Recognizing Objects by Matching Oriented Points

Andrew Edie Johnson and Martial Hebert
The Robotics Institute
Carnegie Mellon University
{aej,hebert}@ri.cmu.edu

Abstract

We present an approach to recognition of complex objects in cluttered 3-D scenes that does not require feature extraction or segmentation. Our object representation comprises descriptive images associated with each oriented point on the surface of an object. Using a single point basis constructed from an oriented point, the position of other points on the surface of the object can be described by two parameters. The accumulation of these parameters for many points on the surface of the object results in an image at each oriented point. These images, localized descriptions of the global shape of the object, are invariant to rigid transformations. Through correlation of images, point correspondences between a model and scene data are established and then grouped using geometric consistency. The effectiveness of our algorithm is demonstrated with results showing recognition of complex objects in cluttered scenes with occlusion.

1. Introduction

For recognition of complex objects, we have developed a representation that combines the descriptive nature of global object properties with the robustness to partial views and clutter of local shape descriptions. Specifically, a local basis is computed at an oriented point (3-D point with surface normal) on the surface of an object represented as a polygonal surface mesh. The positions with respect to the basis of other points on the surface of the object can then be described by two parameters. By accumulating these parameters in a 2-D array, a descriptive image associated with the point is created. Because the image describes the coordinates of points on the surface of an object with respect to the local basis, it is a local encoding of the global shape of the object and is invariant to rigid transformations. To prepare a model for recognition, an image is generated for each point on the model. Since an image is generated at each point in the surface mesh, error prone feature extraction and segmentation are avoided. At recognition time, images from points on the model are compared with images from points in the scene; when two images are similar enough, a point correspondence between model and scene is established. Several point correspondences are then used to calculate a transformation from model to scene for verification.

This research was supported by the US Department of Energy under contract DE-AC21-92MC29104.

Our recognition technique developed from a combination of basis geometric hashing proposed by Lamdan and Wolfson [10] and structural indexing proposed by Stein and Medioni [12]. Because we use information from the entire surface of the object in our representation, instead of a curve or surface patch in the vicinity of the point, our representation is more discriminating than the curves used to date in structural indexing. Furthermore, because bases are computed from single points, our method does not have the combinatoric explosion present in basis geometric hashing as the amount of points is increased. In our algorithm, every point on the model that is visible in the scene can be matched. This is in contrast to geometric hashing where only select feature points can be matched, making its effectiveness dependent on feature extraction.

The idea of encoding the relative position of many points on the surface of an object in an image or histogram is not new. Ikeuchi et. al. [9] propose invariant histograms for SAR target recognition. This work is view-based and requires feature extraction. Guéziec and Ayache [4] store parameters for all points along a curve in a hash table for efficient matching of 3-D curves. Their method requires the extraction of extremal curves from 3-D images.

Chua and Jarvis [2] present an algorithm for matching 3-D free-form surfaces by matching points based on principal curvatures. Similarly, Thirion [14] presents an algorithm for matching 3-D images based on the matching of extremal points using curvatures and Darboux frames. Pipitone and Adams [11] propose the tripod operator which, when placed on the surface of an object, generates a few parameters describing surface shape. Bergevin et. al. [1] propose a registration algorithm based on matching properties of triangles generated from a hierarchical tessellation of an object's surface. Our approach differs from these because the images computed at each point are much more discriminating than principal curvatures and angles between frames measured at a point. The descriptiveness of

1063-6919/97 $10.00

spin-images greatly reduces the number possible correspondences between points.

2. Spin-images

The fundamental shape element we use for matching is an **oriented point**, a three-dimensional point with an associated direction. We define an oriented point O on a surface mesh of an object using vertex position $\boldsymbol{p}$ and surface normal $\boldsymbol{n}$ (defined as the normal of the best fit plane to the point and its neighbors in the mesh oriented to the outside of the object). As shown in Figure 1, an oriented point defines a 2-D basis $(\boldsymbol{p},\boldsymbol{n})$ (i.e., local coordinate system) using the tangent plane $\mathcal{P}$ through $\boldsymbol{p}$ oriented perpendicularly to $\boldsymbol{n}$ and the line $\mathcal{L}$ through $\boldsymbol{p}$ parallel to $\boldsymbol{n}$. The two coordinates of the basis are α, the perpendicular distance to the line $\mathcal{L}$, and β the signed perpendicular distance to the plane $\mathcal{P}$. A *spin-map* S_O is the function that maps 3-D points $\boldsymbol{x}$ to the 2-D coordinates of a particular basis $(\boldsymbol{p},\boldsymbol{n})$ corresponding to oriented point O

