AUTOMATED EXTRACTION OF FEATURES FROM CAD MODELS FOR 3D OBJECT RECOGNITION

Jan BÖHM, Claus BRENNER, Jens GÜHRING, Dieter FRITSCH
Institute for Photogrammetry (ifp), Stuttgart University
Geschwister-Scholl-Strasse 24D, D-70174 Stuttgart
Jan.Boehm@ifp.uni-stuttgart.de

KEY WORDS: Object recognition, CAD, Modeling, Feature extraction

ABSTRACT

In this paper we report on our work on a CAD model-based object recognition system for industrial parts. We present a novel approach which uses information derived from the CAD model in the early process of range image segmentation. This approach gives an exact classification of the range image enabling the development of a CAD based object recognition system. We describe the feature extraction from CAD data and its use in the curvature based range image classification. We carried out experiments with data from multiple sources. The results obtained are presented and discussed.

1 INTRODUCTION

Our work is part of a large scale research project on optical measurement using sensor actor coupling and active exploration. This 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 implement a measurement system flexible enough to handle a large variety of objects from the industrial world. Using different types of sensors, e.g. mono cameras, stereo cameras and stripe projection systems, the measurement process is automatically tailored towards the object presented to the system and measurement tasks specific to the object are executed. A CAD model for each object forms the basis for measurement planning and assessment. Since we allow the objects to be presented to the system in arbitrary position, one of the first steps of the measurement process has to be object recognition determining the pose of the object. For our work we use 3D data, so called range images, as sensor input.

The task of object recognition can 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. Model building by hand is a tedious and time consuming task which is also missing repeatability and is therefore unacceptable in an industrial environment. In our framework a CAD model is available for each measured object. Therefore it is evident that the model used for object recognition should be automatically derived from CAD data.

In the past, several model-based object recognition systems for range images have been reported. A comprehensive review can be found in (Arman and Aggarwal, 1993). Many of the systems rely on the presence of specific features. For example, some systems were able to detect cylindrical objects, others were designed to handle planar surfaces with sharp edges which restricted them to polyhedral objects. 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.

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 (Hoover et al., 1996). 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 (Böhm et al., 1999). 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 arbitrary shaped objects.

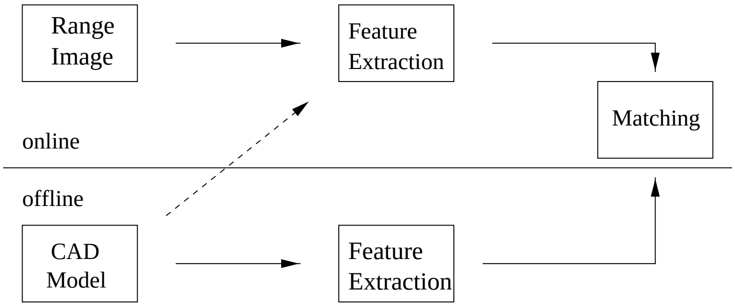


Figure 1: Traditional paradigm in model-based computer vision with our modification (dashed line).

Segmentation algorithms usually are designed to fulfill some basic requirements: completeness, compactness and minimum number of extracted regions. In contrast to many of the reported algorithms we do not need a complete segmentation of the range image for our application. That is, not every measured point has to be assigned a surface label. We only need a few reliably detectable features in the scene to perform the matching. Also we do not require the boundary of the surfaces to be extracted correctly. When dealing with complex shaped objects, surface boundaries are often corrupted by shadowing and self occlusion. So they do not yield reliable information for recognition. The number of detected regions is bound by the number of surfaces present in the CAD model. It becomes clear, that segmentation will benefit strongly from information about the type of features which are expected to be present in the scene.

We propose a model-driven approach for feature extraction based on the CAD model as shown in figure 1. 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 (Newman et al., 1993) have developed a model driven approach, in which they classify range images into planar, spherical, cylindrical or conical surfaces. Then they fit their generic surface model, which is a quadric surface either a plane, cylinder, cone or sphere, to the data. Our system currently handles 15 different surface types.

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 (Besl, 1988) 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.

