3D OBJECT CATEGORIZATION OF LOGISTIC GOODS FOR AUTOMATED HANDLING

Hendrik Thamer, Daniel Weimer, Henning Kost, Bernd Scholz-Reiter
Bremer Institut für Produktion und Logistik at the University of Bremen
Hochschulring 20, 28359 Bremen, Germany
{tha,wei,kos,bsr}@biba.uni-bremen.de

ABSTRACT
The automated handling of universal logistic goods through robotic systems requires suitable and reliable methods for categorizing different logistic goods. They must be able to detect the pose of different types and sizes of logistic goods in order to identify possible gripping points or for selecting a suitable gripping system for the detected object type. For this purpose, Time-of-Flight or Structured Light sensors can deliver a dense 3D representation of the investigated scenario. This paper presents a 3D object categorization system for logistic goods based on synthetically generated model data. We generate the model data by using a sensor simulation framework for different TOF-sensor types. The framework creates point clouds of self-defined geometric models of logistic goods or CAD data. Afterwards, we use these synthetic point clouds for generating a suitable model database offline. In order to evaluate our approach, we describe the synthetic point clouds by global point feature description techniques to distinguish between different types of logistic goods. Finally, we evaluate our concept with real sensor data from different logistic goods.

KEY WORDS
3D and Range Data Analysis, Sensor Simulation, Pattern Recognition, Robotics.

1. Introduction
Increasing the performance of logistic processes requires flexible and intelligent robotic systems that are able to recognize and to handle with different objects in various logistic application scenarios. Therefore, robotic systems need suitable vision systems for categorizing and localising all relevant objects within their field of view. The vision system must be robust regarding obstacles like occlusion and measurement uncertainties. Usually, automated handling tasks need spatial information of the objects within the relevant area of interest. Therefore, 3D sensing techniques like Time-of-Flight (TOF) sensors as well as systems that use structured light are used for capturing a dense point cloud representation of the scene. For instance, within the logistic application scenario of automated unloading of standard sea containers, sensors like laser scanners or TOF-cameras are used for getting spatial information about different packaging scenarios [1]. The main reason of using TOF sensors is a high degree of independence from lighting conditions and reflectivity properties of the investigated objects. Additionally, no reconstruction step as applied in stereo vision systems is necessary.

Recognizing and categorizing of different types of objects needs a model or a complete model database that contain a suitable description about the objects of interest. In today’s logistic applications which use vision systems for categorizing goods, single and simple models for specific types of goods are used [1,2]. Therefore, these systems are not very flexible recognizing new objects or dealing with occlusions and local deformations of goods.

In this paper, we present a flexible 3D object categorization system for logistic goods based on synthetically generated model data by using a sensor simulation framework. The framework creates point clouds of partial views regarding the position and orientation of the sensor as well as specific sensor characteristics like noise or uncertainties due to the measurement principle of TOF sensing. Logistic goods can be divided into three main shape classes: cubical, cylindrical, and free-form surfaces (e.g. sacks) [3]. For each shape class, the sensor simulation framework can generate point clouds of various models from different virtual viewpoints. The resulting point clouds are influenced by a realistic sensor noise. Afterwards, all point clouds are described by global point feature descriptors and are stored in a model database that is generated offline. Therefore, we use View-Point Feature Histograms (VFH) introduced by Rusu et al. on each simulated point cloud [4]. For evaluation of our approach, we compare the model database with real sensor data of different logistic goods.

2. 3D-Object Categorization
The recognition and categorization of different shaped objects in 3D representations is a crucial task in the field of robot vision. Fulfilling this task, robotic vision systems require some previous knowledge of all relevant objects within the specific working environment. Figure 1 shows an example structure of a model based 3D vision system. Here, suitable models of all objects that can appear in the environment of the robots must be trained in an offline
training step. This requires the extraction of suitable features of the training data. Therefore, feature description techniques are used which efficiently describe different surface shapes of an object. Finally, these feature descriptions are stored in a model database. After preprocessing and segmentation, the extracted feature description of real sensor data is compared to the model database. In the final categorization step, the object is classified and its pose in the scene can be estimated.

