Curvature based range image classification for object recognition

Jan Böhm*, Claus Brenner†
Institute for Photogrammetry (ifp),
University of Stuttgart, Germany

ABSTRACT

This work focuses on the extraction of features from dense range images for object recognition. The object recognition process is based on a CAD model of the object. Curvature information derived from the CAD model is used to support the feature extraction process. We perform a curvature based classification of the range image to achieve a segmentation into meaningful surface patches, which are later to be matched with the surfaces of the CAD model.

Keywords: CAD, curvature, feature extraction, object recognition.

1. INTRODUCTION

In 1998 the University of Stuttgart started a research project on optical measurement using sensor actor coupling and active exploration. This large scale project is a collaboration of researchers from seven institutes of the University of Stuttgart including mechanical engineers, electrical engineers and computer scientists. The goal of our work is to design and implement a measurement system, which can handle a large variety of objects from the industrial world. The system is currently equipped with three sensors, a mono camera, a stereo camera and a 3D sensor. The current implementation of the measurement system developed by our group is depicted in figure 1(a).

We aim at a system which is capable to make use of the different types of sensors and automatically tailor the measurement process towards the object presented to the system by selecting optimal sensor configurations and combinations and executing measurement tasks specific to the object. A CAD model for each object forms the basis for measurement planning and assessment. The system is to achieve high flexibility in two respects. First we require the system to handle a variety of objects. Second, we do not restrict the pose of the objects, rather we allow the objects to be presented to the system in arbitrary position. To meet these requirements one of the first steps of the measurement process obviously has to be object recognition in order to identify the object and determine its pose. In our work we use 3D data, so called range images, as sensor input.

1.1. Object Recognition

In the past, several model-based object recognition systems for range images have been reported. A comprehensive review on these systems has been given by Arman and Aggarwal.[1] As Grimson[2] states "virtually all machine recognition systems structure the recognition problem as one of searching for a match: specifically, how to associate components or parameters of the sensory data with corresponding components or parameters of the model". We can further categorize recognition approaches into systems that use global properties of the object for matching and systems that use local properties. A general layout for an object recognition system is shown in figure 2.

The use of global properties has the advantage that no feature extraction has to be performed. In many cases a transformation of the model and the sensor data into a common parameter space is carried out. Because range images contain only a certain view (aspect) of an object, parameters derived from such data can often not be directly compared to those derived from a full 3D model. This leads to approaches where the model is broken down into several views and the 3D object is thus represented by a collection of parameters for each view. Examples for this approach are the works by Campbell and Flynn[3] and Johnson and Herbert.[4] To achieve a most complete representation of the object the view space has to be sampled densely, i.e. a large number of views has to be generated and stored. This leads to systems with high computation costs for both the modeling and recognition stage.

*Email: jan.boehm@ifp.uni-stuttgart.de
†Email: claus.brenner@ifp.uni-stuttgart.de

(a) (b)

Figure 1. The experimental measurement center and a view into the measurement volume.

The fundamental idea behind the second approach which uses local features is to split up the object into smaller parts and try to establish correspondences between these parts. Using geometric constraints one can expect to reduce the exponential search space sufficiently to achieve acceptable performance. One of the crucial parts in this approach is obviously the extraction of local features. In our context of range images this is usually referred to as segmentation or more precisely segmentation into surface patches. The better the segmentation is, the smaller the number of possible correspondences will be and therefore the smaller the search space. However, the more advanced segmentation techniques tend to be more time consuming.

We propose a method in which we combine the idea of a common global parameter space introduced above and the idea of local feature matching. We use the information from the model transformed into a global parameter space, the HK space which is introduced below, to aid the local feature extraction. The HK space forms the link in-between the model and the raw sensor data. It allows us to transfer information from the model to the data before the feature extraction is carried out. This in effect leads to a model-driven feature extraction process also shown in figure 2.

The task of object recognition by local feature correspondence an be formalized as follows: Finding the correspondence of features f_i from the scene $S = \{f_1, f_2, ..., f_n\}$ with features F_i from the model $M = \{F_1, F_2, ..., F_n\}$. The collection of features $\{F_1, F_2, ..., F_n\}$ form the description of the model. In our framework the model used for object recognition is automatically derived from CAD data.

