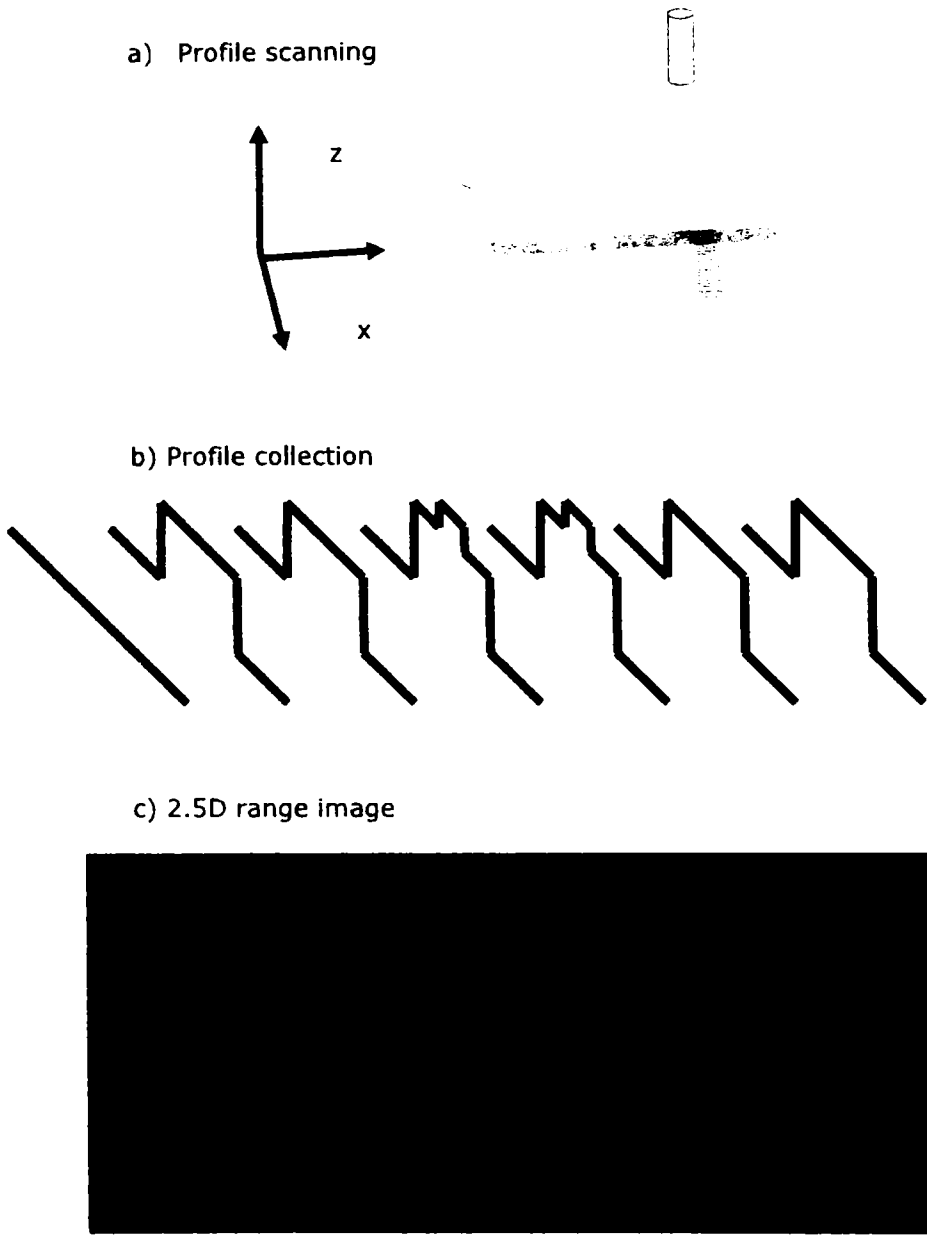

Figure 18 Object scanning
a) the measurement setup for profile scanning. b) Collecting the profiles along the transportation axis (Y) leads to a full 2.5D range image (see c).

The result is a collection of profiles, where each profile contains one profile at a certain location along the transportation direction. The term "scan" here is a synonym for a collection of profiles acquired by the camera. The camera sends the profiles to a computer, which then stores the profiles and processes the profile data

for the application.

Here, the left-handed coordinate system is placed in such a way that Y is the transportation direction (Figure 18). The speed of transportation is connected to the acquisition speed of the camera. Therefore, the accuracy in Y depends mainly on the camera shutter speed and the speed of transportation. The camera shutter speed is often controlled by the exposure time parameter, which can be adjusted, for example, in the parameter file of the camera. The higher the exposure time, the higher the total amount of light falling on the image sensor is. If the scanning process is as slow as possible, the minimum of the resolution is limited by the laser line width, so most applications achieve a Y-resolution of about 0.1mm. The resolution in X is mainly defined by the camera characteristics; namely, the chip resolution and optics such as the distance resolution.

The distance resolution in Z depends on the angle between the laser and the camera. If the angle is very small, the location of the laser line does not change very much in the sensor images, even if the object varies a lot in height. Nevertheless, if the angle is large, a small variation in height would be enough to move the laser line in the camera image (Figure 10). That said, a large angle leads to occlusions, which occur when there is no laser line for the camera to detect in the sensor image. This happens in two different ways: the first type of occlusion is the "camera occlusion". This happens when the laser is hidden by the object. A "laser occlusion" occurs when the laser cannot properly illuminate parts of the object. Adjusting the angles of the Ranger and the laser can reduce the effect of occlusions.

To make the best measurement, the laser line and the camera lens should be optimally adjusted. In addition to the geometric order of the measurement system, the characteristics of the camera play an important role in the analysis of the measuring precision for the X and Z coordinates. Therefore, in many industrial applications special high speed cameras (like RANGER E55 from Sick AG, www.sick.de) are used [58].

2.3 State of the art in robotic bin picking

Object localization is often used in industrial fields such as depalletizing or robotic bin picking[59]. The process of bin picking is commonly separated into different steps, which are shown in Figure 19. First, a visual capture device takes a picture of the scene. In many cases, cameras are used for image acquisition, but more and more range sensors are available in the market to provide three-dimensional data with a high resolution. Second, the most important component of the bin picking problem is the algorithm to recognize and localize the objects in a scene. The object localization results in the position of an object.

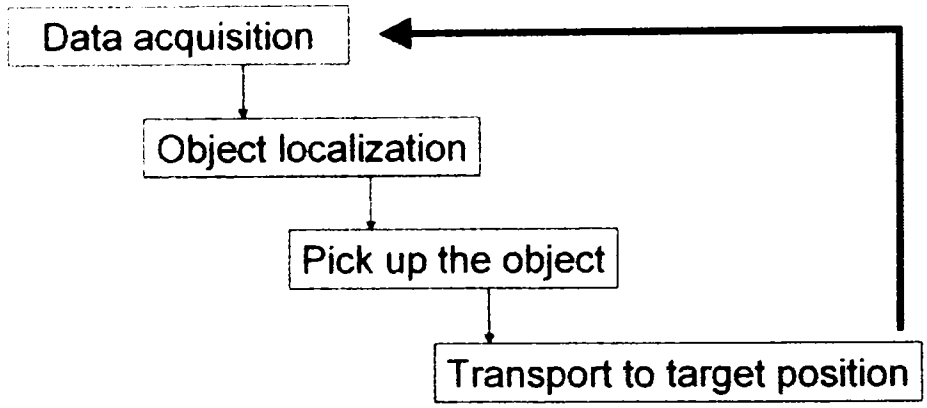

Figure 19 Parts of bin picking
The process of bin picking is started with data acquisition. If the object is localized the robot pick up the object and transport it to the target position.

In order to pick up the object with the gripper, the robot has to know the exact position and orientation (spatial pose) of the object. The spatial poses of the objects in a scene are transformed into the robot's coordinate system and are transferred to the robot controller. Additionally, an adequate grasp point must be defined. After that, the system has to detect collisions with the surrounding environment and find a way to guide the robot to the target position where the object is to be placed. After this step, the robot picks up the object and transfers the object to the target position. All these steps are repeated iteratively for each object in a scene.

The bin picking problem has different levels of requirements and solutions at the current state of the art. The main problem is the performance of the included object recognition and localization. In spite of the availability of features, the search of an object in a scene with all six degrees of freedom could be separated into the following levels:

- Find representations of known, rigid objects
- Find representations of known, rigid objects with (self-) occlusions
- Find representations of known, non-rigid objects
- Find representations of known, non-rigid objects with (self-) occlusions
- Find representations of unknown, rigid objects
- Find representations of unknown, rigid objects with (self-) occlusions
- Find representations of unknown, non-rigid objects
- Find representations of unknown, non-rigid objects with (self-) occlusions

Most of the known solutions of object recognition and localization deal with a

priori known objects. Many of the known solutions are limited to simply shaped objects or objects with specific features. The most developed and well known situation is defined by the first level: finding known rigid objects in a scene is a solvable task, even if these objects are partially occluded. Major problems occur when the objects in the scene are non-rigid or deformable [8]. In addition, if these objects are unknown to the object recognition and localization algorithms, the abstraction level and power of such an algorithm are similar to a human brain. The level of the proposed solution in this thesis deals with explicitly known rigid objects, but can also extended to limited, non-rigid objects. All problems with unknown objects are reduced to the problem of object separation. Due to this, the combination of separation, recognition and localization of an unknown deformable object in a cluttered box, is the most desirable, supreme discipline in the case of robotic bin picking.

"Bin picking" is a term often used in the research activities of robotics. Unfortunately, only a few approaches cover the whole bin picking process [60], [2], [3], [4], [44], [61], [62], [51]. To classify previous work concerning the bin picking problem, these parts are used to separate known solutions. The following list classifies the solutions for comparing different solutions in related work.

- Data acquisition
- Reference representation
- Kind of correspondences
- Matching algorithm

Other classifications depend on the focus of the author and their topic. Campbell and Flynn [9] add surveying methods of object recognition to their classification results of complexity and recognition rates. Matabosch et al. [63] compare registration methods by the kind of correspondences, matching algorithms and registration strategies.

The type of data acquisition is a very important component of this work. The input data influences the object representation as well as the following steps in the process.

The first step in robotic bin picking is data acquisition. The first decision a researcher has to make is to select the form of input data from the scene. Gray value images with two-dimensional (2D) data are made with cameras. This 2D data acquisition is often used in industrial environments [44], [64]. In different range image applications many further methods belong to 2D Template Matching [18], [65], [66]. This work is based on range data (2.5D) data is created with the help of range sensors. Data structures for 3D surfaces arise from moving 2D sensors[67] but also from 3D distance sensors. Most applications use line sensors [68], which are moved on external axes. 2.5D data is characterized by a 2D-array of distance values, i.e. every pixel represents a value for the distance between the sensor and the object. This kind of data acquisition is getting more attractive in industrial application because of the growing number of range sensors in the market. Range sensors should not be mixed with real sensors delivering real three-dimensional (3D) data as they are very sparse and also very expensive. Therefore, only a very

small number of applications can take advantage of them [69].

In many object recognition and localization systems the acquired image is compared to a model. The reference or model representation can be stored in the system in different ways, ranging from low level features in 2D images to high level 3D CAD-models. In the approach by [70], 3D-Models are projected in the image plane and compared to the image of the scene in order to estimate a possible pose. The proposed pose estimation extends this approach to range images. Industrial laser range sensors are modeled to transfer a Computer-aided Design (CAD) model to a 2.5D range data representation. This virtual range data is compared to the real range data of the scene. Therefore, efficient data structure[39], [71] and object representations are needed. To keep the representation as simple as possible, point-based object representations store a collection of points. Wheeler [12] uses this kind of model because of its efficiency and universality.

Objects can be described by a list of features characterizing their shape, orientation or appearance. Local features like vertices, edges and more complex features are grouped in Local Feature Sets. Many approaches try to find the matching feature vector [72],[73], and if this feature vector is found, objects in the scene can be identified and their pose estimated with alignment [74]. This representation can be used if the objects have view-independent features, but fails with convex featureless objects.

The first category is known as *model-based object representations*. One group of the high level geometric model-based object representations is known as *generalized cylinders*, introduced by Binford [75]. Furthermore, parameter-based representations like *geons* and *superquadrics* [76] belong to this group. Parameterized surface-based representations are characterized by a function defined with parameters describing their surface and their pose. Well known examples are polygon meshes and NURBS [77].

The second category of object representations is called *appearance-based*. Appearance-based object representations consist of a set of global object attributes or features [2], and benefit from their computational efficiency. However, the advantage of model-based object representations is their robustness against noise. In addition, they can be generally used.

View-based models are often used in 2D object recognition and localization systems. A collection of images is acquired from different views of the 3D object and stored in an image set, and a full three-dimensional object representation is rebuilt with these images. Approaches using view-based models are often combined with view-based feature extraction [78], or use global features [79]. The disadvantages of view-based models are sensitivity to illumination changes and variations in cameras [12]. Approaches can be classified by the type of correspondences that are compared to each other. This is a typical classification in model-based object recognition [72], [12].

Possible classes are:

- 2D image to 2D model comparison[80]
- 3D image to 2D model comparison [12], [70]
- 2D image to 3D model comparison [81]
- 3D image to 3D model comparison [12]

Within each class, there exist many possibilities for comparison. In the class of 3D to 3D, on the one hand, it is possible to extract high level features and estimate high level descriptions like parameter-based superquadrics from 3D images in order to compare their parameter with the parameter from the 3D-model. This is known as a *bottom-up* approach. On the other hand, the 3D model can be degraded to extract low level features like edges or corners, which are compared to the 3D image. This is a so-called *top-down approach*.

A huge number of approaches are made to estimate a 3D object's position by extracting points [74], lines[82], vertices and other features from 2D-Images [83].

Dickinson [80] describes techniques in 2D character recognition and transfers the results to 3D object recognition. A good overview is also given by Zitova [84] for 2D image registration.

Mataboscha et al. [63] provide a survey of the range image registration process. They separate coarse pose estimation and pose refinement. The well known Iterative Closest Point [6] is introduced in Chapter 4.

The most critical and time consuming part in the bin picking process is object localization. Many surveys for object recognition and localization can be found from the last few decades [85], [80], [1], [83], [86], [87]. Campbell and Flynn [9] show free-form object representations and recognition techniques developed in the 1990s.

Many approaches exist to solve this for a small group of objects. Even this object localization is very important to the robotic bin picking problem but, unfortunately, only a few approaches cover the whole process. A few solutions aim at solving the bin picking problem for a group of objects [88], [44],[89], [2], [4], [90], [3], [91], [92], [93], [62], [61]. A selection of solutions for the whole bin picking process is introduced in the following paragraphs.

Only a small number of papers include the whole process for bin picking. Brady et al. [87] give an overview of solutions concentrating on object localization from the early 1980s to 1988. Kak and Edwards [1] surveyed the state of art of bin picking in 1997. They focused on feature-based methods and optimal tree search algorithms.

One of the early papers related to the bin picking problem was published by Ikeuchi et al. [4] in 1983. Surface orientation histograms are matched to histograms of simulated orientations of CAD-models (represented as unit sphere projections by

Extended Gaussian Image (EGI)). These surface orientations are calculated from multiple views of a convex object. An experimental setup evaluated the results with a process time of one minute for a few torus-shaped objects.

Bolles and Horaud [94] introduce the grouping of features. They extracted features from range images and object models. These features are partitioned in subsets of features with "intrinsic properties". To find the pose of an object in the scene, the features are compared with the help of an ordered tree and a "grow a match" strategy to decrease the processing time. This algorithm is evaluated in an industrial application called 3DPO, using local edge-based features. This paper discusses the bin picking process and offers an overview of related work in this decade.

Rahardja and Kosaka [44] extract features of complex industrial objects from intensity images. These features are separated into so-called seed features and supporting features. At first, regions are extracted and verified with a priori knowledge (2D appearance of the objects) to get seed features. The depth of the objects in the bin is determined with stereo vision. An iterative process searches the space of the corresponding seed features to find the position of the grasp point. Rahardja and Kosaka give an orientation error of less than 7mm and a rotational error of 10° within a processing time of 1.5 minutes on a SUN sparc 1000 server.

Hashimoto and Sumi [3] propose an object recognition system that uses range images and intensity images. The process is separated into two parts. First, information from the range image is used to estimate the rough position of the object. A template of the object shape is matched with the representation of the object in the binarized range image. An intensity image verifies the results of the estimation and refines the position to get an accurate pose with contour matching. The depth of the scene is determined with a structured light stereo vision system. Experiments with box-like objects (600mm to 400mm) have shown that the accuracy of object position has been smaller than 30mm. The recognition process takes about 5 seconds with an unknown computer system.

In the paper of Boughorbel et al. [62], an imaging system works with range image and intensity image data to find accurate object positions and to solve the bin picking problem. The geometry of the objects is either reconstructed from the range data or given in the form of CAD models. The objects in the scene are modeled with a pre-segmentation and parameter recovery of superquadrics object representation. A camera validates the position and provides additional information for tracking the robot arm. The authors give no information of the experimental results.

Katsoulas [2] describes a robotic bin picking system in his PhD Thesis. He focuses on a model-based object localization system. The model uses geometric parametric entities of deformable superquadrics and is compared to scene data, which is preprocessed by edge detection in the input range data. The comparison algorithm performs maximization of the posterior parameters of all known objects to recover the object. Kasoulas proposes a variety of advantages in robustness, accuracy and efficiency. He reported a total processing time of over 100 seconds with a 2.8Ghz Pentium4 PC for one scene.

Chapter 3 Pose Estimation

In this chapter the sensor simulation and the pose estimation algorithms are explained in detail. The object model is transferred to a virtual range image with the sensor simulation and compared to the range data acquired by laser range sensor, which are introduced in the last chapter.

3.1 Object model representation

The object model is a mathematical description of a three-dimensional object, and is described with a defined data representation. This data structure mainly contains the geometry and the texture information of the object model.

The most common representation for object models is a polygonal mesh. Polygonal meshes are a collection of unstructured simple polygons such as triangles or quadrilaterals. One common form of polygonal meshes is the triangle mesh. Three points in the model coordinate space (vertices) and define a triangle. A list of triangles (triangle mesh) defines the geometry of the model. Additionally, every triangle in this structure contains further information like face normals, texture and reflection parameters, which are used to improve the performance of the process. There are more advantages of this object model representation. First, this kind of representation is well known and easy to understand. Triangle meshes are convex [95] and because of this, simpler rendering algorithms in computer graphics can be used. Figure 20 shows an object that is modeled with triangles.

The precision of the object depends on the number of triangles used to model the object. If the number of triangles is too small, the real object does not fit to the model anymore. Even if this issue can be a disadvantage, the reduction of triangles raises the performance and yield to a hierarchical approach. This scalability between accuracy and performance is one of the main contributions of this work and will be described later (Section 3.5).

The second advantage is the fact that the representation is common in different disciplines. The proposed system uses the *computer aided designed* (CAD) models stored in the *structured triangle list* file format (STL). The proposed system aims to simulate industrial parts, where in many cases the geometry is not known at all. Often there exist CAD-models of the target objects, so the decision to use a polygonal mesh-based object model representation suggests itself. This representation is converted because the sensor simulation in this work uses data structures for graphics processing units (GPU), which are encapsulated in the libraries of Microsoft DirectX© or OpenGL©. These libraries provide interfaces for

importing the vertices of triangles. The scale-dependent triangle lists are built from the vertices and face normals. The results are meshes of triangles, which are described in Section 4.4.1.

The simplicity of intersection tests is a further advantage of this object model representation. The intersection tests of the sensor simulation can be reduced to a simple ray-triangle intersection test by dividing the object into single triangles.

3.2 Sensor model simulation

Sensor models should adopt all properties of the available sensors for the industrial applications introduced in the previous sections. The sensor simulation must be separated in TOF laser distance sensors and the triangulation-based laser sensors because both measurement principles deliver different range images, even when they are observing the same scene. The characteristic of triangulation-based laser distance sensors is the fact that their receiving device is not placed in the same position as the laser, which sends a laser line to the scene. Due to this displacement, most depth images of triangulation-based laser distance sensors suffer from occlusions. This is the most important difference of triangulation-based sensors compared to scanning TOF laser distance sensors. Moreover, all the required parameters and properties of the sensors must be known. This requirement can be met in many important situations but, on the other hand, there are some restrictions. The sensor simulation workspace is defined in a right-handed coordinate system, similar to the coordinate system of the scene scanning process.

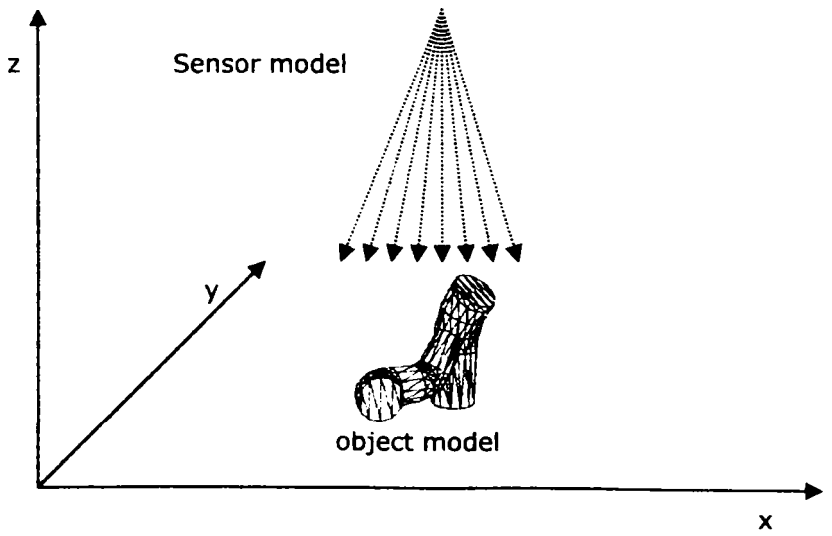

Figure 20 Virtual scene
The virtual scene consists of a sensor model and the object model.

In most workspaces in the research field of computer graphics, the distance to an object in an image is aligned to the Z coordinate axis (see Microsoft DirectX©, OpenGL© and Z-buffer techniques [96]). The sensor model for the TOF sensor contains the light source and the model of a perfect receiving unit. Therefore, the sending aperture and receiving unit has no greater size than zero. The sensor is reduced to a theoretical point in the 3D space. In this optimum case, every other point in the workspace forms exactly one line with the receiving unit. This is one fundamental difference to the real scenario. The object is placed in the sensor simulation workspace in the sensor's view port/ frustum (see next section). All range sensors that deliver a distance profile are moved over the object in order to get distance maps [97], which is also necessary for the sensor simulation. To compare the results from simulation to the real images, the resolution of the data in the moving direction and step resolution should be similar [98]. Therefore, the properties of the scanning process of the objects in the scene must be known. This includes mainly the distance between the ground and sensor and the direction of scanning. This setup is defined and used as the standard sensor simulation setup in the following sections.

3.2.1 Distance map sensor model

This model can be used for unknown sensors or all sensors that cannot be modeled because of their complexity. This simulation results in an equidistant depth map of the scene.

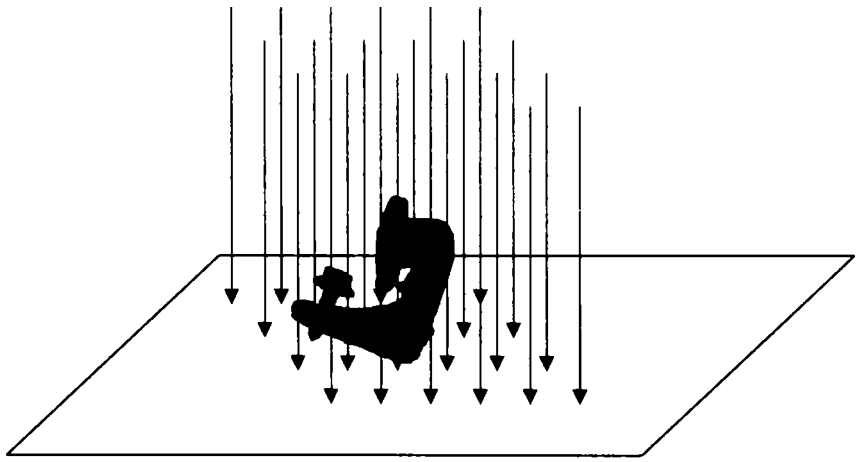

Figure 21 Distance map sensor model
The distance map sensor model is an array of one-dimensional distance in small distance to each other.

This sensor model delivers for every X and Y coordinate one distance value in a Cartesian coordinate system (Figure 21). The implementation of such a sensor model can be fully supported by extremely fast and commonly used algorithms. The simulation is reduced to the Z-buffer/Depth-buffer [96] calculation of a virtual scene. Especially, the usage of GPU-based hardware fits to this task almost perfectly. In computer graphics 3D programming with software development kits like Microsoft DirectX© (www.microsoft.com/directx), the Z-buffer of virtual

cameras is defined by a frustum.

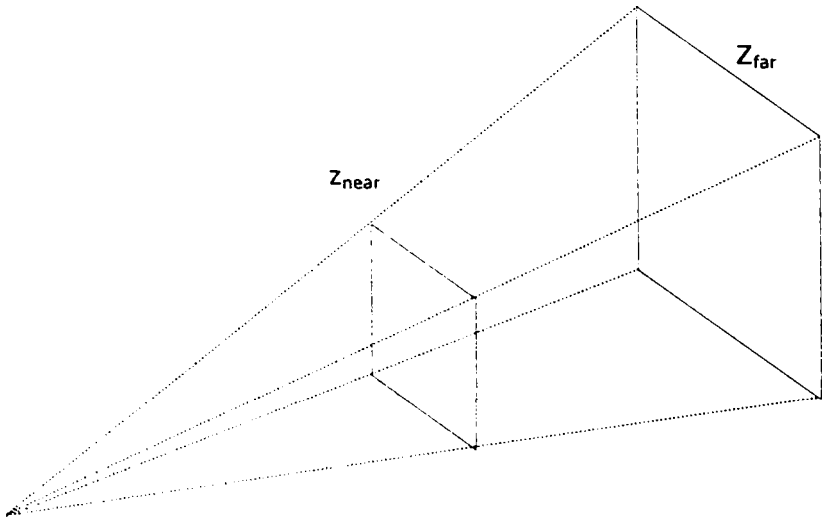

Figure 22 Frustum of virtual cameras
The frustum defined by a rectangle pyramid includes the field of view of virtual cameras.

The shape of the frustum is a rectangular pyramid (Figure 22). The distance between the camera and the scene Z_{near} is defined by the sensor position in the virtual coordinate system. In this case, this is initially set to zero, which means that the scene begins immediately in front of the camera. The distance Z_{far} between the camera and the maximum distance point of the scene can be set to infinity but, in most cases, the limit can be set by knowing the physical limits of the real laser sensor. The boundaries are defined by the maximum number of lines m in X and Y.

3.2.2 Sensor model of Time-of-flight sensors

Time-of-flight (TOF) sensors measure the distance between the object and the light source along a light beam. This beam is moved incrementally with a parameterized angle step width in the orientation of X (see Figure 20). The result is a radial distance. For each angle, a distance value is measured, and to ensure the comparability between different sensor data, these values are converted into Cartesian coordinates.

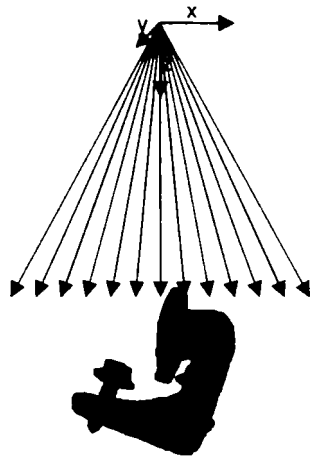

Figure 23 TOF sensor simulation
The sensor model sends out laser beams in the coordinate system of the sensor.

In Figure 23, a simulated sensor measurement is shown. This model is characterized by the sensor coordinate system. The simulated light beams start in the origin of this coordinate system and the distance values are calculated between this point and the closest point to the object. The angle steps can be found in the rotation R_y of the light beam. A commonly used industrial TOF sensor like the SICK LMS400 (www.sick.com) can be modeled in this way. The real sensor delivers each angle step width, start angle and the distance. These properties are adapted to the sensor model as well.

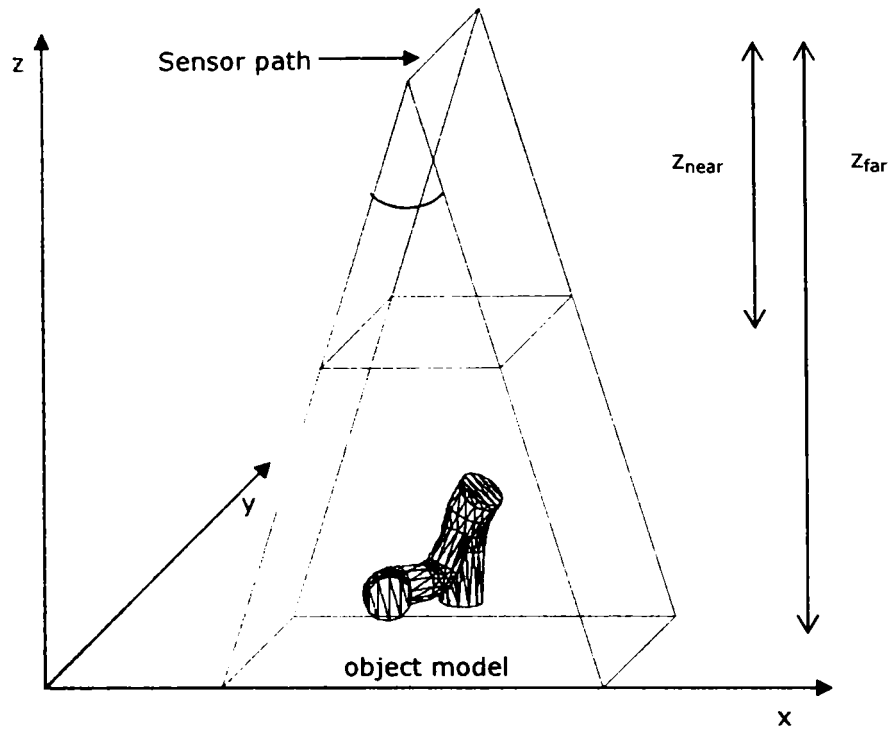

Figure 24 3D TOF sensor frustum
The sensor is moved along the sensor path and records the data in the sensors field of view between z_{near} and z_{far}

This Figure 24 describes the volume of the sensor simulation in the model coordinate system. The sensor model is incrementally moved in the direction of the Y coordinate axis along the sensor path. The minimum distance to the sensor defines the Z_{near} plane, and the maximum distance value defines the Z_{far} plane. With the view angle α, the sensor frustum is defined as a frustum of a rectangular pyramid. The resolution of the resulting depth image/distance field defines the number of rays starting in the camera's point of view. Rays are produced with a normalized length in one line of the distance field. According to the principle of real laser distance sensors (see Section 2.2.2), one scanline is separated into angle steps in the scan direction X.