Feature description techniques can be separated into local or global feature descriptions techniques. Global descriptors describe the whole geometry of the surface of a partial view of an object and are more robust against noise comparing to local feature descriptors but are strongly influenced by missing parts or occlusion [5]. In contrast, local descriptors describe the local neighborhood of a single point on a surface. Due to their operating principle they are more robust against occlusions or clutter but suffer from a higher influence to measurement noise or point density variations.

In contrast, global feature description techniques describe the surface of the partial view of a whole object. Therefore, a prior segmentation step must be performed in order to segment an object candidate from the complete point cloud. Examples for global feature description techniques are Point Feature Histograms [10], Viewpoint Feature Histograms [4], Clustered Viewpoint Feature Histograms [5], and Ensemble of Shape Functions [11]. The VFH descriptor shows very good results with respect to computation time and recognition rate in simultaneous recognition of an object and its pose in a segmented scene [4]. It encodes the geometry of a surface and the viewpoint by creating a local reference frame with a Darboux coordinate system and measuring features that represent angular distributions between surface normals and the viewpoint [12]. Due to its dependence to the viewpoint, training data from different views of a single object is necessary. The final feature distribution is stored in a histogram with 308 bins.

In order to create a suitable model database for an object categorization system, data for model generation are required. This can be realized by using a real sensor setup and acquire real data or by creating synthetic training data. Using a real sensor setup can be a very time consuming step in order to generate sufficient data from different viewpoints. The following sections present a sensor simulation framework for generating geometric models of different shape classes of logistic goods. Based on these models, point clouds from different views can be simulated depending on specific sensor characteristics. The simulation is fully parameterizable and can automatically simulate point cloud representations in different resolutions depending on the used sensor type as well as position and orientation of the sensor in the virtual scene. It can simulate point clouds according to characteristics of different types of commercially available TOF-sensors like laser scanners or TOF-cameras. We validate our approach with experiments using VFH description of the generated synthetic point cloud data. Then, we evaluate our approach with real sensor data that are acquired by the same sensor we have used for synthetic model data generation.

Figure 1. Structure of a model based vision system

Commonly used local feature descriptors are Signature of Histograms [6], Fast Point Feature Histograms [7], 3D Shape Context descriptors [8], and Spin Images [9]. For instance, the spin image technique uses oriented points for calculating a unique local feature description by transforming the 3D object recognition problem to 2D image matching. An oriented point consists of its coordinates and a surface normal. The spin image representation of an oriented point consists of two parameters. All points in a defined vicinity of the oriented point are described by these parameters. The accumulation of the parameters is stored in a histogram with 225 bins, whereby each bin represents the distribution for a unique combination of the two parameters. The spin image computation is performed for all oriented points of the object. In the following recognition step, spin images of a model and a scene object are compared using image correlation techniques. By selecting a support distance that covers the whole object, the spin image technique can also be used as global feature descriptor.
3. Sensor Simulation Framework

This section presents our sensor simulation framework for synthetic model and training data generation. It is implemented using MATLAB in combination with C. It provides opportunities for simulating point clouds from single geometric models of logistic goods or of complete scenarios like container loads or pallets in high rack warehouses. Additionally, the framework could generate multi-view sensor data by placing several sensors in the virtual scene and performing and automatic point cloud merging within a registration step. The following subsection describes geometric modeling of the three relevant shape classes of logistic goods. This includes techniques for deforming the geometric models in order to represent realistic shape distributions. Then, the simulation method of point clouds depending on sensor characteristics is presented. Finally, the last part of this section describes the generation of the training set.

