



# Model and Range Image Features for Free-Form Object Recognition

Richard Campbell      Patrick J. Flynn  
School of EECS  
Washington State University  
Pullman WA 99164-2752 USA  
rcampbel,flynn@eecs.wsu.edu

## Abstract

Recognition of free-form objects in range data is hampered by the difficulty of extracting robust and reliable features. This paper develops two approaches for detection of local features for recognition from both polyhedral mesh approximations to free-form geometry and from range data. Several experiments with models of varying complexity are proposed and specific recommendations are given for appropriate weighting of feature detectors based on the ‘quality’ of the polyhedral approximation.

## 1 Introduction

The topic of this paper is the detection of point and contour features on *free-form* 3D geometry and range images of that geometry. These features are intended to be used in a model-based free-form object recognition system. The area of model-based recognition has a rich history in the computer vision literature, leading most recently to a large number of research systems for model-based recognition from range data [1]. The recognition of objects best described as *free-form* has proved to be a formidable research problem, because of the lack of simple and segmentable descriptions of object geometry in models and images of those models. While general principles for free-form recognition have been proposed [3] and a few prototype systems have been developed [22, 18, 8], little sustained work on this problem has yet appeared in the literature. The lack of progress is particularly vexing given recent advances in computer-aided design technology, which has made free-form design more intuitive and rapid than in the early days of CAD. The development of computer aided design methods for arbitrary object geometries has

led to an increased use of more complex surfaces. These surfaces often offer better mechanical properties (reliability, weight, strength, aerodynamics) than similar parts made using the traditional surfaces (such as natural quadric surfaces). One reason for the interest in (and use of) free-form surfaces in manufacturing and design is that stress in a part concentrates in place where there are sharp corners. Part failure will often occur in these areas of high stress [9].

In computer aided design and manufacturing (CADM) systems, non-uniform rational B-spline surfaces (NURBS) are used to represent free-form surfaces. These parametric surfaces are specified in terms of *control points* and associated weights. The designer can move the control points or change the weights to manipulate the shape of the surface. This is useful when a designer is manipulating the surface to meet certain design specifications.

Besl [3] highlighted the difficulties with generating a matching scheme based on NURB surface types. Parametric surface forms like NURBS do not seem practical for direct use in a model-based recognition scheme. In most applications, occlusion and pose variations will make it difficult to determine the boundaries of the NURB surfaces visible in the image. Without a reliable method to *align* the model surfaces (patches) and the corresponding surfaces in the image, matching of NURBS based on detected control points and estimated weights is impractical.

Object models in model-based object recognition can be broadly classified either as *descriptive* (the representation can be used to generate synthetic imagery) or *discriminatory* (the representation can be used to distinguish between different objects, but not to generate synthetic imagery) [10]. Piecewise-NURBS objects and polygonal surface meshes are examples of descriptive models.

---

\*This research was supported by the National Science Foundation under grants IRI-9209212, IRI-9506414 and CDA-9422044.

Networks of ‘distinctive’ edges, points and regions obtained in a view-dependent or view-independent fashion are components of discriminatory models. In our work, we employ both types, and obtain the discriminatory model of an object from the descriptive model of the same object.

In this paper we will describe a new method for processing descriptive models (polygonal surface meshes) produced from CAD systems to obtain low level features for a discriminatory model database in an object recognition system. In Section 2 some existing strategies for free-form object recognition are described. Then we will introduce our notation for polygonal surface meshes (Section 3) and some measures of local surface orientation change on the mesh; these measures identify ‘salient’ points and curves on free-form objects and are the basis of our discriminatory model. We will then explore the sensitivity of these features to polygonalization methods (Section 4). Then we will describe a similar technique for extraction of compatible features in 3D images (range images) of 3D objects (section 5).

## 2 Review

Most object representations schemes are labeled view-dependent (view-centered) and/or object-centered. In a viewpoint-dependent representation, the information about the object encoded by this scheme is specialized to one of several views of the object. The overall representation is the union of features computed independently from multiple views of the object. It is typically assumed that the features from the objects in an input image will match more closely to one of the views in the corresponding model than any view of any other model. To insure this, these databases must encode large numbers of views to insure adequate coverage of the possible orientations of the object. This will typically cause both the object database and the number of database entries examined in the recognition process to be large.