Many of the early systems that were reported rely on the presence of specific features. For example, some systems were able to detect cylindrical objects,[5] others were designed to handle planar surfaces with sharp edges which restricted them to polyhedral objects.[6] In our context we are required to handle industrial parts of complex shape, including rounded edges and free-form surfaces. Object recognition for industrial parts is a pretentious task, despite the fact that explicit three-dimensional CAD models are available. One reason for this is that CAD models may in general not be used directly for object recognition, since it is often impossible to extract CAD specific high-level features from sensor data. Rather, an intermediate representation is needed which forms the interface between CAD representation and features which can be extracted from sensor data. In the work we present here, the CAD model is broken down into single surfaces. For each of these surfaces the fundamental curvature characteristics are derived and stored. This information is then used to extract features from the sensor data. The crucial steps and representations of data of our proposed method can be seen in figure 3: a CAD model forms the complete geometric representation of the object. The curvature characteristics of the object, extracted from the CAD data, are used to support the extraction process. We can clearly identify the correspondence of the peaks in the curvature histogram computed from a range image to the CAD derived curvature clusters.

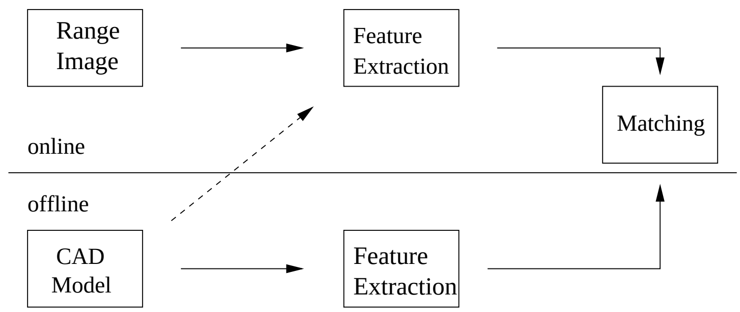


Figure 2. Traditional paradigm in model-based computer vision with our modification (dashed line).

As this paper focuses on the feature extraction process, the object recognition itself is not shown here. We have previously reported on our approach using constrained tree search to match scene and model features.[7] A similar approach can be combined with the feature extraction presented in this work.

1.2. Segmentation and curvature

Segmentation or feature extraction of range images has been a popular research topic for the past years. Several techniques have been developed including clustering, region growing and split and merge, see also the review by Hoover et al..[8] But also problems became evident concerning reliability and curved surfaces. In our previous work we have obtained good results with the segmentation of polyhedral objects using region growing.[9] However, this approach relies on the edges of planar surfaces to terminate the growing of a region. The method could not easily handle rounded edges. It is also not extendible to arbitrarily shaped objects.

We propose a model-driven approach for feature extraction based on the CAD model as shown in figure 2. This approach has the potential to avoid gross over-segmentation and misclassification. For every point in the range dataset we compute the fundamental curvature characteristics. The information extracted from the CAD model is then used to classify the point accordingly. Our approach differs from the work of others in that we do not use a generic surface model, but a fixed model specific to the object processed. This changes the traditional paradigm in model-based computer vision, because we already use the CAD model data in the feature extraction stage. As an example of a more traditional system Newman, Flynn and Jain[10] have developed a model driven approach, in which they classify range images into planar, spherical, cylindrical or conical surfaces according to the sign of curvatures. Then they fit their generic surface model, which is a quadric surface either a plane, cylinder, cone or sphere, to the data. In contrast to this, the method we propose distinguishes all individual surface patches present in the CAD model according to the exact values of the curvature estimates. Our system currently can handle 15 different surface types, including advanced types such as tabulated cylinders and NURBS.

Surface curvature has been a favorite criterion for segmentation among researchers in related fields for some time. In most works only the signs of curvatures have been used to classify the patches. Paul J. Besl[11] has introduced a method which uses eight different curvature classes based on the signs of mean and Gaussian curvature. We believe that advances in sensor technology which have brought us high resolution and high quality range images, enable us to use an exact measurement of curvature to classify the points into more complex surfaces types corresponding to those of the CAD model. The necessity of exact curvature measurement has an implication on our curvature estimation scheme as shown below.