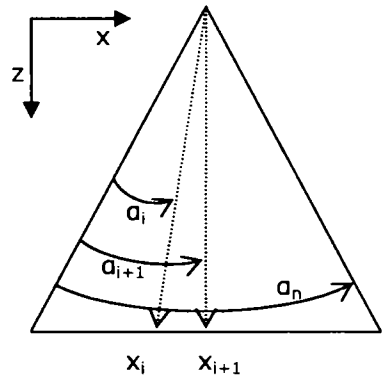

Figure 25 TOF angle resolution
The sensor sends out one beam per defined angle. The number and the angle step width are parameters of the sensor model.

The sensor coordinate system is defined as shown in the Figure 25. The sensor determines the distance in direction of the Z coordinate axis. One line of rays is defined to be in the direction of the X coordinate axis. For one line with *n* distance, values *n* rays are created. The direction of each ray is calculated with the help of the angle *α*. For every step, the normalized direction vector for X_i is calculated with the following equation:

$$v_{DIR(x_i)} = \begin{bmatrix} z_{far} * \cos(\alpha) \\ y_j \\ z_{far} \end{bmatrix} \qquad 3.1$$

The Y coordinate is fixed to the position of the sensor in the scene and is constant for one scan line. Z_{far} is also constant in this consideration. The origin of the rays is equal to the sensor position in the scene:

$$v_{POS} = \begin{bmatrix} \frac{x_n}{2} \\ y_j \\ z_{near} \end{bmatrix} \qquad 3.2$$

The perspective view of the sensor simulation has a range of

$$\alpha = \pm\left(\frac{n}{2} * \Delta\alpha\right) \qquad 3.3$$

where *n* is the number of distance values in one line and *Δα* is the constant angle step width. If the angle *α* is zero, the ray is orthogonal to the X coordinate

axis and parallel to the Z coordinate axis. The direction of the ray is towards the object. For every scan line a spread of rays is calculated depending on the needed resolution in X. Every ray is tested for its intersection with the object in scene.

The proposed sensor model is related to a real TOF sensor. To acquire a full three-dimensional distance image, the sensor model is moved virtually in the direction of Y, as shown in Figure 26. The step width depends on the application requirements. The resolution in Y is connected to the resolution of hardware encoders in different applications. This ensures the compatibility between the real sensor setup and the sensor simulation. To find the correct relation between the real dimensions and the model coordinate system, the setup must be calibrated.

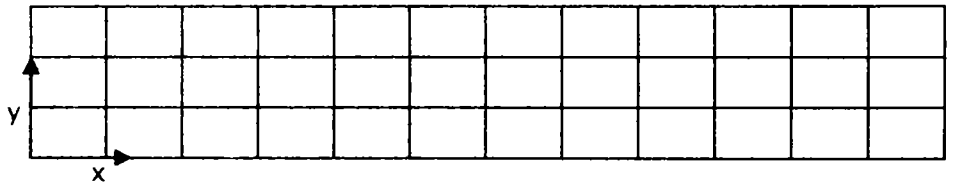

Figure 26 Movement step resolution
The simulation results in equidistance movement lines for each scan.

The Figure 26 shows a resulting distance image in the sensor coordinate system.

The transformation to the model coordinate system is:

$$ {}^{SCS}T_{MCS} = \begin{pmatrix} 1 & 0 & 0 & Vpos \\ 0 & 1 & 0 & Vpos \\ 0 & 0 & 1 & Vpos \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad 3.4 $$

3.2.3 Sensor model of Triangulation sensors

Triangulation-based laser sheet of light sensors consist of a stripe projector to emit a laser line to the object. A camera grabs the projected line, and with the help of the geometric calculation, the distance can be acquired according to Section 2.2.2. With its simulated geometric configuration, the sensor model determines the distance between the object and the laser source in the same way. The distance is calculated for every maximum value in the camera's column index. This results in a distance vector for every projected and acquired laser line. The output data is a distance profile. The resolution in X is fixed by the resolution of the CCD-Chip in the real sensor. Therefore, the model must deliver exactly the same data. For this implementation, the amount of received light in each pixel of the camera must be calculated. Multiple reflections of the emitted laser light must be considered additionally to the main reflection on the object in real sensor setups. Consequently, the amount of light of each (multiple) inter-reflected laser light beam with shadows and refractions for each pixel must be calculated. This leads to a very complex simulation of real triangulation sensors [99], and is otherwise known in computer

graphics as *global illumination* [100]. In most real time rendering algorithms, simpler illumination model simulations are used such as *flat shading, phong shading* or *gouraud shading* [101]. Therefore, the trade-off between performance and accuracy is changed to increase the performance of the sensor simulation. The triangulation simulation in this thesis uses a trick and changes the approach of triangulation without the loss of accuracy or by producing incorrect results. Similar to the TOF sensor simulation, the triangulation simulation separates the emitted sheet of light into single beams. These beams are produced according to Figure 25 in a laser projection plane. For the triangulation sensor model, the discrete laser line angle step's width at the maximum distance must be at least two-times greater than the pixel-wise projection in the camera coordinates to ensure that this simplification results in a distance surface. According to the Nyquist-Shannon-Theorem, the sampling frequency of the camera must be:

$$\Delta y_{Pixel} \times \bar{A}_{Scene}^{Camera} \geq 2 \times z_{\max} \cos(\alpha) \qquad 3.5$$

This requirement can be easily met because the angle step width can be calculated before the triangulation simulation. For each laser beam, the intersection point with the object representation in the virtual scene is determined. This intersection point is checked again if the camera is able to see it. Therefore, the line between the intersection point and the center of the lens is analyzed if there are any occlusions by scene objects.

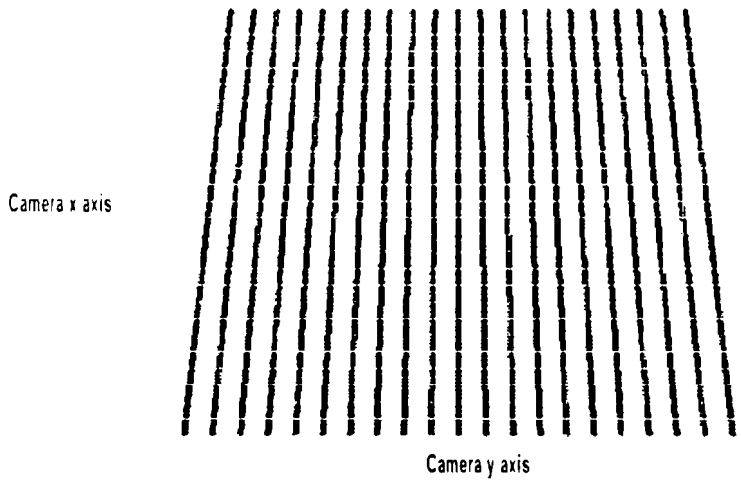

Figure 27 Laser line projection in the camera's CCD chip
The separated laser beams in the sensor model are related to the camera pixel, where it intersects a models surface.

Figure 27 depicts the projection of the laser lines in the camera's field of view. If any object is in the field of view of the camera, the laser line stops at a certain position. This is the intersection point, where the laser intersects a triangle of the object in the virtual scene. Following that, the intersection points are projected to the camera chip and the related pixels are determined. In Figure 27, the pixels are shown as yellow rectangles. The displacement of the pixels in each Y column is proportional to the height of the object in the scene according to triangulation principle in Section 2.2.2.2. If the intersecting point with the highest Z coordinate cannot be projected to the camera CCD chip plane, the related distance value in the height profile is marked as invalid. With known extrinsic parameters (respectively the distance between the camera and the laser and the angle between them), the distance to the object can be determined. The laser is virtually mounted in the similar coordinate system as shown in Figure 24. The position of the virtual camera depends on the application. For most cases, the camera is in the same quadrant of the coordinate system shifted in transportation direction as shown in Figure 24. All invalid distance values in a complete surface scan represent the shadow or self-occlusion.

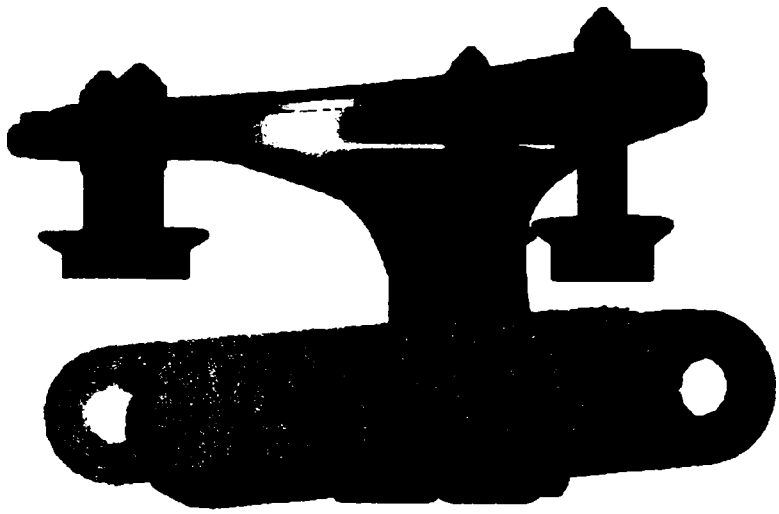

Figure 28 Occlusions in triangulation simulation
This figure shows a virtual scan result of a triangulation based sensor model. The black regions represent the occlusions, where the camera cannot see the laser line.

The occlusions in Figure 28 are in transport direction Y because the camera and laser source are set up like described above. The occlusions are the important difference between TOF and triangulation-based sensors.

3.3 Virtual Range Image simulation

The purpose of simulating a distance sensor is much related to fields in computer graphics. For example, rendering is a process that generates an artificial image of a scene with a computer program. To produce a two-dimensional image from a three-dimensional virtual scene, it is necessary to have a camera (or eye), a light source (e.g. the sun) and a model (or object). *Rendering* is in general a computationally intensive process because it has to take many mathematical and physical restrictions and principles into consideration. According to [100] there exist more or less 4 different methods for producing an image from a virtual scene. *Rasterization* converts objects to an image by applying 3D to 2D transformation and divides the result in image pixels without consideration of the physical principles of light sources. This method is very fast and efficient. *Radiosity* takes the finite element method for rendering scenes with purely diffused surfaces. *Ray casting* is a very fast method, which is very similar to ray tracing. *Ray tracing* is one of the most promising and often used methods for obtaining nearly realistic images. The principle of ray casting and ray racing is the analysis of rays of light between the camera, the objects and the light sources. Ray casting defines rays that are virtually sent out from the point of view — the camera. If the ray intersects with an object on its way into the scene, the intersection point defines the resulting color in the image. Simplified ray casting uses the color defined by the texture of the object. The main difference between ray casting and ray tracing is the determination of the resulting pixel value. In addition to ray casting, reflected rays from other objects in the scene can also change the value of this pixel. This approach has some advantages compared to other methods of rendering. Ray tracing enables effects like shadows and complex reflections. Especially, the shadow effect is very important for the laser sensor simulation. Unfortunately, ray tracing has high computational costs depending on the level of details and complexity of the scene. To get a realistic image, this method has to analyze every possible ray of light in the scene, but this process is extremely inefficient and is not necessary in many applications at the moment.

Most of the methods for virtual image production are supported by hardware or, more specifically, Graphical Processing Units (GPUs). The GPU is the core of commonly available graphic cards for personal computers and can be equipped with user specified code via software development kits. Today, GPU supports fast perspective transformations, depth buffer extraction, rasterization and several ray casting/ray tracing techniques. Furthermore, many processors (CPUs) of personal computers offer specific instruction sets for parallel data processing, with a single machine instruction like Streaming SIMD Extensions (SSE, www.intel.com) or 3D-Now!(www.amd.com) [102]. One key aspect of distance measurement is the determination of the intersection point between a ray (laser) and the object surface. It is very important to have a high performance for virtual range image (VRI) simulation. For example, for each emitted laser beam of the TOF sensor model, one intersection test should be made much faster than in reality. The TOF sensor system fits almost perfectly with ray tracing systems to acquire realistic images. With this in mind, several solutions for object mesh intersection tests are discussed in the next section.

3.3.1 Polygonal mesh intersection

The intersection of 3D objects is still a challenging task in computer graphics because of the requisite performance for increasing the requirements of computer games and scientific applications. In many scenarios, the mesh intersection can be reduced to a ray-object or even to a ray-triangle intersection test. A lot of research has been carried out since Glassner [103] introduced intersection calculation issues. To speed up the ray-scene intersection process, mainly additional data structures, such as *grids, octrees, bounding volume hierarchies*, or *BSP trees* are used. (see [103], [104] for an overview).

3.3.1.1 Triangle normals intersection method

To prove the concept and have a basis algorithm as a reference, a very simple method is created to calculate the intersection. For every pixel of the resulting distance field with the defined resolution, one ray is created, as described in Section 3.2.1. Every created ray is intersected with the planes, which are defined with the three points of every triangle. In the second step, every intersection point is checked to see if it lies inside the triangle. This is realized by solving the equation of the parametric form of the triangles and the line:

$$\begin{pmatrix} Sensorposition_1 \\ Sensorposition_2 \\ Sensorposition_3 \end{pmatrix} + \left(\begin{pmatrix} Sensorposition_1 \\ Sensorposition_2 \\ Sensorposition_3 \end{pmatrix} - \begin{pmatrix} \cos(angle\_start(i * angle\_stepwidth)) \\ Sensorposition_2 \\ 1 \end{pmatrix} \right) * \begin{pmatrix} x_{1i} \\ y_{1i} \\ z_{1i} \end{pmatrix} + \left(\begin{pmatrix} x_{2i} \\ y_{2i} \\ z_{2i} \end{pmatrix} - \begin{pmatrix} x_{1i} \\ y_{1i} \\ z_{1i} \end{pmatrix} \right) * u + \left(\begin{pmatrix} x_{3i} \\ y_{3i} \\ z_{3i} \end{pmatrix} - \begin{pmatrix} x_{1i} \\ y_{1i} \\ z_{1i} \end{pmatrix} \right) * v \quad 3.6$$

Due to the requirement to be a proof of concept solution, this procedure is not very efficient, but it shows the function of the sensor simulation. Therefore, this solution is used as reference and basis only.

3.3.1.2 Möller/Trumbore ray intersection

To compare the performance to a more efficient ray-triangle implementation in C/C++, a slightly modified intersection test function, based on the work of [105] is implemented. Möller and Trumbore proposed a non-SIMD algorithm for fast ray-triangle intersection tests. Their main contribution is that they do not calculate the plane equation, but use a series of transformations to translate the triangle to the origin and transform it to a unit right angled triangle. The ray direction is changed to align in the direction of the X-axis. The barycentric coordinates and distance can be calculated according to the following equation:

With

$$E_1 = \begin{pmatrix} x_{2i} \\ y_{2i} \\ z_{2i} \end{pmatrix} - \begin{pmatrix} x_{1i} \\ y_{1i} \\ z_{1i} \end{pmatrix} \quad 3.7$$

$$E_2 = \begin{pmatrix} x_{3i} \\ y_{3i} \\ z_{3i} \end{pmatrix} - \begin{pmatrix} x_{1i} \\ y_{1i} \\ z_{1i} \end{pmatrix} \quad 3.8$$

where E_1 and E_2 are two non-parallel vectors in the triangle plane. Möller and Trumbore proposed:

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{P * E_1} \begin{bmatrix} Q * E_2 \\ P * T \\ Q * D \end{bmatrix} \quad 3.9$$

with

$$T = \begin{pmatrix} Sensorposition_1 \\ Sensorposition_2 \\ Sensorposition_3 \end{pmatrix} - \begin{pmatrix} x_{1i} \\ y_{1i} \\ z_{1i} \end{pmatrix} \quad 3.10$$

$$P = \begin{pmatrix} \cos(angle\_start(i * angle\_stepwidth)) \\ Sensorposition_2 \\ 1 \end{pmatrix} \times E_2 \quad 3.11$$

$$Q = T \times E_1 \quad 3.12$$

This results in the distance to the triangle and the intersection point coordinates inside the triangle. More details can be found in their paper [105]. Because they do not have to consider storing the plane equation, this is a very fast algorithm for this intersection test. Some other approaches show the potential of this algorithm [106], [107], [108], and this algorithm is used to check if the current ray of the sensor simulation intersects with any triangle of the object. This leads to a slightly better performance of the sensor simulation and still offers many further improvements like a *bounding box calculation* and *backface culling*[109].

3.3.1.3 Fast mesh intersection with axis aligned bounding box

For ray tracing tasks, the usage of *axis-aligned bounding boxes* (AABB) is quite common in computer graphics. Simple AABB intersection tests compute the distances to each of the six planes defined by an axis aligned bounding box around the mesh [110].

For the pose estimation, an axis-aligned bounding volume is used to decide if the ray shoots in the direction of the model before an intersection test is made.

The bounding volume is aligned with the axis of the model's coordinate system. The AABB contains the full object model and is defined by 8 points. This is shown in Figure 29.

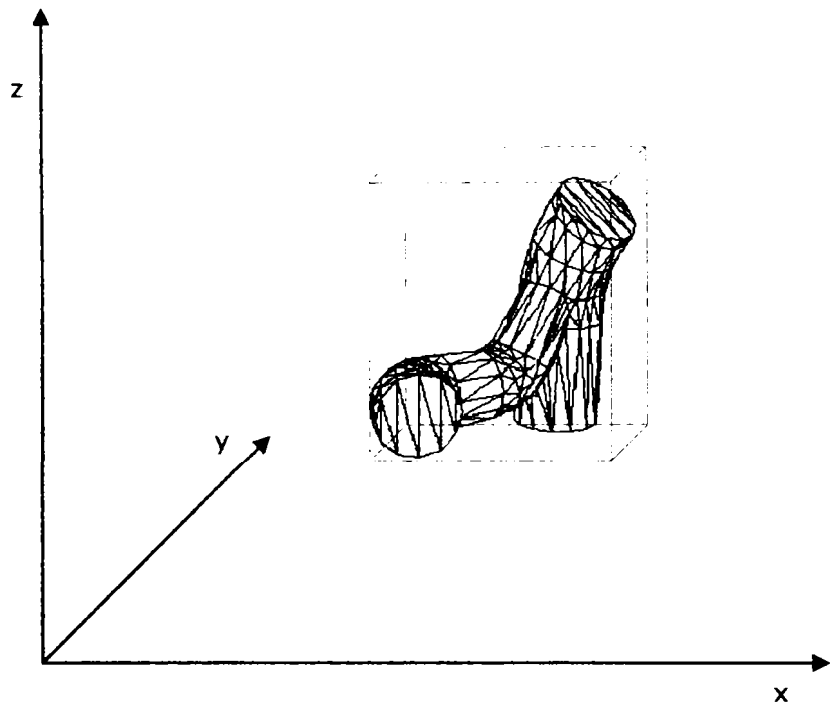

Figure 29 Axis Aligned Bounding Box
The Axis Aligned Bounding Box test enhances the performance by rejecting regions out of view.

Many further improvements for AABB tests are shown in[104], [111]. For example, AABB-based *backface culling*, as introduced by Woo[109], determines whether the polygon of a graphical object is visible to the camera, which leads to a test of three planes instead of six. The intersection test for the pose estimation is made with the SIMD-supported function of Microsoft DirectX©, which is described later in the implementation (Section 5.3). To ensure the compatibility and optimize the performance, the AABB is converted to a triangle list beforehand. Due to the presence of this axis aligned bounding box (AABB), the intersection test integration is very simple and efficient because of the fact that the DirectX© mesh intersection function includes presumably many new improvements as proposed in [111].

3.3.2 Virtual Range Image representation

The purpose of pose estimation is the comparison of real range images (RRI) and simulated virtual range images (VRI). To satisfy the contribution to simulate the sensors as good as possible, the resulting data of the sensor simulation must fit to the acquired data. Because of this, the representation of the VRI is

similar to the representation of the RRI described in the next section.

$$M = \{m_i\} \quad and \quad m_i = (x_i, y_i, z_i, I_i), i = 0,1,...N_i \qquad 3.13$$

Each point in the VRI consists of X, Y, and Z coordinates in a Euclidean coordinate system. The number of points N_i depends also on the resolution of the simulated sensor and the number of scans in the direction of X. The intensity value I is set by the simulation, but only if the real sensor is able to deliver the intensity and the properties of the object material are known. The red VRI in Figure 30 shows a low density VRI in contrast to Figure 28. The density of points depends on the resolution of the sensor simulation. Sensor models with a high resolution deliver a detailed representation of the object surface with a huge number of points. Therefore, all VRI representations are stored in a relational database. For each VRI, the points themselves and additional parameters are stored in one database entry. The VRI points are indexed with the pose parameter of the object, i.e. the translation and rotation in the virtual scene. The database offers fast queries for more than one process. Parallel database queries are needed so that it is possible to distribute the pose estimation and pose refinement steps over different computers. Distributed computing is one key feature of the bin picking system in this thesis. Therefore, the input for the object pose estimation and comparison is given by a data set of points in a Euclidean coordinate system, delivered by the database interface. This also includes a speed up in performance and reduces the requirements of free memory for one estimation step.

3.4 Real Range Image representation

In general, a range image consists of a simple list of points. The smallest part of a 3D scan is a scan point. This abstraction has the great advantage that the representation of the sensor data is completely independent from the selected laser sensor. Additionally, an intensity value I is delivered by many range sensors and can be assigned to every point. To compare the range image representations, the distance data must be transformed to a world coordinate system. For every sensor, the introduced calibration method can be used to calculate the corresponding units in millimeters. For the introduced range sensors, the X values are given by the resolution and the provided measurement range and the Z values are given by the distance. A point in range images is generally defined in the following way:

$$P = \{p_j\} \quad and \quad p_j = (x_j, y_j, z_j, I_j), j = 0,1,...N_j \qquad 3.14$$

The introduced sensors deliver information of a height profile with:

$$p_a = (x_a, 0, z_a, I_a) \qquad 3.15$$

For this reason, a point differs from a height profile by the missing Y coordinate, and in most cases by the additional intensity value I. The missing information for the Y coordinate of a laser line can be assigned from external sources such as robot axes and incremental encoders, or by manual calibrations. In most instances, the X-axis corresponds to the world X-axis according to Figure 24. The distance to the sensor corresponds to the Z-axis. A Y value can independently

be assigned to every point. In the simplest case, Y is incremented if the sensor is orthogonally moved to the X Z plane. The movement in Y can be made with an independent resolution. In this case, the Y coordinate is equal in the data structure for each laser line. A 2.5D range image is constructed with the sequential storage of laser lines. The number of points N in one range image depends on the number of entries *n*, as a sensor can resolve in X direction and the number of scans in direction of Y. The sensor acquires a 2.5D surface. For each X and Y position there exists one distance value Z. There are well known data structures in the computer graphics sciences for 3D range images. The most general and often used data structure is the storage of unordered point clouds. These data structures could be used in simple regression methods up to the complex 3D registration and reconstruction algorithms of modern image processing[17], [112]. The main goal of these complex algorithms in computer graphics is the reconstruction of surfaces from given point clouds. Unfortunately, these reconstruction algorithms are very inefficient in most cases, so the point cloud representation of the acquired real range image (RRI) is suitable to the pose estimation. For the pose refinement, the point clouds are converted from small parts of the RRI to a surface representation.

3.5 Object pose estimation and comparison

Each data set acquired by the laser range sensor is a surface with a huge amount of data points. The simulated object pose (VRI) must be found in the application scene (RRI). The VRI consisting of the surface's representation of the 3D model is compared to the real range image (RRI) of the scene made by the sensor. One example is shown in Figure 30.

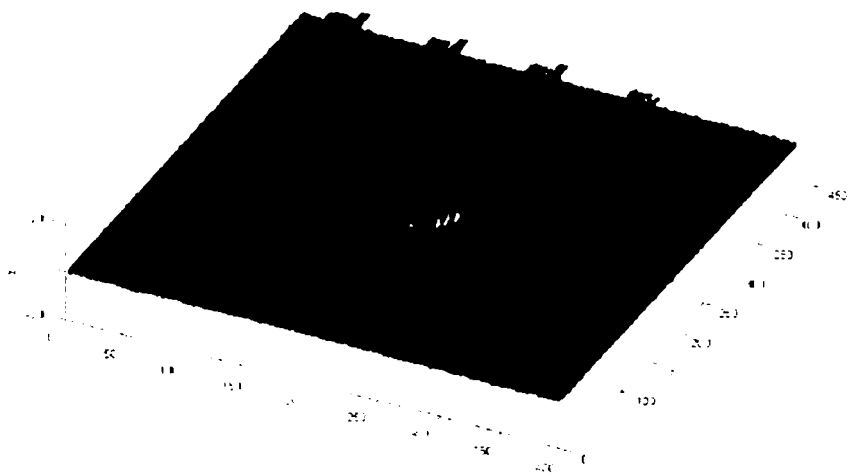

Figure 30 Object correlation:
According to the step width the VRI objects are virtually put in the real scene for pose evaluation.

The application example in Figure 30 shows an application of bin picking for automotive component suppliers. It is characterized by the fact that the objects are often of the same shape. In this application, the pose estimation can be limited to a fixed increment of ΔX, ΔY and ΔZ, which depends mainly on a priori knowledge of the object position in the scene. For example, in the case of the door joints in Figure 30, it can be assumed that the distance Z does not have to be changed in major steps because all the door joints are lying on the bottom of the box. The process of coarse pose estimation is a preliminary search for a best-matching object pose. In general, different VRIs for different kinds of objects are compared to the RRI in the same way. Therefore, the object classification is integrated in the step of object localization. The VRI, consisting of the surface's representation of the 3D model, is compared to the real range image (RRI) of the scene made by the sensor. One advantage of this pre-selection of matching positions is the fact that all VRIs can be calculated offline and stored in a database. Due to this, the process for the coarse pose estimation can be summarized in the following way:

- The RRI is delivered by the sensor
- All VRI in the database (one for each possible pose) are compared to the RRI
- The best VRI candidates are selected for pose refinement

Because all VRIs are compared to the RRI, each VRI gets an error value. The VRI candidates with the lowest error values are selected for pose refinement. The number of best matching VRIs can be limited by an error-threshold or a fixed number of VRI candidates or a combination of both. The error threshold depends on the object size and the increment size. It can be determined by taking the VRI candidates within the best 10-20% of all error values in the coarse pose estimation process. The aim of coarse pose estimation is the reduction of possible solutions that will be found in the pose refinement. The pre-selection results in a few VRI candidates. These candidates are delivered to the pose refinement process, starting with the best matching candidate. The alignment of the VRIs to the RRI data can be done in different ways. The next sections introduce different possibilities.

3.5.1 Brute Force Pose Estimation

The position and orientation of the object is estimated by range data comparison. The goal of the coarse pose estimation is the pose estimation of all visible objects in the RRI. Figure 30 shows the determination of the best matching VRI in RRI. The Brute Force Pose Estimation compares every VRI to the RRI with a defined error function. The error function returns an error value. If the error value is low, the VRI matches with the RRI. Due to the fact that all VRIs are compared to the RRI, each VRI gets an error value. The VRIs with the lowest error values are selected for pose refinement. The error value is the scalar that specifies the level of the correlation between VRIs and the RRI. In the first implementation of this system, the error function is defined as:

$$Error = \frac{1}{N}\sum_{i=0}^{X}\sum_{j=0}^{Y}\left|Z_1(i,j) - Z_2(i,j)\right| \qquad 3.16$$

The error value is calculated for each pixel of the object in the VRI, and is defined as the mean of the difference between every distance value Z_1 of the simulated object and the distance value Z_2 of the scene. The error depends mainly on the positions X, Y, Z and the rotation around the axes R_x, R_y, R_z of the simulated object, and the limits of degrees-of-freedom of the object. The error value is always higher than zero, except for the perfect match of VRI and RRI if the remaining sensor error is taken out of consideration. In the real world, the perfect match is nearly impossible because of the inaccuracy of sensor measurement and errors of bounding pixels. Outliers or invalid points must be filtered in the preprocessing steps, and this error function provides a rate of how good the pose of the model matches with the real image in the distance measurement.

```
//Moving all poses over the surface
For all positions in X
 For all positions in y
  For all positions in z
   For all rotations in Rx
    For all rotations in Ry
     For all rotations in Rz
      For all distance values in object surface
       SumError = Zi1-Zi2
      Next distance value
     Next rotation step
    Next rotation step
   Next rotation step
  Next position step
 Next position step
Next position step
```

Figure 31 Brute Force Pose estimation Pseudo Code
This Pseudo Code example shows the inefficiency of a brute force matching with a high complexity.