One example of a viewpoint-dependent system for free-form object representation is the COSMOS system developed by Dorai and Jain (*e.g.*, [8]). This system uses a histogram of a local *shape index* to summarize a single view of an object. This histogram is computed independently from 320 views of the object. The similarity between an object in a range image and a view in the database is determined by calculating and comparing the first ten moments of the shape index histogram for the two views. This similarity measure is also used to cluster views in the database using a traditional clus-

tering algorithm. This system demonstrated good recognition ability but has yet to be tried on a large database on more than 100 objects and does not include studies of multiple objects in a scene.

Another category of object models is the viewpoint-independent representation. Features in this sort of model are designed to be invariant to translation and rotation; hence, the matching of these features will only be affected by occlusion. An example of this method is Stein and Medioni’s work based on “splash” features [18]. These local features encode the difference between the surface normal at the point of interest and the normal on a contour of constant geodesic radius away. This information is transformed into a periodic curve and encoded into a discriminatory model through a polygonal approximation. The representation of an object using these features is generated by identifying locations on the object which yield ‘interesting’ contours, such as curvature extrema. Each model in the database is stored as a list of the polygonal curves generated from the contour features around these curvature extrema. The recognition of an object from an unknown image is then determined by obtaining splash features from the image in question and searching the database for similar splashes. The database entry that best matches the splashes from the unknown image is determined to be the correct model. Their database of nine objects was built from images rather than a solid models (like a CAD model).

The family of Gaussian image representations (*e.g.*, [13]) are another view-independent family of descriptions for 3D models. The various representations used a discrete unit sphere as a basic data structure for storing information about the object being modeled or matched. The EGI maps surface patches from the object being modeled or matched to the discrete unit sphere. The elements of the unit sphere contain information on how many patches were mapped to the element thus histogramming the aggregate surface area as a function of orientation. This original representation was limited to convex objects and complete views of the object. Recent variants of the Gaussian Image representation alleviate these restrictions. The applicability of these representations to large databases has yet to be shown.

Edges or boundaries between different surface types can also be useful in 3D object recognition. Using surface normal and curvature measures the edges of an object (as determined by surface curvature sign and magnitude) can be used to obtain discriminatory information about the object.

Pankanti *et al.* [15] used a Hausdorff distance to align edges/boundaries of the image with an model edge map for each member of the database of images. The image in the database with the minimum distance will correspond to the most likely match to the object in the scanned image.

### 3 Polygonal Surface Meshes

Besl proposed the use of triangular polygonal surface meshes as a representation for objects in 3D computer vision [4]. This representation (although potentially verbose) is a complete (or ‘descriptive’ in our earlier discussion) object description, and yields Besl’s Crease Angle Histograms (CAH) global feature for free-form object recognition. These histograms contain information about the change in surface normal between adjacent polygons in the mesh. To compensate for varying levels of refinement of the object models the average area or edge length are added as weights to the angle measure. The histograms have been introduced as a possible method for indexing a feature vector, but the effects of occlusion on the proposed histogram matching procedure were not explored. We will begin to consider the effects of occlusion in this paper by using Besl’s crease angle measure to identify local salient features on object models to build a database of local features for model-based recognition.

In the sequel, we will use the polygonal surface mesh as the descriptive model for 3D objects to be recognized. Most modern CAD packages will produce a polyhedral approximation (of a specified resolution or accuracy) to an object being designed. Such representations are used in part since there are well known techniques to rapidly generate and visualize synthetic imagery from a mesh [21].

#### 3.1 Notation

The polygonal surface mesh for an object  $O$  described by:  $O = \langle V, E, F \rangle$  where  $V = \{v_i, i = 1 \dots n_v\}$  is a list of 3D vertices,  $E = \{e_j, j = 1 \dots n_e\}$  is a list of edges, and  $F = \{f_k, k = 1 \dots n_f\}$  is a list of polygonal facets. While a minimal description of a vertex  $v_i$  is simply its 3D coordinates  $(x_i, y_i, z_i)$ , it is valuable to also store with each vertex  $v_i$  the indices of neighboring vertices  $v_i'$  which terminate a facet edge beginning at  $v_i$ , the indices of the edges  $e_j$  for which  $v_i$  is an endpoint, the indices of the facets  $f_k$  which employ  $v_i$  in their edges, and a (perhaps approximate) outward pointing surface normal vector  $\mathbf{n}_i$ . Similarly, the minimal description of each edge  $e_j$  (a pair of vertex indices) is augmented by the indices of adjacent edges  $e_j'$ , the indices of facets  $f_k$  which employ  $e_j$ ,

and the edge length  $l_j$ . Likewise, each polygonal facet  $f_k$  (minimally expressed as a list of vertex indices or a list of edge indices) is actually described in terms of a list of defining vertices, an equivalent list of defining edges, the polygon’s area, and its surface normal  $\mathbf{n}_k$  (calculated using Tampieri’s version of Newell’s method [19]).