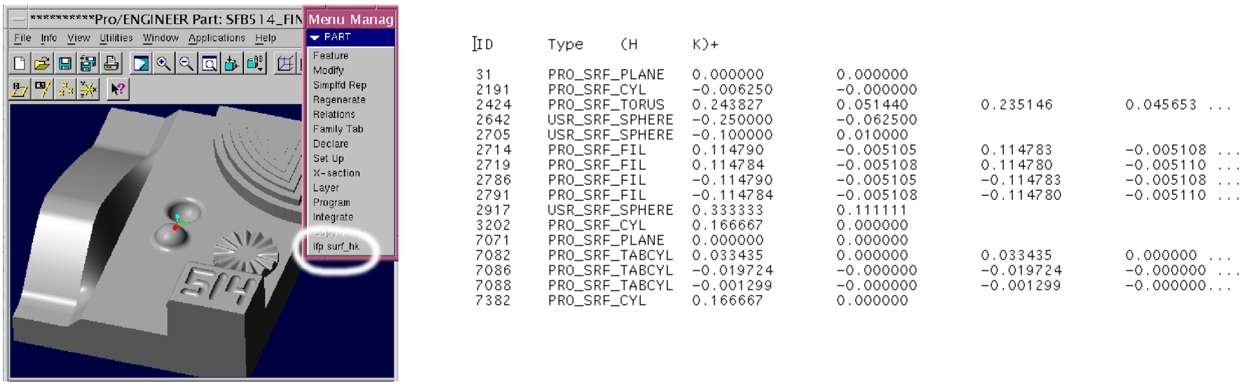
While this paper focuses on the feature extraction process, the object recognition itself is not shown here. We have previously reported on our approach using constrained tree search to match scene and model features. A similar approach can be combined with the feature extraction presented in this work.

2 DERIVING FEATURES FROM 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 2(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 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 3(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



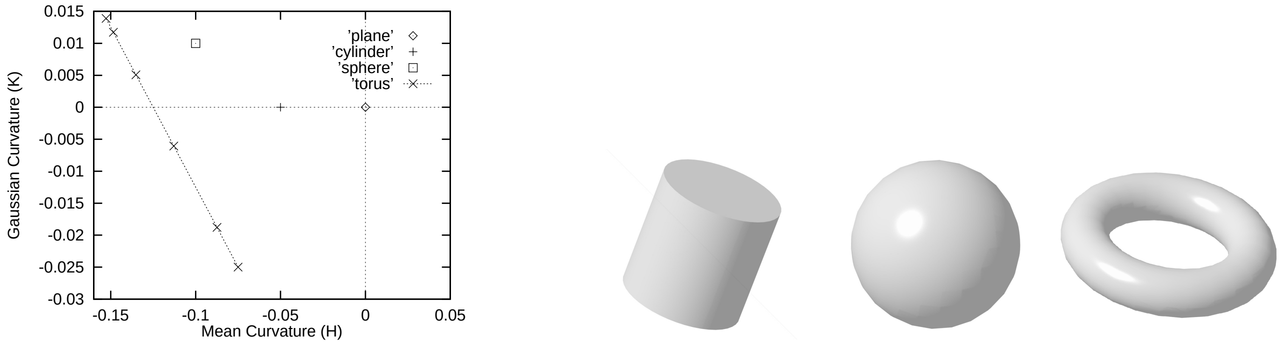
```
ID        Type      (H        K)+

 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
```

(a) screenshot of the CAD system

(b) Example output

Figure 2: The algorithm is fully integrated into the CAD system.



(a) plot in HK space

(b) cylinder

(c) sphere

(d) torus

Figure 3: Some typical surfaces and their signature in HK space.