$$S_O(x) \rightarrow (\alpha, \beta) = \left(\sqrt{\|x-p\|^2 - (n\cdot(x-p))^2},\ n\cdot(x-p)\right) \quad (1)$$

The term spin-map comes from the cylindrical symmetry of the oriented point basis; the basis can spin about its axis with no effect on the coordinates of points with respect to the basis.

Each oriented point O on the surface of an object has a unique spin-map S_O associated with it. When S_O is applied to all of the other points on the surface of the object $\mathcal{M}$, a set of 2-D points is created. We will use the term **spin-image** $I_{O,\mathcal{M}}$ to refer to the result of applying the spin-map S_O to the set of points on $\mathcal{M}$. A spin-image is a description of the shape of an object because it is the projection of the relative position of 3-D points that lie on the surface of an object to a 2-D space where some of the 3-D metric information is preserved. Since spin-images describe the shape of an object independently of its pose, they are object centered shape descriptions.

Correspondences are established between oriented points by comparing spin-images. If spin-images are

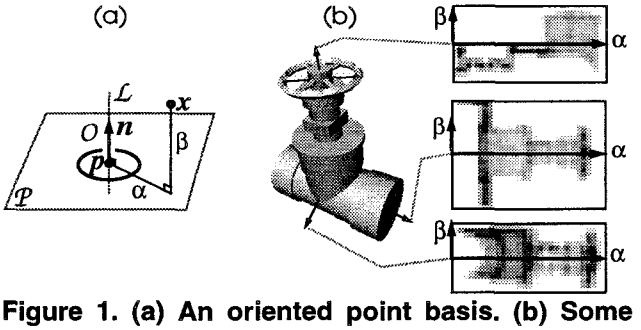

Figure 1. (a) An oriented point basis. (b) Some example spin images generated for three different oriented points on a CAD model of a valve.

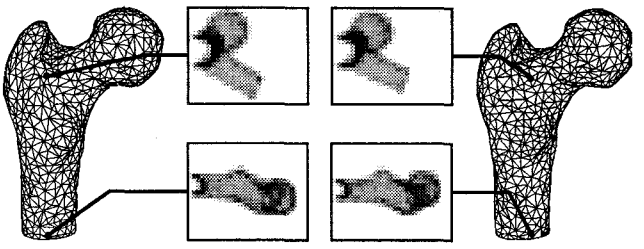

Figure 2. Spin-images generated from different samplings of a model of a femur are similar.

represented as a set of 2-D points then comparisons will have to be made between points sets: a costly and ill-defined operation. Instead, as explained below, spin-images are represented as images that are compared through correlation.

To create the spin-image for the oriented point O on the surface of an object $\mathcal{M}$, the following procedure is invoked. For each point $\boldsymbol{x}$ on the surface of the object, the spin-map coordinates (α,β) with respect to O are computed. Next, the pixel P that the coordinates index in the image is determined by discretizing (α,β). Finally, the array is updated by incrementing the pixels surrounding P in the image. In order to blur the position of the point in the histogram to account for noise in the data and the discrete sampling of the surfaces in the scene, the contribution of the point is bilinearly interpolated to the four pixels surrounding (α,β). In general, the pixel size is set to two times the resolution of the surface mesh (measured as the average of the edge lengths in the mesh). Figure 1 shows some spin images for a CAD object. The darker the pixel, the more points have fallen into that particular bin.

Because a spin-image is a global encoding of the surface, it would seem that any disturbance such as clutter and occlusion would prevent matching. In fact, this representation is resistant to clutter and occlusion, assuming that some precautions are taken. This will be described in detail in Section 4..