#### 3.2 Normal Change Measures

Our goal is to find local features for object recognition employing a correspondence mechanism. Hence we need techniques for processing both models and range images to extract corresponding features. The complexity of free-form geometry makes this problematic, but in keeping with previous work we will focus on locating curvature extrema, which can localize both crease edges (contours of curvature extrema, a common manufacturing feature) and isolated points of high curvature (peaks or pits, which are less common in manufacturing). Extending this to the detection of high-curvature *regions* is an area of future research. In this section we discuss the processing of polygonal mesh models.

We assume the features of interest are located at the vertices  $v_i$  or along the edges of the mesh  $e_j$ . To find interesting points and edges on the surface we use an approximation to surface curvature at these locations to filter the vertices and edges in the mesh. This could be viewed as the development of an edge detector for a polygonal mesh. Edge detectors and related techniques for range data has been explored for almost two decades, but little sustained work has been reported [14][20].

Besl [4] proposed the crease angle measure as an approximation of surface curvature localized at the edge for a polygonal surface mesh. The crease angle (CA)  $\theta_{CA_j}$  at an edge  $e_j$  is defined as the signed angle between the normals  $\mathbf{n}_{j1}, \mathbf{n}_{j2}$  of the two mesh facets  $f_{j1}, f_{j2}$  which share  $e_j$ . The sign is positive if the two polygons form a convex surface and negative otherwise. Two variants of the crease angle measure were also proposed to incorporate variations in the resolution of the mesh approximation.  $\theta_{CAA_j} = \frac{2\theta_{CA_j}}{A_{j1} + A_{j2}}$  normalizes the crease angle by the average area of the two incident facets.  $\theta_{CAL_j} = \frac{\theta_{CA_j}}{L_j}$  normalizes the crease angle by the length of the edge itself. Besl envisioned use of a histogram of the crease angles as an object view descriptor suitable for indexing; our use of the measures is lower-level, being a mechanism for identification of points and curves of interest.

The features above are associated with mesh edges; we developed the following features defined

at mesh vertices to supplement the edge-based features. Associated with each mesh vertex  $v_i$  and its normal  $\mathbf{n}_i$  is a set  $\mathcal{S}_i = \{(v'_i, \mathbf{n}'_i)\}$  of neighboring vertices sharing an edge with  $v_i$  and their normals. For vertices there will be two different measures of curvature localized at the vertex. The first measure  $\theta_{AMN_i}$  (Angle Measure Normal) is found by averaging the signed angles between  $\mathbf{n}_i$  and each of the  $\mathbf{n}_i \in \mathcal{S}_i$ . The second measure  $\theta_{AMCA_i}$  (Angle Measure Crease Angle) is the average of the crease angles of the edges  $\mathcal{E} = \{e_i\}$  that use  $v_i$  as an endpoint. We found it useful to consider weighted versions of these features.  $\theta_{AMNL_i}$  is a weighted version of  $\theta_{AMN_i}$  where each normal difference's contribution to the average is divided by the length of the corresponding edge.  $\theta_{AMCAL_i}$  is computed by averaging the values of  $\theta_{CAL}$  involving edges incident on  $v_i$  rather than values of  $\theta_{CA}$ .

## 4 Experimental Results

In the previous section we reviewed three existing procedures for computing a quantity related to surface curvature on polyhedron mesh edges, and proposed four vertex-based measures. In this section we will develop point and curve detectors based on these measures and examine their performance on a selection of polyhedral models. We will also describe a simple technique for crease edge detection in range imagery and apply it to range images of a subset of the same set of models, to demonstrate the compatibility between model and image features.

