FAST OBJECT LOCALIZATION WITH REAL TIME 3D LASER RANGE SENSOR SIMULATION

Kay Boehnke
Department of Electrical Engineering, University of Cooperative Education, Coblitzweg 1-7, 68163 Mannheim
Kay.Boehnke@gmail.com

Keywords: 3D laser range sensors, Real time object localization, Iterative Closest Points, Surface registration, Progressive Mesh, Robotic bin picking

Abstract: 3D laser range sensors provide range data of objects in a scene. Most of the known approaches for object recognition and localization in range data try to extract features and determine the position of the object from this information. In opposite we create a virtual scene, make a virtual range image with a virtual 3D laser range sensor and compare the result with range data of the real 3D laser range sensor. This idea is integrated in a system for automated robotic bin picking. Our simulation process is combined with a hierarchical registration process using the iterative closest point algorithm and Progressive meshes to refine our simulation results.

1. INTRODUCTION

Object recognition and object localization has a long history in two dimensional image processing[1],[2]. Due to the lack of the third dimension the position of an object in the scene can not be fully determined. By using data provided by laser range sensors our approach is able to find objects in three dimensional (3D) scenes. There exist many advantages to use range data provided by 3D laser range sensors. We achieve a better accuracy of the object poses compared to stereo camera solutions. Many of the known solutions are limited to simple shaped objects or objects with specific features. But in many industrial automation processes the handling of objects with complex shapes without any specific features is still an unsolved problem in the field of robotic automation. [3],[4].
Object recognition and pose estimation is used in industrial fields like depalletizing or robotic bin picking. Almost all processes for robotic bin picking can be divided into different steps. First of all a visual capture device takes a picture of the scene. After that an algorithm has to recognize and localize the objects in that representation of the scene. This is the most important component of the bin picking process. The position and pose are transformed into scene coordinates and afterwards transferred to the robot. The robot picks the object considering adequate grasp points and possible collision points with the surrounding environment. The object must be guided to the target position, where the object has to be placed. These steps are repeated iteratively for each object in the scene.
As mentioned above the object localization step is the most challenging step in the whole process. In 1991 Lowe[5] projects 3D-Models in the image plane and compares them with the image of the scene to estimate the object pose. We extend this approach by comparing range images. We simulate industrial laser range sensors to transfer cad aided design models to a scene representation.
As we compare the shape appearance of every single object in the simulated scene with the real scene, we are able to handle nearly all kind of objects without any feature extraction. With new computational improvements in computer hardware like parallel computing on Multi-Core processors and graphical

processing units (GPU) our simulation will overcome many performance problems and lead to a better accuracy and robustness of the whole system.
Our second contribution is the possibility to adjust the process time of the system. We implement a flexible coarse-to-fine algorithm by changing the threshold for the needed accuracy.
We give a short overview of our system in section 2. The coarse pose estimation is introduced in section 3. We describe the laser sensor simulation, the comparison of the virtual representation and the real sensor image. Section 4 shows the modified Iterative Closest Point (ICP) algorithm to refine the coarse solutions to increase the accuracy. After that we conclude with upcoming extensions of our approach.

2. SYSTEM OVERVIEW

Figure 1 shows an overview of our object localization system. We use a two step object localization approach. The pose estimation provides a number of the best position candidates to the refinement step. This pre-selection helps to decrease the number of candidates for the refinement process and leads to a reduction of high computational costs. This hierarchical coarse-to-fine object localization is related to hypotheses and verification approaches like [5].

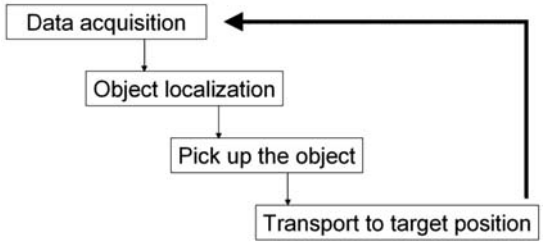

Fig. 1. After the data acquisition follows the object localization. When the object position is known, the robot picks up the object and transports it to the target position.

In the pose estimation step we compare the input data from range sensors with our simulated range data from the "object pose simulation". This "object pose simulation" includes the main contribution to our system: the virtual sensor. This virtual sensor delivers a virtual range image of the objects pose for our object pose database. Every virtual range image is compared during the "comparison" step with the real range data in the scene. The best matching poses are transferred to the refinement step. The refinement step uses a modified Iterative Closest Point (ICP) algorithm combined with hierarchical level of detail (LOD) representations of our object candidates. The next chapters will further explain the steps of our proposed system.