This implementation compares all poses in the VRI database to every possible position in the scene's data set. Figure 30 shows a translation of the VRI over the RRI scene. With a certain step width, the simulated VRI for this position is loaded from the database and put in the RRI scene to compare these representations according to the error function. This algorithm offers the following advantages. Appearance-based comparison can be used for every kind of object and any type of sensor independent from the resolution of the data. Additionally, it does not use any feature detection. Most known algorithms use features for object recognition [113] and localization in any form. The problem with these algorithms is the fact that they must be parameterized for every new object that needs to be learnt. Features that can be hidden by other objects or features are not found in the

scene for any other reasons.

Since all poses and positions in the sensor field of view of a CAD object are stored in a database, the simulation takes the exact appearance representation of the simulated object in consideration. Once more, it is important to mention that this also includes the position of the object in the sensor field of view because this leads to a more exact VRI representation explicit with, for example, self occlusions. These self occlusions result from the setup and position of the virtual sensor compared to the object. This leads to a high number of very similar VRIs in the database but increases the accuracy.

On the other hand, the brute force search over all poses and positions is highly computationally expensive. The maximum complexity is $O(k \times DOF \times N)$ for k objects for the error calculation in equation 3.16. In many applications it is possible to limit the DOF to at least three.

Nevertheless, this algorithm is used in the evaluation tests as a reference test to show the accuracy and resolution in contrast to the other introduced algorithms in the next sections.

3.5.2 Advanced iterative pose estimation

The proposed brute force pose estimation tries to match the position and orientation of the VRI with the RRI in defined steps of DOF resolutions. In pose estimation, all the poses of the object are created beforehand and stored in the database. The opposite of this is the advanced iterative pose estimation that creates the poses of the CAD model within the pose estimation process. The pose is changed iteratively to find the best alignment.

In every iteration step

- The pose of the 3D model is changed
- The VRI is created
- The VRI and the RRI are compared

The error value is used to define a minimization problem of the alignment process. Figure 32 depicts the iterative extension of this principle.

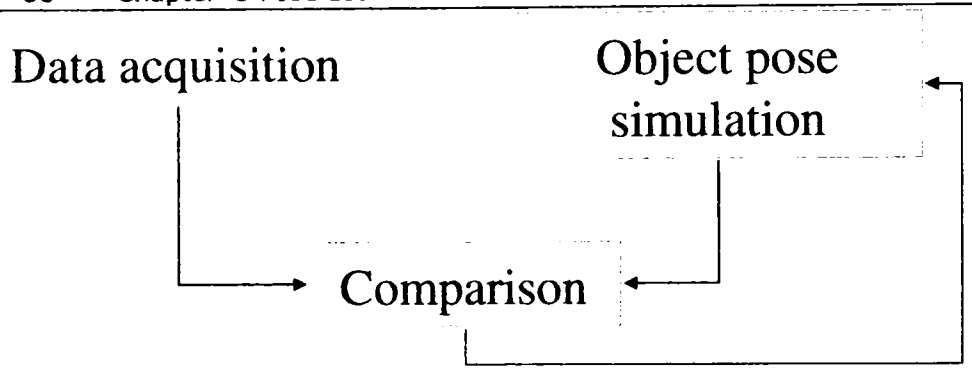

Figure 32 Advanced iterative pose estimation
The iterative pose estimation refines the pose step width for each degree of freedom iteratively.

The error depends mainly on the positions X, Y and Z and the orientations R_x, R_y, R_z, so the equation 3.2 can be written as:

$$Error(X^{VRI}, Y^{VRI}, Z^{VRI}, R_X^{VRI}, R_Y^{VRI}, R_Z^{VRI}) = \frac{1}{N}\sum \left| Z_{RRI}(i,j) - Z_{VRI}(i,j) \right| \qquad 3.17$$

Here, Z is the projection of the 3D model to a 2.5D range image made by the sensor simulation. The projection function of the real sensor is assumed to be identical to the projection function of the simulated sensor. By varying the values of X, Y, Z, R_x, R_y, and R_z in smaller steps compared to the pose estimation, the error is minimized. The implemented search optimization algorithm follows the maximum gradient in the six-dimensional error hyperspace for each DOF individually. The iteration is stopped if a specified threshold for the error value or a maximum number of iteration steps is reached. This algorithm is related to the optimization technique called *hill climbing* [114], which belongs to the family of local search. However, this algorithm is also vulnerable to wrong alignments because of local minima. In this thesis, only a basic search algorithm is used, but this minimization process can be accelerated applying minimization methods like *simplex minimization* and other minimization algorithms such as *Monte Carlo methods*. This approach was planned to be integrated in the refinement step, but the performance of the sensor simulation is much slower than the proposed registration algorithm.

3.5.3 Accelerated brute force pose estimation

To improve the performance of the brute force pose estimation process, the comparison is extended with a common algorithm in signal processing. This cross correlation is generally used to measure the correlation between two signals — especially in image processing correlations that are calculated between sliding two-dimensional search windows (template) and the image. These matching functions only rely on the intensity values. One of the best known methods is so-called "block matching". Block matching is a commonly used process in regions-based approaches for the solution of the correspondence problem in stereo vision. Even if it belongs to the area-based approaches it is, nevertheless, local; surrounding one pixel of a significant region in the image. The procedure of block matching for stereo

vision is very simple in first approximation. Every picture is subdivided into blocks, which are then compared with blocks in the other image. The template is moved pixel-wise over the other image and a quality value is calculated from a matching metric. Hence, the cross correlation determines how similar the two signals are. The cross correlation for two-dimensional signals I_1, I_2 is defined as:

$$CC(x,y) = \sum_{x,y} I_1(x,y) \cdot I_2(x-u, y-v) \qquad 3.18$$

where u,v is the shift of the pattern in the image I_1. This cross correlation could be used to compare a pattern with an image. The comparison of two blocks is usually based on the similarity of the intensity values (but not limited to them). Unfortunately, this procedure is mainly not invariant to lighting changes in the image (see [115]). The corresponding pixel lies in the center of this block. The papers of Roma [116] and Brown et al. [117] give an overview of different matching metrics for the calculation of the similarities between the blocks, one popular example is *Least Square Matching*. The basic concept of *Least Square Matching* is the minimization of the sum of the squares of the gray value differences between the pixels in the two sizable blocks. Sufficient texture in the blocks with a bandwidth-limited signal leads to good approximation values at convergence within less iterations [118]. Another possible matching metric is given by the normalized cross correlation (NCC). Using the normalized cross correlation (NCC),

$$NCC(x,y) = \frac{\sum_{x,y} \left(I_1(x,y) - \bar{I}_1\right) \cdot \left(I_2(x-u, y-v) - \bar{I}_2\right)}{\sqrt{\sum_{x,y} \left(I_1(x,y) - \bar{I}_1\right)^2 \cdot \left(I_2(x-u, y-v) - \bar{I}_2\right)^2}} \qquad 3.19$$

most of the disadvantages of the two-dimensional cross correlation can be overcome. The mean of signal I_1 and I_2 is taken into consideration. Since the cross correlation is related to the convolution process, the calculation can be made in the frequency domain to increase the speed of the calculation process. To calculate the NCC in the frequency domain, the signals must be normalized, as described in [115]. In the implementation of Matlab (www.mathworks.com), the normalized cross correlation is limited to two-dimensional signals (images). The resulting cross correlation coefficient is high at this position where the pattern fits to the image. Figure 33 shows the peak in the cross correlation "surface" at the position of the best matches.

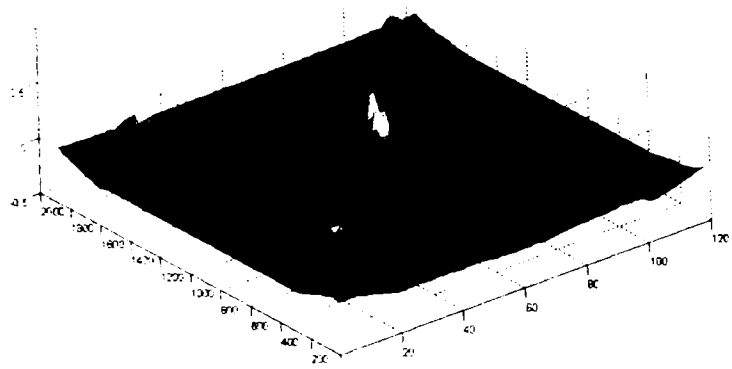

Figure 33 Cross correlation result
The normalized cross correlation is an efficient way to find the best position of a representation in the scene.

This test is made for one object in the scene and one pattern with a fixed distance to the sensor and fixed rotation. These restrictions make this solution by itself nearly useless for pose estimation in common industrial applications. However, in combination with the already introduced "brute force" pose estimation, the performance is increased dramatically. In this case, the process can be modified; for every rotation step, the VRI is created and used as a cross correlation template. The advantage of cross correlation calculation is given by the performance of the frequency domain calculation for large patterns and images. The two-dimensional cross correlation can be further extended to an n-dimensional cross correlation to deal with all degrees of freedom [119].

3.6 Feature-based object localization

3.6.1 Feature correspondences

A feature is a distinctive part of an object that helps to identify the object and determine its pose. Features range from simple edges to high level features in 2D image object recognition and localization. Additionally, they could be invariant to image translation, scaling, rotation; or they could be partially invariant to illumination changes and affine or 3D projection like the Scale Invariant Feature Transform (SIFT) developed by Lowe [113]. With the extracted features, an object can be localized by the search of only the representation of this feature in a scene. The most significant features of an object are selected because they can be very well identified in a scene representation. On the other hand, the feature is very specific and depends on the kinds of objects in the application, so the feature-based approaches often suffer from the lack of universality.

Because features mostly do not consist of single pixels, features are usually

extracted by the different processing layers of the image processing. If the points with identifying gray value and extracted features like edge points, lines or even regions are merged together, the ambiguities can be solved easier and faster in correspondence analysis. This well known field of image processing offers numerous methods of features detection in 2D imaging, but a basic problem of feature-based methods is still the skilful selection of a suitable detector and criteria for an automatic, objective quality assessment.

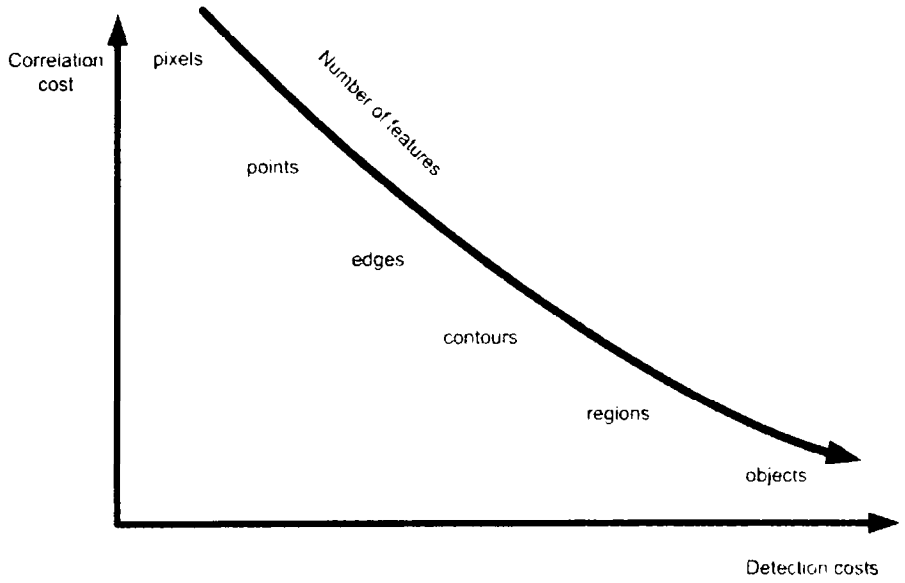

Figure 34 Feature correlation
With an increasing complexity of features the correlation costs are increased.

In Figure 34, common features are arranged with their qualities in relation to the correspondence analysis. The reliability and simplicity of the feature with major information content are better because of their smaller number, but the computational costs are therefore higher. For many applications, the selection of the right features is extremely important and can be made according to the following criteria [120]:

- Reliable extraction
- High information content
- Perspective invariance
- Robustness against occlusions and noise
- Universality

Feature-based procedures can be roughly separated in intensity or pixel-based methods, edge-based, corner-based methods and feature extractions of higher complexity. Point- or intensity-based methods can make a correlation between the intensity of every single pixel or the position properties of points to find correspondences. In the case of the pixel-based methods in image processing, the feature with the smallest information content is selected as an allocation feature — a pixel with his intensity value $I(X,Y)$. Ordinarily, the number of intensity values in a picture is very small in comparison with the pixel numbers, so a direct assignment is almost not possible (unless there are very small pictures). Most pixel-based methods have problems in different illuminations. Nevertheless, most approaches use other methods because a pixel-based method has very high computational costs.

Many feature-based methods work with edge detection, which is a common method of image processing related to image segmentation. An edge is, by definition, a discontinuity in the image separating different areas. These edge-based methods are inspired by biological stereo vision [121]. There exist a large number of image processing operations that make it possible to recognize edges, including the well known edge detection of Canny [122]. A comprehensive summary of common edge detections can be found in [123].

Image operators to locate edges in images could be separated like this [124]:

- Simple local operators
- Template-matching operators
- Optimal edge operators
- Morphological operators
- Parameterized edge models

A huge amount of edge operators exist for feature-based correspondence analysis. Some important high level interest operators use the results of edge detection methods and other methods extract feature points (Moravec-Filter [125], Harris corner detection [126]). With these corner detection algorithms, the maximization of the gray-scale value differences in directions all around the point is attempted. One example is the usage of the endpoints of recognized edges whose quality is computed by the form, size and position of the edge segments, as discussed by Dhond and Aggerwal in [127]. Tomasi and Birchfield [128] use regions in images that are extracted through segmentation. Like most feature-based methods, this reduces the match sensitivity to depth discontinuities. Modern algorithms are extended to scale invariant features and many other improvements [129], [130]. Based on this, SIFT features have a great success in interest point detection [131]. However, the main drawback of feature-based methods is that they are limited to good-natured object with features for 2D image processing. The area between the features remains unconsidered and must be processed in further steps. A fundamental other approach for the disparity calculation pursues surface-based or area-based techniques. These techniques match the corresponding regions with

very similar appearances. Unlike feature-based techniques where in a preprocessing step point, lines or regions are extracted, area-based methods use a cost function-based directly on the gray value in the images or distances. A disadvantage of feature-based methods lies in the fact that features must be extracted and must exist in a sufficient number in the input data. Therefore, area-based methods can be used with every kind of data reliably, but suffer from a smaller convergence range as well as smoothing effects in object discontinuity, which leads to a localization inaccuracy.

3.6.2 Example for feature-based object pose estimation

In many applications of industrial robotic bin picking, it is possible to reduce the complexity of the pose estimation process by feature-based object localization. Features are often used in camera-based systems [132] to detect specific parts of an object. This object localization is being further extended to 3D range data object localization problems[133]. There exists a variety of applications where feature-based localization is the first choice. Therefore, this section gives an example for a feature-based pose estimation application for robotic bin picking.

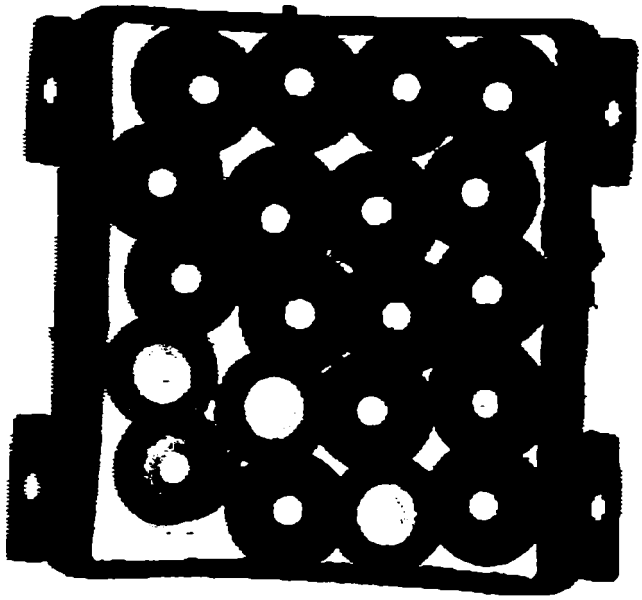

Figure 35 Brake disk application example
The brake disks in the box must be found in the range image and picked up by a robot.

In Figure 35 the distance values are represented as gray values in the picture. Darker pixels represent closer distance values, and objects with brighter pixels are farther away. This picture shows brake disks lying in a bin. Most of the brake disks are arranged planar, so the object localization is an easy task for feature-based localization. The proposed object localization finds the best matching brake disk in the bin. To find the hole in the middle of each brake disk, this section describes a fast and simple algorithm. The first part of the algorithm is based on

commonly used 2D image processing methods. Therefore, two circles with a fixed radius must be found that represent the topmost ring of each brake disk. There exist many different methods such as contour-based matching or Generalized Hough transform (GHT) to find the known pattern in an image [134]. In this application [135], a commonly used library PATMAX© from COGNEX (www.cognex.com) finds the best matching brake disk in the image by taking orientation, scale and occlusion into consideration. The matching algorithm requires the number of objects to be known as well as the template form of the object. The search can be minimized because of the rotational invariance and the given maximum slope of the objects in this application. The library delivers the best matching templates in the image with positions X and Y and information about their similarity, scale and slope referring to the template. Additionally, the coverage is determined so that occlusions can be found very easily. All these results are taken into consideration for one quality value which is calculated for each found brake disk.

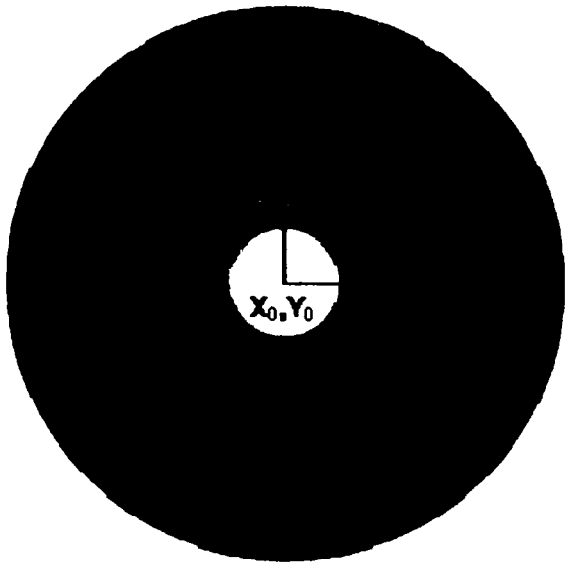

Figure 36 Template matching
The template given by the circle representation provides four regions to form the front plane of the brake disk.

The object position X Y in the image coordinate system is similar to the center point X_0, Y_0 in the object coordinate system. Around the center point of the chosen disk, four small regions (4x4 pixels) between the two circles are used to find the orientation of that disk. The regions are shifted by 90 degrees within the two circles (in the object coordinate system: $\pm\Delta X$, $\pm\Delta Y$). Inside these rectangular regions, the median gray values are extracted.

$$Z(r)_{median} = \frac{1}{2} \cdot \left(Z_{\frac{n}{2}}(x,y) + Z_{\frac{n}{2}+1}(x,y) \right) \qquad 3.20$$

For each 4x4 region *r* the median is calculated with the equation 3.20. The value r =0,2 define the two regions in direction of X in the Figure 36; r =1,3 define the regions in direction of Y in the Figure 37. If the brake disk is planar all sensor distance values are in the same plane. With these four points this plane is fully defined and the orientation is calculated with the direction vectors:

$$\vec{V}_X = \begin{pmatrix} X_0 + \Delta X \\ Y_0 \\ Z(0)_{median} \end{pmatrix} - \begin{pmatrix} X_0 - \Delta X \\ Y_0 \\ Z(2)_{median} \end{pmatrix} \qquad 3.21$$

$$\vec{V}_Y = \begin{pmatrix} X_0 \\ Y_0 + \Delta Y \\ Z(1)_{median} \end{pmatrix} - \begin{pmatrix} X_0 \\ Y_0 - \Delta Y \\ Z(3)_{median} \end{pmatrix} \qquad 3.22$$

The cross product of this direction vectors delivers the orientation of the plane, given by the four points representing the brake disk:

$$\vec{V}_{Z'} = \vec{V}_X \times \vec{V}_Y \qquad 3.23$$

Each gray value is assigned to a specific distance value. The distance Z to the sensor can be recalculated with the help of this gray value.

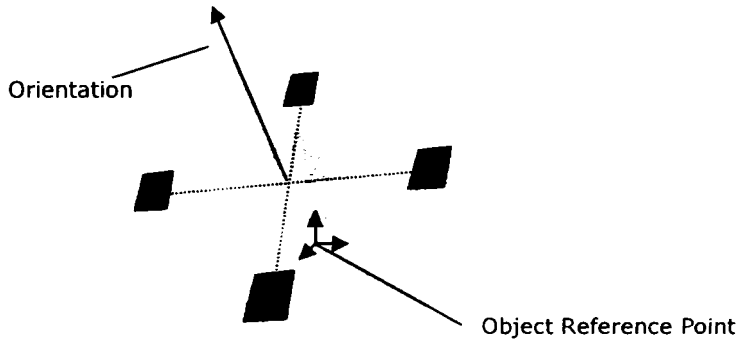

Figure 37 Object orientation determination
The orientation of the brake disk is calculated from the normal vector of the plane formed by the four regions.

With the known depth *d* of the brake disk, the object reference point (*ORP*) can be calculated with the following equation:

$$ORP = \begin{pmatrix} X \\ Y \\ \frac{1}{4}\sum_{r=0}^{3} Z(r) - d \end{pmatrix} \cdot \vec{V}_{Z'} \qquad 3.24$$

The *ORP* is transformed into the robot coordinate system, so the position of the object is known for the robot or the pose refinement step. The proposed solution combines well known algorithms of 2D image processing with 3D range data from a laser distance sensor to find a successful solution for this application. Furthermore, this feature-based pose estimation gives the results to the pose refinement step for better accuracy. Depending on the objects and their features, it could be far more feasible to find an easy feature-based solution, but this would increase the cost and complexity for the installation and adaptation to the specific application in robotic bin picking.

Chapter 4 Pose refinement with ICP

The term "pose refinement" is used when the initial position and rotation of an object in a scene is roughly known. A very common problem in range image processing is the reconstruction of 3D models of real world objects. Hereby, data sets of points are available in different orientations or views of the real object and should be merged together. It is the task of the registration algorithm to find the relative position and orientation of a data set of points A to another data set of points B. The registration can be defined as:

"The determination of a geometrical transform that aligns points in one view of an object with corresponding points in another view of that, or another, object."

The proposed solution uses the registration approach to find the exact match between the real range data set and the model range data set. Therefore, corresponding points must be found and matched with a pairwise rigid alignment technique. The algorithm is based mostly on a minimization process of the distance. The literature gives examples of the most relevant registration algorithms [136] and Iterative Closest Points algorithm (ICP) is the most important and fundamental registration algorithm. The ICP algorithm and its variants are approaches often used in registration tasks. The ICP was introduced by Besl & McKay in 1992 [6] and in parallel by Chen & Medioni[137]. It belongs to the 3D to 3D registration process [23] and, generally, determines a transformation between two or more object representations [8] based on the distances of the closest points of these data sets. Due to the exhaustive search for the closest point, several improvements on the ICP method have been developed[138], [13], [139]. Only compatible and view point independent feature points are extracted in [140] to reduce the number of points. Therefore, many approaches try to extract features from the range data that can be used for matching, as shown in[141]. These strategies are advantageous in terms of better performance and also in terms of the robustness if features are available in range data. There are many other feature-based ICP approaches that can be found in the literature [31]. In this chapter the algorithm is generally described and some application fields were introduced. The chapter describes the algorithm in detail and shows some expansions and improvements. An innovative fast variant of ICP is introduced and evaluated in the following.

4.1 Fields of applications of Iterative Closest Points algorithms

The registration of point data is found in many different areas of daily life. In the next sections, some examples are shown for the usage of the ICP algorithm in different areas. The applications are not limited to the following application areas. Even further applications in the areas of education/study, games, e-commerce and more are possible.

4.1.1 Medicine

Registration methods support doctors attending to the diagnosis and therapy of patients. 3D data from different image sources (MRT + PET) are merged with registration methods. The combined result is used to fight against cerebral tumors or to support surgery. Depending on the applications, different methods of graphical data registration are used. The registration of graphical data plays a major role in different systems. Image data must be transformed by:

- taking from different image sources to each other
- taking from one point of view to another point of view
- taking at different times
- taking to compare different patients

In this context, the ICP algorithm plays a major role. For example, Feldmar et al. [142] use an extended ICP algorithm. They used a non-landmarked approach and extended the data with an additional dimension to include the intensity. These methods were proven for the comparison of MRT brain photos or for the detection of heart ischemia.

Another variant of ICP is used by Stewart, Tsai and Roysam [143] to register retinal pictures. The so-called Dual Bootstrap ICP registers small segments, based on the qualities of the blood systems, and applies the transformations to the overall picture. The applications in medicine or biometrics [144] are various. Many further medical applications are reviewed in the work of [145].

Jain, Chen and Demirkus [146] use ICP to align and match fingerprints by extracting level 3 features of pores and ridges in the fingerprints. These features are automatically extracted using wavelet transform and Gabor filters and are locally matched using the ICP algorithm.

4.1.2 Engineering

In computer graphics, the registration of three-dimensional data is also a major topic of research. CAD models of solid objects are produced with the help of sampling or scanning (reverse engineering). Here especially, laser distance sensors are used. The ICP algorithm is often used to align the acquired data of two- or three-dimensional object scans.

An interesting approach proposed by Gutmann and Schlegel [147] is the self localization of mobile robot agents. Two-dimensional laser scans are taken by a mobile robot. A general term for mobile robots is "SLAM" (Simultaneous Localization and Mapping). The resulting transformation of the registration yields the position of the mobile agent. An often cited approach comes from Lu and Milos [18], and is used in further developments and provides stable localization [147], [17], [148], [149].

Another application is the digital reconstruction of cultural objects. The sculpture of Michelangelo's David was digitized by a complex procedure [150] by the University of Washington. For this purpose, many parts of small laser scans of the sculpture were brought together again to produce a complete model. Kari Pulli [20] uses a modified variant of the ICP algorithm to arrange the single scans. With these methods, a full model can be put together automatically. Further improvements in accuracy are made with ICP-based non-rigid alignment by Brown [8].

4.1.3 Photogrammetry

Photogrammetry is a scientific discipline that deals with the passive reconstruction of models in topographic mapping and measurement.

Huber [151] describes an application for 3D map construction of areas and buildings using a variant of ICP from different scans of distance sensors with a long range, resulting in a three-dimensional map. In this application, a large amount of data arises with all sorts of resolutions and dimensions. With the help of the registration, large maps merge from single scans. In [27], map matching uses ICP to refine the multilevel surface maps for the extraction of building and city reconstruction.

4.2 The Iterative Closest Points algorithm

The ICP algorithm was developed by Besl and McKay [6] and in parallel by Chen and Medioni [137]. The application of the ICP algorithm is not restricted to a representation of an object. Points, lines or triangle sets, implicit curves/areas and parametric curves and areas could be used as geometrical representations of the objects. Matching methods can generally be distinguished according to the result in transformation. The transformation of the ICP algorithm is part of the class of the rigid transformations that only moves and rotates the object. The ICP registration is a rigid transformation that maps one object to another as well as possible. In general, this is a complex task because the correct mapping between the points is not known beforehand. As the name already implies, ICP is an iterative algorithm, which determines the result with the help of a mathematical minimization. Therefore, the knowledge of an approximate initial solution is important for the success of the method. An iteration of ICP is separated into four steps (Figure 38). Initially, the corresponding points in each data set must be found. Every point of the scene is then assigned to the Euclidean closest point of the model. With the help of a minimization function, an alignment transformation can be calculated, and the algorithm minimizes the mean square error of the point distance in several iteration steps. An iteration step finishes by applying the resulting transformation to the

scene and recalculating the distance error between the data sets. To abort the iterations, a threshold or a maximum number of iterations can be implemented. ICP converges monotonously to a minimum. The next section introduces the base algorithm of ICP. It starts with mathematical basics followed by expansions and modifications of the ICP algorithm.

4.2.1 Model and scene

The set of points of the model is defined as:

$$M = \{m_i\} \quad with \quad m_i = (x_i, y_i, z_i), i = 0,1,...N_i \tag{4.1}$$

The set of points of the scene is defined as:

$$P = \{p_j\} \quad with \quad p_j = (x_j, y_j, z_j), j = 0,1,...N_j \tag{4.2}$$

Model M can consist of points, line segments, parametric curves, implicit curves, triangles, parametric surfaces or implicit surfaces. The representation becomes important at the calculation of the distance values, where all these representations are re-converted back to point representations again. Therefore, the model and scene data sets are assumed to be points in the Euclidean space in this thesis.