2. PROCESSING THE CAD MODEL

In this work we use Pro/ENGINEER, a widely used solid modeling CAD system to perform CAD related operations. The system has an application interface, which allowed us to integrate our own software into the system. Interfacing to the system relieved us from some tedious programming work, such as reading CAD files and identifying individual surfaces. While the implementation is specific to the system, the basic idea of our work is general to all CAD data. Figure 4(a) shows the integration of our software into the user interface of the CAD system.

A CAD model typically consists of several individual surfaces which were generated during the design process. We have implemented a routine which iterates over all surfaces of the model. For each surface we output the surface

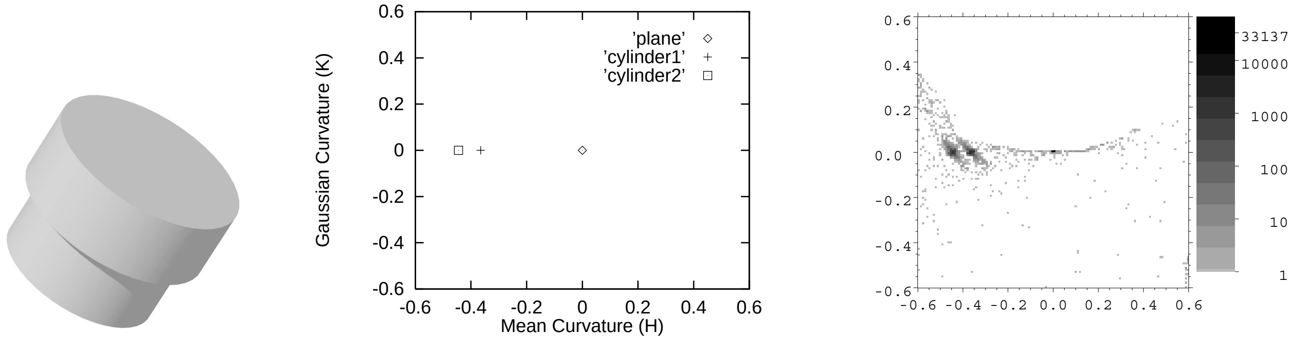


Figure 3. A simple part "adapter" (left), its curvature characteristics as derived from the CAD model (center) and a histogram of curvatures measured from range data(right).

ID, the surface type and the curvature characteristics. For the curvature we concentrate on the mean and Gaussian curvature H and K. Mean and Gaussian curvature form a two dimensional space, let us call it the HK space. Each surface has a distinct footprint in the HK space. Some surface types such as plane, cylinder and sphere occupy only a single point in HK space, i.e. the mean and Gaussian curvature is constant across the surface. Others occupy areas of arbitrary shape in HK space. For example the mean and Gaussian curvature of a torus lie along a line in HK space. The values of mean and Gaussian curvature can be displayed in a two-dimensional plot as seen in figure 5(a).

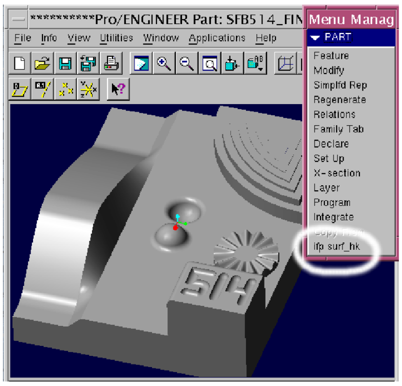
During iteration when we encounter a surface of constant curvature we compute the curvature according to the geometric parameters. In detail this generates the values $(0, 0)$ for a plane, $(1/(2*R), 0)$ for a cylinder and $(1/R, 1/R^2)$ for a sphere. For other surface types we retrieve the parametric representation $F : x(u, v)$ of the surface. We evaluate the surface at a discrete set of points $\{(u_1, v_1), ..., (u_i, v_i)\}$. The coordinates $[x, y, z]$ of the point, the surface normal $\mathbf{N} = [N_x, N_y, N_z]$ and the derivatives $\mathbf{x}_u, \mathbf{x}_v, \mathbf{x}_{uu}, \mathbf{x}_{uv}, \mathbf{x}_{vv}$ are computed at each sampled point. From these, the mean and Gaussian curvature are computed using the well known formulas 1 and 2 shown in figure 6.[12] The list of HK values form the footprint of the surface in HK space. After all surfaces have been visited, we have obtained a list of surface features for our CAD model. Figure 4(b) shows a typical output of our algorithm.