3.1 Geometric Modeling

Geometric models for the three predefined shape classes cubical (boxes, packages), cylindrical (barrels), and free-form surfaces (sacks) must fulfill different requirements. They must be scalable in order to represent different sizes of the shape classes. Additionally, they need to be locally deformable in order to simulate deformations caused by defects of the package material or by deformable goods like sacks. The modeling is realized by using Bézier surfaces [13]. A set of control points define the shape of the surface. Equation 1 shows the definition of a parametric surface defined by the Bézier technique. Thereby, the points of a Bézier surface are created by using \((n+1)(m+1)\) Bézier control points \(P_{ij}, i=0,...,n, j=0,...,m\) and \(u,v \in [0,1]\). \(B\) represents the so called Bernstein polynomial that is necessary for computing the points on the Bézier surface.

\[
P(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} P_{ij} \cdot B_i^n(u) \cdot B_j^m(v)
\]

\[
B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i}, i=0,1,...,n
\]

(1)

Figure 2 illustrates an example surface with a net of sixteen Bézier control points. The shape of the surface depends on the position of the control points. Changing the position of a control point results in a different shaped surface.

Figure 2. Bézier surface with control net

In general, cubical goods are not very complicated to model. However, in order to simulate fine defects in the package of the cubical good, a dense surface representation is necessary. Therefore, we decompose a cubical good into six single surfaces with selectable granularity. Subsequent, local deformations like dents and bumps can be easily simulated. Figure 3 visualizes a cubical good with local deformations.

Figure 3. Cubical good with local deformations

Compared to cubical goods, cylindrical goods like barrels consist of different form types. In order to get the coordinates of the lateral surface a Bézier control net is spanned vertically from the bottom to the top. A quarter of the lateral surface of a cylindrical good is modeled and mirrored in order to simulate the entire cylindrical good. The sensor simulation framework is able to simulate different shape types of cylindrical goods as shown in Figure 4. Usually cylindrical goods are made of undeformable material like wood or metal. Therefore, the sensor simulation framework does not provide the opportunity of applying local deformations of the goods for cylindrical objects.

Figure 4. Different types of cylindrical goods
Free-form surfaces for modeling of sacks or bags are the most complex type regarding geometric modeling. In logistic application scenarios, their shape depends on the type of product that is transported. They can be locally deformable depending on the environment of the good. Other goods that are placed near the object can influence its shape. In order to simulate different realistic sack goods, we construct a standard model of a sack as shown in Figure 6. Therefore, a 3x3 net of Bézier control points is developed and the resulting Bézier surface is computed.

3.2 Point Cloud Simulation

The sensor simulation framework is able to transform the presented geometric models into point cloud representations from a virtual viewpoint by considering specific sensor characteristics. At first, a logistic object is modeled as presented in the previous subsection. Following, a TOF sensor is placed in the virtual scene and its parameter like field of view, angular resolution and noise settings are specified. According to these settings, the resulting point cloud and range image are simulated. Therefore, the sensor simulation framework uses techniques from the field of computer graphics like rendering. In the first step, the surfaces are decomposed into single triangles. Then, ray tracing is applied regarding field of view and angular resolution of the simulated sensor. For each ray, the intersection point with each triangle in the virtual scene is computed. The intersection point which is the closest one to the sensor origin is stored in a z-buffer and considered as intersection point for the resulting point cloud.

In order to improve the computation time, the number of triangles in the virtual scene is reduced by applying visible surface determination algorithms. For instance, the application of back face culling leads to reduction of computation time by half [13]. This algorithm removes all triangles from the computation process that are not visible from the sensor location. This is the case when the angle between the normal of the triangle and the vector from sensor origin to the triangle exceeds 90 degrees. The point clouds are influenced by an artificial characteristic sensor noise from TOF-measurement systems. TOF sensors measure the time from emitting and detecting the reflected laser beam. We have divided the noise influence into Signal to Noise Ratio (SNR) depending on the measurement distance, mixed pixels and the heating time of the sensor. For simulating a characteristic noise depending on measurement distance we distinguish between a laser scanner and TOF-cameras. In contrast to laser scanners, the integration of the angle of incidence in the simulation is important for TOF-cameras. They capture the entire scene with a single laser pulse. Equation 2 describes our distance dependent noise for TOF-cameras.