When processing models, one is tempted to use an extremely high resolution of the polygonal mesh in order to ‘capture’ all the curvature changes on the object. We recommend care in such an approach. Extremely dense polygon meshes require similarly extreme computation time to process for feature extraction, and the resulting features (which are expressed in terms of vertices and edges of the original mesh) will be verbose.

To demonstrate and compare the use of the angle measures a simple single threshold filter will be applied to the vertex-based and edge-based angle measures. If the magnitude of the angle measure under consideration is above the threshold, the vertex or edge is determined to be of “interest”. Histograms of the surface areas of the mesh polygons and the lengths of mesh edges can be used to choose between weighted and non-weighted angle measures. We will also demonstrate the utility of a hysteresis-based threshold detector for meshes; hysteresis was pioneered by Canny for intensity images [7]. As an enhancement to hysteresis we will

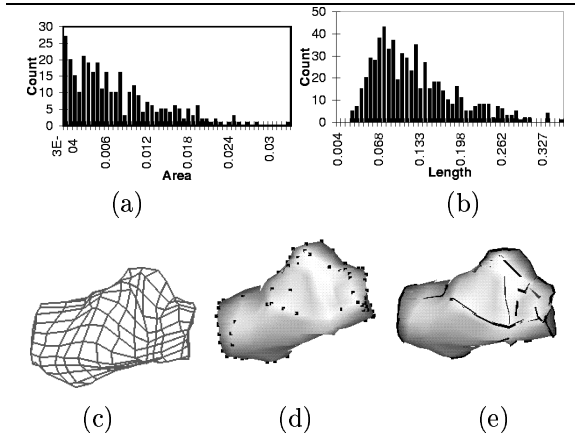


Figure 1: Calcaneus mesh. (a): Mesh facet area histogram. (b): Mesh edge length histogram. (c): Mesh. (d): Point features. (e): Edge features.

examine the use of non-maxima suppression [12] for polyhedral meshes.

The first experiment employed a free-form mesh model of a foot bone (Calcaneus) with 321 polygons. Figures 1(a) and 1(b) are histograms of histogram the mesh facet areas and edge lengths. Figure 1(c) shows the edges of the polyhedral mesh, and Figures 1(d) and 1(e) show point and line features (respectively), which are clustered primarily on crease edges or ridges of the mesh. The  $\theta_{AMN_i}$  angle measure was used to locate the points and  $\theta_{CA_i}$  was used to locate the edges. These features were chosen based on the presence of thick tails (*versus* a broad mode) in the face area and edge length histograms.

The next experiment highlights the need for a high-quality mesh generator. In this experiment a simple ellipsoidal disk was defined by the implicit equation  $x^2 + 16y^2 + z^2 - 1.0 = 0$  and a polygonal mesh was generated using Bloomenthal’s [5] implicit surface polygonizer. The mesh is generated by placing a fixed element size 3D lattice over the surface and finding the intersection between the lattice points and the surface. This is a disadvantage when fine partitioning is required for accuracy; polygons with small areas and edges with small lengths will appear everywhere in the mesh – even in places where the surface is nearly planar! This effect is shown in the histograms of facet area and edge length of the generated mesh (Figures 2(a) and 2(b)). The method produces a large number of small polygons (Figure 2(a)) for the ellipsoid surface, while producing almost a uniform distribution of edge lengths (Figure 2(b)). Table 1