3. POSE ESTIMATION

The purpose of the object pose estimation is to find adequate coarse positions of an object in the scene and can be divided into three parts. At first the scene is acquired by the laser range sensor in the data acquisition step (see figure 1). The Object pose simulation delivers the virtually scanned scene which is compared with the real scene.

3.1 Data acquisition

One of our contributions is that we take features of real range sensors into consideration to adapt the simulated range sensor to real range sensors. Therefore this chapter will shortly introduce the data acquisition with industrial range sensors.
Non-contact visual range measurement methods are proportionally fast and efficient. They allow us to obtain information about substances which may be hot, chemically aggressive, sticky or sensitive, provided that sufficient light is reflected from the surface.
Visual data sources in the industrial environment can be separated into active and passive sources. The most common passive data sources for industrial applications are still camera based systems. Cameras provide a 2-dimensional projection of a scene without depth information. With the help of two or more cameras and the known transformation between these cameras distance values can be determined[6]. Other active sensor solutions use defined light patterns which are projected to the scene. Unfortunately many camera based solutions suffers from surrounding lightning conditions and have a lower accuracy[4].
At the current state-of-the-art active visual data sources like non- contact industrial laser range sensors are superior to other industrial measurement methods regarding their accuracy, costs and robustness [4],[7]. Additionally active methods produce dense sampling points compared to passive methods. Two major principles of laser based distance measuring methods are used in industrial applications: Triangulation and time-of-flight (TOF). For active triangulation, 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. More or less TOF and phase measurement methods are long range technologies (over 1.0 meter) and triangulation based methods belong to close range methods.

3.1.1 Triangulation based sensors

The principle of triangulation bases on simple geometrical constraints. In the case of a laser triangulation sensor an active triangulation system contains a light source and a receiving unit (usually a CCD- or CMOS camera). A laser diode emits a laser beam with a defined angle towards the object. The surface of this object reflects this beam to the receiver. The base length between laser source and the receiving unit is known from calibration. The distance from the object to the instrument is geometrically determined from the recorded angle and base length.

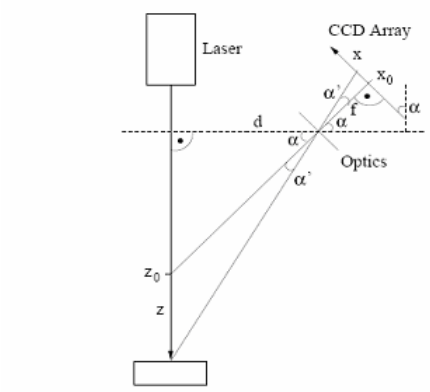

Fig. 2. With the triangulation the distance depends only on the angle *a* given through *x* and x_0.

Figure 2 shows an arrangement for a one dimensional distance measurement. The beam is reflected to a defined position on the CCD- Array. With known extrinsic parameters of the arrangement the distance to the object z can be calculated with the following equation.

$$z = d \cdot \frac{f \cdot \tan(\alpha) + x}{f - x \cdot \tan(\alpha)} - z_0 \qquad (1)$$

All variables (like the focal length f, the distance d and the angle a between the laser source and the camera) of the arrangement are known. Smaller angles leads to bad sampling and large angles lead to occlusions, so the angle of emitted beams usually ranges from 15° to 45°. The distance between the object and the sensor z mainly depends on the position x where the reflected beam intersects the CCD array. This principle is extended to two dimensions by using a CCD- or CMOS camera instead of an array and the laser beam is split by lens optics to a laser line.

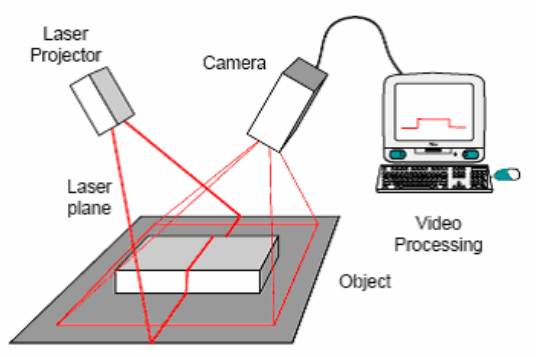

Fig. 3. The laser line extends the triangulation to a 2D contour recognition. Moving the object orthogonally to the laser line while recording provide a 2.5D surface representation.

The accuracy (usually ~1:1000) depends on the distance between laser and receiving unit and the object distance. Active triangulation is usually used in measuring range of 0.1 to 5.0 meters. For larger distances the distance between laser and receiving unit must be increased, in turn this leads to a vulnerability of occlusions.