3. Comparing spin-images

Spin images generated from the scene and the model will be similar because they are based on the shape of objects imaged. However, they will not be exactly the same due to variations in surface sampling and noise from different views. For example, in Figure 2 the vertex positions and connectivity of two models of a femur are different, yet the spin-images from corresponding points are similar. A standard way of comparing linearly related images is the correlation coefficient. Because the correlation coefficient can be used to rank point correspondences, correct and incorrect correspondences can be differentiated.

The linear correlation coefficient provides a simple way to compare two spin-images that can be expected to be

similar across the entire image. In practice, spin images generated from range images will have clutter (extra data) and occlusions (missing data). A first step in limiting the effect of clutter and occlusion, is to compare spin images only in the pixels where both of the images have data. In other words, the pixels used to compute the linear correlation coefficient are taken only from the region of overlap between two spin images.

Since the linear correlation coefficient is a function of the number of bins used to compute it, the amount of overlap will have an effect on the correlation coefficients obtained. The more bins used to compute the correlation coefficient, the more confidence there is in its value. The variance of the correlation coefficient is included in the calculations of the similarity between two images so that the similarity measure between pairs of images with differing amounts of overlap can be compared. An appropriate similarity function C which we use instead of the correlation coefficient to compare spin-images P and Q where N is the number of overlapping bins is

$$C(P,Q) = (\text{atanh}(R(P,Q)))^2 - \lambda\left(\frac{1}{N-3}\right) \quad (2)$$

This similarity function will return a high value for two images that are highly correlated and have a large number of overlapping bins. The change of variables, a standard statistical technique ([3] Chapter 12) performed by the hyperbolic arctangent function, transforms the correlation coefficient into a distribution where the variance is independent of the mean. In Equation 2, λ is a free variable used to weight the variance against the expected value of the correlation coefficient. In practice λ is set to three.

4. Limiting the effect of clutter and occlusion

In real scenes, clutter and occlusion are omnipresent. Any object recognition system designed for the real world must somehow deal with clutter and occlusion. Some systems perform segmentation before recognition in order to separate clutter from interesting object data. In our case, the effects of clutter are manifested as a corruption of the pixel values of spin-images generated from the scene data. To some extent, the effect of clutter and occlusion can be limited by setting two thresholds that determine which points contribute to spin-image generation. The first threshold sets the maximum distance between the oriented point basis and a point in the mesh contributing to the spin-

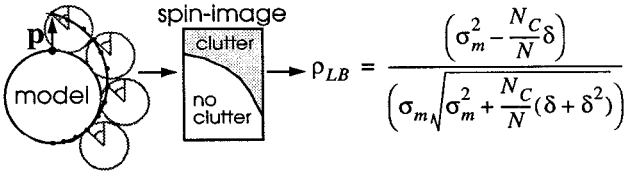

Figure 3. Theoretical clutter model.

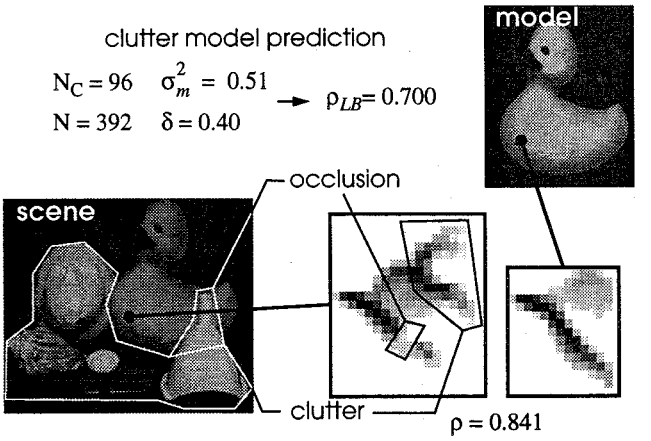

Figure 4. Verification of clutter model.

image. This parameter localizes the support of the spin-image to a sphere around its oriented point. In general this distance threshold is set to the size of the model (average distance of points on the model from its centroid). The second threshold sets the maximum angle between the oriented point basis surface normal and the surface normal of other points on the surface. This threshold prevents most points that will be self-occluded from contributing to the spin-image without specifying a viewing direction. This angular threshold is usually set to 90 degrees.