4.2.2 Point correspondence and distance function

Every point of scene p_i must be assigned to one point of *M*. Due to simplicity purposes, the number of points in the two sets of points is assumed to be equal in the following equations. Besl and McKay [6] describe in their work an assignment of *P* and *M* with the smallest distance. The distance is calculated with the help of the Euclidean distance:

$$d(p_j, M) = \min(p_j \varepsilon P \| m_i, p_j \|) \tag{4.3}$$

This includes $m_i \varepsilon M$ where *M* can be a set of points, lines or curves. The function always provides the point with the smallest distance to the point p_i of the scene *P*. The resulting point correspondences calculated with the distance function are summarized in:

$$Y_k = C(p_k, M) \tag{4.4}$$

C is the closest point operator, *p* is the transformed point set per iteration *k* and *M* the point set of the model. Finding the smallest distances in every iteration step is the most complex and time-consuming function of the ICP algorithm. This operation has a complexity in the not optimized implementation of $O(N_i \times N_j)$. This fact leads to an unintentionally high calculation time if the set of points are very large.

4.2.3 Closest points error minimization

The distances of all points of the model M and the transformed scene must be minimized. This reflects the following error function

$$E = \frac{1}{N_j} \sum_{i=0}^{N_j} \left\| m_i - T(p_i) \right\|^2 \qquad 4.5$$

The transformation T can be separated in a translation t and a rotation R. This leads to:

$$E = \sum_{i} \left\| m_i - R(p_i) - t \right\|^2 \qquad 4.6$$

Different strategies can be separated into direct and indirect methods with (closed and not closed form) solutions to find the minimum of $E(R,t)$.

Four methods for the closed form transformation calculation can be used [152]:

- The transformation estimate under the use of the unit quaternions
- The transformation estimate by means of the singular value reduction of a matrix
- The transformation estimate with the help of orthogonal matrices
- The transformation estimate with dual quaternion

They have examined and compared the closed methods. The best known and most important transformation estimate employs the method of unit quaternions, which is explained in the following section.

4.2.4 Quaternion-based approach according to Horn

Rotations can be described very effectively with the help of quaternions, making sequences of rotations extremely simple. In addition, no reflections in solution appear. Besl and McKay [6] uses the quaternion-based approach introduced by Horn [10]. With the unit quaternion:

$$q_r = [q_0 \; q_1 \; q_2 \; q_3]^T \qquad 4.7$$

with $q_0 > 0$ and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 0$ the rotation is shown in the following:

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 - q_0q_3) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix} \quad 4.8$$

Substituting this in the error function mentioned above, which has to be minimized, this results in:

$$E = \sum_i \|m_i - R(q_r) - q_t\|^2 \quad 4.9$$

With the translation vector

$$q_t = [q_4\ q_5\ q_6]^T \quad 4.10$$

the vector

$$q = \begin{pmatrix} q^r \\ q^t \end{pmatrix} \quad 4.11$$

describes the current situation of the registration process. The closed form solution, which is used in this work, was originally proposed by Sanso[153]. It agrees with the solution introduced by Horn [10], which is briefly shown here. The required rigid transformation is represented by the formula:

$$T(p_j) = R(p_j) + t \quad 4.12$$

The rotation matrix is usually a matrix of the dimension 3x3. The translation vector t is a vector of the quantity 3, and the solution of the problem is separated into two tasks — the calculation of the rotation matrix R and calculation of the translation of vector t. With the "center of mass" of the given set of points

$$\mu_p = \frac{1}{n}\sum_{i=1}^{n} p_i \quad 4.13$$

$$\mu_m = \frac{1}{n}\sum_{i=1}^{n} m_i \quad 4.14$$

the relative point data μ_m and μ_p are calculated. With this result the covariance matrix can be built:

$$\sum_{PM} = \frac{1}{n}\sum_{i=1}^{Np}\left[(p_i - \mu_p)(m_i - \mu_m)^T\right] = \sum_{i=1}^{n}\left[p_i m_i^T\right] - \mu_p \mu_m^T \qquad 4.15$$

With help of the covariance matrix $\sum_{PM}$ the anti-symmetrical help matrix *A* is built according to:

$$A = \sum_{PM} - \sum_{PM}^{T} \qquad 4.16$$

The cyclic components of the anti-symmetrical help matrix turns into the vector:

$$\Delta = \begin{bmatrix} a_{23} \\ a_{31} \\ a_{12} \end{bmatrix} \qquad 4.17$$

and a symmetric matrix Q is built:

$$Q(\sum_{PM}) = \begin{bmatrix} tr(\sum_{PM}) & \Delta^T \\ \Delta & \sum_{PM} + \sum_{PM}^{T} - tr(\sum_{PM})I_3 \end{bmatrix} \qquad 4.18$$

The term $tr(\sum_{PM})I_3$ is hereby the multiplication of the sum of the main diagonals of the covariance matrix and an identity matrix in $\Re^3$ (trace of the covariant matrix). The elements of the normalized eigenvector of the greatest, positive eigenvalue correspond to the elements of the unit quaternion (Equation 4.7) representing the target rotation. The associated rotation matrix *R* can be calculated after equation 4.19. With a known rotation matrix *R* the translation vector *t* is calculated as:

$$q_t = \mu_m - R * \mu_p \qquad 4.19$$

The transformation for the new scene is given by Formula 4.12.

Summarizing the ICP algorithm, an iteration of the method contains the steps according to Figure 38. After initialization, the corresponding closest point m_i to every point p_j must be found. After that, the algorithm calculates the transformation from $Y_k = C(p_k, M)$ by minimizing the square error between the points of the scene and the corresponding points of the model.

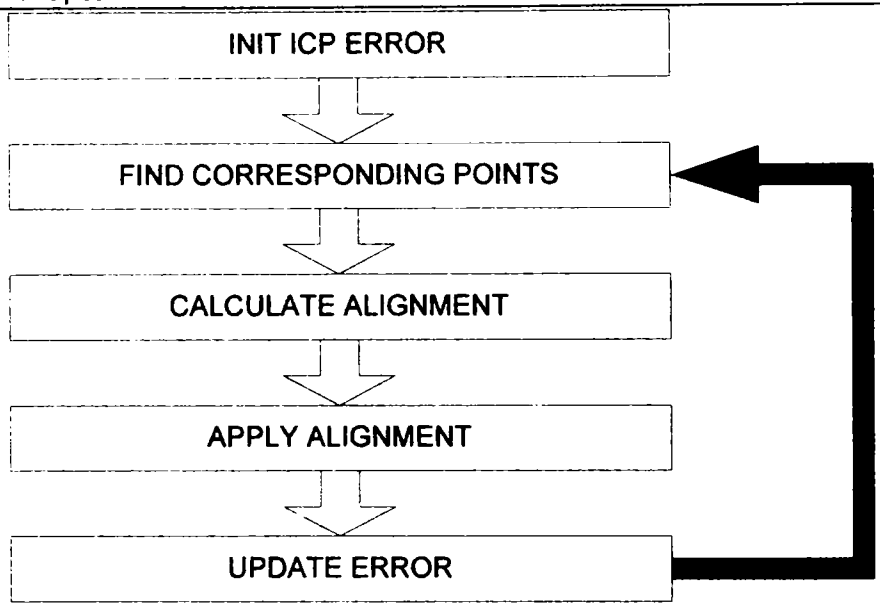

Figure 38 Iterative Closest Points
The process of ICP can be separated into several iterative steps after the initialization. The alignment is calculated in each iteration step from the corresponding points and applied to the data set.

This transformation is applied to all points of scene *P*, and this process is repeated iteratively up to the point where it reaches a threshold. Otherwise, the iteration starts again. For example, if the error difference is less than a certain bound, the iteration is stopped. The choice of a threshold value is dependent on the qualities of the objects involved. Alternatively, a maximum number of iteration steps also interrupt the minimization process.

The iteration stop criterion could be extended to many other possibilities [11], [154], [155]:

- Fixed number of iterations
- Absolute error threshold
- Error difference threshold
- Pose change threshold
- Complex error criterion

In general, all of these iteration stop criterions can be implemented simultaneously. Whenever one of these criterions is met, the iteration stops. The algorithm converges monotonous, as proven by Besl and McKay [6], and the alignment accuracy is defined by the minimum of the standard square error. If the iterative minimization always converges towards a local minimum, it is not

necessarily the global minimum. Therefore, the choice of a suitable initial start position is an important factor for scene *P* and model *M* to find the global minimum. The major problem of the base algorithm is the high number of iterations finding the correct transformation. This leads to a time-consuming registration system. The introduced ICP algorithm of Besl and McKay [6] has been modified and improved several times. Some of the most popular and efficient results of further approaches are shown in the following sections.

4.3 Modified ICP algorithms

The ICP algorithm of Besl and McKay [6] does not perform particularly well in some cases if an optimal transformation has to be found between two surfaces. There exist different scenarios where the ICP method has profound problems and arrives at only bad or incomplete solutions. Special attention has to be placed on the choice of the initial position. The ICP algorithm is vulnerable to the convergence to local minima, which do not have to correspond to the global solution. A good initial solution is particularly important if a smaller surface should be transformed completely on a part of a bigger surface.

The ICP algorithm works well if the point sets have sufficient overlapping and continuous surfaces. Overlapping surfaces cause problems because the calculated transformation is no longer identical with the actually desired transformation. In this case, the unmodified ICP algorithm cannot provide the desired result. The weaknesses of the ICP algorithm concerning these problematic cases cannot be eliminated completely. Numerous variants have been developed that work substantially better in such cases, as well as also improving the performance and precision. Furthermore, the base algorithm is vulnerable to outliers in the point data, which destroy the guaranteed convergence of ICP [156]. Many extensions exist to increase the robustness against outliers in scenes and models. The biggest problem of the ICP algorithm is the time-consuming calculation of the transformation due to the exhaustive search for the nearest point [13]. Therefore, most approaches, expansions and comparisons concentrate on the acceleration of the algorithm itself.

According to [13], the ICP algorithm can be separated into the following variants:

- Selection: Selection of points from one or the two sets of points
- Matching: Finding corresponding points to the selected points
- Weighting: Add weights of the point pairs
- Rejecting: Excluding some of the point pairs
- Minimizing: Assigning and minimizing an error metric to the corresponding points

All these variants are described as follows.

4.3.1 Selection

To enhance the speed, a reduction of the used points is logically consistent. Besl and McKay [6] use all points of the scene to find the closest point in the model. A reduction of the points brings a great improvement in the time intensive correspondence search given by large sets of points of the scene. Turk and Levoy [138] use uniformly distributed samples of points. Only 1-5 percent of the complete set of the points are sufficient to reduce the costs drastically without any effects on accuracy. Masuda et al. [139] change the selected points at every iteration step randomly. Additionally, this increases the robustness because a bad initial selection causes the algorithm to not necessarily converge. Normal-space sampling selects points to cover a uniform distributed spectrum of the point's normals [13]. In contrast to the described selection of points, some approaches use a selection of points with special features[157]. Gobin et al. [158] select the points from the set of the scene due to the correspondence of the intensity values. These are provided by the used laser scanner and search for the next respective neighbor in the model set of points. Sharp et al. [159] add curvature and moments of a point beside further characteristic qualities for the reduction of the vulnerability of outliers in the data. The experiments of [13] show that there are no great differences in convergence with varying selection methods (uniform sampling [138], random sampling [139], or normal-space sampling [13]). Nevertheless, the selection of points dramatically increase performance without (or at least minor) a loss of convergence quality [41]. As a consequence of this, the core ICP refinement algorithm in this thesis will extend this by using an optimized sub-sampling of points for every iteration step (Section 4.4).

4.3.2 Matching

When finding corresponding points, there is the possibility to search the closest points directly, as shown in Besl and McKay [6]. This method is very complex, because all points of the scene and the model must be compared to one another. The complexity has to be valued with $O(N_j \times N_i)$ and is the most time-consuming step of the whole ICP algorithm. This complex method can be improved by many different methods [33]. Simon [11] describes a high increase of the speed by using k-d trees and closest point caching, as was suggested by Zhang [160] and improved by [161]. The use of a k-dimensional binary tree search permits excluding big regions in the search space [162]. At every decision in a tree node, one side of the hyper plane can be rejected. Compared to the base algorithm, the use of a suitable tree search reduces the required time by more than 90 percent[11], [163]. Simon combines further methods such as closest point caching with the tree search, and these have been proven in additional articles[164] and[163]. Especially when large point data sets exist, the use of a binary selection tree is important. The complexity can be reduced to $O(N_j \times \log N_i)$ in comparison with the brute force search in [6]. In his comparison studies, Zinsser [163] comes to essentially the same result. The search with a k-d tree requires less than 10 percent of the time that is needed for a complete search.

Other possibilities for matching strategies can be found if not using the closest point as the corresponding point. The matching of suitable points can also be improved by other methods such as projection methods. An example of a projection

method is so-called "normal shooting"[137]. Therefore, Chen and Medioni find the intersection of orthogonal lines from each point in the set of the scene with the destination surface.

With depth pictures, it is probably better to project the starting point along the principal axis of the camera. Blais and Levine [165] describe this as "reverse calibration".

Instead of using the intersection point at the projection procedures for itself, the closest point can be searched close to the intersection point. For example, it can be used only by that closest point, where its normal differs only around a certain angle [20]. In addition to all these methods, compatibility properties can be considered as a type of quality. A typical quality is the color of the point [37]. The methods of "normal shooting compatible" and "reversed calibration" converge very well in the tests of Rusinkiewicz and Levoy [13]. The profound effect compared to all methods can be found by the acceleration of the k-d tree search [166], but includes the generation of the kd-tree before the ICP algorithms starts.

4.3.3 Weighting

Weighting of the correspondences is important in reducing the influence of worse correspondences and to avoid systematic errors at data acquisition under certain circumstances. The weighting primarily increases the improvement on the robustness. In addition, the convergence is accelerated because outliers at the point pairs can influence the quality of the solution. The base algorithm gives every pair a constant weight (factor) of 1. In their approach, Turk and Levoy [138] give an isotropic weighting to every point pair, which calculates itself at the orientation of the assigned points. If the normals of the points are oriented in the same way as the sensor's principle axis, these points get a higher weighting value. These weightings especially increase the robustness of ICP with defective input data. Dorai et al. [167] weight the point pairs in accordance to their reliability in the adjustment. The value of the weights can also be connected to the distance values. This method is very similar to the exclusion of point pairs with too large distances. No major advantages have been recognized by Rusinkiewicz and Levoy [13] while analyzing the weighting of points in their tests. In comparison with the original "constant weight" (weight =1) by Besl and McKay [6], no weighting is significantly better. Further, increasing the noise in the data does not change the results. Therefore, no explicit weighting was included in the proposed Progressive Meshes-based ICP introduced later in this thesis.

4.3.4 Rejection

Rejecting point correspondence is the best method to eliminate outliers at the correspondences. Outliers have a bad influence when determining the optimal transformation. Some approaches [165], [20], [13], [168] pursue a simple strategy by excluding point pairs when the distance between their points is larger than a specified threshold. Specifically, a certain percentage of the point pairs with the greatest distance are usually removed and unwanted outliers are filtered out. If a slower convergence under certain circumstances is not important, this method has minor disadvantages and can be used at all times. A simple threshold for the

rejection of point pairs is given through the addition of multiple standard deviations to the mean value; so, large deviations can be simply removed. Correspondences that are located at the edge of a surface can be additionally excluded [138] if the surfaces are relatively flat and smooth. Dorai et al. [167] exclude correspondences by comparing them with the respective neighbor correspondences; taking their distance and orientation into account. Rusinkiewicz and Levoy [13] compare three methods of rejection of point pairs with the base algorithm. If 10 percent of the point pairs with the largest distances are rejected, an improved convergence behavior arises. A similar behavior can be achieved by rejecting the point pairs — the distances of which exceed a 2.5-times standard deviation. The method of Dorai et al. [167] also influences the convergence behavior in a similar way. The use of rejections has nearly no effect on the convergence speed, but it is a good way to increase the robustness and accuracy of the registration.

4.3.5 Minimizing of point-to-point metrics

Before determining the optimal transformation, the choice of the minimization metric is important. If the well known Euclidean metric is used to minimize the distance between two points, several closed form solution procedures can be used. One of these procedures is the unit quaternion method of Horn [10], as described in Section 4.2.4. Further methods such as SVD, orthonormal matrices or dual quaternions have been examined in [152]. Furthermore, other optimization strategies like non-closed form solutions for the minimization of the distances at the point to point metric could be used [169]. Examples for an indirect solution are search methods (e.g. gradients or simulated annealing, physical system simulation [170], [171]). These open form solution methods are opposite to the closed form solution methods that analytically calculate the transformation directly from the point pairs. Lorusso et al. [152] compare the algorithms in closed form with the point to point metric. In this work, they conclude that the methods bring the same results in precision and speed. As stated in [13], the difference between the solutions is not significant. All of them converge approximately in the same time ($O(N)$ with similar constants) and have similar precisions (the floating point precision) and stabilities against outliers.

4.3.6 Minimizing of point-to-plane metrics

Another possibility is the usage of the point-to-plane metric. Here, the distance of the points is defined by the distance between a point and the plane, as explained in Chen and Medioni [137]. The error metric is defined as:

$$E = \sum_i \left\| \left(m_i - R(p_i) - t \right) \cdot n_i \right\|^2 \qquad 4.20$$

where the normal n_i of the point correspondence m_i is integrated. The determination of the transformation between the two data sets is given by the minimization of the alignment error. In general, the solution of the point-to-plane minimization is non-linear and is usually solved using standard non-linear least squares methods [172]. Seeger et al. [173] have tested three different solution

methods to minimize the distance. Compared to the SVD (singularity value reduction) and the complex Levenberg-Marquardt algorithm, a LU-decomposition is proven to be much faster. The use of point to plane metrics with a non-linear solution [35] gives a significantly improved algorithm. Gelfand et al. [174] analyze the non-linear rotation according to [175]. Assuming a small incremental rotation in one iteration of the ICP, the rotation matrices (for example, for the X axis in equation 4.21) can be linearized:

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{bmatrix} \qquad 4.21$$

Applying this linearization to all rotation matrices, the full rotation can be written as:

$$R_{X,Y,Z} \approx \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} \qquad 4.22$$

With a substitution of this equation with respect to Equation 4.20, the error minimization is written as ([175], [174]):

$$\begin{aligned} E = \sum \big(& (m_i - p_i) \cdot n_i + t \cdot n_i \\ & + \alpha \cdot (m_{i,y} \cdot n_{i,z} - m_{i,z} \cdot n_{i,y}) \\ & + \beta \cdot (m_{i,z} \cdot n_{i,x} - m_{i,x} \cdot n_{i,z}) \\ & + \gamma \cdot (m_{i,x} \cdot n_{i,y} - m_{i,y} \cdot n_{i,x}) \big)^2 \end{aligned} \qquad 4.23$$

The minimization by setting the partial derivatives:

$$\frac{\delta E}{\delta \alpha} = \frac{\delta E}{\delta \beta} = \frac{\delta E}{\delta \gamma} = \frac{\delta E}{\delta t_x} = \frac{\delta E}{\delta t_y} = \frac{\delta E}{\delta t_z} = 0 \qquad 4.24$$

to zero, leads to a "covariance"-matrix in the form of:

$$\sum_i \begin{pmatrix} c_{i,x}c_{i,x} & c_{i,x}c_{i,y} & c_{i,x}c_{i,z} & c_{i,x}n_{i,x} & c_{i,x}n_{i,y} & c_{i,x}n_{i,z} \\ c_{i,y}c_{i,x} & c_{i,y}c_{i,y} & c_{i,y}c_{i,z} & c_{i,y}n_{i,x} & c_{i,y}n_{i,y} & c_{i,y}n_{i,z} \\ c_{i,z}c_{i,x} & c_{i,z}c_{i,y} & c_{i,z}c_{i,z} & c_{i,z}n_{i,x} & c_{i,z}n_{i,y} & c_{i,z}n_{i,z} \\ n_{i,x}c_{i,x} & n_{i,x}c_{i,y} & n_{i,x}c_{i,z} & n_{i,x}n_{i,x} & n_{i,x}n_{i,y} & n_{i,x}n_{i,z} \\ n_{i,y}c_{i,x} & n_{i,y}c_{i,y} & n_{i,y}c_{i,z} & n_{i,y}n_{i,x} & n_{i,y}n_{i,y} & n_{i,y}n_{i,z} \\ n_{i,z}c_{i,x} & n_{i,z}c_{i,y} & n_{i,z}c_{i,z} & n_{i,z}n_{i,x} & n_{i,z}n_{i,y} & n_{i,z}n_{i,z} \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{pmatrix} = -\sum_i \begin{pmatrix} c_{i,x}(m_i - p_i) \cdot n_i \\ c_{i,y}(m_i - p_i) \cdot n_i \\ c_{i,z}(m_i - p_i) \cdot n_i \\ n_{i,x}(m_i - p_i) \cdot n_i \\ n_{i,y}(m_i - p_i) \cdot n \\ n_{i,z}(m_i - p_i) \cdot n \end{pmatrix} \qquad 4.25$$

The decomposition of the linearized equations can be made with standard

decomposition algorithms for the form of $A*t=b$. Because matrix A is symmetrical, the *Cholesky decomposition* is suggested by [175]. In [174], this approach is extended for stability analysis of the ICP algorithm. According to [13], [176], it turns out that the minimization method is a very efficient point-to-plane metric. The comparison of the convergence behaviour of the different metrics is proposed in [13]. It is shown, however, that the point-to-plane metric of Chen and Medioni [137] converges faster. Using point-to-plane metrics is not the only advantage of improving the speed of the ICP algorithm. In addition, the vulnerability of the convergence to local minima is reduced, as shown by Pottmann et al. [177]. In Section 5.2, the results of the point-to-plane metric compared to the point-to-point metric are shown in detail.

4.3.7 Extrapolation

Independently to the metric, there are several variants that carry out the minimization as robustly and as fast as possible. The base algorithm has already been combined with an extrapolation method by Besl and McKay [6]. The result of the transformation vectors q_0, q_1, ...q_n forms a path in the seven-dimensional space. This describes the translation of the initialized position up to the definite optimal orientation. To reduce the large number of iterations, where the status vector Q changes almost linearly, an extrapolation procedure based on the last transformation vectors q_{k-1}, q_{k-2} can be used. This accelerates the calculation of the next vector q_k. Therefore, the difference vectors:

$$\Delta\vec{q}_k = \vec{q}_k - \vec{q}_{k-1} \qquad 4.26$$

are calculated by representing the angel *k* in the seven-dimensional room

$$\Delta\vec{q}_k = \vec{q}_k - \vec{q}_{k-1} \qquad 4.27$$

When the last q_k are located on an approximate straight line, this linear (or quadratic) function is used to calculate the zero point (or the extreme value).

It is checked at every iteration if this applies to a sufficiently small angel $\theta_k < \delta\theta$ and $\theta_{k-1} < \delta\theta$. This condition is true for the last status vector q_k and if d_{k-1}, d_{k-1}, d_{k-2} are the accompanying standard square distance errors of the last three status vectors. The angles along the status vectors are defined as:

$$v_k = 0, \quad v_{k-1} = -\left\|\Delta\vec{q}_k\right\|, \quad v_{k-2} = -\left\|\Delta\vec{q}_{k-1}\right\| \qquad 4.28$$

With the help of equation 4.28 a linear approximation and a square interpolation can be calculated. The estimated value for the smallest error arises from either the zero crossing of the line d_1 or the minimum of the curve d_2. The smaller value is finally used for the calculation of the next status vector

$$\vec{q}_{k+1} = \vec{q}_k + v\frac{\Delta\vec{q}_k}{\|\Delta\vec{q}_k\|} \qquad 4.29$$

An upper threshold is usually set at this extrapolation to prevent over-fitting. On average, this method saves a third or up to a half of the iterations. In contrast to the standard method, the extrapolation method minimizes the square error faster, as already suggested by Besl and McKay [6]. The combination of a suitable metric with simple extrapolation procedures leads to the reduced number of iterations.

Further minimization strategies exist besides this proposed acceleration. Simon [11]changes the process by electing different initial conditions (initial positions) at the beginning of the algorithm, processing one iteration step and selecting the best one with which to proceed. With this approach, a certain resistance to the local minima is given in the error function if the point-to-point metric is used. The selection of parts of the point clouds changing in every iteration step is used by Masuda[139]. There are many further variants discussed [165], [11]. Trucco et al. [178] describe a robust registration algorithm called RICP. To reduce the effect of outliers in the data, they integrated Least Median of Squares inside the ICP algorithm as a replacement for the Least Squares rotation estimation. A point pair rejection removes outliers from the closest points to increase the robustness.

4.4 Progressive mesh ICP algorithm

As shown in the previous section, one of the major problems of the ICP algorithm is its low performance when calculating a huge among of points in scenes and models, as well as its sensitivity to outliers. Rusinkiewicz and Levoy [13] (2001) reviewed several approaches to reduce the number of points in the closest points search process. In every iteration step, all points of the two data sets (meshes) must be compared to each other with a complexity of $O(N_j \times N_i)$where N_j and N_i are the numbers points. With k iterations in one refinement step, the complexity will be $O(k \times N_j \times N_i)$[179]. The proposed Progressive Mesh ICP algorithm uses a hierarchical system to match the point sets. This idea reduces the complexity by comparing only low resolution representations of each mesh and not the whole mesh points to each other. The obvious advantage is the increased performance, but the profound effect is the increased robustness against outliers. The next section describes the basics for Progressive Meshes followed by the combination with the ICP algorithm.

4.4.1 Progressive meshes

Hierarchical levels of detail (LOD) systems like Progressive Meshes are related to hierarchical pyramids in image processing. An early approach was introduced by Tanimoto and Pavlidis in 1975 [180] for an image representation with resolution levels. The resolution level of a pyramid in image processing is generated by the average determination of neighboring points, so quadratic non-overlapping

regions are created with one intensity value. In every pyramid level, small image structures disappear with decreasing resolution and the information content decreases with increasing reduction. Because of the unsafe localization of feature points in higher levels, every pair of points is projected on the next pyramid level. Hierarchical techniques are very quick and, nowadays, are used for many image processing tasks. However, corresponding points can be found more efficiently. The idea of a hierarchical system is transferred to 3D data structures as well. One of most efficient algorithms is the Progressive Mesh representation introduced by Hoppe et al. in 1993 [71]. An important application for Progressive Meshes is given by the progressive compression of 3D meshes. This is desirable for the data transmission of complex meshes over networks with limited bandwidth. First, a coarse mesh is transmitted over the network. After that, refinement data are transmitted to enhance the mesh representation until the mesh has its full resolution. For this reason, a hierarchical progressive data structured is required. The Progressive Meshes technique is closely related to the work on mesh simplification [181] and belongs to the energy function optimization methods [182]. This data representation has the following advantages:

- Highly efficient mesh representation
- Level of Detail representation
- Mesh simplification with noise reduction

The mesh representation consists of faces (triangles) defined by three vertices (points). An edge is the line between two vertices of two adjacent faces. The representation of Progressive Meshes is given by a set of meshes M_0 to M_n. M_0 is the mesh with the lowest resolution and M_n is the mesh with the highest resolution.

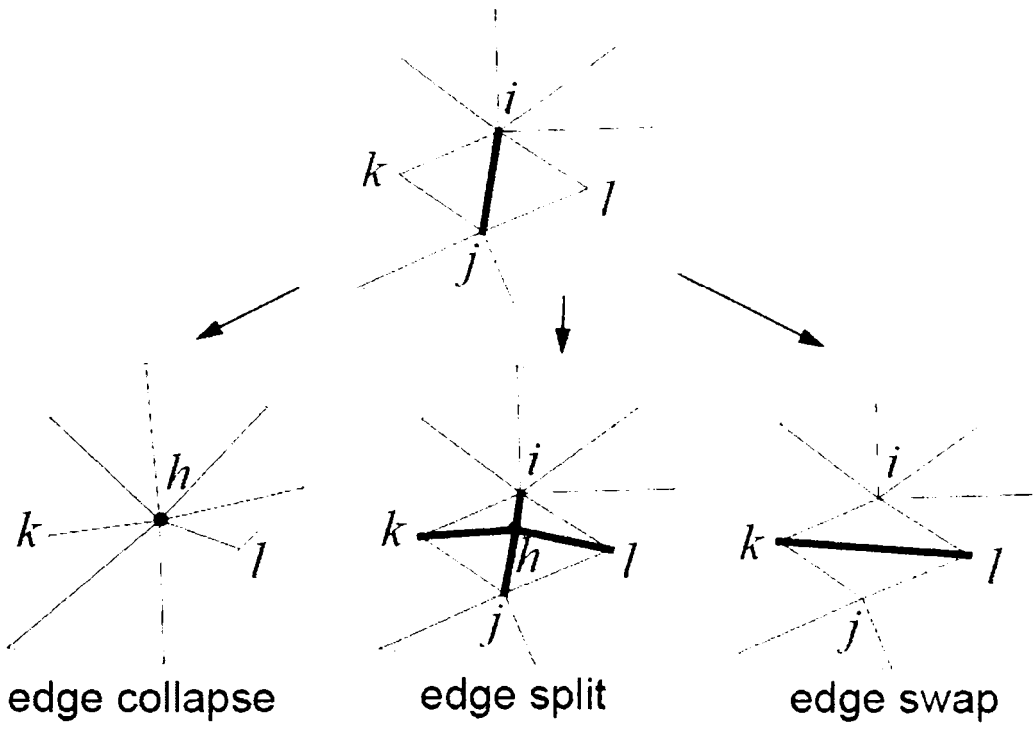

Figure 39 Progressive Mesh operations[71]
Edge collapse: The number of edges is reduced with this operation. Vertex split: The opposite operation of edge collapse adds a vertex to the mesh to increase the level of detail. Edge swap: The edge between vertices is swapped according to the error function.