3.1.2 Time of flight sensors

Time-of-flight (TOF) sensors send out a light beam towards an object. The time that light needs to travel from the laser diode to the object surface and back is measured. The existing principles of TOF can be separated into the following categories[7]:

- Pulsed time-of flight
- Continuous wave phase shift measurement

A pulsed time-of flight sensor emits a light pulse and starts high accuracy stopwatch. The light pulse travels to the target, is diffusely reflected by the surface of the object and a part of the light returns to the receiver. When the light pulse arrives, the stopwatch is stopped in order to determine the time of flight. With the known speed of light the distance to the object is determined.
In the case of pulsed TOF the travel time is directly proportional to the distance, like shown in the following equation:

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

The constant c is the velocity of light and Δt is the time needed by the signal to travel from the source to the object and back. The equation contains the refraction index n of the involved medium ($n_{air} \sim 1$) and the factor ½ for the way to the object and back.
The second method for TOF distance measuring is the measuring of the phase shift. This method measures the difference between emitted and received signal. A continuous wave laser emits light continuously with a modulated phase.
The distance information is extracted from the received signal by comparing its modulation phase with that of the emitted signal. The distance can be calculated with the equation:

$$d = \frac{\lambda_m}{4\Pi n}\Delta\varphi \qquad (3)$$

The range of these TOF sensors depends on wavelength of the modulated signal so the distance resolution of these sensors can be increased. A short wavelength leads to a smaller maximum range[7]. The quality of the received signal must have an adequate level to calculate a valid distance. In some field of application this leads to invalid results, because of reflection properties of the object or different external interferences.
Phase shift measurement has a higher precision than the conventional TOF measuring, so a combination of these two procedures is usually used. The higher precision is achieved by using modulation with a variable frequency or using more than one frequency simultaneously (Frequency modulated continuous wave). Another possibility is an amplitude modulated continuous wave (AMCW) modulated in amplitude by varying the power.
One of the major advantages of TOF is that it is free of corresponding problems of passive triangulation and the range ambiguities of the passive triangulation. In most existing arrangements the active light source and the receiver are located very closely to each other. So illumination and observation directions are approximately collinear to avoid occlusions.
Compared to Triangulation based sensors TOF sensors have advantages regarding accuracy and resolution in measurement ranges up to 100m and do not suffer from occlusions. Theoretically the accuracy of the depth measuring is independent to the distance of the object of the camera and only depends on the precision of measuring the travel time. Precision in sub-millimeter range requires pulse lengths of a few tens of picoseconds and the associated electronics (at least with over 100Ghz). Mainly the pulse rate and the amount of reflected photons which reach the detector influence the maximum range for TOF sensors. Some long range sensors use the pulsed TOF method to measure distance up to a few kilometers for cartographical mapping [8]. But in our case of applications they are less accurate especially at close range setups. The accuracy is between some millimeters and two or three centimeters, depending on the distance between the object and the sensor, because of the lack of electronical realization for high accuracy time measurement at the moment.

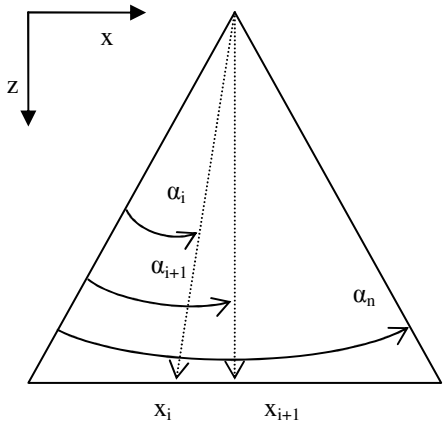

Fig. 4. For every angle step a light beam is emitted along the x axis.

To create a two dimensional distance contour common TOF laser sensors (like Sick LMS sensors) redirect the beam in defined angle steps. This can be achieved by a rotation mirror inside the sensor. For every emitted and received beam the mirror is rotated an angle step. This principle is shown in figure 4. With a parameterized scanning frequency a distance for every angle step is collected and results in an array of distance values along a line.

3.1.3 Creating 3D information

Most of the industrial laser range sensors deliver a two-dimensional distance contour. To get a three dimensional representation of the scene the laser range sensor must be moved over the real scene preferably in a linear way [9]. This step is often called the scanning process (refer to fig. 3). The sensor moves from the start point to the end point with specific incremented steps. The step width is connected with the scan frequency of the sensor and depends also on the properties of the mechanical

hardware (i.e. resolution of linear axis encoder units). In industrial applications the step width is often fixed due to the process cycle time. Beside the number of scans and scan frequency the step width is directly proportional to the data acquisition time of the whole process.
Moreover, the step width in the sensor simulation influences the simulation time significantly, which is shown in the next section.