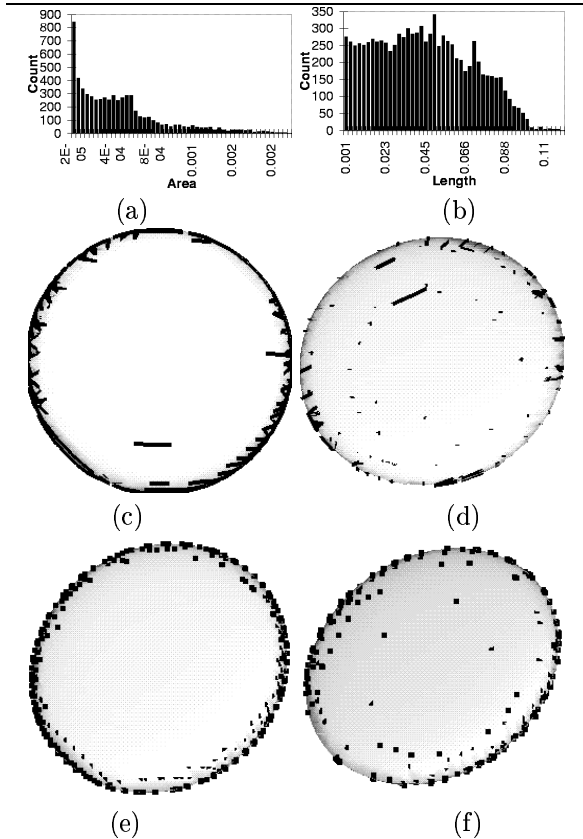


Figure 2: Implicit disk. (a): Facet area histogram. (b): Mesh edge length histogram. (c): Edge features ( $\theta_{CA_i}$ ). (d): Edge features ( $\theta_{CAA_i}$ ). (e): Point features ( $\theta_{AMN_i}$ ). (f): Point features ( $\theta_{AMNL_i}$ ).

lists the type of features sought, the number found, the angle measure used, and the figure number for six of the measures developed in this paper. The “MSD” column is explained below. The effects of the poor polygonization of the ellipse can be seen on the edges of the disk when we try to detect edge features by thresholding  $\theta_{CA_i}$  (Figure 2(c)) or  $\theta_{CAA_i}$  (Figure 2(d)). Even with a relatively dense mesh of 6127 polygons and using a conservative threshold to detect the edge features edges oriented toward the center will survive simple thresholding. For the edges oriented toward the center of the disk the magnitude of the curvature measure  $\theta_{CA_i}$  is as large as the edges oriented along the true edge of the disk. Figures 2(e) and 2(f) show the locations of point features obtained by thresholding two of the point angle measure outputs.

We conducted experiments with this ellipsoid because the location of the edge of the disk is known

Feature	Number	Method	Figure	MSD	Threshold
Edge	341	$\theta_{CA_i}$	2(c)	0.0019	0.504
Edge	346	$\theta_{CAA_i}$	2(d)	0.0013	1542
Point	335	$\theta_{AMN_i}$	2(e)	0.0027	0.337
Point	339	$\theta_{AMNL_i}$	2(f)	0.0532	7.735
Point	339	$\theta_{AMCA_i}$	-	0.0119	0.190
Point	338	$\theta_{AMCAL_i}$	-	0.0570	7.825

Table 1: Experimental parameters and measured MSD between disk edge and detected features

(its implicit equation is  $f_e(x, z) = (x^2 + z^2 - 1) = 0$ ). The features that we are detecting on the mesh can therefore be compared with the known location of the edge of the disk to find the mean squared distance (MSD) between the known edge and located features. If a vertex (point) feature with coordinates  $(x_d, y_d, z_d)$  is detected, its MSD to the correct edge is the squared algebraic distance obtained by plugging the recovered point into the implicit equation of the edge; if an edge defined by two vertices is labeled an ‘interesting’ edge, its MSD is the average of the two point-wise MSD’s of its endpoints. A table of experimental results for the disk are given in Table 1. The table also gives references to figures where the features are displayed and how many features were found on the polygonal surface mesh. The number of features is important when making a comparison between the different methods.

For Bloomenthal’s polygonization method the best feature measure is  $\theta_{AMN_i}$  (Figure 2(e)). This method for finding feature points does not place feature points in the center of the disk, since this method does the most averaging in finding the surface normals and therefore the difference of surface normals. With a fixed cell size the “noise” in the mesh is more noticeable. The other methods for finding feature points all yield points in the center of the disk, and also exhibit larger mean squared error values. The  $\theta_{AMCA_i}$  measure of curvature is more localized and less smoothed than  $\theta_{AMN_i}$ . When edge lengths are used to weight both  $\theta_{AMCA_i}$  and  $\theta_{AMN_i}$  the errors in the polygonization are magnified.