Generating a Progressive Mesh means applying *edge collapse* transformations to the mesh M_n. The Progressive Mesh generation algorithm applies a series of *edge collapses* to a mesh, simplifying the model by reducing the number of vertices and faces. The edge collapse process is shown in figure 38, where the edge *{i,j}* is reduced to one vertex. The most simplified model obtained after this process is called the base model M_0. The opposite of the edge collapse transformation is the edge split transformation. The base model can be reconstructed to an original representation by the reversed operation of edge collapse through *vertex splits*. In this case, one triangle is split into two triangles with the vertices *ijkl*. Each *vertex split* operation replaces one vertex with two edge-connected vertices and creates one additional vertex and two additional triangles. By incrementally adding new vertices and faces to the base model, the original mesh is reconstructed. *Edge swapping* is made if an edge collapse will lead to a higher inaccuracy of the mesh representation. To construct a Progressive Mesh, it is important to find a proper edge to be collapsed at each step. The decision can be made with the calculation of an energy function which takes number, distance, accuracy attributes of vertices, discontinuity curves and a regularization term into consideration [71], [183], [184]. According to the calculated energy costs, each edge is put into a priority queue. After the algorithm has calculated the priority

value of each edge, the edge with the smallest priority value is selected to collapse. With every iteration one edge is collapsed, the nearby edges are reevaluated and the priorities are reordered in the priority queue. This process repeats until topological constraints prevent further simplification. Thus, the original mesh with full resolution is simplified by edge collapse operations in batches — creating a series of intermediate meshes M_i of decreasing resolution. The simplification stops at a simple model M_0. The complete process can be described in the following equation:

$$M_0 \underset{ecoll_0}{\overset{esplit_0}{\rightleftarrows}} M_1 \underset{ecoll_1}{\overset{esplit_1}{\rightleftarrows}} \cdots \underset{ecoll_{n-1}}{\overset{esplit_{n-1}}{\rightleftarrows}} M_n \qquad 4.30$$

Every single step M_i is restored to recover the mesh of this step. So, every mesh M_i with the required resolution can be retrieved in a very efficient way. Therefore, a Progressive Mesh is defined as the representation of the base model and the series of *vertex splits*. Many arbitrary multi-resolution models provide a fixed number of levels of details organized level by level. The main advantage of Progressive Meshes is the fact that they allow the refinement steps to be done, *vertex split* by *vertex split* [185]. Some reviews of further developments of Progressive Meshes like [186], [185] can be found in [184]. Thus, the Progressive Mesh representation provides a powerful framework for polygonal simplification and object representation in different levels of details.

4.4.2 Combination of Progressive Mesh and the ICP algorithm

ICP is a time-consuming algorithm that depends on the number of points in the scene and model. The major problem of the ICP algorithm is its low performance when calculating a huge number of points in scenes and models. The proposed algorithm is a hierarchical system including Progressive Meshes — introduced in the previous section in order to speed up the convergence. This combination reduces the complexity of the neighbor search by comparing only the M_i representations of each mesh and not the whole mesh points to each other. The obvious advantage of multi-resolution ICP is the increased performance, as already shown in [14], [160] and the robustness against outliers. By reducing the mesh up to M_0, outliers can no longer affect the result of the distance calculation.

Multi-resolution mesh registration has been introduced in only a few papers so far. Different mesh representations do exist and each has its own advantages and disadvantages. An approach for a multi-resolution mesh registration is introduced by Jost in [14]. Jost increases the number of points by a fixed factor in one LOD step depending on the number of points in the data sets. The author states an improvement in multi-resolution registration by factor 8, increasing the number of points by sub-sampling when an error criterion [155] is met. Due to this, the number of ICP iterations varies in each resolution step.

One year later Zinsser et al. [179] used a uniform subsampling technique in a hierarchical context. They use only every 2^h-th points in the data sets, where h is the increase after the data points are aligned with the ICP algorithm. This is combined with robust outlier thresholding and an extrapolation of motion

parameters in their Picky ICP.

Ikemoto et. al. [187] align warped range data in their coarse-to-fine hierarchical approach. The data sets are separated into small pieces to compensate for the global warp, which comes from sensor errors. Each piece is rigidly aligned with respect to other pieces in the data set, leading to a better convergence. To gain the advantage of better convergence and accuracy, the total performance is reduced. Their registration tests result in a pairwise match with 1200 points, with a total convergence of several minutes on a 2.8Ghz Pentium4.

The Progressive Mesh representation in combination with the ICP algorithm is introduced by [188]. They register each LOD representation of the mesh with a full ICP registration step in the following way. They create a list of mesh pairs (model and scene) with different LODs. Each pair is registered with the help of the ICP algorithm, starting with the lowest LOD, and the resulting transformation of each LOD registration is used as the initial transformation for the next LOD registration. In their experiments, they use two LODs for a data set. They report a slightly better performance than the original ICP algorithm (about 5% in convergence time, without considering the time to create the mesh representation). Unfortunately, they do not give any information about the error stop criterion they used.

Low [189] smooth their data sets into multiple resolutions by two-dimensional cubic uniform B-spline wavelet transform. They use a pseudo-point-to-plane minimization metric to implement a closed form solution for transformation determination. To change the level of detail, the average distance between the points in the current resolution is compared to the average distance between the matched points. When the maximum distance between the correspondence pairs is less than 25% of the size of the support of the cubic B-spline basis function used to produce the current smoothed surface, the next level of detail is used. In opposite to the described approaches so far, the approach of [189] changes the LOD inside the registration steps. Most approaches determine one LOD and try to find the best transformation for this exact model and scene representations with the help of the ICP algorithm. After the best transformation is found with the ICP algorithm, the LOD is increased and the next ICP process is started in order to find the best transformation for the next LOD.

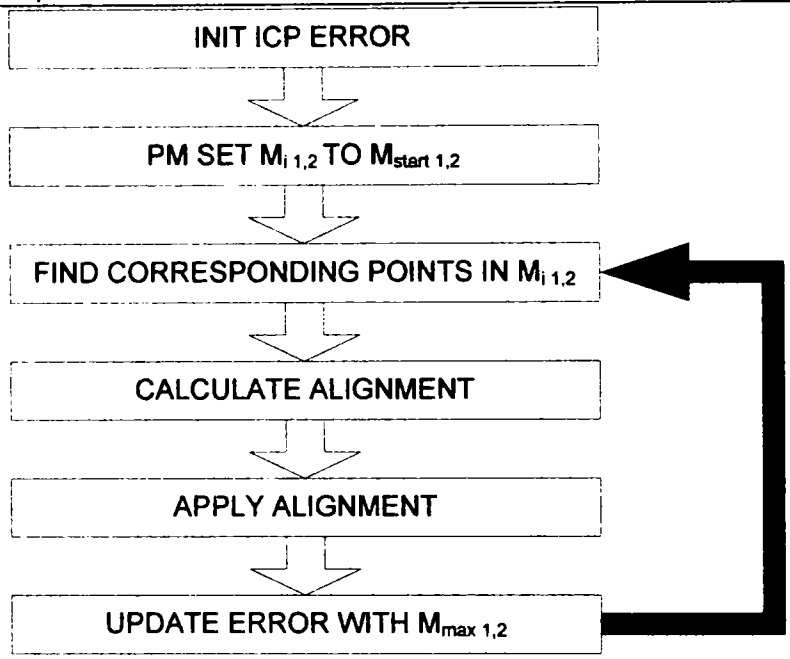

Figure 40 PMICP iteration steps
The progressive mesh LOD adjustment is integrated inside the ICP iteration.

The Figure 40 shows the integration of the Progressive Mesh data representation in the iteration steps of the ICP algorithm. Before the iteration starts, the model (M_{i1}) and the scene (M_{i2}) data sets are transformed to a coarse representation with a low level of detail (M_{start1} and M_{start2}). The number of vertices in M_{start1} and M_{start2} is not the minimum M_0 of the mesh. According to the experiments, it transpires that the optimal number of faces is given by

$$5 \le F_{start} < 0.05 * F(M_n) \qquad 4.31$$

Thus, the maximum faces at the start of the iteration are five percent of the maximum faces of the mesh. With only one face in each data representation, misalignments could occur because the triangle could flip over (face normals are opposite). Due to this, the minimum of 5 faces is used in the proposed algorithm based on the experimental results. The next step is the search of corresponding points in the current LOD data representation M_i. This step incorporates the selection, matching, weighting and rejection strategies described in previous sections.

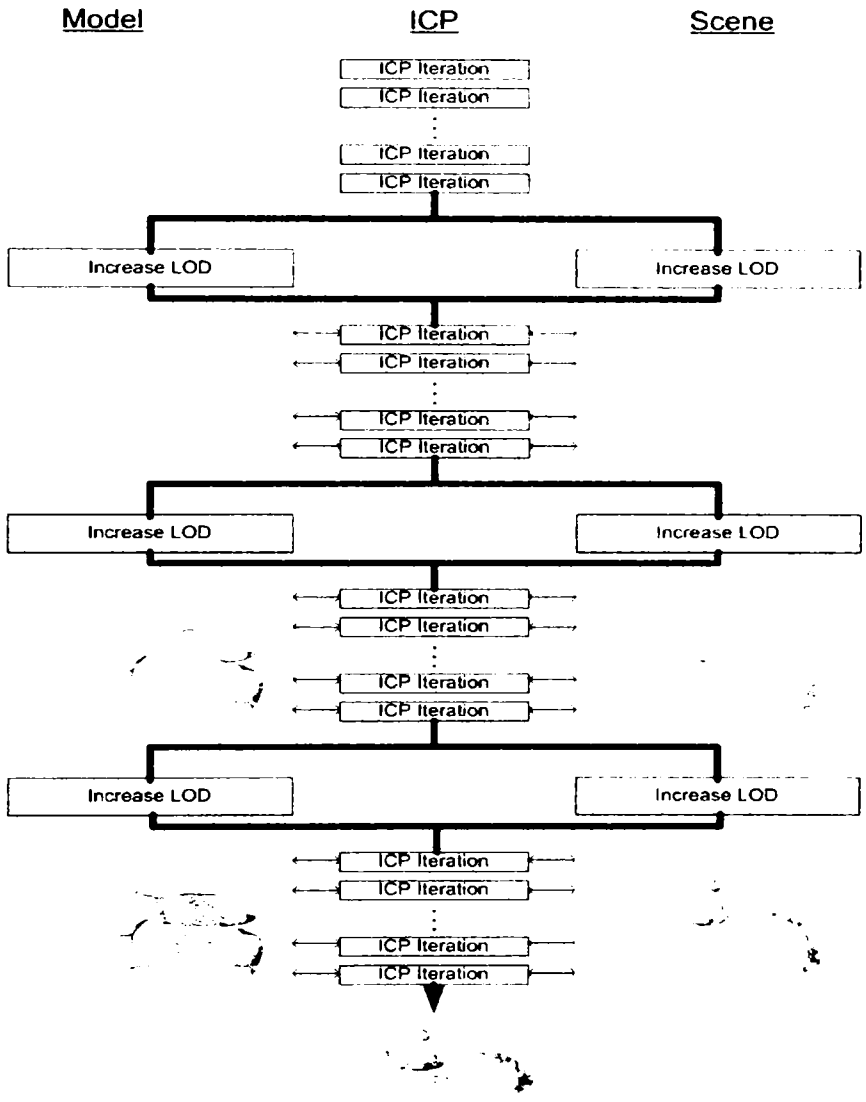

Figure 41 LOD integration
This is an overview of the LOD integration. A reduced data model is aligned and the LOD is increased after several iteration steps.

All points of the current LOD data representation (which is a subset of all points) are used to find the corresponding point pair. Figure 41 shows the integration of the LOD data representation in the ICP algorithm. In the first ICP iterations, the coarsest LOD data representation of the scene is matched against the model. The LOD is increased with an increasing number of iterations until the scene and model data reaches the maximum number of their vertices. Each iteration compares the two meshes and determines their closest points. The point-to-point metric allows for the accelerated closest point search in a kd-tree implementation.

Because the Progressive Mesh operations vertex split and edge collapse, the surface differs in each LOD representation and the kd-tree in the same way. As a consequence of this, the basis implementation to the non-accelerated closest point search with complexity of $O(N_j \times N_i)$ is implemented and tested and compared to the point-to-plane metric of Chen and Medioni [137]. This leads to a closed form solution for the eigenvalue calculation in the transformation calculation step of the ICP algorithm (see Section 4.3.5). The points of the M_n representation of the scene data set are transformed with the resulting transformation matrix to the new position. The distance error of the alignment is calculated over all points in the data sets (M_n), so the iteration stop criterion of Section 4.2.4 cancels the iteration. The convergence rates of the whole data sets are shown in the tests in the following Chapter 5.

Chapter 5 System evaluation and implementation

To quantify the accuracy and performance of the whole system, it is necessary to analyze the object localization steps in detail. Therefore, the next sections will analyze the performance of the coarse pose estimation and, later, the performance of the pose refinement. All tests are run on the reference PC with 1,73Ghz Pentium M processor, 1GB of RAM and an ATI Radeon graphic chipset (Microsoft DirectX© 9.0c Support).

5.1 Evaluation of pose estimation

5.1.1 Pose estimation test

The test is made with the data set for an industrial doorjoint depalletizing application. One doorjoint is laying on the conveyor belt with an unknown position and orientation, as shown in Figure 18. The object model representation consists of 180 vertices and is transformed into a VRI with about 4000-5000 vertices (depending on position and orientation). The scene data has about 150000 vertices in total. In every pose, an amount of about 10000 vertices of the scene data are compared to the VRI.

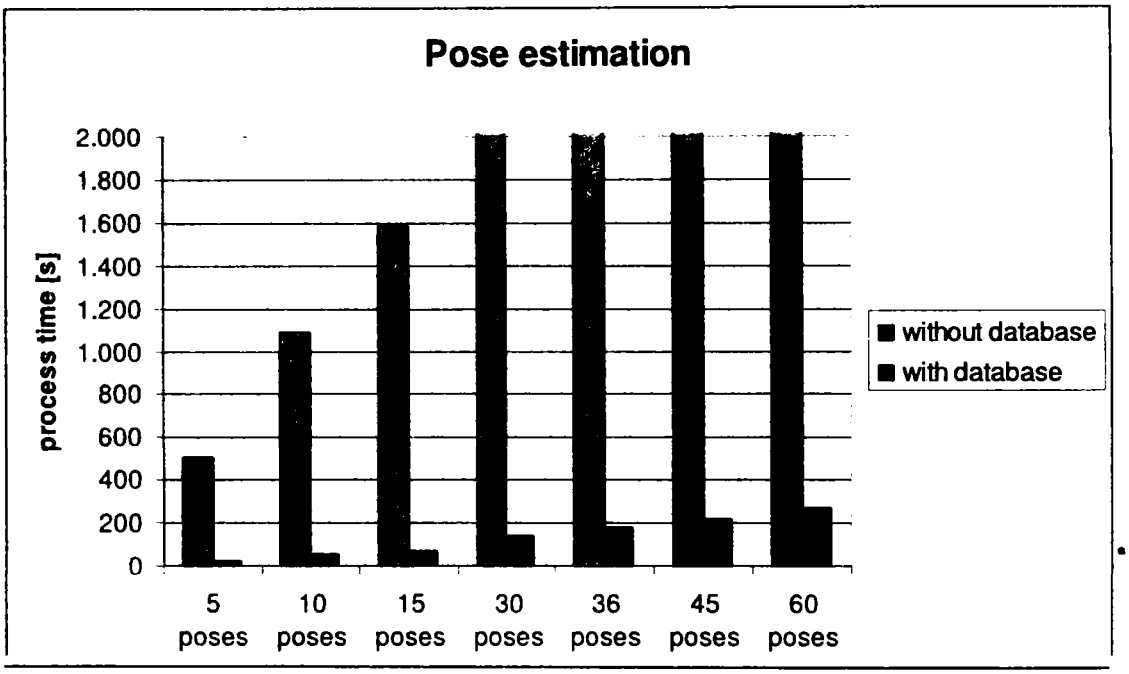

Figure 42 Pose estimation database performance
The database decreases the calculation time up to 5% of the original calculation time.

Figure 42 shows the performance enhancement of the database integration. The process time is reduced to about 5% of the process time of the pose estimation without the usage of the database. The total process time includes the VRI transformation to the respective pose, the database query and the distance calculation. The distance calculation is the most time-consumable part, which needs on average more than 70% of the total processing time. Therefore, the high number of vertices in this test results in a high total process time. If the data sets cannot be found in the database, the sensor simulation must be performed. The sensor simulation has a high potential to improve the calculation process time of the pose estimation. The TOF-based sensor simulation is based on the ray-triangle intersection test, which is tested in the following section.

5.1.2 Pose estimation triangle intersection tests

The pose estimation algorithm is based on fast ray-triangle intersection tests, as stated previously in Section 3.3.1.3. The different intersection methods are tested in detail in the following. Table 1 shows the average results of mesh intersection tests:

Table 1 Intersection performance

	Vertices		
	36	1850	31752
DirectX© Ray-Mesh Intersection (AABB)	0,3s	0,9s	14,7s
Ray-Mesh Intersection with DirectX©	0,4s	1,12s	18,4s
Matlab (C++ Ray-Triangle Intersection Test)	5,3s	221,3s	3750,6s
Matlab (proof of concept)	10,3s	271,6s	-

A non-optimized Matlab proof of concept implementation has a performance of about 5 ray-triangle intersection tests per second. That said, even with an implementation of [105] in a compiled C++ function, the intersection is not significantly faster. The high-optimized intersection test in Microsoft DirectX© outperforms all other intersection tests, especially with huge data sets — the performance increased by a factor of 200 compared to the other implementations. The DirectX©-based implementation handles over 2000 vertices per second with the reference PC, using high-optimized intersection functions. Nevertheless, ray-triangle intersection tests have a lot of potential for optimization in the future. Known high performance software renderers [190] achieve several billion vertices per second. These solutions gain advantages from the development of modern hardware (progress in CPU and GPU design), parallelization and high performance software concepts like *spatial data structures*, *Culling* and more [191].

5.2 Evaluation of pose refinement

The performance of pose refinement is mainly influenced by the implementation details of the iterative closest points algorithm, combined with the Progressive Mesh data representation. Therefore, the performance of this algorithm is shown in detail in the following figure. In order to compare the test results, the standard ICP algorithm is implemented according to [6], without approximation. The PMICP is implemented based on the Progressive Mesh implementation of Hoppe in Microsoft DirectX© [192]. To evaluate the proposed algorithm, the reference data sets of Rusinkiewicz and Levoy [13] are used. These synthetic meshes have about 100,000 vertices in total, added with Gaussian noise and outliers. The "incised" data set has two lines in the shape of an "X" in the middle of a planar surface.

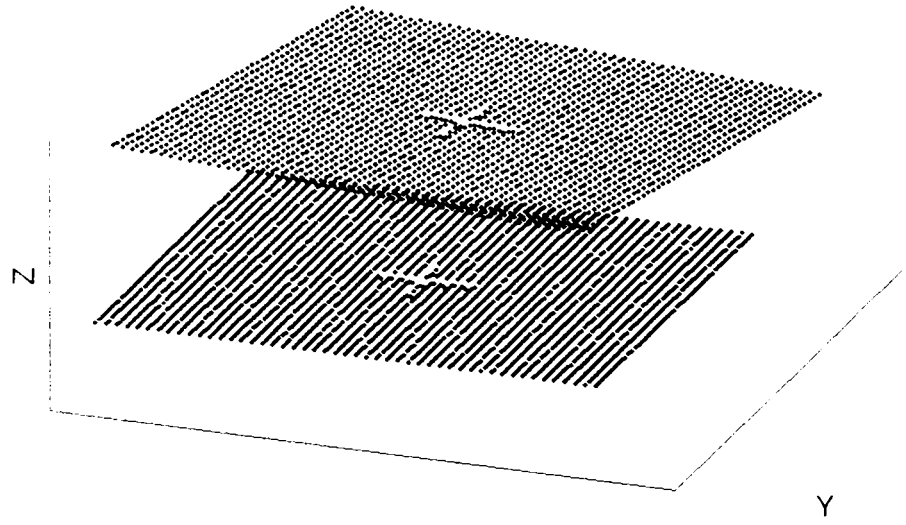

Figure 43 "incised" data sets
The incised dataset is a plane with a small X inside. Therefore, the most important information is given by this structure.

The two data sets' scenario shown in Figure 43 is a difficult task for the ICP algorithm, because there is only one good feature (the "X") in the data set to align them correctly. As with the "incised" data set, Gaussian noise and outliers are added to the "wave" data set (Figure 44). This is known as an easy scenario for low frequency features and a smooth surface [13].

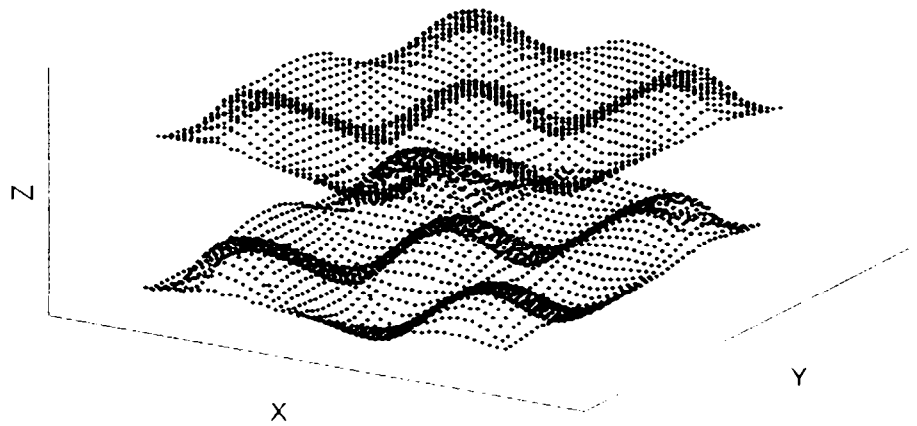

Figure 44 "wave" data sets
The wave is well defined with bumps, which gives the ICP the chance to perform very good in the first iterations.

Opposite to the wave scenario, the "fractal" data set has features in all levels of detail. Therefore, this data set is a good reference for the experiment setup. The "fractal" data set represents landscape data of the terrain registration.

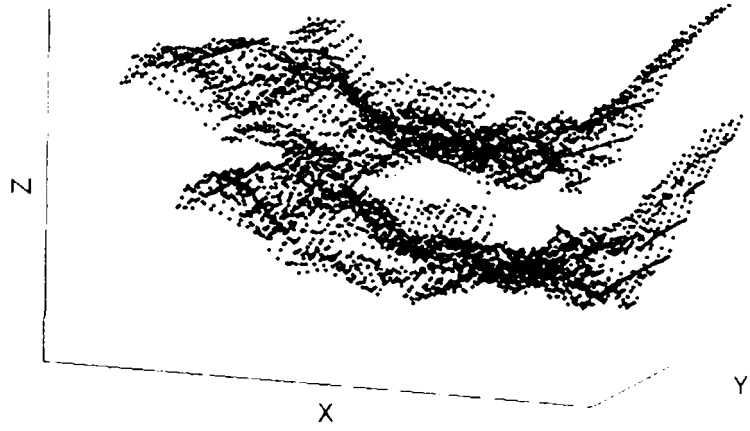

Figure 45 "Fractal" data sets
The fractal data set represents a terrestrial map with a signal of high frequency.

These three test scenarios do not cover all possible classes of object surfaces, but they do provide the opportunity to compare the results with different kinds of ICP modifications such as the ones that Rusinkiewicz and Levoy [13] or Zinßer et al. [163] or Matabosch et al. [63] carried out in their experiments. In [193], the test scenarios already show the potential to be a good reference for comparison.

5.2.1 Convergence tests

The convergence behavior of the alignment algorithms is an important fact that can be used to compare the results of new algorithms to already known solutions. To compare the results, the standard ICP algorithm is implemented according to [6], without approximation. The original ICP algorithm of Besl and McKay [6] is not changed or improved with any other possible strategies such as selecting, matching, weighting and rejecting points or usage of different metrics and minimizations in this certain test scenario in order to avoid side effects and show the pure results of the contributions in this thesis. The original algorithm uses only the position of the vertices without considering normals. To have the exact conditions as the tests in [13], the introduced data sets are changed in the following way. The number of vertices is reduced to 2% (2000 vertices). At first, the experiments use the simplest way to adapt the hierarchy of the Progressive Mesh representation — for each iteration, the current LOD representation M_i is increased by a fixed increment and starts with a defined level of detail.

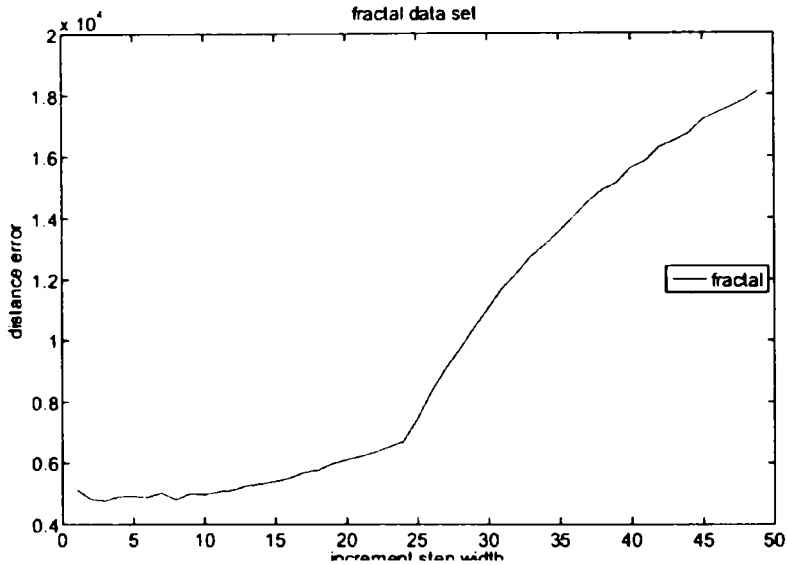

Figure 46 Influence of LOD step width in "fractal" data sets
The number of LOD steps increases the resulting distance error for the fractal datasets starting from a step width of about 25.

It is obvious that the results depend on the used registration data. There are influences regarding the number of points, error in the data sets and their initial pose. The optimal iteration step width in the fractal scenario can be estimated from the graph shown in Figure 46. The distance error of the algorithm is measured until it reaches the maximum number of 100 overall ICP iterations. The graph shows the influence of the step width of the LOD of the Progressive Mesh in every ICP iteration step. This experiment shows two results. First, the quality of alignment depends on the number of points used to calculate the transformation in each iteration step. With each increasing step width, the maximum number of points in the mesh (maximum LOD) is reached and the PMICP converges to the original ICP algorithm. The number of total iterations is increased, so the process cannot converge to the global minimum with an increment size over 25, because of the iteration number limit. However, the optimal LOD step width for this scenario can be estimated. The optimal LOD step width is the minimum of the graph, which is mostly greater than 1. For example, when increasing the number of faces by 1 (increment size 1 in Figure 46) in each iteration step, there will be only a minimum of additional information in the next step, which leads to a slower convergence rate with more iteration steps. As a result of this, a higher total convergence time is needed. With an incremental size over 10, the calculation time of every iteration step is increased, which leads also to a higher total convergence time. Therefore, the optimal LOD step width is about 5 in this case. The LOD is incremented every iteration step of both Progressive Meshes (scene and model) with this value in order to achieve the best total convergence time in this experiment. The initial number of vertices depends mainly on the number of points in the data set; it is set to five triangles at the beginning. The sum of squared distances of the closest points over all input vertices is calculated in every iteration step to compare the results to the original ICP. The distance error calculation is changed from the Euclidian distance to:

$$E = \sum (\vec{v}_1 - \vec{v}_2)^2 \qquad 5.1$$

to avoid computational floating point errors and increase calculation performance[11]. The convergence theorem of the standard ICP [6] also applies to the PMICP given by the "internal" distance error of the current point data set M_i. If the correct registration transformation is known, a ground-truth-based distance measurement can be calculated. Usually, when validating a registration approach with unknown data sets, no ground truth distance error is available. The squared distance error cannot be zero because Gaussian noise is often added to the model and scene data set independently. Due to this fact, the iteration process is stopped if the maximum of iteration steps is reached.

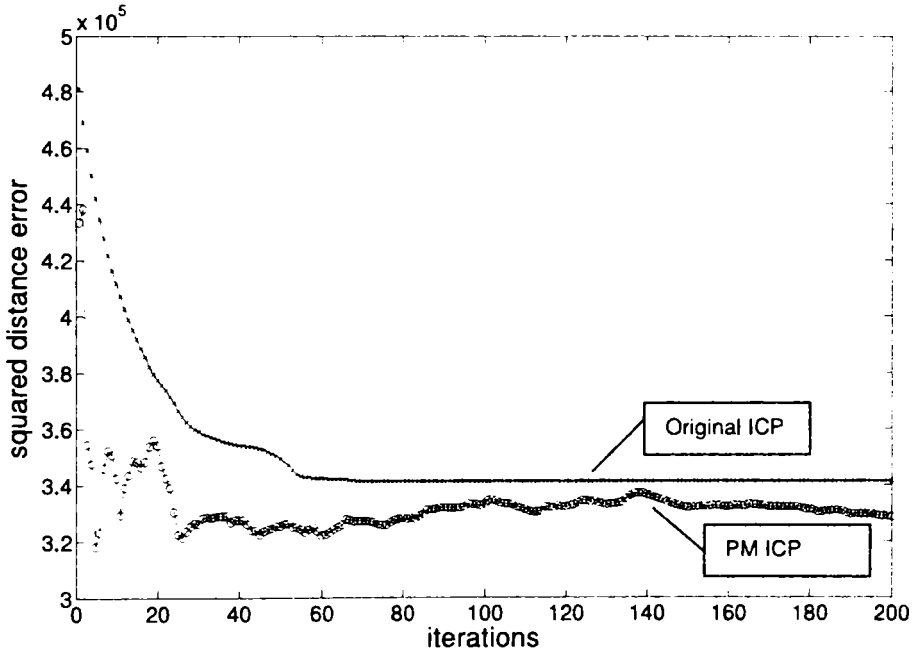

Figure 47 Convergence of "fractal" data set refinement
The fractal data convergence rate of the progressive mesh based ICP algorithm outperforms the original ICP in each iteration step.