However this list has to be post-processed. Features which are too similar in HK space can not be distinguished and have to be merged. This is particularly true for features which are exactly the same. For example, if the CAD model contains several planar surfaces or several cylindrical surfaces of the same radius, each will be reported separately. Merging of features introduces ambiguities in the subsequent matching process. More formal, if the model contains features F_i and F_j which can not be distinguished they are merged temporarily to feature T_i. After classification, when a feature f_i from the scene falls into the category of T_i it has a possible match to both F_i and F_j or $\Omega(f_i) = \{F_i, F_j\}$. While pattern matching processes are designed to handle a certain amount of ambiguity, the complexity increases exponentially with the number of possible pairings. So clearly if the object is a polyhedron thus containing only planar surfaces (all with $H = K = 0$) our approach will perform poorly. We depend on the surfaces having distinguishable curvature characteristics.

3. FEATURE EXTRACTION

3.1. Computing Curvature

As for the CAD model, the curvature has also to be computed for the range image. Several schemes have been proposed for curvature estimation. Some techniques were developed especially for triangulated surfaces. However, these techniques usually use only a very small neighborhood of the inspected point and therefore deliver unreliable results. Other methods assume raster organization of the data. The simplest uses convolution to determine the derivatives needed to compute H and K. However, second order derivatives are susceptible to noise. More elaborate techniques fit second order bivariate polynomials of the form $z = f(x, y)$ to the data using least squares techniques. The fitting is done in a small rectangular neighborhood of the point. Using the coefficients of the polynomial the derivatives are computed analytically. The results are then stored for the center pixel of the window which is moved over the data.

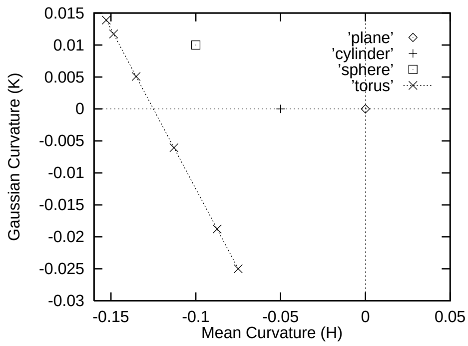


(a) screenshot of the CAD system

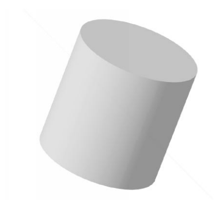
ID	Type	(H	K)+		
31	PRO_SRF_PLANE	0.000000	0.000000		
2191	PRO_SRF_CYL	-0.006250	-0.000000		
2424	PRO_SRF_TORUS	0.243827	0.051440	0.235146	0.045653 ...
2642	USR_SRF_SPHERE	-0.250000	-0.062500		
2705	USR_SRF_SPHERE	-0.100000	0.010000		
2714	PRO_SRF_FIL	0.114790	-0.005105	0.114783	-0.005108 ...
2719	PRO_SRF_FIL	0.114784	-0.005108	0.114780	-0.005110 ...
2786	PRO_SRF_FIL	-0.114790	-0.005105	-0.114783	-0.005108 ...
2791	PRO_SRF_FIL	-0.114784	-0.005108	-0.114780	-0.005110 ...
2917	USR_SRF_SPHERE	0.333333	0.111111		
3202	PRO_SRF_CYL	0.166667	0.000000		
7071	PRO_SRF_PLANE	0.000000	0.000000		
7082	PRO_SRF_TABCYL	0.033435	0.000000	0.033435	0.000000 ...
7086	PRO_SRF_TABCYL	-0.019724	-0.000000	-0.019724	-0.000000 ...
7088	PRO_SRF_TABCYL	-0.001299	-0.000000	-0.001299	-0.000000...
7382	PRO_SRF_CYL	0.166667	0.000000		

(b) Example output

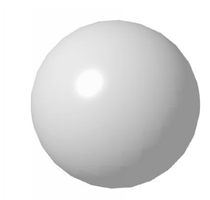
Figure 4. The algorithm is fully integrated into the CAD system.