3.2 Object pose simulation

The object pose simulation creates a virtual range image (VRI) with help of a simulated sensor and a virtual scene.

3.2.1 Sensor simulation

The sensor model for the simulation adopts all properties of the real sensors introduced in the previous chapter. Therefore, the properties of the scanning process of the objects in the scene must also be known. Beside parameters and properties of the sensors itself, this mainly includes the distance between ground and sensor and the direction of the scanning process.
The simulation of laser sensors in the computer is related to the field of computer graphics. We simulate a TOF sensor by using commonly known algorithms and principles (like Raytracing) of the so called virtual reality. Virtual scenes are often used in computer games. This gives us the chance to use all hardware accelerations and ready to use programming libraries like Microsoft ©DirectX or ©OpenGL.
In most workspaces like defined in Microsoft ©DirectX or ©OpenGL in the research field of computer graphics the camera workspace is defined in a right-handed coordinate system. The distance to an object in an image is aligned to the z coordinate axis[18]. Simulating a TOF sensor, the sensor model unifies the light source and the model of a perfect pinhole camera. Therefore we have no lens or the sending aperture with a size greater than zero. The sensor is reduced to a theoretical point in the 3D space. In this case every other point in the workspace forms exactly one line with the camera.
In virtual scenes beams of the light source are called rays. Creating rays like a real TOF sensor and tracing them to their intersection with the object is one major principle of raytracing. We reduce the known algorithms to a simple distance calculation. The resolution of resulting depth image/distance field defines the number of beams starting in camera point of view.

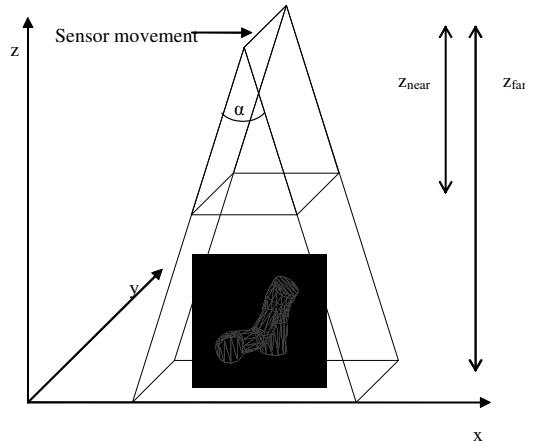

Fig. 5. The virtual scene is given by the frustum of the virtual TOF sensor. The movement in y enables the scanning process to provide 2.5D surface data.

As shown in figure 5 the distance between camera and the scene Z_{near} is define by the sensor position and the object space (frustum) in the virtual coordinate system. The distance Z_{far} between the camera and the maximum distance point of the scene can be set to maximum range of the real laser sensor. According to the principle of real laser distance sensors one scanline is separated into angle steps along the scan direction x like already shown in figure 4. 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 i the normalized direction vector for x_i is calculated with the following equation:

$$v_{DIR(x_i)} = \begin{bmatrix} z_{far} * \cos(\alpha) \\ y_j \\ z_{far} \end{bmatrix} \quad (4)$$

The y coordinate is fixed to the position of the sensor in the scene and is constant for one scan line. For every scan line a spread of rays are calculated depending on the needed resolution in x. Every ray is tested for intersection with the object in scene. To describe our scene in the virtual scene our objects must be modelled in a defined representation.

3.2.2 Virtual scene object representation

The object model is a mathematical description of a three dimensional object. The model can be described with different defined data representation (i.e. Boundary representations, polygonal meshes or

implicit NURBS). The most intuitive representation for object models is polygonal mesh. This data structure contains mainly the geometry and the texture information of the object model. Geometries of polygonal meshes are a collection of unstructured simple polygons like triangles or quadrilaterals. One common form of polygonal meshes is the triangle mesh. Three points in the model coordinate space (vertices) 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. The sensor simulation in this work uses data structures for graphics processing units (GPU) which are encapsulated in the libraries of DirectX© or OpenGL©. These libraries provide interfaces to import vertices. The intersection tests of the sensor simulation can be reduced to a simple ray-triangle intersection test by dividing the object into single triangles. So the simplicity of intersection tests is a further advantage of this object model representation.