In order to analyze the effects of clutter, we have developed a simple model of the effect of clutter on spin-images under the assumption that objects are spherical. The clutter model combines the angular and distance thresholds explained above with the fact that objects of non-zero thickness cannot intersect to show that clutter is limited to connected regions in spin-images. Because of limited space, we cannot include a derivation of the clutter model, but our approach to clutter analysis is sketched in Figure 3.

Clutter and occlusion manifest as extra and missing points in the scene where the number of these points is bounded. Therefore, it is reasonable to assume that the total change of any pixel in a scene spin-image δ_i that is corrupted is bounded $|\delta_i| \le \delta$. Let the number of corrupted pixels in the scene spin-image be N_C and the total number of pixels be N. If the model and scene pixel values are normalized on [0,1], then the lower bound on the correlation coefficient when comparing model and scene spin-images is

$$\rho_{LB} = \left(\sigma_m^2 - \frac{N_C}{N}\delta\right) \Big/ \left(\sigma_m\sqrt{\sigma_m^2 + \frac{N_C}{N}(\delta+\delta^2)}\right) \quad (3)$$

where σ_m^2 is the variance of the pixels in the model spin-image. Hence, the worst case effect of clutter and occlusion grows sub-linearly with the area of corruption in the scene spin-image. Since clutter and occlusion cannot corrupt an entire spin-image and the effect of the corruption on the correlation coefficient is bounded, it can be concluded that matching of spin-images is only moderately affected by

clutter and occlusion. Figure 4 validates our clutter model using spin-images from a a real scene with clutter and occlusion. 96 of 392 pixels in the scene spin-image are corrupted by an amount δ less than 0.40. The correlation coefficient for the two images (0.841) is well above the lower bound (0.700) predicted by the clutter model.

5. Generating point correspondences

The similarity measure (Equation 2) provides a way to rank correspondences so that only reasonable correspondences are established. Before recognition (off-line), spin-images are generated for all points on the model surface mesh and stored in a **spin-image stack.** At recognition time, a scene point is selected randomly from the scene surface mesh and its spin-image is generated. The scene spin-image is then correlated with all of the images in the model spin-image stack and the similarity measures (Equation 2) for each image pair are calculated and inserted in a histogram. As explained below, the images in the model spin-image stack with high similarity measure when compared to the scene spin-image produce model/scene point correspondences between their associated oriented points. This procedure to establish point correspondences is repeated for a random sampling of scene points that adequately cover the scene surface. Depending on the complexity and amount of clutter on the scene, this number can vary between one tenth and one half of the points in the scene. The end result is a list L of likely model/scene point correspondences that are then filtered and grouped in order to compute transformation from model to scene.

Possible corresponding model points are chosen by finding the upper outliers in the histogram of similarity measures for each scene point. This method of choosing correspondences is reliable for two reasons. First, if no outliers exist, then the scene point has a spin-image that is very similar to all of the model spin-images, so definite correspondences with this scene point should not be established. Second, if multiple outliers exist, then multiple model points are similar to a single scene point, so should be considered in the matching process. We use a standard method for detection of outliers in a histogram ([3] Chapter 1); correspondences that have similarity measures that are greater than the upper fourth plus three times the fourth spread of the histogram are statistical outliers. Figure 5 shows a similarity measure histogram with detected outliers.

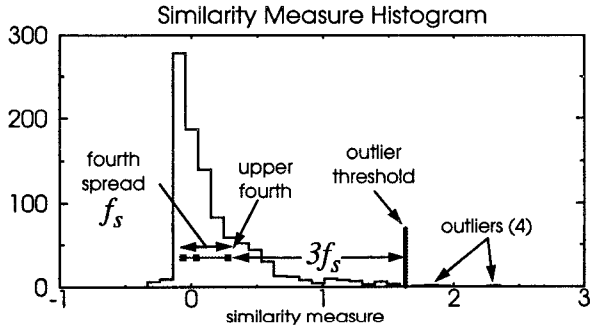

Figure 5. Similarity measure histogram.