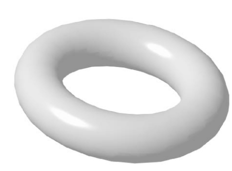
(a) plot in HK space



(b) cylinder



(c) sphere



(d) torus

Figure 5. Some typical surfaces and their signature in HK space.

$$\begin{array}{lll} E = \mathbf{x}_u \mathbf{x}_u & F = \mathbf{x}_u \mathbf{x}_v & G = \mathbf{x}_v \mathbf{x}_v \\ L = \mathbf{N} \mathbf{x}_{uu} & M = \mathbf{N} \mathbf{x}_{uv} & N = \mathbf{N} \mathbf{x}_{vv} \end{array} \tag{1}$$

$$H = \frac{EN + GL - 2FM}{2(EG - F^2)} \quad K = \frac{LN - M^2}{EG - F^2} \tag{2}$$

Figure 6. Formulas for derivation of curvature.

A very fast implementation of this approach has been reported by Paul J. Besl.[11] He formulated the surface fitting process as a series of convolutions thus speeding up processing. Two reasons prevented us from using his approach: First the range image samples are assumed to be equally spaced in x and y direction. Unfortunately, due to the nature of most sensors the distance in x and y of two neighbor pixels is not constant across the image, but is usually dependent on the z value. The second reason concerns the coordinate system of the data to be fit. Every result of a 3D measurement is stored in a certain coordinate system usually determined during sensor calibration. Let us call this coordinate system the implicit coordinate system. For range images it is quite typical for x and y to lie in the image plane and z perpendicular to it, for example pointing towards the viewer. When a function of the form $z = f(x, y)$ is fit to the data the error criterion $\sum(f(x_i, y_i) - z_i)^2$ is minimized. It is important to notice that the residuals of the regression are taken along the z axis. They do not necessarily represent the orthogonal distance of the point to the estimated surface. The more the surface normal deviates from the z axis, the less the residuals represent true geometric distances. This leads to a certain bias in surface fitting. A review of curvature estimation techniques is given in.[13] However the presented techniques suffered from the quantization of the range images which is not common today. An update on the methods for triangulated surfaces is given in.[14]

3.2. Model-based Classification

After mean and Gaussian curvature have been computed for each valid pixel in the range image, the output from the CAD system is used for classification of the range image. A simple minimum distance classification has been implemented. At the beginning all the pixels are unclassified. Then each pixel of the range image is transformed into a feature vector $p = (H_i, K_i)$ in HK space. The Euclidean distance of this feature vector to each of the features derived from the CAD model is computed. If the CAD feature F contains more than one HK sample, such as $\{(H_1, K_1), ..., (H_n, K_n)\}$, the distance of the feature vector p to feature F in HK space is determined as $d(p, F) = \min(d(p, (H_i, K_i)))$. The pixel is assigned the surface label of the closest feature. If the minimum distance is larger than a threshold, no assignment takes place and the pixel remains unclassified. As it has been noted above, this is not an object recognition process. Only the features for object recognition are being extracted. For example several planar patches may be present in the scene. They will all be assigned the same label. After the classification connected component analysis is performed. This step assigns different labels to unconnected patches of the same surface type. The resulting regions define the scene features f_i. Searching techniques are then needed to resolve the ambiguities for the object recognition process.

4. EXPERIMENTS

First experiments were made based on data from the SAMPL database maintained by P. Flynn.[15] The data base provided us with CAD models and range images for various simple parts. Most of them contain only cylindrical and planar surfaces and have few surfaces (less than 10). We chose two objects from the database, the first dataset is called “adapter”. It is a simple combination of two cylinders (see figure 3).