The "fractal" convergence results of the test scenario are shown in Figure 47. Looking at the squared distance error, the PMICP outperforms the standard ICP in every iteration step, and the standard ICP is locked in a local minimum. Note that the PMICP found the absolute minimum in the very first few iterations (as stated above, the iteration is not stopped when the algorithm finds the correct pose in this experiment). By increasing the iteration steps and the number of points in the data sets, the PMICP implementation degenerates progressively further to the standard with a better robustness. The initial M_i does not suffer from outliers like the standard

ICP does.

The convergence performance of the PMICP in the wave scenario is similar to the previous scenario, especially in the first few iteration steps where the PMICP aligns the data sets to a good initial pose. The squared distance error over all the points in the data set is always smaller in every iteration step compared to the standard ICP.

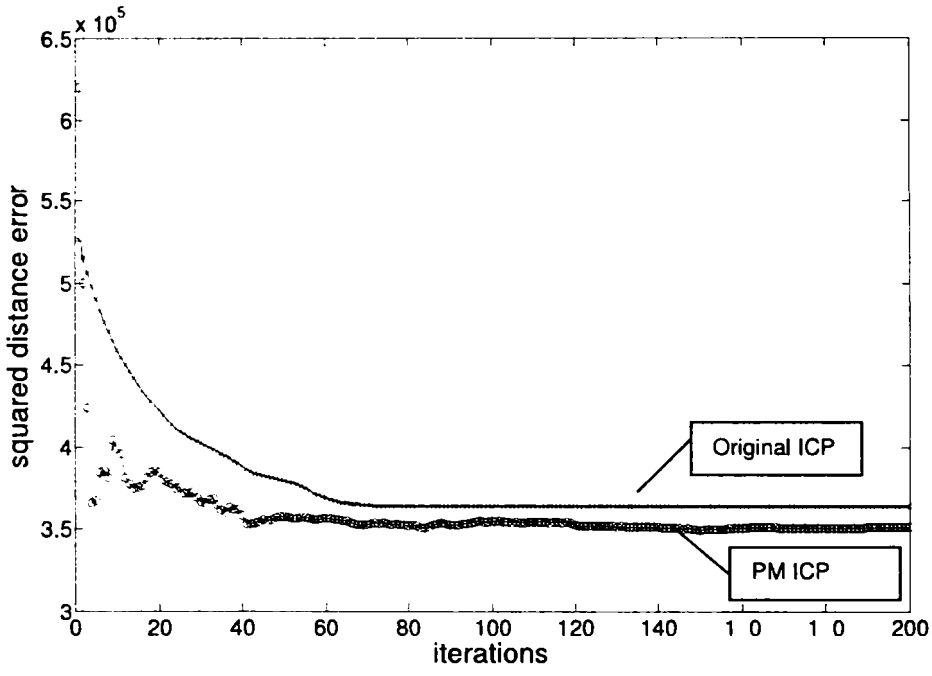

Figure 48 Convergence of "wave" data set refinement
The wave data convergence rate of the progressive mesh based ICP algorithm is always better the original ICP.

The "incised" scenario has an interesting behavior in the convergence graph (Figure 48) of the PMICP implementation.

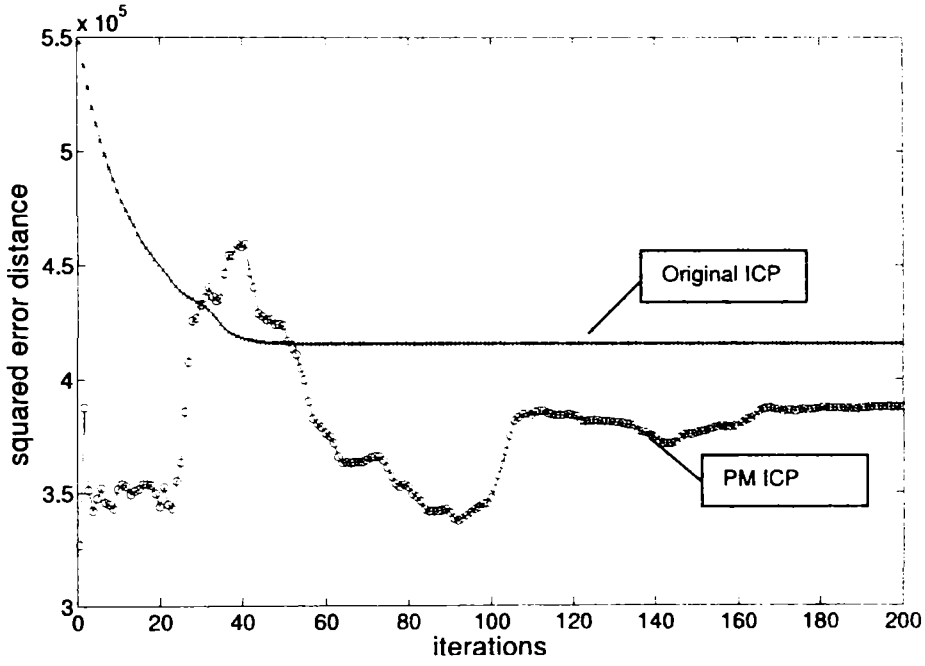

Figure 49 Convergence of "incised" data set refinement
The rate of original ICP algorithm is only better in a specific range of iterations.

Figure 49 shows an increasing error between the iteration steps 20 and 40. In these iteration steps, the planar section is split into triangles. In general a plane can be divided into a triangle in many different ways.

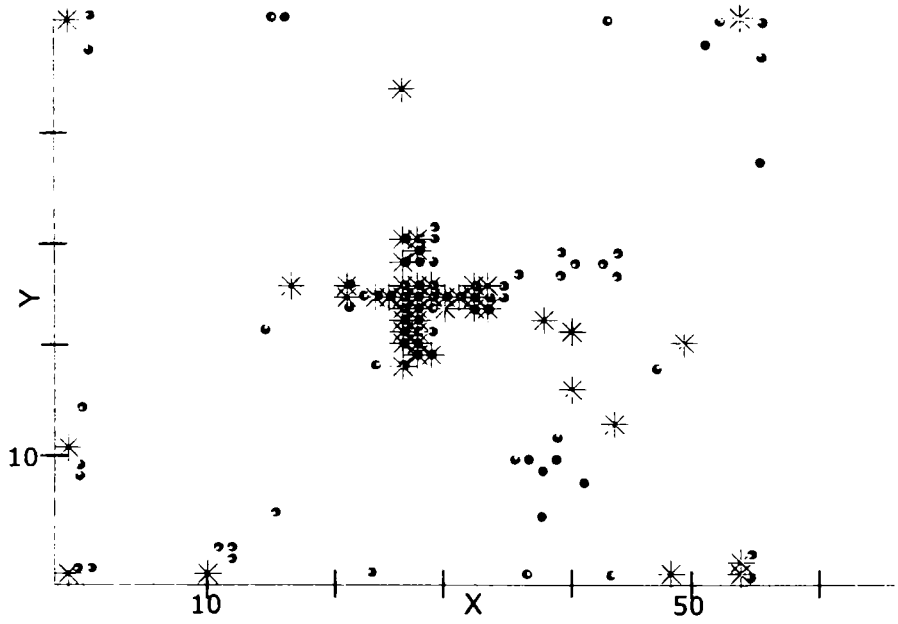

Figure 50 "Incised" M_{30} Mesh representation
The full mesh is reduced to a mesh with many points around the "X" and only a few mesh points in the plane. The points inside the plane are leading to the increased alignment error.

Taking the different Gaussian noise of the planar shapes into consideration, it turns out that points are created at quasi-random positions lying in the planar section. This results in the wrong rotation of the aligned data set and increases the ground truth error, as shown in Figure 50. With the increasing number of points, the alignment is corrected and the iteration process performance enhanced. The M_{30} Progressive Mesh representation shows a very good distribution of the "feature" points in this data set. All of these experiments concentrate on convergence robustness and final distance error issues. The influence of Gaussian noise in the data is analyzed in the following section. After that, the next experiments show that the overall performance of the refinement process can be increased significantly with the proposed solution.

5.2.2 Influence of noise

To quantify the influence of noise on the proposed algorithm, Gaussian noise is added independently to the model data set and the scene data set with increasing amplitudes.

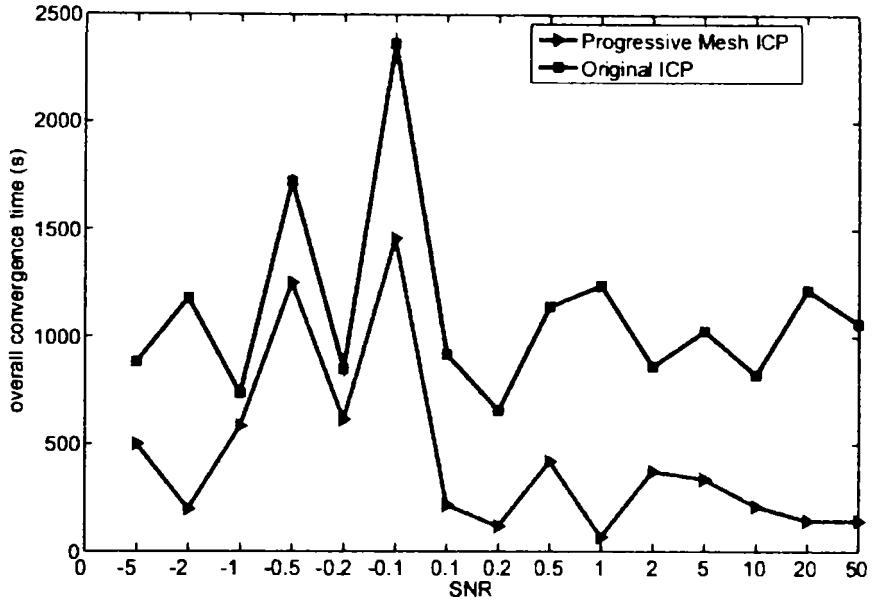

Figure 51 Noise influence to registration
Due to the error reduced representation the progressive mesh based approach converge always better the original approach.

As shown in Figure 51, the measured iteration time is compared to the level of noise. Over all the experiments [193], the PMICP has a 2 to 3 times lower iteration time than the original ICP algorithm. The convergence time of ICP mainly depends on the number of iterations and nearest neighbor calculation time, as already shown in [14] and in the experiment data in table 2 where the number of iterations of ICP is reduced in total.

Table 2 Noise influence

SNR	Original ICP	Inc. PMICP	PMICP
5	1027.7 ms	2623.1 ms	335.8 ms
2	861.8 ms	3099.7 ms	371.7 ms
1	1238.8 ms	4933.1 ms	65.2 ms
0.5	1140.5 ms	3035.3 ms	419.0 ms
0.2	654.7 ms	1149.4 ms	115.1 ms
0.1	916.2 ms	2497.4 ms	214.6 ms

This can be achieved in addition to the higher computational performance because of the reduced number of points in the nearest neighbor calculation step. The results of the experiments in Table 2 show that the incremental Progressive Mesh ICP of [188] in column 3 is slower than the original algorithm. This is caused by the overhead of changing the resolution of the mesh back to the lowest LOD step in every iteration step. In the test, the error difference iteration stop criterion and the maximum iteration stop criterion are used for their algorithm. The implementation does not include any further improvements to the incremental Progressive Mesh approach, as introduced in [188]. The complexity of the ICP algorithm depends mainly on the number of points in the data set. The search of the closest points has a computational complexity of $O(N_j \times N_i)$. The number of points in the data set is reduced, starting with only a few points and increasing the number in every iteration step. The computational complexity is reduced to the average complexity of $O((0.5\times N_j)\times(0.5\times N_i))$ if the iteration is not stopped until the end (M_n mesh) has been reached. If the iteration process is stopped, because ICP reached the minimum, the performance of the PMICP implementation is always better than $O((0.5\times N_j)\times(0.5\times N_i))$. The PMICP (using a linear M_i delta) needs on average 25% of the time of the standard ICP implementation. Adding the overhead of the LOD adjustment into every iteration step, the proposed solution of this thesis (column 4) is always more than 3 times faster than the original ICP, which is already shown in the convergence time experiments (table 1).

5.2.3 Evaluation of alignment methods with Progressive Meshes

To find other test scenarios for a better evaluation, the next test scenario is taken from the Stanford 3D scan repository. The well known and most commonly used data set of the "Stanford Bunny" consists of a reconstructed collection of about 69,451 triangles in total. The model was assembled from the range images of 10 different views.

Figure 52 "Stanford bunny" registration test scenario
This test scenario consists of two scans from the dataset of range scans of a bunny figure.

Each of the scan data sets has over 30000 points. The range images of the statue of the "Stanford Bunny" are acquired by a laser range sensor. In Figure 52, the two test data sets are shown. On the left, the bunny in scanned from the right front; on the right the bunny is scanned more from the left front side; because of this, there are more range points on the bunny's right side than on the left. This scenario requires a correct global alignment [8] with overlapping regions. Due to the fact that this thesis does not concentrate on this topic of research in detail, the data sets are only used for alignment to the global minimum without pre-calculation of overlapping regions or multi-scan global alignment. The threshold to find the global minimum is given by the correct transformation between the scans (here the files *bun270.ply* and *bun315.ply*) from the implementation of the application *trimesh2* [194]. These tests concentrate on the convergence and performance of the alignment.

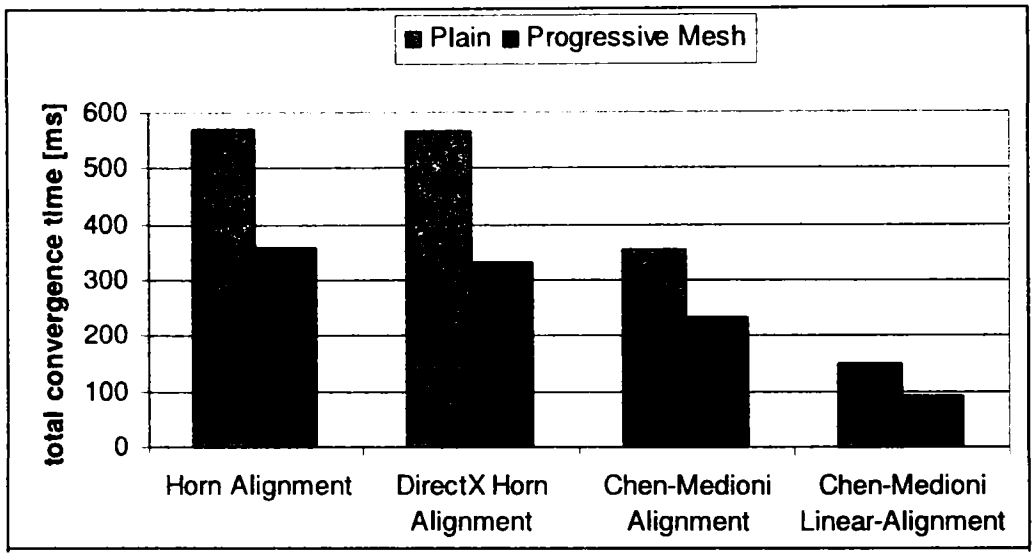

Figure 53 Alignment convergence time of the "Stanford Bunny"
The convergence time is reduced in the Chen-Medioni based alignment because of the excessive search of nearest neighbor of Horn-Alignment.

Figure 53 shows the alignment convergence times of the implemented algorithms. All alignment methods according to Section 4.3 are implemented in C++, with and without Progressive Mesh support. The first thing to mention is the dramatic improvement of the Progressive Mesh data representation with a linear increased level of detail. This is compared to the full corresponding point search implementation. The total convergence time includes the search of the corresponding points in the data sets, the calculation of the transformation and the transformation of the data set. These are the most time-consumable steps in the refinement process. It is shown that the Progressive Mesh Iterative Closest Points combination is a good method for all kinds of transformation calculation methods. The Horn-based minimization on the left of Figure 54 has the most stable convergence rate but is, on the other hand, the slowest because of the exhaustive search to the closest points. The closest point search of the Progressive Mesh algorithm is not accelerated by a kd-tree implementation, because it turns out that the creation of a kd-tree for each LOD of the Progressive Mesh is not generally applicable. According to the minimization of Section 4.3.6, the implementation of the Chen Medioni algorithm [137], converges faster to the minimum [20]. The DirectX©-based Horn minimization is faster because the data must not convert from the DirectX© data representation to the required data representation of the functions provided by Pulli [20]. The linearized implementation of [174] provides a similar performance, with or without Progressive Meshes, because the performance is more influenced by the number of iterations.

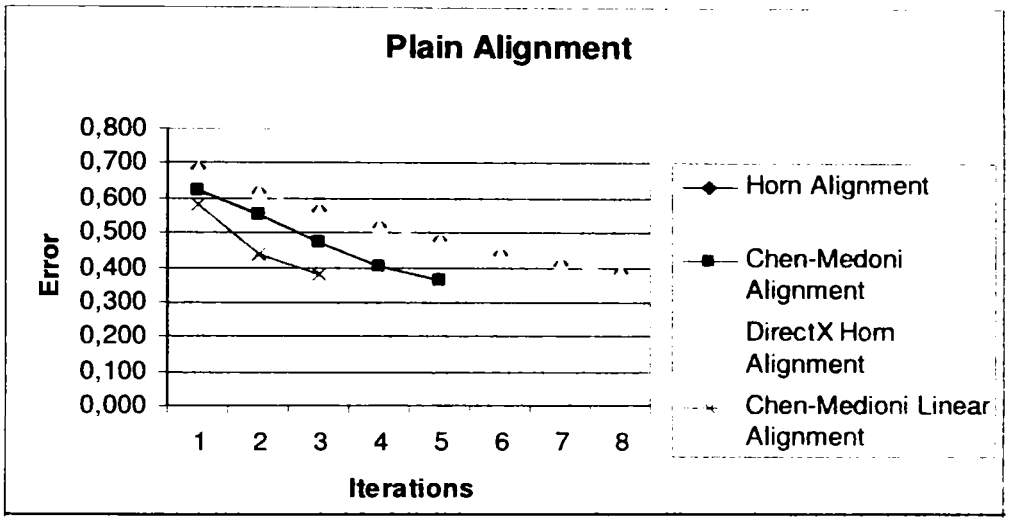

Figure 54 Convergence behavior without Progressive Mesh
The Chen-Medioni Linear Alignment is the fastest because it converges in lesser iterations.

The convergence rates of the alignment methods are shown in the diagram of Figure 54. As supposed before the Chen-Medioni Linear Alignment converges to the minimum error threshold of 0.4 within 3 iterations in this application scenario. The other algorithms need up to 8 iterations to converge.

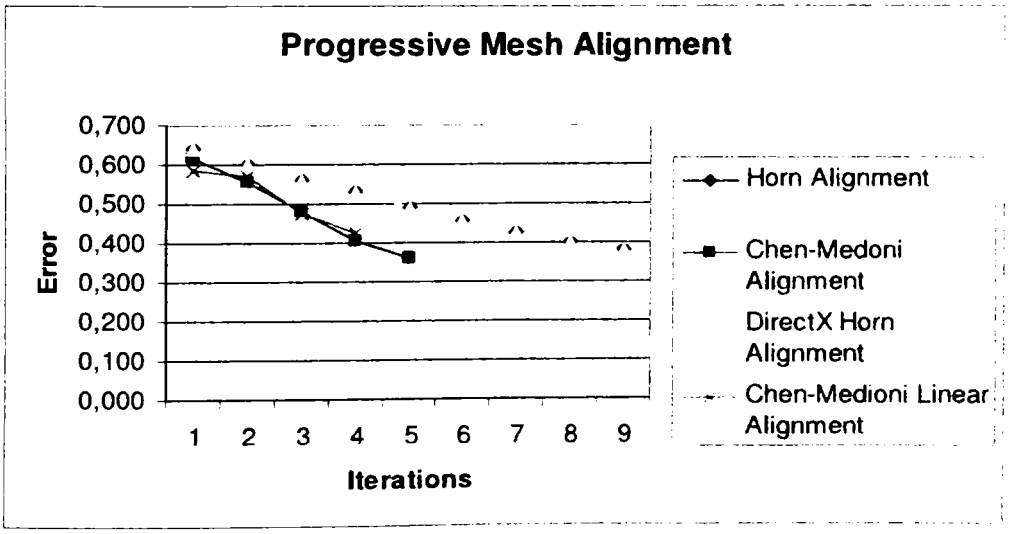

Figure 55 Convergence behavior with Progressive Mesh
The use of Progressive Meshes leads to more iteration steps because of the lack of points in the first iteration, but increases the performance dramatically.

Comparing these results to the Progressive Mesh-based alignment, the Chen-Medioni Linear Alignment needs one more iteration due to the low number of points in the first iteration step. Nevertheless, the Progressive Mesh-based Chen-Medioni Linear Alignment (PMICP CML Aligment) is faster than the Chen Medioni Linear Alignment without Progressive Meshes. This can be proven in experiments with other data sets.

To compare the best alignment method of the previous experiments to further algorithms, the fractal data set from [13] is used again. In the comparison of [195], many more registration methods are tested in detail, focusing on accuracy. Nevertheless, they included the total convergence in their paper. Based on the paper and the source code of Matabosch et al. [63], the PMICP CML Alignment method is compared to the implementation of other algorithms.

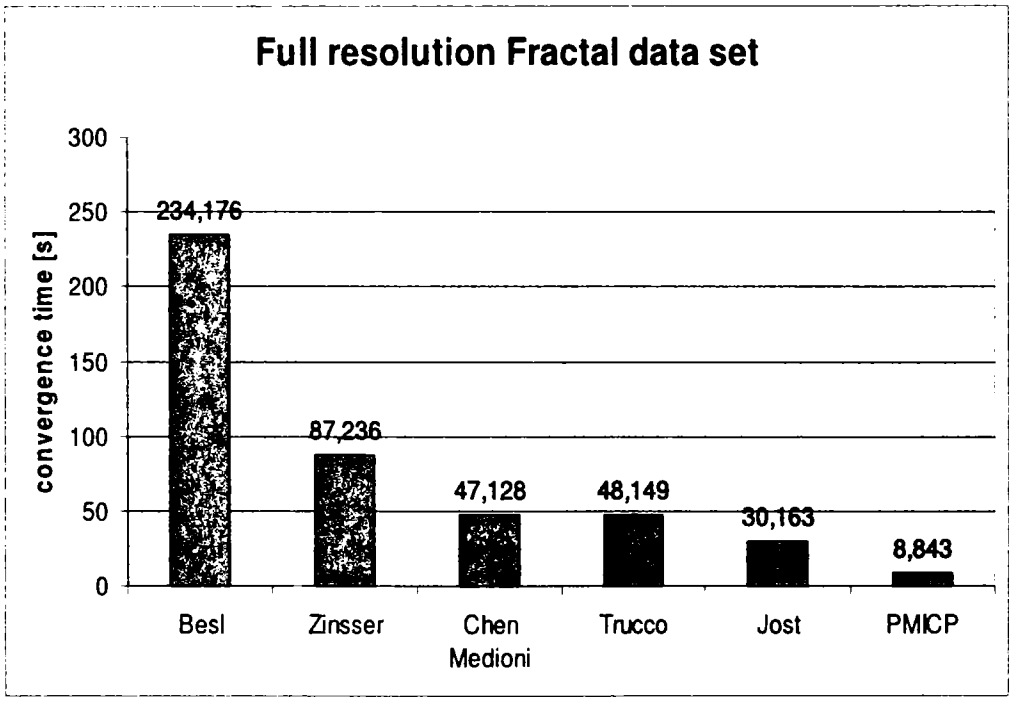

Figure 56 Comparison of total convergence time
This diagram compares the results of the implementation of the most important modifications of the ICP algorithms to the implementation of this work (PMICP). The PMICP is over 2.5 times faster than the fastest implementation of Jost.

In Figure 56, the total convergence time of implementations of [63] are shown. This diagram is created with similar preconditions to their tests. The fractal data set of [13] is rotated in one degree of each axis and translated in one unit. With this test, the exact transformation matrix is known. Like the test of [63], the iteration is stopped when the mean error falls below the threshold of 0.00001, or if the maximum number of 20 iterations is reached. No further conditions and further improvements such as sub-sampling or kd-tree implementation are integrated to meet the requirements for the comparison to the results in [63]. In contrast to these tests, however, the full resolution of the mesh with 4096 points in total is

used. The basic algorithm is not surprisingly the slowest in this test scenario. The point-to-plane approach of Chen and Medioni [137] is known to perform well on plain surfaces. Extending this with the multi-resolution-based algorithm of this thesis, it turns out that the PMICP CML Alignment performs about 3 to 4 times better than the fastest algorithm of the test implementations of Matabosch [63], based on the multi-resolutional ICP of Jost [14].

In the test data, it is shown that the PMICP is the most efficient implementation in all test scenarios provided in [63]. The proposed reduction of the influence of outliers of the PMICP is proven by the comparison of the total convergence time for synthetic tests of data sets to real data sets. The PMICP outperforms the fast algorithm by Jost [14] with about a factor of 7. These results of the PMICP include the creation of the Progressive Meshes, the iteration process itself and the final transformation of the data sets.

5.3 System implementation details and evaluation

This section goes into the implementation details for the object localization system, which is generally described in Section 2.1. The system is separated into two hierarchical steps — the pose estimation and the pose refinement.

5.3.1 Pose estimation implementation details

As already shown in Chapter 3, the pose estimation can be different depending on the application. The implementation of the pose estimation is introduced in detail in this section. The overview in Figure 57 shows the steps in the implementation of the system. On the left side of the diagram, the data acquisition is shown. The object in the scene is usually known by an identifier or name. The sensor and its main properties (e.g. measurement principle) and parameters (resolution, coordinate system and position) are also known. All this information is needed for the data acquisition step to result in the acquired points in the current coordinate system. The pre-processing of the point list filters outliers and makes the scene data set available for the pose estimation algorithm, which compares the acquired points with the simulated data sets. This simulation is shown on the right side of the diagram. All steps for a certain set of object poses are already done beforehand (offline). The known CAD representation (e.g. in the well known STL file format) is the input data to the simulation step. Furthermore, the sensor model is selected according to the sensor used in the data acquisition. First, the points and normals are extracted from the STL file to transform them into the required pose. The transformed data set is given to the sensor simulation function.

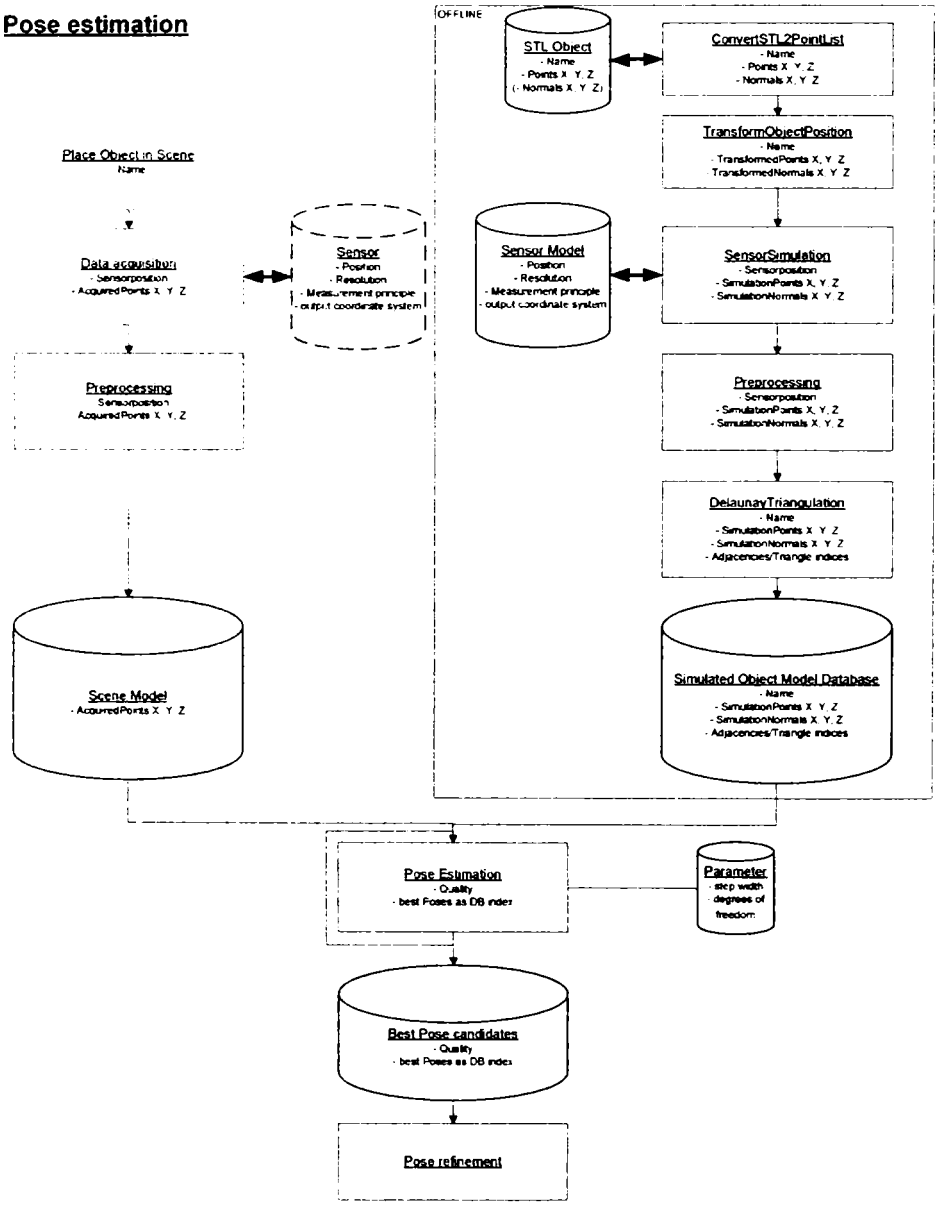

Figure 57 Pose estimation implementation diagram
The pose estimation is separated in the processing of the real sensor data on the left and the sensor simulation on the right. The comparison results in the best matching candidates.