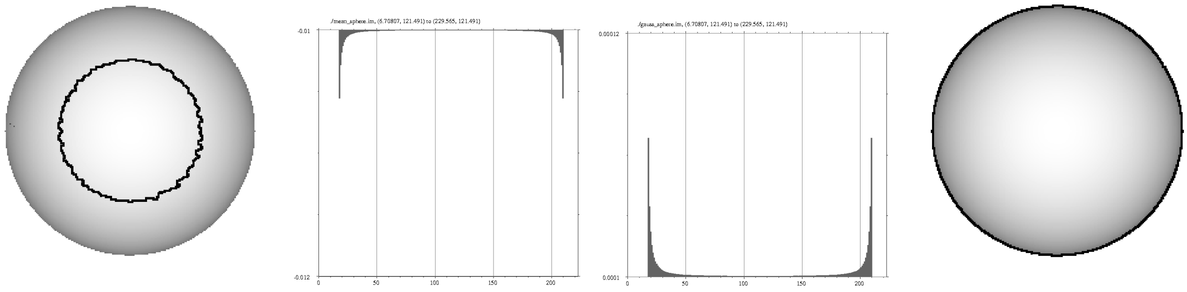
sphere. For other surface types we retrieve the parametric representation $F : x(u, v)$ of the surface. We evaluate the surface at a discrete set of points $\{(u_1, v_1), ..., (u_i, v_i)\}$. The coordinates $[x, y, z]$ of the point, the surface normal $\mathbf{N} = [N_x, N_y, N_z]$ and the derivatives $\mathbf{x}_u, \mathbf{x}_v, \mathbf{x}_{uu}, \mathbf{x}_{uv}, \mathbf{x}_{vv}$ 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 4, see for example (do Carmo, 1976). 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 2(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.

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

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

Figure 4: Formulas for derivation of curvature.



(a) surface fitting using implicit coordinate system

(b) plot of mean curvature across the sphere

(c) plot of Gaussian curvature across sphere

(d) local coordinate system aligned to surface normal

Figure 5: Curvature on a sphere. The images display the z (depth) values. The superimposed marking indicate the areas where mean and Gaussian Curvature have correct values. The quality of curvature estimation can be greatly increased when a local coordinate system for surface fitting is used which is aligned to the normal of the surface.

3 DERIVING FEATURES FROM RANGE IMAGES

3.1 Curvature

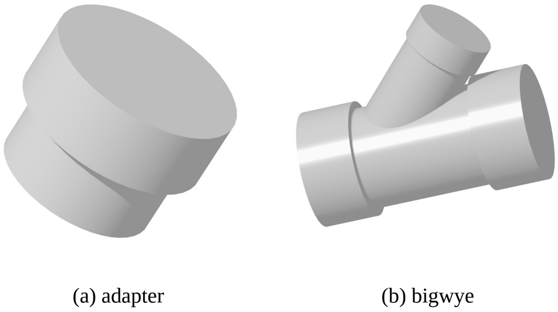
As for the CAD model the curvature has also to be computed for the range image. Several schemes have been proposed for curvature estimation. Some techniques were developed especially for triangulated surfaces. However, these techniques usually use only a very small neighborhood of the inspected point 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 very fast implementation of this approach has been reported by Paul J. Besl (Besl, 1988). 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 is 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 errors in surface fitting.

One way to solve this problem is to use implicit polynomials of the form $F(x, y, z) = 0$. However true orthogonal distance regression of implicit polynomials is a non-linear process as shown by Nicholas J. Redding (Redding, 2000). Our way to overcome the problem is to transform the points to a local coordinate system which has its z axis aligned to the local Normal vector as proposed in (Flynn and Jain, 1989). This also requires additional computations. First the local normal vector has to be estimated. Second all the data points within the window have to be transformed. Figures 5(a) through (c) show some experiments with a synthetic sphere. At the center of the sphere the surface normal is in alignment to the z axis of the implicit coordinate system. Therefore the local surface fitting is correct and so are the derived values for mean and Gaussian curvature. But towards the edge of the sphere Gaussian curvature starts to deviate greatly from the theoretical value. This can be prohibited with the proposed method of local transformation as shown in figure 5(d).

3.2 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 distance of this feature vector to each of the features derived from the CAD model is



(a) adapter (b) bigwye

Figure 6: Shaded view of the CAD models.