Models with superquadric surfaces have been proposed to represent natural forms [16][17]. The parametric form for these surfaces is given by:

$$\vec{x}(\eta, \omega) = \begin{bmatrix} a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\omega) \\ a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\omega) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix},$$

where  $\eta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$  and  $\omega \in [-\pi, \pi]$  are the paramet-

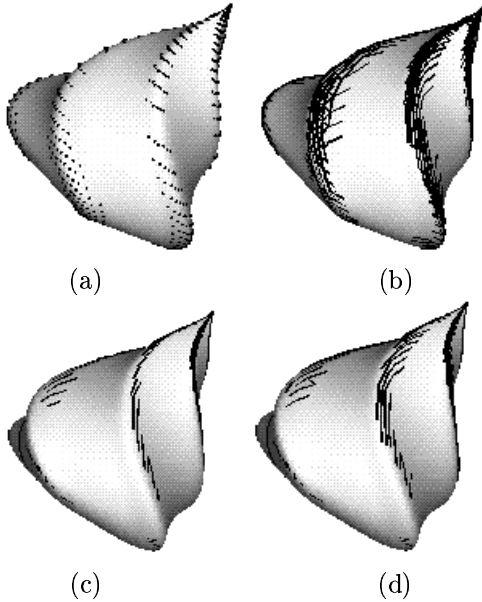


Figure 3: Superquadric with taper and twist deformations. (a): Point features. (b): Edge features (simple thresholding). (c): Edge features (upper threshold of hysteresis procedure). (d): Edge features (output of hysteresis procedure).

ric variables. This form of the superquadric surface easily yields a simple polygonization method. By sweeping out the parametric variables  $\eta$  and  $\omega$  in discrete steps over their domains the vertices, edges, and polygonal faces of the meshes are found. If the polygonization is done with uniform discrete steps in the parametric variables the mesh that is generated has some interesting properties. In the areas where the surface curvature is changing most rapidly, the areas of the polygonal faces and lengths of their edges is the smallest. This is used to our advantage when detecting edge and point features by using the angle measures divided by the length of an edge  $\theta_{AMNL_i}$ ,  $\theta_{AMCAL_i}$  and area of the polygons  $\theta_{CAA_i}$ . In that sense these three angle measures are well suited to meshes whose resolution adapts to surface complexity.

Deformations have been applied to superquadrics to enhance their ability to describe natural objects [2]. Figure 3 shows point and edge features detected on a superquadric shape ( $\epsilon_1 = 1.0$ ,  $\epsilon_2 = 0.3$ ) deformed by twisting and tapering transformations. While hysteresis-based thresholding did ‘clean up’ the curvature edge maps, a single contour was not produced.

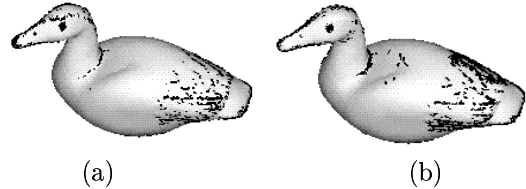


Figure 4: Duck Model (10,000 Polygons). (a): point features ( $\theta_{AMNL_j}$ ). (b): edge features ( $\theta_{CAA_i}$ ).

Earlier, we highlighted the value of edge- and area-weighted edge and vertex features for meshes of varying resolutions. Mesh optimization procedures (which generally attempt to reduce the number of facets in a mesh while preserving the ‘character’ of the surface) often yields a mesh with a wide range of facet sizes. Hence, the modified angle measures  $\theta_{CAA_j}$  and  $\theta_{AMNL_i}$  are appropriate feature detectors for optimized meshes.

Figure 4 summarizes our experiments with a polygonal mesh shaped like a duck. This model was assembled from multiple range views by the Innovmetric **Polyworks** software suite and reduced to 10,000 polygons from the original 150,000. Parts (a) and (b) of the figure show the detected point and edge figures, respectively.

## 5 Range Image Feature Detector

In this section we summarize a previously developed technique [6] for detection of crease and curvature edges in range data. This technique is demonstrated on several range images of the objects in our database to confirm that the features it detects and those the earlier angle measures produced are compatible for correspondence purposes.

A range image

$$\mathcal{R} = \{\vec{r}_{ij} = [x_{ij}, y_{ij}, z_{ij}]^T\}$$

is the input to this procedure, along with a surface normal image

$$\mathcal{N} = \{\vec{n}_{ij} = [n_{x_{ij}}, n_{y_{ij}}, n_{z_{ij}}]^T\}$$

calculated from  $\mathcal{R}$  using Flynn and Jain’s principal component method [11]. From the normal image we calculate an *angle image*