3.2.3 Virtual Range Image creation

To create a Virtual Range Image(VRI) every ray of the virtual sensor has to be intersected with every triangle of the scene. We use efficient variants of ray-triangle intersection tests. To compare the performance of our implementation, we implemented a slightly modified intersection test based on the work of Moeller and Trumbore [10]. They proposed an algorithm for fast ray/triangle intersection test. Their main contribution is that they do not require the calculation of the plane equation. Instead they use a series of transformations to translate the triangle to the origin and to transform it to a unit right angled triangle.
We use a bounding box or bounding volume to decide whether the ray shoots in the direction of the model or not. The bounding volume is aligned with the axis of the model coordinate system. Due to this axis aligned bounding box (AABB) the intersection test integration is very efficient. We convert the bounding box into a triangle list to ensure the compatibility to the ray-mesh intersection function. After that we test the intersection of the rays, which hit the boundary intersection test, and model triangles. This intersection test is made by a DirectX function. The performance is about 200 times faster than the reference algorithm[10], which must be traced back to the fact, that the DirectX function is heavily optimized and uses the computational power of the GPU. This tremendous performance boost allows us to do our sensor simulation in an adequate time.
As already mentioned above range sensors which deliver a distance profile are moved over the object in order to get distance maps[9]. This is also necessary for the sensor simulation. To compare the results from simulation to the real images the resolution of the data in moving direction should be similar. To acquire a full three dimensional distance image, the sensor model is moved virtually in the direction of y as shown in figure 5. The step width depends on the application requirements. To ensure the compatibility between the real sensor setup and the sensor simulation, the resolution in y is connected to the resolution of hardware encoders in different applications.

3.3 Comparison

The sensor simulation results in a virtual range image (VRI). If there are many objects in the real scene, we choose a simple solution in order to compare the VRI to the real range image(RRI). We simulate only one object of the scene in its possible position and pose and compare these VRIs with the RRI. The process of our initial application robotic bin picking allows us to reduce the object recognition and localization to the search for the best candidate. Therefore we do not need to simulate all objects in the scene. Figure 6 illustrates the idea of the comparison.

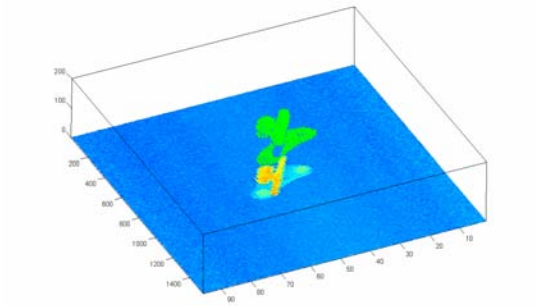

Fig. 6. The simulated door joint is compared to the real scene data in the comparison step.

In the pose estimation step we try to reduce the number of possible numbers of candidates and select the best candidate in the pose refinement step. Therefore we compare every possible VRI with the RRI. To decrease the computational costs we create all possible VRI's within fixed increment of ΔX,

ΔY, ΔZ and angles of one object offline before and store them in a database. The density of positions and the rotational degree of freedom of the object to create a VRI is the most important parameter to adjust the accuracy and the performance in our system. Different VRI's for different kind of objects are compared with the RRI in the same way. So the object classification is integrated in the step of object localization assuming we can separate the appearance of the different objects in the real scene. Every VRI is compared with the RRI with a defined error function returning an error value. If the error value is low the VRI matches with the RRI. Because of the fact, that all VRI are compared to the RRI, each VRI gets an error value. The error value is a scalar specifying the level of the correlation between VRI and RRI.

$$Error = \frac{1}{N} \sum_{i=0}^{X} \sum_{j=0}^{Y} \left| Z_1(i,j) - Z_2(i,j) \right| \qquad (5)$$

The error 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 within the area of the simulated object distance values.
This error function provides a rate of how good the pose of the model matches with the real image in the distance measurement. Often we can reject degrees-of-freedom of the object mainly due to a-priori knowledge of the object position in the scene or appearance. To increase the performance in special application like shown in figure 6 the comparison can be reduced to a two dimensional search. This provides the possibility to use well known and highly optimized algorithms of signal and image processing. To measure the correlation between two signals usually the cross correlation is used in image processing. Taking the mean of the signal into consideration we use the normalized cross correlation (NCC)

$$NCC(x,y) = \frac{\sum_{x,y} \left(Z_1(x,y) - \bar{Z}_1\right) \cdot \left(Z_2(x-u, y-v) - \bar{Z}_2\right)}{\sqrt{\sum_{x,y} \left(Z_1(x,y) - \bar{Z}_1\right)^2 \cdot \left(Z_2(x-u, y-v) - \bar{Z}_2\right)^2}} \qquad (7)$$

Due to the fact that the cross correlation is related to the convolution process the calculation can be done in the frequency domain, to increase the speed of the calculation process. The resulting cross correlation coefficient is high at this position where the pattern fit with the image.