\[
\sigma_{\text{TOF}}^2(r, \rho) = \frac{1}{\rho} \cdot k_1 \cdot r^2 \cdot (1 + \sin(x) \cdot \cos(y) + \cos(x) \cdot \sin(y))
\]
\( \lambda \) represents the wavelength of the laser, \( \rho \) the reflectance of the target, \( x \) and \( y \) are the position values and \( r \) is the distance between the sensor and the reflecting object. The range dependent simulation of laser scanners is presented in Equation 3.

\[
\sigma_{LS}^2(r, \rho) = \frac{1}{\rho} k_r \cdot r^2
\]  

(3)

The coefficients of different TOF-sensor types are listed in Table 1. After calculating the variance, we multiply it for each dimension of the calculated intersection point with a Gaussian distributed random number. Then, this value is added to the real values of the intersection point.

<table>
<thead>
<tr>
<th></th>
<th>LMS 200</th>
<th>LMS 500</th>
<th>CamCube</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_r )</td>
<td>300 1/m³</td>
<td>73 1/m³</td>
<td>215 1/m³</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>905 nm</td>
<td>905 nm</td>
<td>870 nm</td>
</tr>
</tbody>
</table>

In addition to range dependent noise, we simulate the TOF specific characteristic of mixed pixels. Mixed pixels are pixels which receive reflectance power of two different objects in the local neighborhood from different ranges. We use a standard median filter with a nearest neighborhood of nine points for each point, to simulate the influence of mixed pixels to our synthetic point clouds. Since the emitted wavelength is a function of temperature, the measurement should only begin in a steady state, which is reached after approximately three hours [14]. After this time, the drift of the characterization parameters is reduced. In our approach, we added a random noise to the calculated intersection points, that is influenced by a given operation time. However, the longer the operation time the less noise is added to the intersection point. The resulting point clouds are stored as point list and in specific point cloud format used in the Point Cloud Library (PCL) [15]. Additionally, the RGB image resulting of the field of view of the sensor is simulated. Thereby, the texture information can be mapped directly to the point cloud. In this way, 2D and 3D (RGB-D) data of a logistic good could be simulated and referenced to each other.

### 3.3 Model Database Generation

In order to generate a suitable model database, we simulate point clouds from several viewpoints. The amount of viewpoints can be set by defining a fixed sensor position and a rotation angle that rotates the good relative to the sensor origin. For each angle, the resulting point cloud is simulated according to the sensor type and parameter settings. This is performed for all predefined shape classes and on different model instances. Figure 7 visualizes the concept of the presented synthetic dataset approach. After creating the synthetic point clouds, a suitable feature representation for describing the shape type of the point clouds must be chosen. In our approach, we select the description by VFH because of fast computation time and very good recognition rates in other application scenarios [4]. All VFH descriptions of our simulated point clouds form the model database. The whole model database generation procedure is performed in an offline training step. Figure 5 illustrates our model database generation concept.

### 4. Experiments

In order to validate our model generation within the sensor simulation framework, we performed experiments with real sensor data of the same sensor used in our sensor simulation. The VFH description of a logistic good is compared with the descriptions of our synthetic point clouds. The database is generated simulating a PMD CamCube 3.0 sensor. The sensor parameters are described in Table 2. We create two model databases. The first one is influenced by our TOF noise model, the other is generated by ideal point cloud data without the influence of noise. The databases consist of 20 objects and 80 distinguishable partial views for each predefined shape classes.
Databases are constructed without knowing which test objects are used for the evaluation process. Therefore, they do not directly match with the geometric sizes or deformation properties of the test objects. The test objects are placed in the field of view of the CamCube with an arbitrary orientation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame rate (3D)</td>
<td>40 fps (200x200 pixels)</td>
</tr>
<tr>
<td>Measurement range</td>
<td>0.3 - 7m</td>
</tr>
<tr>
<td>Repeatability</td>
<td>&lt; 3mm</td>
</tr>
<tr>
<td>Illumination Wavelength</td>
<td>870 nm</td>
</tr>
<tr>
<td>Resolution</td>
<td>200x200 pixels</td>
</tr>
</tbody>
</table>