During matching, a single point can be matched to more than one point for two reasons. First, symmetry in the data and in spin image generation may cause two points to have similar spin-images. Second, spatially close points may have similar spin-images. Furthermore, if an object appears multiple times in the scene, then a single model point will match multiple scene points.

To reduce computation and remove less likely correspondences, the correspondences in L are filtered based on similarity measure. Correspondences in L with similarity measures that are less than some fraction of the maximum similarity measure of the correspondences in L are eliminated. In practice, this fraction is set to one half. The end result is a list of correspondences that are the most likely to be correct. In practice this number is between 20 and 50 correspondences The next step is to group these correspondences into sets that can be used to compute transformations.

6. Computing transformations

Single correspondences cannot be used to compute a transformation from model to scene because an oriented point basis encodes only five of the six necessary degrees of freedom. We use geometric consistency to group correspondences into a few groups from which plausible transformations are computed. We use the spin-map coordinates (Equation 1) to measure the geometric consistency between two correspondences $C_1 = [s_1, m_1]$ and $C_2 = [s_2, m_2]$

$$d_{gc}(C_1, C_2) = 2\frac{\left\|S_{m_2}(m_1) - S_{s_2}(s_1)\right\|}{\left\|S_{m_2}(m_1) + S_{s_2}(s_1)\right\|} \qquad (4)$$

$$D_{gc} = max(d_{gc}(C_1, C_2), d_{gc}(C_2, C_1))$$

Normalized distance d_{gc} between spin-map coordinates is used because it is a compact way to measure the consistency in position and normals. Since d_{gc} is not symmetric, the maximum of the distances is used to define the geometric consistency measure D_{gc}.

We group correspondences based on a measure of geometric consistency W_{gc} that is the geometric consistency distance between two correspondences (Equation 4) augmented by a weight that promotes grouping of correspondences that are far apart.

$$w_{gc}(C_1, C_2) = \frac{d_{gc}(C_1, C_2)}{1 - e^{\left\|S_{m_2}(m_1) + S_{s_2}(s_1)\right\|/2}} \qquad (5)$$

$$W_{gc}(C_1, C_2) = max(w_{gc}(C_1, C_2), w_{gc}(C_2, C_1))$$

W_{gc} will be small when two correspondences are geometrically consistent and far apart. The measure of geometric consistency between a correspondence C and a group of correspondences $\{C_i, ..., C_n\}$ is

$$W_{gc}(C, \{C_1, ..., C_n\}) = \max_i (W_{gc}(C, C_i)) \quad (6)$$

Given a list of correspondences $L = \{C_1, ..., C_n\}$, the grouping procedure is as follows: For each correspondence C_i in L, initialize the group $G_i = \{C_i\}$ with one correspondence. Find the correspondence C_j in L, for which $W_{gc}(C_j, G_i)$ is a minimum. Add C_j to G_i if $W_{gc}(C_j, G_i) < T_{gc}$ where the threshold T_{gc} is set to the size of the model. Repeat until no more correspondences can be added to G_i. This grouping procedure is performed for each correspondence in L, resulting in n groups, one for each correspondence in L. This grouping algorithm allows a correspondence to appear in multiple groups which is necessary to handle model symmetry. For example, the CAD model in Figure 1 has a plane of symmetry resulting in two feasible transformations. Correspondences along the plane of symmetry contribute to two distinct transformations.

A plausible transformation from model to scene is calculated from each group of correspondences using the algorithm in [5]. Verification of transformations is performed by transforming the model surface mesh into the scene. Then for each model vertex, its closest scene vertex