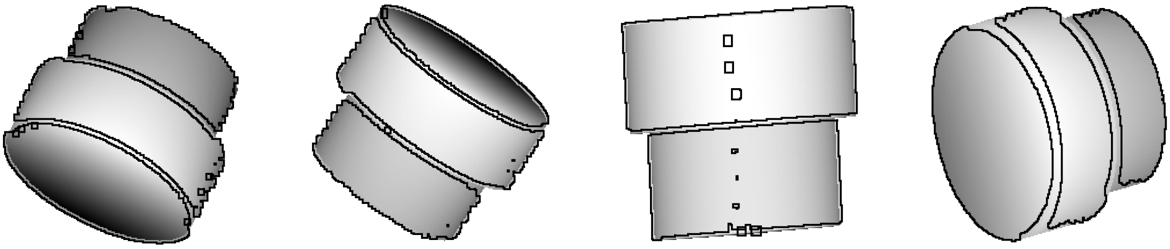


Figure 7: Results on test dataset “adapter” at different poses.

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. The assignment is thresholded to prevent severe misclassification. Pixels which were not assigned any label remain 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 (Campbell and Flynn, 1998). 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 6(a).

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 6(b) 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 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

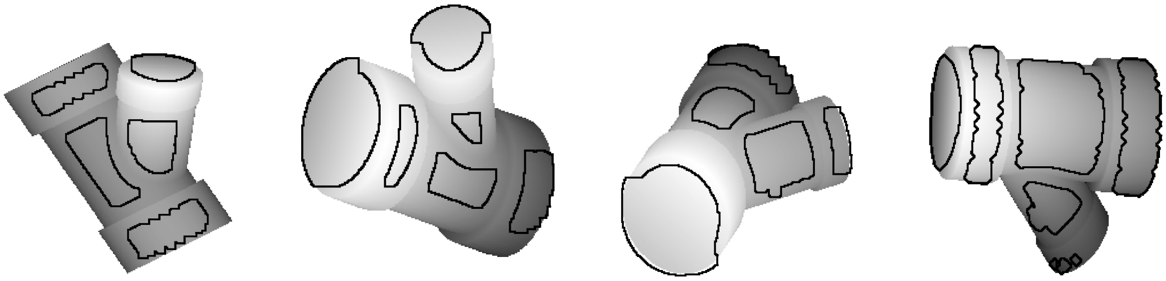
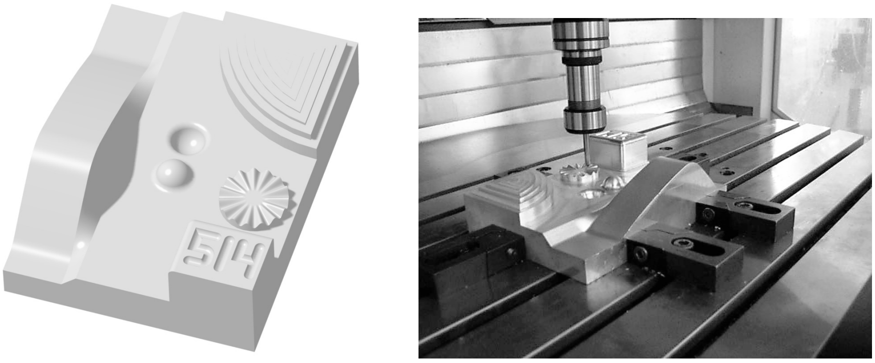


Figure 8: Results on test dataset “bigwye”. Surfaces can be detected independent on the pose of the object.