$$\mathcal{A} = \{(\theta_{r_{ij}}, \theta_{c_{ij}})\}$$

with components

$$\theta_{r_{ij}} = \pm \cos^{-1} \left( \frac{\vec{n}_{xz_{i,j-\frac{K-1}{2}}} \bullet \vec{n}_{xz_{i,j+\frac{K-1}{2}}}}{\|\vec{n}_{xz_{i,j-\frac{K-1}{2}}}\| \|\vec{n}_{xz_{i,j+\frac{K-1}{2}}}\|} \right)$$

$$\theta_{c_{ij}} = \pm \cos^{-1} \left( \frac{\vec{n}_{yz_{i-(\frac{K-1}{2}),j}} \bullet \vec{n}_{yz_{i+(\frac{K-1}{2}),j}}}{\| \vec{n}_{yz_{i-(\frac{K-1}{2}),j}} \| \| \vec{n}_{yz_{i+(\frac{K-1}{2}),j}} \|} \right),$$

where the negative sign is chosen for  $\theta_{r_{ij}}$  ( $\theta_{c_{ij}}$ ) if:

$$n_{x_{i,j-(\frac{K-1}{2})}} > n_{x_{i,j+(\frac{K-1}{2})}}$$

$$(n_{y_{i-(\frac{K-1}{2}),j}} > n_{y_{i+(\frac{K-1}{2}),j}}),$$

respectively and  $\vec{n}_{xz_{i,j}}$ ,  $\vec{n}_{yz_{ij}}$  are the normalized projections of the surface normals onto the  $xz$  and  $yz$  planes (respectively) for pixel  $(i, j)$ . The two angles  $\theta_{r_{ij}}$  and  $\theta_{c_{ij}}$  at pixel  $(i, j)$  measure the difference in the projected surface normals between two ends of a strip  $K$  pixels long centered at  $(i, j)$  and aligned with the rows or columns (respectively) of the images.

A combined measure  $\theta_{ij}$  of the orientation change in these two strips is then found from  $\theta_{r_{ij}}$  and  $\theta_{c_{ij}}$ :

$$\theta_{ij} = \cos^{-1} \left( 1 - \frac{\sin^2(\theta_{r_{ij}}) + \sin^2(\theta_{c_{ij}})}{2} \right)$$

This measure is only an approximation to the angle change of the normals in the direction of highest curvature. The angle measures  $\theta_{r_{ij}}$  and  $\theta_{c_{ij}}$  in the rows and columns may describe the angle change over different distances along the surface for a given pixel in the image. The different distances are caused from the varying distances between adjacent samples in the image.

To find areas of interest on an range image we will use the auxiliary *angle image*  $\mathcal{A}$  to determine pixels in the image where the surface orientation is changing rapidly. Our definition of rapid change for a pixel is:

$$\text{if } |\theta_{ij}| > Tc$$

where  $Tc$  is a preset threshold. For our test cases  $Tc = 0.7$  radians was used.

Figure 5 is a synthetic range image of the duck model processed earlier as a polygonal mesh. Dark pixels denote locations where the detector flagged ‘interesting’ pixels. Differences between the features marked in Figures 4 and 5 can be seen in the areas near the edges of the range images and where the object occludes itself. To eliminate this the local features in a range image that are to be used in comparison with the model features found through the normal change measures (Section 3.2) should be localized in regions away from object boundaries for the view.

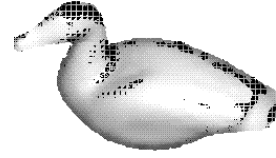


Figure 5: Duck range image: pixels of interest.

## 6 Conclusion

In this paper we have tested a number of feature detectors for both free-form objects and range images. Successful generic approaches to model-based free-form object recognition will likely employ the polygonal mesh as a primary low-level representation; hence features defined on the mesh can serve as the basis for correspondences in recognition. These estimates can be used to localize points and edges on the mesh where change in surface orientation is above a threshold indicating edges and points of ‘interest’. We have shown that depending on the type of surface polygonization different methods of estimating the models surface curvature should be used. For fixed grid polygonization the surface orientation change should be measured using the techniques not modified by the length of the edges or the area of the polygons. The meshes whose polygonal faces vary in area depending on if the polygon is near a region where the surface shape is change rapidly should use modified measures.

Future work will focus on the development and testing of a model-based recognition system which employs these features and higher-level features such as networks of curves for recognition and pose estimation. We also hope to improve these feature detectors. New techniques need to be devised for non-maxima suppression on polygonal meshes in order to thin the edges into a single contour. A set of these higher level features ideally should uniquely characterize the object from an other models in a database. The sets of higher level features in the database need to be compared with the features that can be found from range images to determine if the view-independent database is able to identify the correct free-from model in an image.