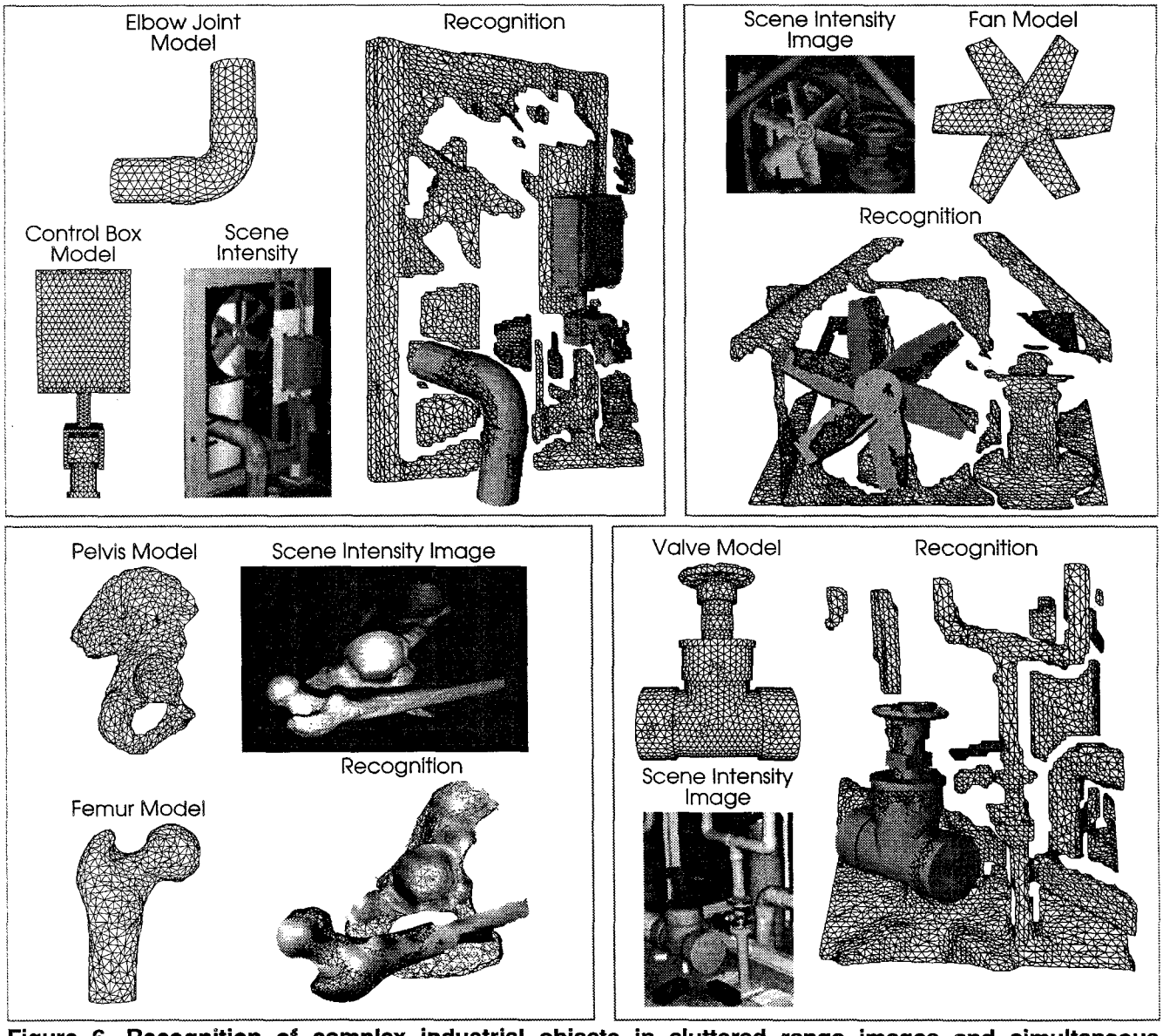

Figure 6. Recognition of complex industrial objects in cluttered range images and simultaneous recognition of models of a femur and pelvis in a range image.

in the six dimensional space of vertex positions and surface normals is determined. If a large number (e.g., one third the vertices in the model) of models points and corresponding closest scene vertices have a closest distance that is less than two times the mesh resolution of the model, then the transformation is verified because it brings a large number of model points in alignment with scene points.

7. Results

A strong property of our recognition algorithm is that it permits simultaneous recognition of multiple models. Recognition with multiple models is similar to recognition with one model except that each scene point is compared to the spin-images of all of the models. The rest of the algorithm is the same except that correspondences with model points from different models are prevented from being clustered.