The sensor simulation function is a very important step in the pose estimation. At first the sensor simulation was implemented in Matlab as proof of concept. The detailed function header shows the need parameter for the sensor simulation implemented in Matlab:

```
function dist = SensorSimSICKLDX9(transformedObject,
Sensorposition, angle_start, angle_stepwidth, numofbeams,
encoder_steps, encoder_stepwidth)
```

Figure 58 Pose Sensor Simulation function declaration
The CAD object (transformedObject) and the parameters according to the sensor model is provided to the function and result in a list of points

The first parameter is a multidimensional array of triangle vertices. This array contains the object points in the sensor simulation coordinate system. The transformation from the model coordinate system to the sensor simulation coordinate system is already applied to the points. The array is a list of triangles defined with at least 3 three-dimensional points. One triangle t_i consists of a 3x3 double precise floating point matrix:

$$t_i = \left[\begin{pmatrix} x_{1i} \\ y_{1i} \\ z_{1i} \end{pmatrix} \begin{pmatrix} x_{2i} \\ y_{2i} \\ z_{2i} \end{pmatrix} \begin{pmatrix} x_{3i} \\ y_{3i} \\ z_{3i} \end{pmatrix} \right] \qquad 5.2$$

The object array is a list of *n* triangles, so the dimension of the array is 3x3x*n*. The ray-triangle intersection implementation in Microsoft DirectX© requires a compatible mesh representation. The mesh is generated from the triangle list implemented with the function *D3DXCreateMesh()* in the underlying efficient C++ implementation.

The next parameter *Sensorposition* of the function *SensorSimSICKDX9* defines the position of the sensor in the sensor simulation coordinate system. This is a 3x1 vector according to the sensor model definition (see Section 3.2.2).

The next parameters define the view direction and field of view of the sensor. Starting from the value of the parameter *angle_start*, one ray per *angle_stepwidth* is created. The number of rays in one scan line is limited with the value of *numofbeams* here. With these parameters, an array of rays is calculated according to the following equation:

$$Ray_i = \left[\begin{pmatrix} Sensorposition_1 \\ Sensorposition_2 \\ Sensorposition_3 \end{pmatrix}, \begin{pmatrix} \cos(angle\_start(i * angle\_stepwidth)) \\ Sensorposition_2 \\ 1 \end{pmatrix} \right] \qquad 5.3$$

The last two parameters setup the movement of the scan line in the direction of the Y coordinate axis. The $Sensorposition_2$ or Y position of the sensor is incremented in steps of *encoder_stepwidth*. This is done *encoder_steps* times, so the resulting distance map is an $m \times p$ image where *m* is equal to *numberofbeams* and *p* is equal to *encoder_steps*. All these parameters are transferred to a DirectX©-based C++ function. The core of the simulation is the ray-triangle

intersection tests. This is made with the SIMD function *D3DXIntersect*:

```
HRESULT D3DXIntersect(LPD3DXBASEMESH pMesh, CONST
D3DXVECTOR3 *pRayPos,CONST D3DXVECTOR3 * pRayDir, BOOL *
pHit,DWORD *pFaceIndex, FLOAT *pU, FLOAT *pV, FLOAT *pDist,
LPD3DXBUFFER *ppAllHits, DWORD *pCountOfHits);
```

Figure 59 Intersection function declaration
The most important parameters are the definition of the ray and the mesh. In this implementation only the distance value is used to calculate the resulting point.

The parameter *pMesh* and the ray (defined by a point in the coordinate system *pRayPos* and direction *pRayDir*) are provided to the function as inputs. The result the function provides the index of the intersected triangle in the mesh, the distance to the intersection point and its barycentric coordinates. With equation 6.4 the intersection point is determined.

$$\begin{pmatrix} Intersec tionPos_1 \\ Intersec tionPos_2 \\ Intersec tionPos_3 \end{pmatrix} = \begin{pmatrix} Sensorposition_1 \\ Sensorposition_2 \\ Sensorposition_3 \end{pmatrix} + pDist \times \begin{pmatrix} pRayDir_1 \\ pRayDir_2 \\ pRayDir_3 \end{pmatrix} \qquad 5.4$$

The vector *pRayDir* is created according to equation 6.4 and linearly moved in Y-direction. All intersection points together form a distance map. This distance map is returned by the function *SensorSimDX9*, and the sensor simulation then finishes. The distance map is converted in the function *ConvertPointList2ProgressiveMesh()* to a Progressive Mesh representation. This step is not necessary in this state of implementation because the Progressive Mesh representation is not required in the pose estimation function. Due to a better performance in the pose refinement step, the generation of Progressive Meshes of the simulated objects is prepared in the offline calculations beforehand. This step mainly includes a triangulation of the data using a 2D delaunay triangulation [196]. The resulting triangulation indices are stored in the database in addition to the points and normals of the simulated object representation. The database entity relationship model of the database is shown in Figure 60.

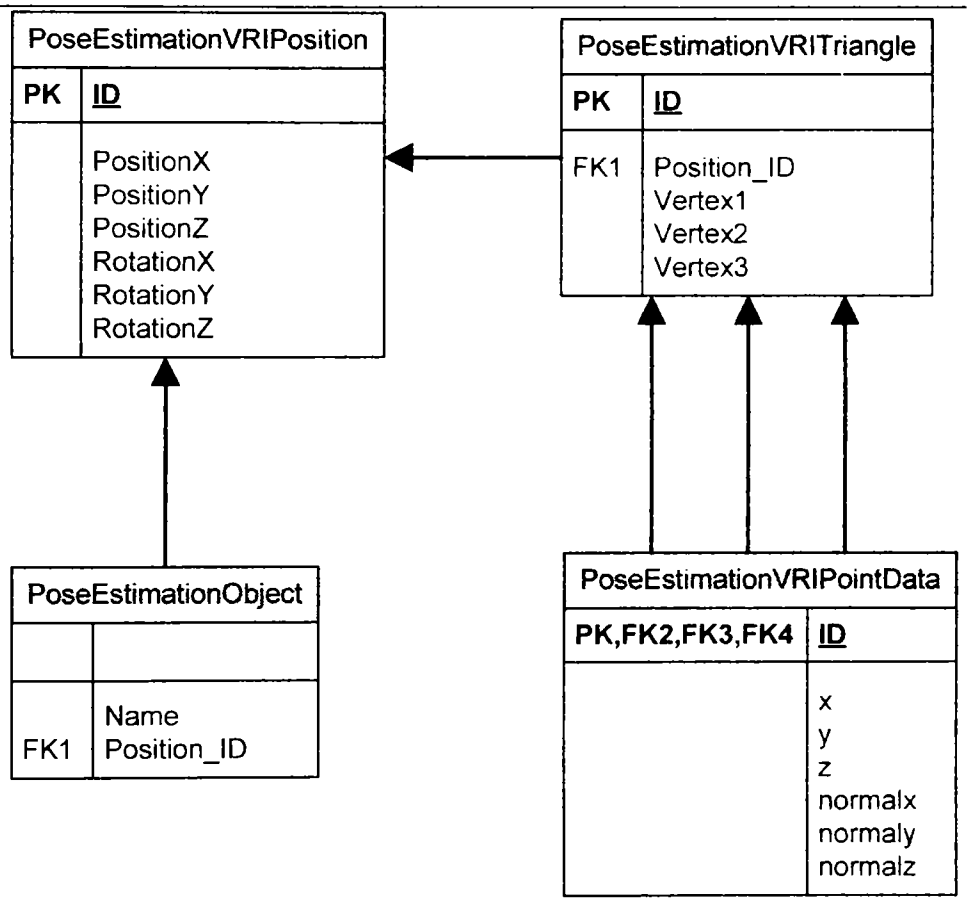

Figure 60 Database entity relationship model
The entity model shows the data stored in the database. For each VRI object the list of triangles and the list of points of these triangles are connected to each other via ID fields.

For every object in the entity *PoseEstimationObject*, the name and the position is integrated. The name identifies the used object and its STL-File and the Position_ID references to the VRI positions in *PoseEstimationVRIPositions*. Each position consists of n triangles with its n*3 points. To determine the indices of the triangles from a points list, the delaunay triangulation is used. To simplify the process of Progressive Mesh generation, the normals in every vertex are calculated in the same function *DelaunayTriangulation()* (see Figure 57). These offline-generated objects in the database are compared to the scene in the function *PoseEstimation()*. The implementation of the brute force pose estimation (see Figure 47) is "straightforward" in the case of looping through the positions in the database with a defined step width in the directions of all degrees of freedom. Depending on the density of the compared positions and the size of the scene, the implementation of the ray-triangle intersection plays a major role, as already shown in Section 3.3.1.

The accelerated brute force pose estimation is based on the normalized

cross correlation implementation in Matlab(www.mathworks.com). The algorithm usually cannot be computed in the frequency domain in a simple and efficient way [115]. To overcome the computationally expensive normalized cross correlation in the spatial domain, the implementation of the function *normxcorr2()* in Matlab uses the improved version of Lewis [115]. The function *normxcorr2()* is a 2D normalized cross correlation function in the image processing toolbox of Matlab and takes two parameters:

```
C = normxcorr2(template, A)
```

Figure 61 2D normalized cross function declaration
The template is searched the data A. The result C is a 2D array, where the highest value represents the best matching position.

The input parameter *template* is a matrix, which has to be smaller than the second input parameter *A*. Parameter *A* is given by the distance map of the scene and the parameter template is the VRI representation. It is obvious that the function is not invariant to scale and rotation. The scaling effect is given in the pose estimation process by the difference in the distance between the sensor and the simulated object. Furthermore, the object is rotated in many cases. Nevertheless, the normalized cross correlation in the Matlab implementation can be used to increase the performance, especially for huge data sets as proven in [115]. The function is used instead of the translational degrees of freedom in the brute force pose estimation steps for this reason. The object is rotated in every direction and the normalized cross correlation finds the best position for this pose in the scene data set. The best matching candidates are stored in an array of position indices from the database representation (primary key of entity *PoseEstimationVRIPosition*, see Figure 60). These candidates are used in the refinement process. The diagram (Figure 62) shows the implementation steps in detail.

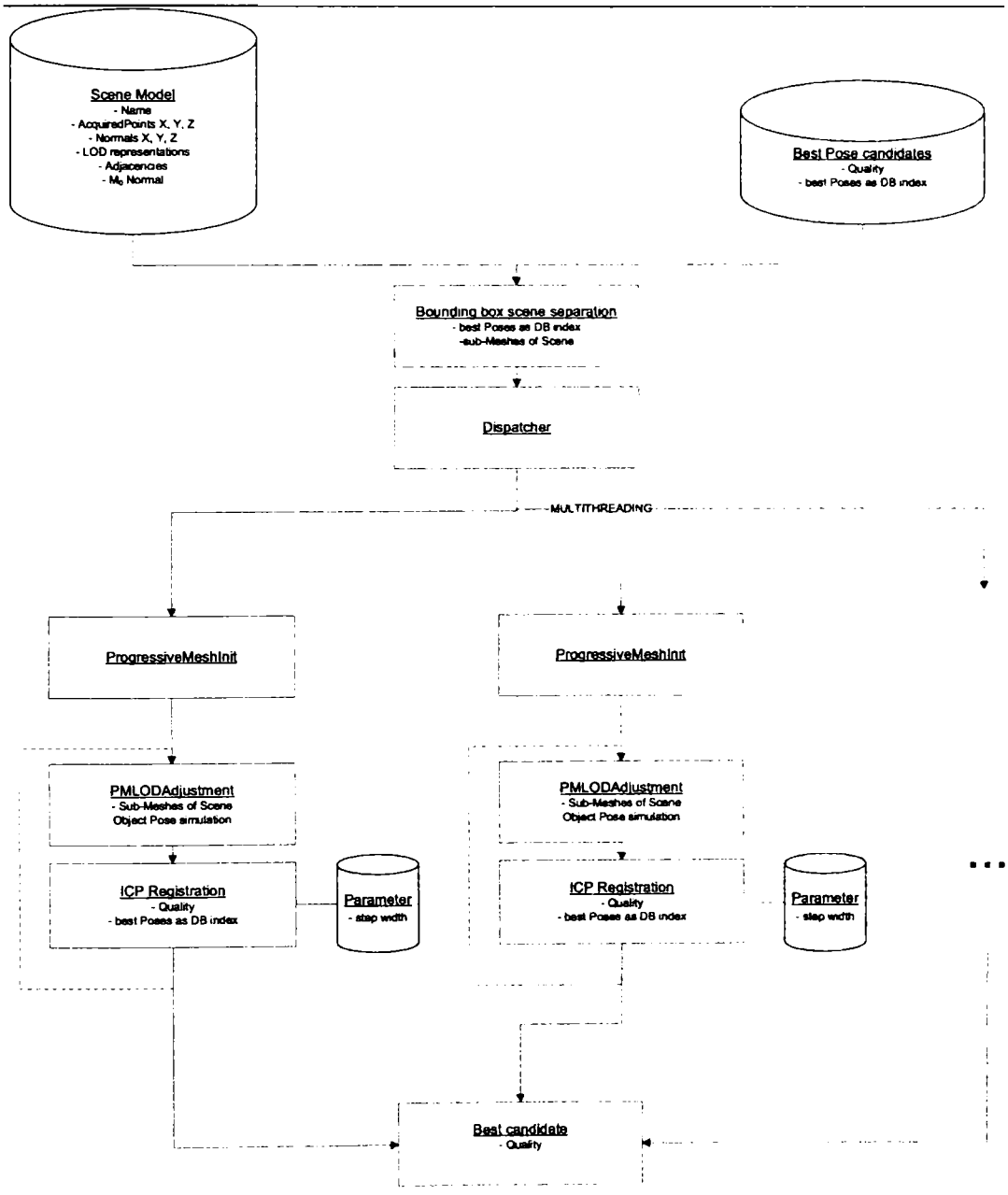

Figure 62 Pose refinement implementation diagram
The pose refinement is realized with a solution to parallelize the data streams. For each pose candidate one thread is started to calculate the alignment of this datasets.

The database indices of the best candidates are delivered with their quality of the pose estimation process. With the index a two-dimensional bounding box in the scene is determined in the function *BoundingBoxSceneSeparation()*. This

function is implemented in the following way. The size of the X-Y boundaries of the model in the scene coordinates is retrieved from the database. The shape of the bounding box of the scene part is created according to:

$$P = \{p_j\} \quad with \quad p_j = \begin{pmatrix} 0.9 \times \min(x_i) < x_j < 1.1 \times \max(x_i) \\ 0.9 \times \min(y_i) < y_j < 1.1 \times \max(y_i) \\ z_j \end{pmatrix} \qquad 5.5$$

Due to this, the size of the scene is increased by 10% in each direction. This is illustrated in the Figure 62.

The new scene's data set and the corresponding model's data set are given to the dispatcher. The dispatcher collects the tasks for registration and distributes them according to the free resources in the computer system. For each registration task, a thread is started with its own data sets. This offers a highly efficient parallel registration on multiprocessor systems. Each path in Figure 62 represents one thread, which handles the registration of exactly one scene/model pair. Each thread instantiates a class which is shown in Figure 63, sets the point data sets, and executes the calculation function.

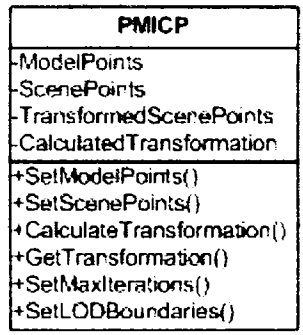

Figure 63 Registration class
The class encapsulates the functions for the registration. Main function is given by the function CalculateTransformation after the model and scene points are set.

The functions *SetModelPoints()* and *SetScenePoints()* convert the model and scene points to an internal Progressive Mesh representation. Therefore, the points and triangle indices are retrieved from the database and converted into a Microsoft DirectX© compatible mesh (D3DXMESH) in the same way as was done in the sensor simulation.

This mesh is transformed to a Progressive Mesh with the function:

```
HRESULT D3DXGeneratePMesh(LPD3DXMESH pMesh,CONST DWORD
*pAdjacency, CONST D3DXATTRIBUTEWEIGHTS
*pVertexAttributeWeights, CONST FLOAT *pVertexWeights, DWORD
MinValue, DWORD Options,LPD3DXPMESH *ppPMesh);
```

Figure 64 Progressive mesh generation function declaration
The progressive mesh is generate from the mesh representation with the predefined adjadency and other optional parameters

The function *SetMaxIterations()* sets the parameters for the iteration stop criterions, and the function *SetLODBoundaries()* sets the minimum and maximum levels of detail (M_{imin} and M_{imax}) of the Progressive Mesh. These parameters (and some other internal hard-coded parameters) tweak the behavior and convergence rate of the registration process. The registration process is started with *CalculateTransformation()*. This core function includes all the steps described in Section 4.4.2 for the Progressive Mesh-based Iterative Closest Point algorithm. After the iteration process has finished, the transformation of every scene/model pair is applied to the scene data set. The thread calculates the quality according to equation 4.20. After all the threads are finished, the returned quality of each thread is compared to the other threads in the function *FindBestCandidate()*. If the quality is better than a defined threshold, the candidate with the best quality is returned. To get the right pose of the object, the position and rotation of the candidate in the database are used.

$$P_{Target} = T_{PoseEstimation} \times T_{registration} \times P_{STL} \qquad 5.6$$

To get the exact target Position P_{Target}, the transformation of the simulated object from the database $T_{PoseEstimation}$ and the transformation of the registration process $T_{Registration}$ must be applied to the STL object model points P_{STL}.

Chapter 6 Conclusion

6.1 Outline of Contributions

This thesis proposes a system to align range data surfaces in the context of industrial process automation. The solution for robotic bin picking deals with range images provided by range sensors and the use of a model-based scalable hierarchical system to find known objects in 3D data.

The pose estimation, introduced in Chapter 3, provides a basic system that can be used with any kind of known objects acquired by laser range sensors. It is shown that the proposed solution solves the object pose estimation with its universality, without the need of segmentation or feature extraction. In general, this approach "simulates" the stimuli of the real world to a sensor in advance and compares these stimuli to data acquired from the real world. This general approach is applied to range sensors and object localization in this work. The simulation of data acquisition offers the possibility to use the proposed solution in many scenarios. New, innovative range sensors with higher resolutions can be modeled without any problem; most of which deliver range images with measurement errors, occlusions, and reflections. To increase the accuracy, the pose simulation takes all the properties and features of real sensors into consideration. By using sensors with high resolutions for range data acquisition, the complexity of the object localization process is increased. The complexity of the object localization depends also on the chosen algorithm and the complexity of the object. To reduce the high computational cost (depending on the application and the used PC), the proposed solution is scalable. For example, the number of pre-calculated VRIs in the database can change this computational time-memory tradeoff. The solution is characterized by a flexible coarse-to-fine algorithm, which can be used to find any kinds of objects in range data, provided these objects are known to the proposed system. The pose estimation is not limited to only one object because the database stores as many different kinds of objects as necessary. Another advantage of the scalability in the coarse pose estimation process is the fact that the required accuracy can be adjusted by simple changes in the position and orientation step width. The density of positions and the rotational degree of freedom of the object to create a VRI are also important parameters for adjusting the level in pose estimation. Additionally, sub-sampling can be integrated if it is necessary due to system limits. Because of the improvements in parallel computing and the support of new innovative approaches in general purpose GPU computing, the proposed solution has a high performance and is able to meet many industrial requirements.

In Chapter 4, the thesis focuses on the improvements in the refinement step of the hierarchical object localization system. The well-known and proven ICP

algorithm, with its variants, is a registration algorithm that tries to transform a scene iteratively into the position of a model. A lot of modifications and improvements have been introduced in the last few years, and some expansions and results have been described in Section 4.3. The ICP algorithm generally offers a proven method to transform two sets of points in a common coordinate system. The input data is independent of the representation and source, and the algorithm does not need a time-consuming and complex feature extraction. In the proposed implementation, the ICP algorithm is combined with the Progressive Meshes. The LOD adjustment inside the ICP iteration steps is a new and innovative approach to solve at least two major problems of the ICP algorithm. The combination increases the robustness against outliers in the early iterations' steps to counter misalignments. The integrated level of detail adjustments reduces computational costs and leads to a very high performance compared to other known improvements of the ICP algorithm. To be sure of meeting the requirements of different applications, the proposed solution is evaluated with test scenarios in Chapter 5, which cover many types of possible range data scenes. These tests prove the advantage over all algorithm implementations known to author. The pose estimation and pose refinement do not use any segmentation algorithms, but instead use 3D information in a hierarchical system. This leads to a high universality for the object localization. The proposed solution was successfully implemented and tested in different application test scenarios to prove the potential of the pose estimation and pose refinement.

6.2 Future Work

A complete system for industrial robotic bin picking includes — beside the object localization — an adequate sensor selection, a robot control interface, a grasp point definition and a collision avoidance strategy. Merging all these components forms a system for many different applications. Nevertheless, the proposed system offers many possible extensions. Due to the incremental process, the object positions can be verified and tracked over all steps of the pose estimation and refinement process. This increases the robustness and reduces the computational costs. In the available literature, many suggestions and approaches for improvement are found, especially for acceleration and the increase of the robustness of object localization.

One of the promising improvements in the future will be a wider interaction between the pose estimation and the ICP algorithm. ICP could use the knowledge and properties of the global sensor position. For example, the point-to-point metric could be a change to the so-called "reversed calibration metric", introduced by Blais and Levine [165].

Many improvements on the robustness and the increase of the convergence speed improve the ICP algorithm. In some cases this cannot ensure that the error of the minimization function decreases monotonously. Important additions described in Section 4.4.2 are made in order to increase the robustness against measuring errors. Thus, the algorithm is not very vulnerable to outliers in the point data. Many researchers pay attention to the high computational time and error convergence to local minima.

Using Progressive Meshes in the iteration steps of the ICP algorithm offers several methods of adjusting the number of used triangles in the Progressive Mesh representation. Currently, the registration algorithm uses the current iteration counter in the iteration process to connect the level of detail in the meshes. In every iteration step of the ICP algorithm the number of faces in the model mesh and the scene mesh are increased with a defined value. Taking the degree of performance of one ICP iteration step into consideration, the number of triangles in the current Progressive Mesh could be adjusted to the current iteration step error as stated in [197].

With rising computer power, the ICP algorithm is also conceivable in video real time environments. Dorai et al. [167] have extended the algorithm of Chen and Medioni [137] and this is the basis for further developments at the multi-view registration[198], [170], [199]. Non-rigid registration methods also offer good potential to accelerate the algorithm [8] and will play an important role in future research.

The ICP algorithm is a robust algorithm for registration tasks in many fields of research. Consequently, there is a large quantity of further expansion capabilities available.

Chapter 7 References

x

[1] A. C. Kak and J. L. Edwards, "Experimental State of the Art in 3D Object Recognition and Localization Using Range Data," in *Proceedings of the IEEE Workshop on Vision for Robots*, 1997.

[2] D. Katsoulas, "Robust Recovery of Piled Box-Like Objects in Range Images.," Albert-Ludwigs-Universität Freiburg, Ph.D. Thesis 2004.

[3] M. Hashimoto and K. Sumi, "3-D Object Recognition Based on Integration of Range Image and Gray-scale Image," in *British Machine Vision Conference*, 2001.

[4] K. Ikeuchi, B. K. Horn, S. Nagata, T. Callahan, and O. Feingold, "Picking up an object from a pile of objects," in *MIT AI Memo*, 1983, pp. 139-166.

[5] M. Kavoussanos and A. Pouliezos, "Visionary automation of sack handling and emptying," *IEEE Robotics & Automation Magazine*, vol. 7, no. 4, pp. 44-49, Dec. 2000.

[6] P. J. Besl and H. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[7] H. Zhou, Y. Liu, and L. Li, "Incremental mesh-based integration of registered range images: robust to registration error and scanning noise," in *Asian Conference on Computer Vision*, vol. 3851, 2006, pp. 958-968.

[8] B. J. Brown, "Registration and Matching of Large Geometric Datasets for Cultural Heritage Applications," Department of Computer Science, Princeton University, Ph.D. Thesis 2008.

[9] R. J. Campbell and P. J. Flynn, "A survey of free-form object representation and recognition techniques," *Comput. Vis. Image Underst.*, vol. 81, no. 2, pp. 166-210, 2001.

[10] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America. A*, vol. 4, no. 4, pp. 629-642, Apr 1987.

[11] D. Simon, "Fast and Accurate Shape-Based Registration," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Ph.D. Thesis

December 1996.

[12] M. D. Wheeler, "Automatic Modeling and Localization for Object Recognition," Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, Ph.D. Thesis 1996.

[13] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. Third International Conference on 3-D Digital Imaging and Modeling*, 2001, pp. 145-152.

[14] T. Jost and H. Hugli, "A multi-resolution scheme ICP algorithm for fast shape registration," in *Proc. First International Symposium on 3D Data Processing Visualization and Transmission*, 19–21 June 2002, pp. 540-543.

[15] A. Nuechter, "Semantische dreidimensionale Karten fuer autonome mobile Roboter," University of Bonn, Ph.D. Thesis 2006.

[16] Wikipedia: The Free Encyclopedia. (2008, Aug.) DARPA Grand Challenge. [Online]. http://en.wikipedia.org/wiki/DARPA_Grand_Challenge

[17] P. Biber and W. Strasser, "nScan-Matching: Simultaneous Matching of Multiple Scans and Application to SLAM," in *IEEE International Conference on Robotics and Automation*, 2006.

[18] F. Lu and E. Milios, "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans," *J. Intell. Robotics Syst.*, vol. 18, no. 3, pp. 249-275, 1997.

[19] D. Huber and N. Vandapel, "Automatic 3D underground mine mapping," in *International Conference on Field and Service Robotics*, July 2003.

[20] K. Pulli, "Multiview Registration for Large Data Sets," *2nd International Conference on 3D Digital Imaging and Modeling*, vol. 0, p. 0160, 1999.

[21] F. Blais and J. A. Beraldin, "Tutorial: Active 3D Sensing," in *Third International Conference on 3D Digital Imaging Modeling*, 2001.

[22] C. Brenner, "Building reconstruction from images and laser scanning," *International Journal of Applied Earth Observation and Geoinformation*, vol. 6, no. 3-4, pp. 187-198, 2005.

[23] D. Huber and M. Hebert, "A New Approach to 3-D Terrain Mapping," in *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS '99)*, October 1999, pp. 1121-1127.

[24] M. Kada, "Zur maßstabsabhaengigen Erzeugung von 3D-Stadtmodellen," University Stuttgart, Ph.D. Thesis 2007.

[25] C. Beder, "Grouping Uncertain Oriented Projective Geometric Entities with Application to Automatic Building Reconstruction," Photogrammetry Department, Institute for Geodesy and Geoinformation, Bonn University, Germany, Nussallee 15, 53115 Bonn, Germany, Ph.D. Thesis Jan 2007.

[26] C. Froehlich and M. Mettenleiter, "Terrestrial Laser-Scanning - New Perspectives in 3D-Surveying," in *Proceedings of the ISPRS working group VIII/2*, 2004.

[27] P. Pfaff et al., "Towards Mapping of Cities," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 4807-4813.

[28] J. Santamar, O. Cord, S. Damas, I. Aleman, and M. Botella, "A scatter search-based technique for pair-wise 3D range image registration in forensic anthropology," *Soft Comput.*, vol. 11, no. 9, pp. 819-828, 2007.

[29] S. Se and P. Jasiobedzki, "Instant Scene Modeler for Crime Scene Reconstruction," p. 123, 2005.

[30] C. S. Fraser, H. B. Hanley, and S. Cronk, "Close-Range Photogrammetry for Accident Reconstruction," in *Optical 3D Measurements VII Techical Univiversity of Vienna*, 2005.

[31] J. Feldmar, N. Ayache, and F. Betting, "3D-2D projective registration of free-form curves and surfaces," in *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, Washington, DC, USA, 1995, p. 549.

[32] M. Aleff, A. Krzizok, W. Neddermeyer, R. Seibel, and W. Winkler, "3D-NaMiS, ein Navigationssystem für den minimal invasiven Eingriff.," in *Bildverarbeitung für die Medizin*, 2005, pp. 257-261.

[33] A. Almhdie, C. Léger, M. Deriche, and R. Lédée, "3D registration using a new implementation of the ICP algorithm based on a comprehensive lookup matrix: Application to medical imaging," *Pattern Recogn. Lett.*, vol. 28, no. 12, pp. 1523-1533, 2007.

[34] H.-Y. Shum, K. Ikeuchi, and R. Reddy, "Virtual reality modeling from a sequence of range images," in *IEEE / RJS International Conference on Intelligent Robots and Systems*, vol. 1, September 1994, pp. 703-710.

[35] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *in*

Proceedings of Image Vision Computer, vol. 21, no. 13-14, pp. 1145-1153, 2003.

[36] K. Boehnke, P. Roebrock, and M. Otesteanu, "Industrielle optische 3D-Messung von Airbaggehäusen mit Laserlichtschnittsensoren," in *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik*, Oldenburg, 2007, pp. 70-78.