## 7 Acknowledgments

The duck and elephant model were provided by Marc Soucy, president of InnovMetric Software (<http://www.innovmetric.com>) and optimized using InnovMetric’s Polyworks software tools for 3D Imaging Systems. The foot bone mod-

els were obtained from the Viewpoint Datalabs WWW site (<http://www.datalabs.com>). Many of these models are also available in Washington State University's 3D model database (<http://www.eecs.wsu.edu/~irl/3DDB/>).

## References

- [1] Anil K. Jain and Patrick J. Flynn (eds.). *Three-Dimensional Object Recognition Systems*. Elsevier, Amsterdam, 1993.
- [2] A. H. Barr. Global and local deformations of solid primitives. *ACM Computer Graphics (Proc. SIGGRAPH '84)*, 18(3):21–30, 1984.
- [3] P. J. Besl. *Machine Vision For Three-Dimensional Scenes*, chapter The Free-Form Surface Matching Problem. Academic, 1990.
- [4] P. J. Besl. Triangles as a primary representation. In *Lecture Notes in Computer Science 994 (International NSF-ARPA Workshop)*, pages 191–206, New York City, NY, December 1994.
- [5] J. Bloomenthal. An implicit surface polygonizer. In P. S. Heckbert, editor, *Graphics Gems IV*, pages 324–349. AP Professional, Cambridge, MA, 1994.
- [6] R. Campbell. Range Image Segmentation Via Presegmentation and Natural Quadric Surface Identification. Master's thesis, School of Electrical Engineering and Computer Science, Washington State University, 1995.
- [7] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 8(6):679–698, Nov. 1986.
- [8] C. Dorai and A. K. Jain. View organization and matching of free-form objects. In *International Symposium on Computer Vision*, pages 25–30, Coral Gables, Florida, November 1995.
- [9] R. A. Flinn and P. K. Trojan. *Engineering Materials and Their Applications*. Houghton Mifflin Company, Boston, MA, 1990.
- [10] P. Flynn and A. K. Jain. Cad-based computer vision: From cad models to relational graphs. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(2):114–132, February 1991.
- [11] P. J. Flynn and A. K. Jain. Surface classification: Hypothesis testing and parameter estimation. In *Proc. IEEE Conf. Comput. Vision and Patt. Recog.*, pages 261–267, Ann Arbor, Michigan, 1988.
- [12] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGraw-Hill, Inc., 1995.
- [13] S. B. Kang and K. Ikeuchi. The complex egi: A new representation for 3-d pose determination. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(7):707–721, July 93.
- [14] A. Mitiche and J. K. Aggarwal. Detection of edges using range information. *IEEE Trans. Pattern Anal. Machine Intell.*, 5(2):174–178, March 1983.
- [15] S. Pankanti, C. Dorai, and A. K. Jain. Robust feature detection for 3d object recognition and matching. In *SPIE Conference on Geometric Methods in Computer Vision*, pages 1–, San Diego, July 1993.
- [16] A. P. Pentland. Perceptual organization and the representation of natural form. In M. A. Fischler and O. Firschein, editors, *Readings In Computer Vision: Issues, Problems, Principles, and Paradigms*, pages 680–699. Morgan Kaufmann Publishers INC, 95 First Street Los Altos, CA. 94022, 1988.
- [17] F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(2), February 1990.
- [18] F. Stein and G. Medioni. Structural indexing: Efficient 3-d object recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2):125–145, February 1992.
- [19] F. Tampieri. Newell's method for computing the plane equation of a polygon. In D. Kirk, editor, *Graphics Gems III*, pages 231–232. Academic Press, San Diego, 1992. includes code.
- [20] M. A. Wani and B. B. Batchelor. Edge-region-based segmentation of range images. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(3):314–319, March 1994.
- [21] A. Watt. *3D Computer Graphics*. Addison-Wesley, 1993.
- [22] B. W. York, A. R. Hanson, and E. M. Riseman. 3D Object Representation and Matching with B-Splines and Surface Patches. In *Proc. 7th. Int. Jt. Conf. Artificial Intell.*, pages 648–651, 1981.