Figure 6 demonstrates the simultaneous recognition of two models of free-form shape. The femur and pelvis model were acquired using a CT scan of the bones, followed by surface mesh generation from contours. The scene was acquired using a K^2T structured light range camera. Both the models and scene data were processed by removing long edges associated with step discontinuities, applying a "smoothing without shrinking" filter [13], and then applying a mesh simplification algorithm that preserves the shape of objects in the scene while evenly distributing the points over the its surface [7]. Our algorithm was able to recognize the objects even in the presence of extreme occlusion; only 20% of the surface of the pelvis is visible.

Our main application domain is interior modeling. In interior modeling, objects are recognized in range images of complex industrial interiors. By recognizing objects, a semantic meaning is associated with the objects in the scene, setting the stage for high-level robotic interaction. For example, by recognizing a valve in the scene, a robot can be given a high-level commands such as "turn off the valve"[8].

Figure 6 shows the result of recognizing four different industrial objects in cluttered industrial scenes. The surface mesh models were generated by CAD drawings using finite element software to tessellate the surface of the objects. The scene images were acquired with a Perceptron 5000 scanning laser rangefinder. Before recognition, the scene data is processed to remove long edges and small surface patches, smoothed and simplified. In all examples, the scene data is complex with a great deal of clutter. Furthermore, all the models exhibit symmetry which makes the recognition more difficult, because a single scene point can match multiple model points.

In addition to these results, we have generated results from multi-view merging and alignment of terrain maps [6].

8 Future work

In the future, we will extend our algorithm to recognize multiple objects simultaneously from a library of models. This will require efficient methods for determining which models in the library are present in the scene. We have determined that, due to redundancy, the spin-images for a model lie in a low dimensional subspace in the high-dimensional spin-image space. Through the use of principal component analysis, this subspace can be computed. Rapid determination of which models appear in the scene can then be obtained by projection of scene spin-images onto subspaces generated for each model in the library.

Acknowledgments

We would like to thank Jim Osborn and all the members of the Artisan project for supporting this work.

References

[1] R. Bergevin, D. Laurendeau and D. Poussart, "Registering range views of multipart objects," *Computer Vision Image Understanding*, 61(1):1-16, 1995.

[2] C. Chua and R. Jarvis, "3-D free-form surface registration and object recognition," *IJCV*, 17(1):77-99, 1996.

[3] J. Devore, *Probability and Statistics for Engineering and Sciences*, Brooks/Cole, Belmont, CA, 1987.

[4] A. Guéziec and N. Ayache, "Smoothing and matching of 3-D space curves," *IJCV*, 12(1):79-104, 1994.

[5] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Optical Soc. Amer.*, 4(4):629-642, 1987.

[6] A. Johnson and M. Hebert, "Surface registration by matching oriented points," *Proc. Int'l Conf. Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97)*, 1997.

[7] A. Johnson and M. Hebert, "Control of mesh resolution for 3-D Computer Vision," *CMU Robotics Institute TR, CMU-RI-TR-96-20*, December 1996.

[8] A. Johnson, R. Hoffman, J. Osborn and M. Hebert, "A system for semi-automatic modeling of complex environments," *Proc. Int'l Conf. Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97)*, 1997.

[9] K. Ikeuchi, T. Shakunaga, M. Wheeler and T. Yamazaki, "Invariant Histograms and deformable template matching for SAR target recognition," *Proc. Computer Vision and Pattern Recognition (CVPR 1996)*, pp. 100-105, 1996.

[10] Y. Lamdan and H. Wolfson, "Geometric Hashing: a general and efficient model-based recognition scheme," *Proc. Second Int'l Conf. Computer Vision (ICCV '88)*, pp. 238-249, 1988.

[11] F. Pipitone and W. Adams, "Tripod operators for recognizing objects in range images; rapid rejection of library objects," *IEEE Robotics and Automation (R&A 1992)*, pp. 1596-1601, 1992.

[12] F. Stein and G. Medioni, "Structural Indexing: efficient 3-D object recognition," *IEEE PAMI*, 14(2):125-145, 1992.

[13] G. Taubin, "A Signal processing approach to fair surface design," *Proc. Computer Graphics 1995 (SIGGRAPH '95)*, pp. 351-358, 1995.

[14] J. Thirion, "New feature points based on geometric invariants for 3D image registration," IJCV, 18(2):121-137, 1996.