For the computation of curvatures from the range image we had to chose the size of the neighborhood for the surface fit. Clearly, choosing the mask size is a trade-off between reliability and accuracy near edges. When choosing a small mask curvature computation will be strongly affected by noise, due to the small number of points considered for regression. With the size of the mask we also increase the reliability of the estimation. However, when we chose large mask sizes the estimation will produce incorrect results near edges. For all the experiments presented here no preprocessing of the images such as smoothing was performed.

On the first object we used a mask size of 7x7. Figure 7 shows the result on this dataset. The z image is overlaid with the contours of the extracted regions. As we can see the algorithm performs well and gives sharp boundaries up to the edges. But as this is a very simple object the curvature characteristics of the three surfaces contained in the object can be distinguished easily and curvature estimates did not have to be extremely precise.

The second object, more complex, contains 5 cylinders and the according planar surfaces. Figure 8 shows this dataset called “bigwye”. Here more surfaces which are quite similar have to be distinguished. We needed a better estimate for the curvatures and therefore chose a mask size of 15x15. Figure 8 shows the results. We can see the effect a large mask size has on the performance of the algorithm near edges. Surface boundaries are not as sharp as before. But still the overall results are quite encouraging as the surfaces can be reliably detected independently of the pose of the object.

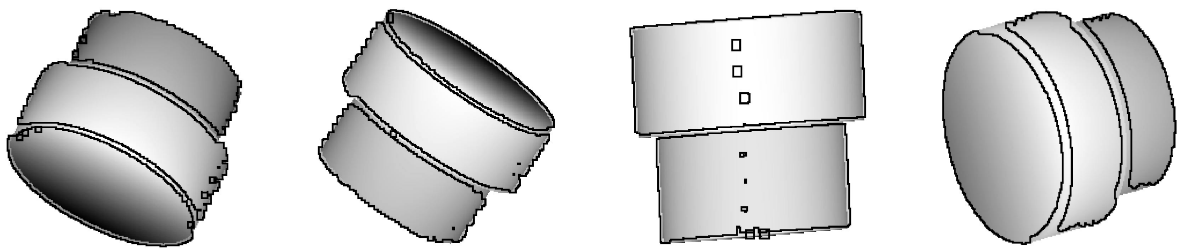
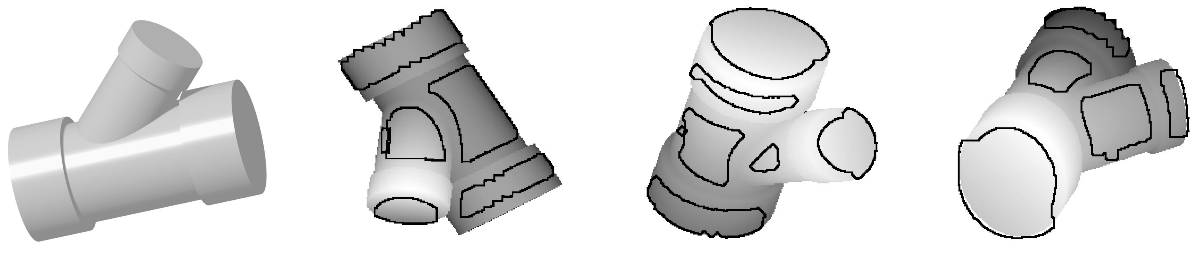


Figure 7. Results on test dataset “adapter” at different poses.



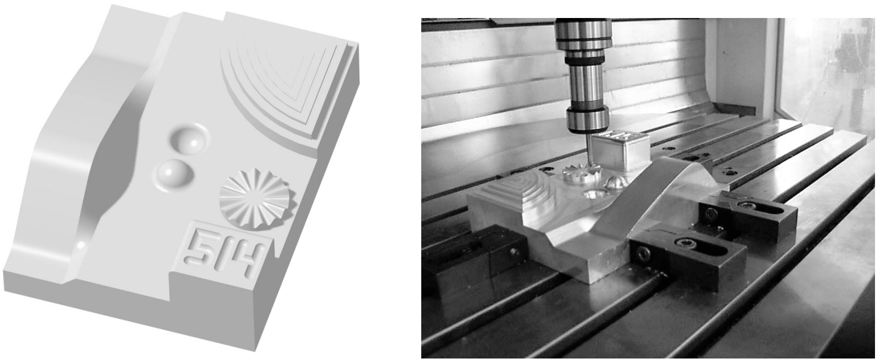
(a) (b) (c) (d)

Figure 8. (a) The CAD model of “bigwye”. (b)-(d) Extraction results on test dataset. Surfaces can be detected in range data independent of the pose of the object.