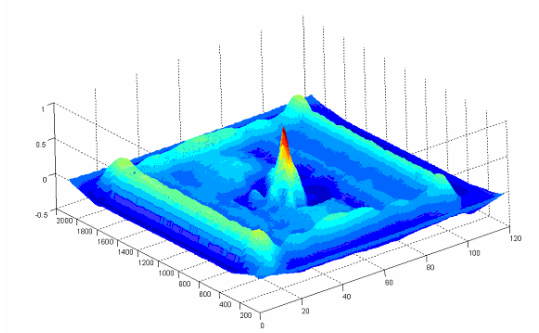

Fig. 7. The maximum of the normalized cross correlation indicate the best matching position of the virtual range image in the scene.

Figure 7 shows the peak in the cross correlation "surface" at the positions of the best match. This test is made for one object in the scene and one pattern with fixed distance and fixed rotation to the sensor (see figure 6). This shows the advantage of cross correlation calculation: The high performance of the frequency domain calculation for large pattern and images. The two dimensional cross correlation can be extended to an n-dimensional cross correlation to deal with all degrees of freedom.
The simulated object poses with the lowest error values are selected for pose refinement. The number of best matching VRI's can be limited by an error-threshold or a fixed number of VRI candidates or a combination. We decided to use a combination of both: As we know the maximum number of simulated object poses, we select the candidates within the best 10% of all error values in the coarse pose estimation process. This seems to be a good initial percentage proven in our experiments. All of these poses must be under an error threshold, which depends on the complexity of the simulated object, the used sensor (with its sensor errors), outliers and invalid points. Due to this the error threshold could not be fixed for every application.
In our experiments this pre-selection results in 10-15 VRI candidates in the application. These VRI candidates are delivered to the pose refinement process, starting with the best matching candidate.
The process for our 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

4. POSE REFINEMENT

The task of the pose refinement is to find an exact matching pose between the object in the scene and the simulated object. The best VRI candidates were chosen regarding their error value for every pose. The pose refinement in our case is very similar to the process of range image registration medicine, face recognition and many other fields of research[11]. The registration process aligns a representation of an object to another pose of this object and determines the transformation between them. We use an iterative registration algorithm to find the exact position for all VRI candidates. Due to the fact that the chosen iterative Closest Point algorithm is computational especially for large point datasets, we combine the ICP with a hierarchical object representation in the pose refinement step. Therefore we use Progressive Meshes introduced by Hoppe [12] in 1996. We call this combination Progressive Mesh Iterative Closest Point Algorithm (PMICP).

4.1 Iterative Closest Points

The most commonly used algorithm for the determination of rigid transformations is the Iterative Closest Point algorithm (ICP) [13],[14] [15]. Real time implementations [15] show the potential of the ICP-Algorithm for small point datasets. As the name already implies, the ICP is an iterative algorithm which transforms the view of an object to another view of the same object or another object. The algorithm minimizes the mean square error of the point-to-point[13] or point-to-plane[14] distance in several iteration steps. The algorithm converts monotonously to a local minimum. Therefore an approximate initial solution is important for the success of the method. The model points M and the scene points P consist of three dimensional points with the coordinates x,y,z in the coordinate system of our simulated sensor. Due to the fact that the simulated and real coordinate systems are equal, we get the rigid transformation between the VRI and RRI by minimizing the error E.

$$E = \sum_i \left\| m_i - R(p_i) - t \right\|^2 \qquad (8)$$

The error between the model points and the scene points is extended to the three dimensional Euclidean distance to increase the accuracy of the solution. To find the rotation R and the translation t we use the closed form solution with the help of unit quaternions[16]. Every point of the model (here VRI) is assigned to the Euclidean closest point of the scene (RRI). We use a point-to-point metric according to Besl and McKay[13]. The object coordinates of best VRI candidate is selected and used as the final result in our object localization step.

4.2 Progressive Mesh

To increase the performance of the iterative closest point algorithm we use an implicit hierarchical dataset for the points given by Progressive meshes[12]. The Progressive Mesh consists of triangles defined by three points and edges defined by the line between two points of two adjacent 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 with the original anymore. This data representation provides a highly efficient implementation for adjusting the level of detail in a point dataset and includes an inbuilt noise reduction. To create a Progressive Mesh we triangulate our point clouds with a two dimensional Delaunay Triangulation. 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 accuracy and M_n is the mesh with the highest accuracy.