[37] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3-D data," in *NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, Washington, DC, USA, 1997, p. 234.

[38] K. Boehnke, "Industrielle optische 3D-Messung von Airbaggehäusen mit Laserlichtschnittsensoren," *German Annual Journal for Engineering (Automatisierungs-Atlas)*, vol. I, no. I, Oct. 2008.

[39] K. Boehnke, "Data structures for industrial laser range sensors," in *Symposium on Electronics and Telecommunications (ETC 2006), University of Timisoara, Faculty of Electronics and Telecommunications & IEEE Romania Section*, Timisoara, Romania, September 2006, pp. 64-69.

[40] K. Boehnke, "3D Sensors," *INSPECT Vision, Automation und Control Journal publisher John Wiley & Sons*, vol. Buyers Guide, no. 1, pp. 44-46, Oct. 2008.

[41] R. B. Fisher and K. Konolige, "Range Sensors," in *Handbook of Robotics.*: Springer Verlag, 2008, p. Chapter 22.

[42] B. K. Horn, *Robot Vision*,. Cambridge, MA: The MIT Press, 1986.

[43] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2th ed.,.: Chapman & Hall, 1998.

[44] K. Rahardja and A. Kosaka, "Vision-Based Bin-Picking: Recognition and Localization of Multiple Complex Objects Using Simple Visual Cues," , 1996.

[45] P. Dias, V. Sequeira, J. G. Goncalves, and F. Vaz, "Fusion of intensity and range data for improved 3D models," in *Proc. International Conference on Image Processing*, vol. 3, 7–10 Oct. 2001, pp. 1107-1110.

[46] F. Blais, J. -a. Beraldin, and S. F. El-hakim, "Range Error Analysis of an Integrated Time-of-Flight, Triangulation, and Photogrammetry 3D Laser Scanning System," in *SPIE Proceedings AeroSense*, 2000.

[47] R. Collier, "Characterization of a Range Scanning System Utilizing a Point Laser Rangefinder," Knoxville, 1998.

[48] J. A. Beraldin, "Integration of Laser Scanning and Close-Range Photogrammetry - The Last Decade and Beyond," in *XXth Congress International Society for Photogrammetry and Remote Sensing*, 2007.

[49] T. Ringbeck and B. Hagebeuker, "A 3D time of flight camera for object detection," in *Optical 3-D Measurement Techniques*, 2007.

[50] H. Rapp, *Investigation of Correlating Tof-Camera Systems*,.: Vdm Verlag Dr. Mueller E.K., 2008.

[51] O. Ghita and P. F. Whelan, "A bin picking system based on depth from defocus," *Mach. Vision Appl.*, vol. 13, no. 4, pp. 234-244, 2003.

[52] Wikipedia: The Free Encyclopedia. (2008, Sep.) Structured Light 3D Scanner. [Online]. http://en.wikipedia.org/wiki/Structured_Light_3D_Scanner

[53] J. Salvi, J. Pagès, and J. Batlle, "Pattern codification strategies in structured light systems," *Pattern Recognition*, vol. 37, no. 4, pp. 827-849, 2004.

[54] SICK AG, *Ranger E/D Reference Manual*. SICK IVP www.sickivp.com, 2008.

[55] R. Y. Tsai, "A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," in *IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4*, 1987, pp. 323-344.

[56] K. Harding, "Calibration methods for 3D measurement systems," in *Conference Machine vision and three-dimensional imaging systems for inspection and metrology*, Boston, 2001, pp. 239-247.

[57] P. Peternek, "3D Laser Messsystem Ranger E55," Mannheim, 2008.

[58] K. Boehnke, "Triangulation based 3D laser sensor accuracy and calibration," in *Symposium on Electronics and Telecommunications (ETC 2006), University of Timisoara, Faculty of Electronics and Telecommunications & IEEE Romania Section*, Timisoara, Romania, September 2008, pp. 224-230.

[59] K. Boehnke, "Object localization in range data for robotic bin picking," in *Proc. IEEE International Conference on Automation Science and Engineering CASE 2007*, 22-25 Sept. 2007, pp. 572-577.

[60] L. Caponetti et al., "A Three-Dimensional Vision System for Bin-Picking," in *Proc. IEEE 1986 Conf. on Compuer Vision and Pattern Recognition*, May 1986, pp. 407-411.

[61] G. Bachler, M. Berger, R. Röhrer, S. Scherer, and A. Pinz, "A Vision Driven Automatic Assembly Unit," in *CAIP '99: Proceedings of the 8th International Conference on Computer Analysis of Images and Patterns*, London, UK, 1999, pp. 375-382.

[62] F. Boughorbel et al., "Laser Ranging and Video Imaging for Bin Picking," *Assembly Automation*, vol. 1, pp. 53-59, 2003.

[63] C. Matabosch, J. Salvi, D. Fofi, and F. Meriaudeau, "Range image registration for industrial inspection," in *Machine Vision Applications in Industrial Inspection XIII*, San Jose, California, USA, January 2005.

[64] K. Boehnke, M. Otesteanu, P. Roebrock, W. Winkler, and W. Neddermeyer, "Neural network based object recognition using color block matching," in *SPPR'07: Proceedings of the Fourth conference on IASTED International Conference*, Anaheim, CA, USA, 2007, pp. 122-125.

[65] L. J. Latecki, R. Lakaemper, X. Sun, and D. Wolter, "Building Polygonal Maps from Laser Range Data," in *Proceedings of the workshop cognitive robotics (CogRob)*, 2004.

[66] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon, "Automatic Line Matching And 3D Reconstruction Of Buildings From Multiple Views," in *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, vol. 3, 1999, pp. 69-80.

[67] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," *Computer Graphics*, vol. 30, no. Annual Conference Series, pp. 303-312, 1996.

[68] J. Weingarten, G. Gruener, and R. Siegwart, "A State-of-the-Art 3D Sensor for Robot Navigation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, vol. 3, 2004, pp. 2155-2160.

[69] Z. Zhigang and H. Yu-Chi, "Gamma/X-Ray Linear Pushbroom Stereo for 3D Cargo Inspection," in *SPIE Defense and Security Symposium*, Orlando, Florida, USA, 2006, pp. 6213-04.

[70] D. G. Lowe, "Fitting Parameterized Three-Dimensional Models to Images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 5, pp. 441-450, 1991.

[71] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," in *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1993, pp. 19-26.

[72] J. Andrade-Cetto and A. C. Kak, , J. G. Webster, Ed.: Wiley Encyclopedia of Electrical and Electronics Engineering, 2000.

[73] V. Rodehorst and A. Koschan, "Comparison and Evaluation of Feature Point Detectors," in *Proc. of the 5th Int. Symposium - Turkish-German Joint Geodetic Days (TGJGD 2006)*, Berlin, March 2006, pp. 1-8.

[74] D. P. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *Int. J. Comput. Vision*, vol. 5, no. 2, pp. 195-212, 1990.

[75] T. O. Binford, "Visual Perception by Computer," in *Proceedings of the IEEE Conference on Systems and Control (Miami, FL)*, 1971.

[76] S. J. Dickinson et al., "Panel report: the potential of geons for generic 3-D object recognition.," *Image Vision Comput.*, vol. 15, no. 4, pp. 277-292, 1997.

[77] Y. Zhang, "Superquadric representation of scenes from multi-view range data," PhD thesis, University of Tennessee, 2003.

[78] A. R. Pope and D. G. Lowe, "Learning Appearance Models for Object Recognition," in *ECCV '96: Proceedings of the International Workshop on Object Representation in Computer Vision II*, London, UK, 1996, pp. 201-219.

[79] C. Rothwell et al., "An Experimental Comparison of Appearance and Geometric Model Based Recognition," in *Proceedings of the International Workshop on Object Representation in Computer Vision II*, London, UK, 1996, pp. 247--269.

[80] S. Dickinson, , E. Lepore and Z. Pylyshyn, Eds.: Basil Blackwell publishers, 1999, pp. 172-207.

[81] X. Chen, T. Faltemier, P. Flynn, and K. Bowyer, "Human Face Modeling and Recognition Through Multi-View High Resolution Stereopsis," in *Proc. Conference on Computer Vision and Pattern Recognition Workshop*, 17-22 June 2006, pp. 50-50.

[82] D. G. Lowe, *Perceptual Organization and Visual Recognition*,. Norwell, MA, USA: Kluwer Academic Publishers, 1985.

[83] D. W. Jacobs and R. Basri, "3-D to 2-D recognition with regions," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 17–19 June 1997, pp. 547-553.

[84] B. Zitova and J. Flusser, "Image Registration Methods: A Survey," in *Image and Vision Computing*, vol. 21, October 2003, pp. 997-1000.

[85] P. J. Besl and R. C. Jain, "Three-Dimensional Object Recognition," *COMPUTING SURVEYS*, vol. 17, pp. 75-145, 1985.

[86] A. R. Pope, *Model-Based Object Recognition - A Survey of Recent Research*,. Vancouver, BC, Canada, Canada: University of British Columbia, 1994.

[87] J. P. Brady, N. Nandhakumar, and J. K. Aggarwal, "Recent progress in the recognition of objects from range data," in *Proc. th International Conference on Pattern Recognition*, 14–17 Nov. 1988, pp. 85-92.

[88] P. G. Mulgaonkar, C. K. Cowan, and J. DeCurtins, "Understanding object configurations using range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 303-307, Feb. 1992.

[89] N. Okada and T. Nagata, "A parts picking system with a range finder and a camera system," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 2, 21–27 May 1995, pp. 1410-1415.

[90] B. K. Horn and K. Ikeuchi, *Picking parts out of a bin*,.: Massachusetts Institute of Technology, Cambridge, Massachusetts, 1982.

[91] O. Ghita and P. F. Whelan, "Robust robotic manipulation," in *Proceedings of the SPIE - Intelligent Robots and Computer Vision XVII*, vol. 3522, Boston, USA, 1998, pp. 244-254.

[92] R. D. Eriksen and I. Balslev, "From belt picking to bin picking," in *Optomechatronic systems III, Proceedings of SPIE*, vol. 4902, 2002, pp. 616-623.

[93] A. Distante et al., "A model-based 3-D vision system for bin-picking," *IEEE Transactions on Circuit and Systems, Volume: 35*, p. 545 – 553, 1988.

[94] R. C. Bolles and P. Horaud, "3DPO: A three-dimensional part orientation system," *Int. J. Rob. Res.*, vol. 5, no. 3, pp. 3-26, 1986.

[95] J. D. Foley, R. L. Phillips, J. F. Hughes, A. v. Dam, and S. K. Feiner, *Introduction to Computer Graphics*,. Boston, MA, USA: Addison-Wesley

Longman Publishing Co., Inc., 1994.

[96] Wikipedia: The Free Encyclopedia. (2008, Sep.) Z-buffering. [Online]. http://en.wikipedia.org/wiki/Z-buffer

[97] J. Park and G. N. DeSouza, *3-D Modeling of Real-World Objects Using Range and Intensity Images.*, B. Apolloni et al., Eds.: Springer, 2005, vol. 7.

[98] K. Boehnke, "Fast object localization with real time 3d laser range sensor simulation," *WSEAS Transactions on Electronics: Real time applications with 3D sensors*, vol. 5, no. 3, pp. 83-92, 2008.

[99] P. Dutre, P. Bekaert, and K. Bala, *Advanced Global Illumination*,. Natick, USA: A K Peters, 2003.

[100] Wikipedia: The Free Encyclopedia. (2008, Aug.) Global illumination. [Online]. http://en.wikipedia.org/wiki/Global_illumination

[101] Wikipedia: The Free Encyclopedia. (2008, Aug.) Shading model. [Online]. http://en.wikipedia.org/wiki/Shading_model

[102] S. Müller and M. Geimer, "A CrossPlatform Framework for Interactive Ray Tracing," in *Proceedings of GI Graphiktag*, Frankfurt, 2003, pp. 25-34.

[103] A. S. Glassner, *An introduction to ray tracing*,. London, UK, UK: Academic Press Ltd., 1989.

[104] I. Wald et al., "State of the Art in Ray Tracing Animated Scenes," in *Eurographics 2007 State of the Art Reports*, 2007.

[105] T. Moeller and B. Trumbore, "Fast, minimum storage ray-triangle intersection," *J. Graph. Tools*, vol. 2, no. 1, pp. 21-28, 1997.

[106] N. Chirkov, "Fast 3D Line Segment—Triangle Intersection Test," *Journal of graphics tools*, pp. 13-18, 2005.

[107] J. Hanika, "Fixed Point Hardware Ray Tracing," Ulm, 2007.

[108] K. Sung-Soo, N. Seung-Woo, K. Do-Hyung, and L. In-Ho, "Hardware-accelerated ray-triangle intersection testing for high-performance collision detection," in *The 15-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, University of West Bohemia, Plzen, 2007, pp. 17-25.

[109] A. Woo, "Fast ray-box intersection," *Graphics gems*, pp. 395-396, 1990.

[110] T. L. Kay and J. T. Kajiya, "Ray tracing complex scenes," *SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 269-278, 1986.

[111] M. Eisemann, T. Grosch, M. Magnor, and S. Mueller, "Fast Ray/Axis-Aligned Bounding Box Overlap Tests using Ray Slopes," *journal of graphic tools*, vol. 12, no. 4, Dec. 2007.

[112] I.-K. Lee, "Curve reconstruction from unorganized points," *Computer Aided Geometry Design*, vol. 17, no. 2, pp. 161-177, 2000.

[113] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, Washington, DC, USA, 1999, p. 1150.

[114] Wikipedia: The Free Encyclopedia. (2008, Aug.) Hill Climbing. [Online]. http://en.wikipedia.org/wiki/Hill_climbing

[115] J. P. Lewis, "Fast normalized cross-correlation," in *Vision Interface*, 1995, pp. 120-123.

[116] N. Roma, "A comparative analysis of cross-correlation matching algorithms using a pyramidal resolution approach," in *2nd Workshop on Empirical Evaluation Methods in Computer Vision*, 2000, pp. 117-142.

[117] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in Computational Stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 993-1008, 2003.

[118] A. Gruen, "Adaptive least squares correlation: a powerful image matching technique," *South African Journal of Photogrammetry, Remote Sensing and Cartography*, vol. 14, pp. 175-187, 1985.

[119] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325-376, 1992.

[120] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*,. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1991.

[121] D. Marr and T. Poggio, *A Theory of Human Stereo Vision*,. Cambridge, MA, USA: Massachusetts Institute of Technology, 1977.

[122] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679-698, 1986.

[123] D. Ziou and S. Tabbone, "Edge detection techniques: an overview," *International Journal on Pattern Recognition and Image Analysis*, vol. 8, no. 4, pp. 537-559, 1998.

[124] R. Steinbrecher, *Bildverarbeitung in der Praxis*, 2th ed., R. Oldenbourg, Ed.: R. Oldenbourg, 2005.

[125] H. Moravec, *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*,.: Stanford University, September 1980.

[126] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," in *4th ALVEY Vision Conference*, 1988, pp. 147-151.

[127] U. Dhond and J. Aggarwal, "Structure from stereo-a review," *IEEE Transactions on Systems, Man and Cybernetics*, pp. 1489-1510, 1989.

[128] C. Tomasi and B. Stan, "Multiway Cut for Stereo and Motion with Slanted Surfaces," in *In International Conference on Computer Vision*, 1999, pp. 489--495.

[129] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," in *International Journal of Computer Vision, 60, 2*, 2004, pp. 91-110.

[130] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63-86, 2004.

[131] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, 2005.

[132] P. Roebrock and K. Boehnke, "Offline Path Correction System for Industrial Robots," in *Proceedings of the 9th WSEAS International Conference on Automatic Control, Modeling & Simulation*, 2007, pp. 276-280.

[133] R. J. Campbell, "Recognition of free-form three-dimensional objects in range data using global and local features," PhD Thesis, The Ohio State University, 2001.

[134] M. Ulrich and C. T. Steger, "Performance Comparison of 2D Object Recognition Techniques," in *Photogrammetric Computer Vision*, 2002, p. 368.

[135] K. Boehnke, M. Otesteanu, P. Roebrock, W. Winkler, and W. Neddermeyer, "An industrial laser scanning system for robotic bin picking," in *8th Conference on Optical 3-D Measurement Techniques*, 2007, pp. 158-164.

[136] E. Batlle, C. Matabosch, and J. Salvi, "Summarizing Image/Surface

Registration for 6DOF Robot/Camera Pose Estimation.," in *IbPRIA (2)*, vol. 4478, 2007, pp. 105-112.

[137] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. IEEE International Conference on Robotics and Automation*, 1991, pp. 2724-2729.

[138] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1994, pp. 311-318.

[139] T. Masuda, K. Sakaue, and N. Yokoya, "Registration and Integration of Multiple Range Images for 3-D Model Construction," in *ICPR '96: Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume I*, Washington, DC, USA, 1996, p. 879.

[140] G. Godin, D. Laurendeau, and R. Bergevin, "A Method for the Registration of Attributed Range Images," *Third International Conference on 3-D Digital Imaging and Modeling*, vol. 0, p. 179, 2001.

[141] A. Hoover et al., "An Experimental Comparison of Range Image Segmentation Algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 7, pp. 673-689, 1996.

[142] J. Feldmar, G. Malandain, J. Declerck, and N. Ayache, "Extension of the ICP Algorithm to Non-Rigid Intensity-Based Registration of 3D Volumes," in *MMBIA '96: Proceedings of the 1996 Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA '96)*, Washington, DC, USA, 1996, p. 84.

[143] C. V. Stewart, C.-L. Tsai, and B. Roysam, "The Dual Bootstrap Iterative Closest Point Algorithm with Application to Retinal Image Registration," *IEEE Trans. Med. Imaging*, vol. 22, no. 11, pp. 1379-1394, 2003.

[144] P. Yan and K. W. Bowyer, "A fast algorithm for ICP-based 3D shape biometrics," *Comput. Vis. Image Underst.*, vol. 107, no. 3, pp. 195-202, 2007.

[145] J. B. Maintz and M. A. Viergever, *A Survey of Medical Image Registration*,.: Department of Information and Computing Sciences, Utrecht University, 1998.

[146] A. K. Jain, Y. Chen, and M. Demirkus, "Pores and Ridges: High-Resolution Fingerprint Matching Using Level 3 Features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 15-27, 2007.

[147] J. S. Gutmann and C. Schlegel, "AMOS: comparison of scan matching

approaches for self-localization in indoor environments," *1st Euromicro Workshop on Advanced Mobile Robots, Eurobot*, vol. 00, pp. 61-67, 1996.

[148] R. Burgard and W. Triebel, "Improving Simultaneous Mapping and Localization," in *Proceedings of the National Conference on Artificial Intelligence*, London, 2005, pp. 1330-1335.

[149] D. Borrmann, J. Elseberg, K. Lingemann, A. Nuechter, and J. Hertzberg, "The Efficient Extension of Globally Consistent Scan Matching to 6 DoF," in *Proc. 4th Internatl. Symp. 3D Data Processing, Visualization and Transmission (3DPVT)*, 2008.

[150] M. Levoy et al., "The Digital Michelangelo Project: 3D Scanning of Large Statues," in *Proceedings of ACM SIGGRAPH 2000*, jul 2000, pp. 131-144.

[151] D. Huber, "Automatic Three-dimensional Modeling from Reality," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Ph.D. Thesis December 2002.

[152] A. Lorusso, D. W. Eggert, and R. B. Fisher, "A comparison of four algorithms for estimating 3-D rigid transformations," in *BMVC '95: Proceedings of the 1995 British conference on Machine vision (Vol. 1)*, Surrey, UK, UK, 1995, pp. 237-246.

[153] F. Sanso, "An Exact Solution to the Roto-Translation Problem," *Photogrammetria, journal of the International Society for Photogrammetry*, vol. 29, pp. 203-216, 1973.

[154] T. Jost, "Fast Geometric Matching for Shape Registration," PhD thesis, University of Neuchâtel, 2003.

[155] C. Schuetz, "Geometric point matching of free-form 3D objects," University of Neuchâtel, Ph.D. Thesis 1998.

[156] G. Dalley and P. Flynn, "Pair-wise range image registration: A study in outlier classification," in *Computer Vision and Image Understanding*, 2003, pp. 104-115.

[157] S. Weik, "Registration of 3-D partial surface models using luminance and depth information," in *NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, Washington, DC, USA, 1997, p. 93.

[158] G. Godin, M. Rioux, and R. Baribeau, "Three-dimensional registration using range and intensity information," *Videometrics III*, vol. 2350, no. 1, pp. 279-290, 1994.

[159] G. C. Sharp, S. W. Lee, and D. K. Wehe, "ICP Registration Using Invariant Features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 90-102, 2002.

[160] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vision*, vol. 13, no. 2, pp. 119-152, 1994.

[161] A. Nuechter, K. Lingemann, and J. Hertzberg, "Cached k-d tree search for ICP algorithms," in *3DIM '07: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling*, Washington, DC, USA, 2007, pp. 419-426.

[162] M. Greenspan and M. Yurick, "Approximate K-D Tree Search for Efficient ICP," *Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM '03)*, vol. 0, p. 442, 2003.

[163] T. Zinsser, J. Schmidt, and H. Niemann, "Performance Analysis of Nearest Neighbor Algorithms for ICP Registration of 3-D Point Sets," in *Vision, Modeling, and Visualization 2003*, Munich, Germany, 2003, pp. 199-206.

[164] M. Greenspan and G. Godin, "A Nearest Neighbor Method for Efficient ICP," *3rd International Conference on 3D Digital Imaging and Modeling*, vol. 0, p. 161, 2001.

[165] G. Blais and M. D. Levine, "Registering multiview range data to create 3D computer objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 820-824, Aug. 1995.

[166] K.-L. Low, *Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration*,.: Technical Report TR04-004, Department of Computer Science, University of North Carolina at Chapel Hill, 2004.

[167] C. Dorai, G. Wang, A. K. Jain, and C. Mercer, "Registration and Integration of Multiple Object Views for 3D Model Construction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 83-89, 1998.

[168] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The Trimmed Iterative Closest Point Algorithm," in *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3*, Washington, DC, USA, 2002, p. 30545.

[169] D. Akca, "Least squares 3D surface matching," Inst. für Geodäsie und Photogrammetrie, Zürich, Ph.D. Thesis 2007.

[170] D. W. Eggert, A. W. Fitzgibbon, and R. B. Fisher, "Simultaneous registration of multiple range views for use in reverse engineering of

CAD models," *Comput. Vis. Image Underst.*, vol. 69, no. 3, pp. 253-272, 1998.

[171] A. J. Stoddart and A. Hilton, "Registration of Multiple Point Sets," in *ICPR '96: Proceedings of the 13th International Conference on Pattern Recognition*, Washington, DC, USA, 1996, p. 40.

[172] S.-Y. Park and M. Subbarao, "An accurate and fast point-to-plane registration technique," *Pattern Recogn. Lett.*, vol. 24, no. 16, pp. 2967-2976, 2003.

[173] S. Seeger, X. Laboureux, and B. Glomann, *Comparision of ICP-algorithms: First Results*,.: Technical Report Chair for Optics, Friedrich-Alexander-Universität Erlangen, 2000.

[174] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, "Geometrically Stable Sampling for the ICP Algorithm," in *Fourth International Conference on 3-D Digital Imaging and Modeling*, 2003, pp. 260- 267.

[175] S. Rusinkiewicz. (2008, Sep.) Szymon Rusinkiewicz. [Online]. http://www.cs.princeton.edu/~smr/papers/icpstability.pdf

[176] P. J. Neugebauer, "Geometrical cloning of 3D objects via simultaneous registration of multiple range images," in *SMA '97: Proceedings of the 1997 International Conference on Shape Modeling and Applications (SMA '97)*, Washington, DC, USA, 1997, p. 130.

[177] H. Pottmann, S. Leopoldseder, and M. Hofer, "Registration without ICP," *Comput. Vis. Image Underst.*, vol. 95, no. 1, pp. 54-71, 2004.

[178] E. Trucco, A. Fusiello, and V. Roberto, "Robust motion and correspondence of noisy 3-D point sets with missing data," *Pattern Recogn. Lett.*, vol. 20, no. 9, pp. 889-898, 1999.

[179] T. Zinsser, J. Schmidt, and H. Niemann, "A refined ICP algorithm for robust 3-D correspondence estimation," *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, p. 695, Sept. 2003.

[180] S. L. Tanimoto and T. Pavlidis, "A Hierarchical Data Structure for Picture Processing," *Computer Graphics and Image Processing*, vol. 4, no. 2, pp. 104-119, June 1975.

[181] M. Garland, "Multiresolution Modeling: Survey & future opportunities," in *Eurographics '99, State of the Art Report*, 1999.

[182] P. Cignoni, C. Montani, and R. Scopigno, "A comparison of mesh simplification algorithms," *Computers & Graphics*, vol. 22, no. 1, pp.

37-54, 1998.

[183] H. Hoppe, "Progressive meshes," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 99-108.

[184] J. L. Peng, C. S. Kim, and C. C. Kuo, "Technologies for 3D mesh compression: A survey," *Journal of Visual Communication and Image Representation*, vol. 16, no. 6, pp. 688-733, December 2005.

[185] W. Cheng, W. T. Ooi, S. Mondet, R. Grigoras, and G. Morin, "An analytical model for progressive mesh streaming," in *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, New York, NY, USA, 2007, pp. 737-746.

[186] R. Pajarola and J. Rossignac, "Compressed Progressive Meshes," *IEEE Transactions on Visualization and Computer Graphics*, pp. 79-93, 2000.

[187] L. Ikemoto, N. Gelfand, and M. Levoy, "A Hierarchical Method for Aligning Warped Meshes," in *Proc. 4th International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2003, pp. 434-441.

[188] X. Chen, L. Zhang, R. Tong, and J. Dong, "Multi-resolution-based mesh registration," in *Proc. 8th International Conference on Computer Supported Cooperative Work in Design*, vol. 1, 2004, pp. 88-93.

[189] K.-L. Low and A. Lastra, "Reliable and Rapidly-Converging ICP Algorithm Using Multiresolution Smoothing," *Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM '03)*, vol. 0, p. 171, 2003.

[190] Wikipedia: The Free Encyclopedia. (2008, Aug.) Software rendering. [Online]. http://en.wikipedia.org/wiki/Software_rendering

[191] I. Wald, "Realtime Ray Tracing and Interactive Global Illumination," Computer Graphics Group, Saarland University, Ph.D. Thesis 2004.

[192] H. Hoppe, "Efficient implementation of progressive meshes," *Computers & Graphics*, vol. 22, no. 1, pp. 27-36, 1998.

[193] K. Boehnke and M. Otesteanu, "Progressive Mesh based Iterative Closest Points for Robotic Bin Picking," in *Proceedings of the International Conference on Informatics in Control, Automation and Robotics*, 2008, pp. 469-473.

[194] S. Rusinkiewicz. (2008) trimesh2. [Online]. http://www.cs.princeton.edu/gfx/proj/trimesh2/

[195] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vision Comput.*, vol. 25, no. 5, pp. 578-596, 2007.

[196] B. N. Delaunay, "Sur la Sphère Vide," *Bulletin of Academy of Sciences of the USSR*, vol. 7, pp. 793-800, 1934.

[197] K. Boehnke and O. Marius, "Progressive Mesh Object Registration," in *Proceedings of IEEE/SICE International Symposium on System Integration*, Nagoya, 2008, pp. 22-28.

[198] S. J. Cunnington and A. J. Stoddart, "N-View Point Set Registration: A Comparison," in *Proceedings of the British Machine Vision Conference*, 1999.

[199] O. Jokinen and H. Haggren, "Statistical analysis of two 3-D registration and modeling strategies," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 53, pp. 320-341, December 1998.

[200] L. Matthies and S. Shafer, "Error Modeling in Stereo Navigation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 3, pp. 239-250, June 1987.

x

Titluri recent publicate în colecția „TEZE DE DOCTORAT"
seria 7: Inginerie Electronică și Telecomunicații

1. **Adrian Lazăr Șchiop** – *Contribuții la studiul convertoarelor utilizate la acționarea motoarelor sincrone, ISBN 978-973-625-409-3,* 2007;

2. **Ioan Gavriluț** – *Contribuții la navigația roboților mobili autonomi utilizând rețelele neuronale celulare, ISBN 9789-973-625-417-8,* 2007;

3. **Marian Constantin Bucos** – *Dezvoltarea sistemelor informatice pentru e-learning și realizarea de organizații educaționale virtuale, ISBN 978-973-625-560-1,* 2007;

4. **Horia – Gheorghe Baltă** – *Contribuții la dezvoltarea și proiectarea turbo-codurilor binare și nebinare, ISBN 978-973-625-601-1,* 2008;

5. **Marin Titus Tomșe** – *Contribuții la studiul teoretic și experimental al surselor de alimentare pentru cuptoarele de încălzire inductivă, ISBN 978-973-625-608-0,* 2008

6. **Radu Dan Mihăescu** – *Concepția unor surse de curent de referință pentru circuite integrate CMOS, ISBN 978-973-625-707-0,* (2008);

7. **Raul Ciprian Ionel** – *Contribuții la localizarea surselor de zgomot utilizând instrumentație virtuală, ISBN 978-973-625-746-9,* (2008).

8. **Corina-Alda Naforniță** – *Contribuții la marcarea transparentă a imaginilor în domeniul transformatei wavelet, ISBN 978-973-625-774-2,* 2009;

9. **Ciprian David** – *Détection d'hétérogénéités lineaires dans les textures directionnelles – application à la detection de failles en sismique de reflexion, ISBN 978-973-625-776-6,* (2008).

EDITURA POLITEHNICA