For the second experiment we used a part of more complex shape. For our research work we created a test object ourselves. Using Pro/ENGINEER we created a model containing several different surface types expected in an industrial object including among others planar, spherical, cylindrical, and free-form surfaces. This model was then manufactured from an aluminum block using a high precision milling machine. The size of the part is approximately 20 cm x 15 cm. We obtained range images from the object using our ABW stripe projector capable of projecting at a resolution of 640 x 640 stripes. Our camera system is a Basler A113 at a resolution of 1300 x 1030 pixels. Approximately only half of the image size was used. Based on the experiences from the first experiments we used again a mask size of 15x15. This allows for a precise estimate of curvatures but it prohibits us to detect small details of the part. As we can see in the results of figure 10 small surface details on the left of the object could not be extracted. However, large surface patches can be detected with great reliability. Figure 11 shows that the algorithm indeed identified region borders correctly even at smooth surface transitions. Thus it is capable to achieve a level of abstraction from the data comparable to that of the CAD data. These results are very encouraging for the matching stage where a correspondence of extracted features to those of the CAD model is to be established.

5. CONCLUSION

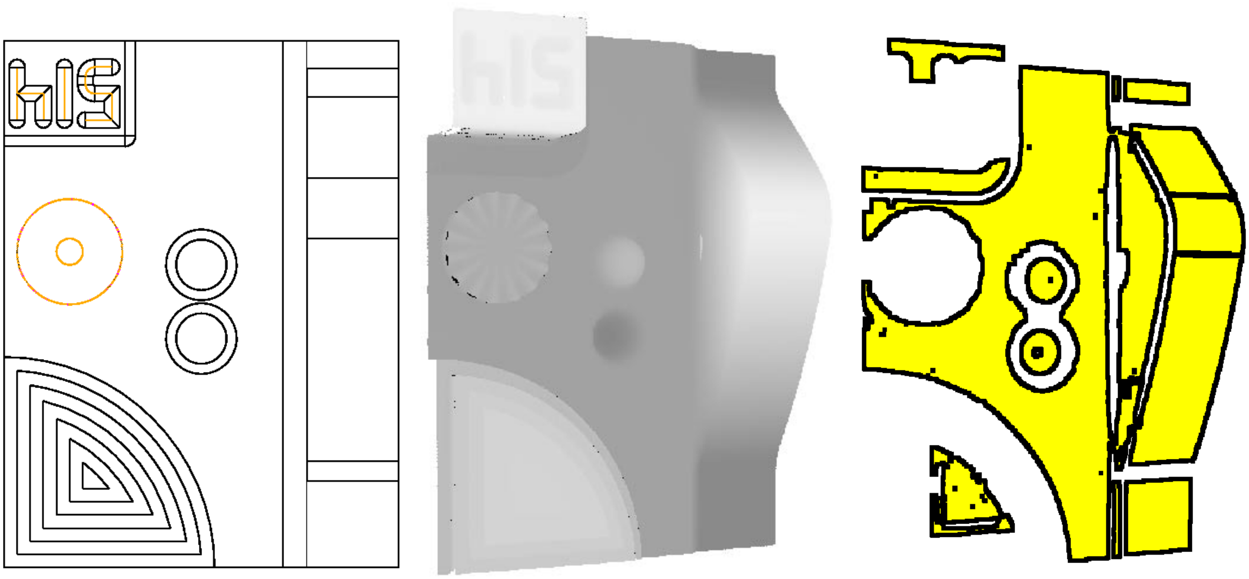
We have demonstrated a system detecting surfaces in a range image independent of the pose of the object. Reliable estimates of surface curvature are obtained from range images using a least squares surface fitting algorithm. A simple minimum distance classification has been shown to be adequate for range image classification when a CAD model is used to derive the 'master' classes for the classification process. The surfaces extracted from the range image closely correspond to those of the CAD model. The procedure is able to process all surface types which can be expressed in the CAD system including critical curved surfaces and free-form surfaces. This is an improvement over previous systems which are restricted to certain surface types. These results are an important step towards our goal of establishing a CAD model based object recognition system for industrial parts which is able to process arbitrarily shaped objects.