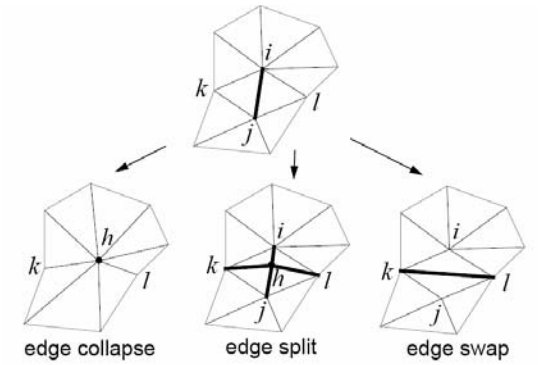

Fig. 8. The three Progressive Mesh operations.

Generating Progressive Meshes means to apply edge collapse transformations to the mesh M_n, which is shown in figure 8. The edge {i,j} is reduced to one point or vertex. The opposite of the edge collapse transformation is the edge split transformation. In this case one triangle is split into two triangles with the vertices ijk and l. Edge swapping is made if an edge collapse will lead to a higher inaccuracy of the mesh representation. The decision can be made with the calculation of an energy function which takes

distance of vertices, their number and a regularization term into consideration (See [17] for further details). The complete process is described in the following equation:

$$M_0 \underset{ecoll_0}{\overset{esplit_0}{\rightleftarrows}} M_1 \underset{ecoll_1}{\overset{esplit_1}{\rightleftarrows}} \ldots \underset{ecoll_{n-1}}{\overset{esplit_{n-1}}{\rightleftarrows}} M_n \quad (8)$$

Every single step Mi is restored to recover the mesh of this step. So every mesh Mi with the needed accuracy can be retrieved in a very efficient way.

4.3 The combination PMICP

The ICP algorithm suffers from two major problems. First, the performance is poor if the ICP has to calculate with a huge amount of points in scene and model. Reducing the number of points decreases the complexity dramatically. In every iteration step all points of the two meshes must be compared to each other with a complexity of O(n*m) where n is the number of scene points and m is the number of model points. We reduce this complexity by comparing only the M_i representations of each mesh. The ICP starts with only a few points and increases the number in every iteration step. The computational complexity is reduced in average to O((0.5*n)*(0.5*m)) assuming we do not stop the iteration until we reach the end (M_n). If the iteration process is stopped, because the ICP reached the minimum, the performance of our implementation is always better than O((0.5*n)*(0.5*m)). So our Progressive Mesh ICP needs in average 25% of the time of the standard ICP implementation using a linear ΔM_i.

The second problem of the ICP Algorithm is given by its sensitivity to outliers[15]. With our combination PMICP we can increase the robustness against outliers. By reducing the Level of Detail (LOD) of the mesh up to M_0 all outliers are automatically removed from the object representation of the scene and the model. The M_0 shape of the model and the scene representation is a single triangle. Because of this initial pose transformation are not influenced by outliers. The performance of our pose refinement is shown in our prior experiments [18]. It is shown that results naturally depend on the number of points, error in the datasets and their initial pose. This combination increase robustness and convergence performance over many types of datasets.

5 CONCLUSIONS

We described a system which uses a two step object localization layout. The first step we focused on uses a model-based scalable hierarchical system without the need of segmentation or feature extraction to find a coarse pose of our object in the scene. We introduced the sensor simulation for object pose estimation using well known algorithms from the field of computer graphics to increase the performance. The pose estimation results in a number of pose candidates. These candidates are matched with an improved ICP Algorithm in the following refinement step.

The system will be used in industrial robotic bin picking and includes -beside the 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 described two-step object localization has potential to meet different requirements and cover a high percentage of applications in robotic bin picking. The system does not use any segmentation algorithms, but uses 3D information which is aligned to the input data in a hierarchical system.

Simulating the real world and comparing the appearance of this simulation to the appearance of the real world is a very ambitious goal. Our approach rudimentary includes this general solution and offers many possible extensions. Due to the incremental process object positions can be verified and tracked over all data acquirements of the scene.

The scalability of our approach offers a great potential in the future. Starting with the coarse pose estimation process, we are able to adjust the needed accuracy with simple changes in the position and orientation step width.

The complexity of the object localization also depends on the chosen algorithm and the complexity of the object. The main problem of our algorithm is the high computational cost, if we have very complex objects models and sensors with high resolutions. But depending on the application and the used PC, we can change this computational time-memory-trade off by increasing the number of pre-calculated VRIs in our database. We are not limited to only one object, because we can store as many objects in our database as we want. This is one big advantage of the whole system.