Table 2: Sensor parameters of CamCube 3.0 [16]

Due to the global nature of the VFH feature description technique, the online categorization step includes a previous segmentation step. Thereby, the background is segmented from the scene by using dominant plane subtraction and the resulting points represent only an object candidate. Additionally, the descriptor is very sensitive regarding missing parts or occlusions. Therefore, we place the goods with sufficient space to each other. The whole recognition process is programmed using PCL and the FLANN library [15]. The final histogram comparison is realized by using Chi-Square metric. In order to improve the pose estimation step, the ICP registration algorithm could be applied for refining the alignment [17]. Figure 8 illustrates an example experiment setup with real sensor data acquired by the PMD CamCube for each predefined shape class of the logistic objects as well as the corresponding VFH feature description histograms.

We have performed fifteen different experiments with boxes, barrels and sacks in arbitrary orientations. We received a total correct classification rate of 91%. The resulting single classification rates for each shape class are listed in Table 3.

<table>
<thead>
<tr>
<th>Classification rates for both databases</th>
<th>Ideal Data</th>
<th>Noisy Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boxes</td>
<td>75 %</td>
<td>75 %</td>
</tr>
<tr>
<td>Barrels</td>
<td>100 %</td>
<td>100 %</td>
</tr>
<tr>
<td>Sacks</td>
<td>100 %</td>
<td>100 %</td>
</tr>
</tbody>
</table>

For both databases, two boxes were categorized as sacks. This could be caused by the arbitrary models in our model databases, which do not directly fit to the test objects. Comparing the resulting mean measurement distances of the histogram comparisons, the usage of the noisy database gets better results than the ideal database. Table 4 represents the mean measurement distances for each predefined shape class.

In order to improve the classification rates, we will focus our further research work on generating larger databases with more different model instances to improve classification rates and mean measurement distances. Furthermore, we will train machine learning algorithms like neural networks or support vector machines with our synthetic data. This will result in a much more efficient object classification. Thereby, real sensor data is only compared to the learned model of the machine learning algorithm, instead of comparing it with the whole model database.

5. Conclusion

This paper has presented a sensor simulation framework for generating synthetic model data for a 3D model-based categorization system. The framework can simulate point cloud representations of self-defined geometric models or from imported CAD-data. The simulation depends on specific sensor parameters and an artificial TOF measurement noise model. We have developed geometric models for three different shape classes of logistic objects. Afterwards we have generated synthetically different point clouds from several distinguishable virtual viewpoints. Afterwards, we have described the shape of these point clouds by using the VFH global feature description technique. All generated VFH descriptions are stored in model database. We evaluate the suitability of our approach by comparing the model database with real sensor data from the same sensor type used in our sensor simulation procedure. Our first results show very promising results for using our synthetic model data for fast and efficient model database generation for various application fields.

In our further research work, we want to use different feature description techniques for matching sensor data and the model database. Additionally, we will use the generated feature descriptions of the model database for training a machine learning system like support vector machines or neural networks. Therefore, the real sensor data is only compared with the learned model of the machine learning method instead of using the whole database. This would result in a faster computation time within the online recognition step. Additionally, we want to improve a measurement noise model by adding the influence of the reflectance coefficient, the type of the reflecting material and by modelling the ambient light conditions.
Acknowledgement

The authors gratefully acknowledge the financial support by DFG (German Research Foundation) for the EMOSES project.

References


Figure 8. a) Example Experiment Setup b) Captured and Segmented Point Cloud from CamCube
c) Single point clouds of logistic objects d) Corresponding VFH feature histograms