(a) shaded view of CAD model

(b) the part in the milling machine

Figure 9. Self-created model / part for experiments.

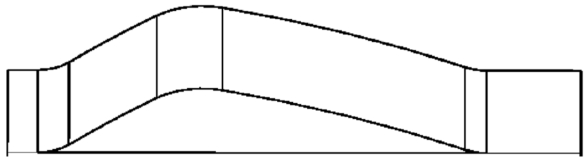


(a) wireframe display of the CAD model

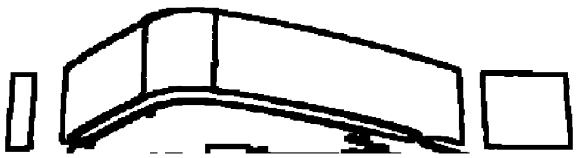
(b) range image of the part

(c) the extracted surface patches from sensor data

Figure 10. Feature extraction on self-created part.



(a) wireframe display of a part of the CAD model



(b) the extracted surface patches from sensor data

Figure 11. The large curved structure consists of three patches with smooth transitions. Still the method is able to extract the three patches correctly.

6. ACKNOWLEDGMENTS

The work presented in this paper was supported by the German Research Foundation under research grant DFG-SFB514.

REFERENCES

1. F. Arman and J. Aggarwal, "Model-based object recognition in dense range images," *ACM Computing Surveys* **25**, pp. 5–43, March 1993.
2. W. Grimson, *Object Recognition by Computer*, MIT Press, 1990.
3. R. Campbell and P. Flynn, "Eigenshapes for 3d object recognition in range data," in *CVPR99*, pp. II:505–510, 1999.
4. A. E. Johnson and M. Herbert, "Recognizing objects by matching oriented points," in *CVPR97*, pp. 684–689, 1997.
5. R. Bolles, P. Horaud, and M. Hannah, "3dpo: A three-dimensional part orientation system," *The International Journal of Robotics Research* **5**(3), pp. 3–26, 1986.
6. W. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range or tactile data," *The International Journal of Robotics Research* **3**(3), pp. 3–35, 1984.
7. J. Böhm, C. Brenner, J. Gühring, and D. Fritsch, "Cad-basierte objekterkennung für ein multisensorielles meßsystem," *Mustererkennung 1999* , pp. 62–69, 1999.
8. A. Hoover, G. Jean-Baptiste, X. Jiang, P. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher, "An experimental comparison of range image segmentation algorithms," *PAMI* **18**, pp. 673–689, July 1996.
9. C. Brenner, J. Böhm, and J. Gühring, "Experimental measurement system for industrial inspection of 3d parts," *Proc. SPIE Machine Vision Systems for Inspection and Metrology VII* **3521**, pp. 237–247, November 1998.
10. T. Newman, P. Flynn, and A. Jain, "Model-based classification of quadric surfaces," *CVGIP* **58**, pp. 235–249, September 1993.
11. P. Besl, *Surfaces in Range Image Understanding*, Springer-Verlag, 1988.
12. M. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, 1976.
13. P. Flynn and A. Jain, "On reliable curvature estimation," in *IEEE Conference on Pattern Recognition*, pp. 110–116, 1989.

14. R. Sacchi, J. Poliakoff, K. Häfele, and P. Thomas, "Curvature estimation for segmentation of triangulated surfaces," in *3DIM99 : Proceedings of the 2nd International Conference on 3-D Digital Imaging and Modeling, Ottawa, Canada, Oct. 4-8*, 1999.
15. R. Campbell and P. Flynn, "A www-accessible image and model database for computer vision research," in *Proc. IEEE Workshop on Empirical Evaluation Methods in Computer Vision*, 1998.