Our approach offers a wide range of possible extensions and improvements. New sensors with higher resolution can be modelled and integrated without any problem. All real range sensors deliver

range images with measurement errors, reflections, and noise[7]. To increase our accuracy we will take these additional features into consideration. The complexity of the ICP algorithm in our refinement step depends mainly on the number of points in the dataset. The search of the closest points has a computational complexity of O((0.5*m)*(0.5*n)) over all iteration steps. This can be reduced using Kd-Trees implementations[19] to O(0.5*log(n)) 0.5*log(m))).
Using Progressive Meshes in the iteration steps of the ICP algorithm offers several methods of adjusting the number of triangles. We used the current iteration counter in the iteration process to connect the level of detail in the meshes. Every iteration step of the ICP algorithm the number of faces in the model mesh and the scene mesh are increased with a define value. Taking the degree of performance of one ICP iteration step into consideration we could adjust the number of triangles in the current Progressive Mesh to the current iteration step error as stated in [18].

ACKNOWLEDGEMENTS

This work was supported in part by the company VMT (Pepperl+Fuchs Group) and University of Cooperative Education in Mannheim/ Germany. The author would like to thank M. Otesteanu, P. Roebrock, M. Kleinkes, W. Neddermeyer, W. Winkler, and K. Lehmann for their support.

REFERENCES

[1] J. Andrade-Cetto and A. C. Kak, “Object recognition”, in *Wiley Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster Ed. New York: John Wiley & Sons, 2000, pp. 449-470.

[2] S. Dickinson, "Object Representation and Recognition", in *What is Cognitive Science?,* E. Lepore and Z. Pylyshyn Ed. Oxford: Basil Blackwell, 1999, pp. 172-207.

[3] M. Hashimoto, K. Sumi, “3d object recognition based on integration of range image and grey-scale image”, *in 1999 Proc. British Machine Vision Conf.*, pp. 253–262

[4] D. Katsoulas, “Robust recovery of piled box-like objects in range images”, Ph.D. dissertation, Dept. Computer Science, Freiburg Univ., Germany, 2004

[5] D.G. Lowe, “Fitting parameterized three-dimensional models to images”, *IEEE Trans. Pattern Anal. and Machine Intell*., vol 13, pp. 441-450, 1991

[6] K. Rahardja and A. Kosaka: “Vision-Based Bin-Picking: Recognition and Localization of Multiple Complex Objects Using Simple Visual Cues”, in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Osaka, pp. 1448-1457, 1996

[7] K. Boehnke, “3D Sensors”, Seminar paper, Univ. Timisoara, Romania, 2006

[8] Fröhlich, C., Mettenleiter, M.: “Terrestrial Laser-Scanning - New Perspectives in 3D-Surveying”, Proceedings of the ISPRS working group VIII/2 Freiburg, 2004

[9] J. Park and G.N. DeSouza, ”3D Modeling of Real-World Objects Using Range and Intensity Images “, in *Innovations in Machine Intelligence and Robot Perception*, S. Patnaik, L.C. Jain, G. Tzafestas and V. Bannore Ed.,New York: Springer-Verlag, 2005

[10] T. Moeller and B. Trumbore, “Fast, Minimum Storage Ray-Triangle Intersection", *Journal of graphics tools*, Vol. 2.1, pp.21-28, 1997

[11] K. Boehnke, “ICP Algorithms”, Seminar paper, Univ. Timisoara, Romania, 2006

[12] Hoppe, H., “Progressive meshes”, *In SIGGRAPH Computer Graphics Proceedings*, pp. 99–108, 1996

[13] J.P. Besl, N.D. McKay, ”A Method for Registration of 3-D Shapes“, in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol 14, pp. 239-256, 1992

[14] Y. Chen, G. Medioni, “Object Modelling by Registration of Multiple Range Images” in *Image and Vision Computing*, vol. 10, pp. 145-155, 1992.

[15] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm”, in *Proc. of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pp. 145–152, 2001

[16] B. Horn, „Closed–form solution of absolute orientation using unit quaternions“, in *Journal of the Optical Society of America*, vol. 4, pp. 629-642, 1987

[17] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W., “Mesh optimization”, In SIGGRAPH Computer Graphics Proceedings, pp 19-26, 1993

[18] K. Boehnke, M. Otesteanu, “Progressive Mesh based Iterative Closest Points for Robotic Bin Picking”, *to be published in Proceedings of International Conference on Informatics in Control, Automation and Robotics*, 2008

[19] T. Jost and H. Huegli, “A Multi-Resolution ICP with Heuristic Closest Point Search for Fast and Robust 3D Registration of Range Images”, in *4th Inter. Conf.on 3-D Digital Imaging and Modeling*, 2003, pp. 427-433