(a) shaded view of CAD model

(b) the part in the milling machine

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

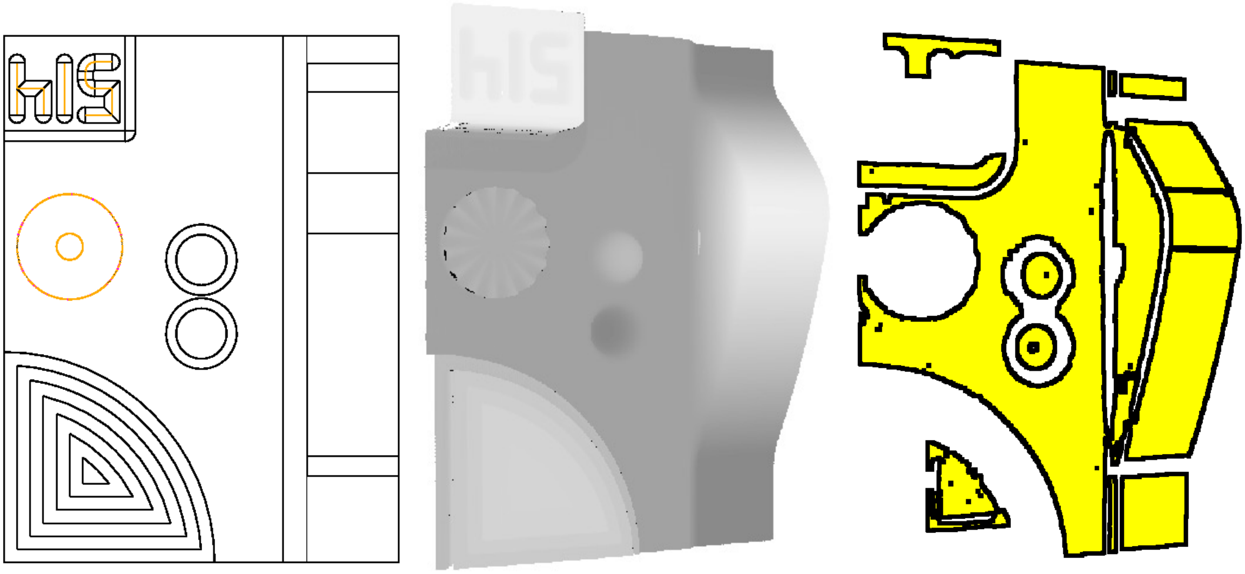
object including among others planar, spherical, cylindrical, and free-form surfaces. This model was then manufactured from an aluminum block using a high precision milling machine. The size of the part is approximately 20 cm x 15 cm. We obtained range images from the object using our ABW stripe projector 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 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 the 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 arbitrary objects.

6 ACKNOWLEDGMENTS

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

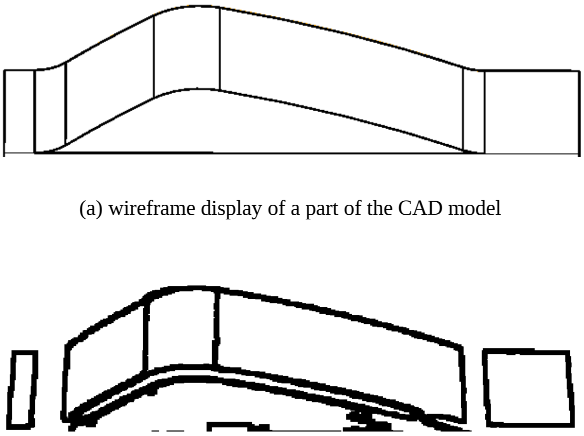


(a) wireframe display of the CAD model

(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 method was able to achieve a level of abstraction from the data comparable to that of the CAD data.

REFERENCES

Arman, F. and Aggarwal, J., 1993. Model-based object recognition in dense range images. Surveys 25(1), pp. 5–43.

Besl, P., 1988. Surfaces in Range Image Understanding. Springer-Verlag.

Böhm, J., Brenner, C., Gühring, J. and Fritsch, D., 1999. Cad-basierte objekterkennung für ein multisensorielles meßsystem. Mustererkennung 1999 pp. 62–69.

Campbell, R. and Flynn, P., 1998. A www-accessible image and model database for computer vision research. In: Proc. IEEE Workshop on Empirical Evaluation Methods in Computer Vision.

do Carmo, M., 1976. Differential Geometry of Curves and Surfaces. Prentice-Hall.

Flynn, P. and Jain, A., 1989. On reliable curvature estimation. In: IEEE Conference on Pattern Recognition, pp. 110–116.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P., Bunke, H., Goldgof, D., Bowyer, K., Eggert, D., Fitzgibbon, A. and Fisher, R., 1996. An experimental comparison of range image segmentation algorithms. PAMI 18(7), pp. 673–689.

Newman, T., Flynn, P. and Jain, A., 1993. Model-based classification of quadric surfaces. CVGIP 58(2), pp. 235–249.

Redding, N. J., 2000. Implicit polynomials, orthogonal distance regression, and the closest point on a curve. PAMI 22(2), pp. 191–199.