ELSEVIER

Image and Vision Computing 16 (1998) 635–651

image AND vision COMPUTING

Surface matching for object recognition in complex three-dimensional scenes

A.E. Johnson*, M. Hebert

The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Received 10 February 1997; revised 5 September 1997; accepted 13 October 1997

Abstract

We present an approach to recognition of complex objects in cluttered three-dimensional (3D) scenes that does not require feature extraction or segmentation. Our object representation comprises descriptive images associated with oriented points 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. Geometric consistency is used to group the correspondences from which plausible rigid transformations that align the model with the scene are calculated. The transformations are then refined and verified using a modified iterative closest point algorithm. The effectiveness of our representation comes from its ability to combine the descriptive nature of global object properties with the robustness to partial views and clutter of local shape descriptions. The wide applicability of our algorithm is demonstrated with results showing recognition of complex objects in cluttered scenes with occlusion.

Keywords: 3D object recognition; Surface matching; Point correspondences

1. Introduction

We have developed a representation for recognition of complex three-dimensional (3D) objects, in 3D data, 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 (3D 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 two-dimensional (2D) array, a descriptive image associated with the oriented point is created. Because the image encodes the coordinates of points on the surface of an object with respect to the local basis, it is a local description of the global shape of the object and is invariant to rigid transformations. Since our representation comprises images that describe 3D points, we can apply powerful techniques from 2D template matching and pattern classification to the problem of 3D object recognition.

The images generated for two corresponding points will be similar, so oriented points can be matched based on a comparison of their associated images. The images are descriptive enough that correspondences between points can be established based on a comparison of images alone, even when the position of the points with respect to the object are not exactly the same. This is in contrast to interpretation-tree approaches to recognition [6] where combinatoric search is needed to match feature points which are assumed to be in the same position. Since images can be generated for any point on the surface of an object, our algorithm does not require or use feature extraction.

Furthermore, the method of comparison of these images is resistant to clutter and occlusion, so segmentation of scene data is not necessary for recognition in cluttered scenes. As shown in Fig. 1, during recognition, several scene points are matched to corresponding model points and a transformation that localizes the model in the scene is computed. The position of the model in the scene is subsequently verified and refined using a modified iterative closest point algorithm.

Our algorithm has been integrated into a complete robotic system including a mobile robot with an on-board range sensor and manipulator [10]. Using this system we have demonstrated the effectiveness of our algorithm in real world settings using tens of models and hundreds of scenes. In addition, the sensor independence of our algorithm has

* Corresponding author.

0262-8856/98/$19.00
PII S0262-8856(98)00074-2

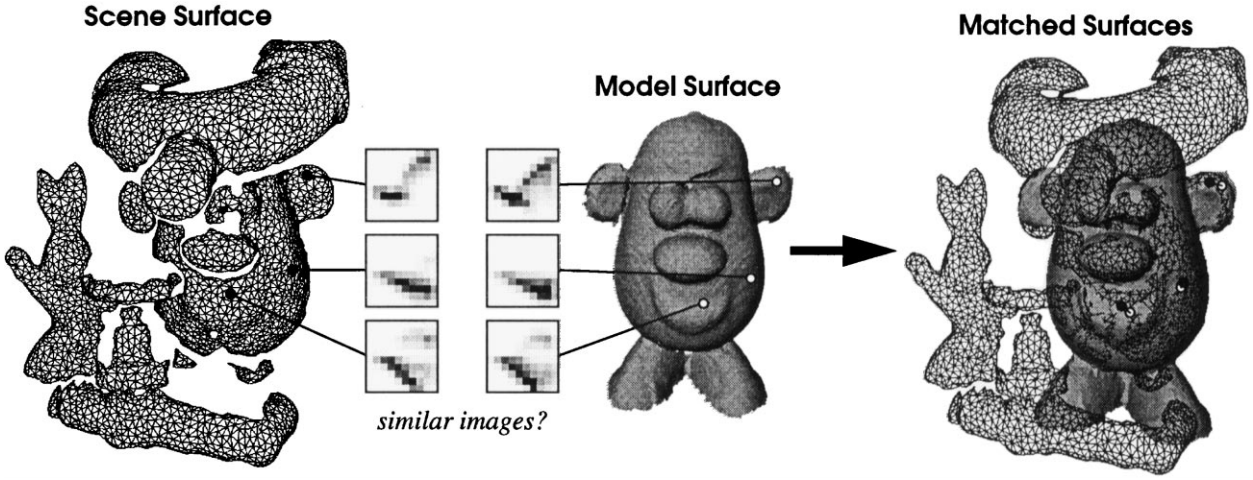


Fig. 1. Recognition algorithm overview. Images generated at oriented points on a model surface mesh are matched to images generated from a scene surface mesh. Matched images determine point correspondences from which a rigid transformation mapping the model into the scene can be computed.

been demonstrated using several different 3D sensing modalities [11].

Our recognition technique was developed from a combination of basis geometric hashing proposed by Lamdan and Wolfson [14] and structural indexing proposed by Stein and Medioni [17]. 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 selected 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. [13] propose invariant histograms for SAR target recognition. This work is view-based and requires feature extraction. Guéziec and Ayache [7] store parameters for all points along a curve in a hash table for efficient matching of 3D curves. Their method requires the extraction of external curves from 3D images.

Chua and Jarvis [3] present an algorithm for matching 3D free-form surfaces by matching points based on principal curvatures. Similarly, Thirion [19] presents an algorithm for matching 3D images based on the matching of extremal points using curvatures and Darboux frames. Pipitone and Adams [15] propose the tripod operator which, when placed on the surface of an object, generates a few parameters describing surface shape. Bergevin et al. [2] 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. Furthermore, dependence on computation of second order surface derivatives makes it difficult to robustly compute principal curvatures on surfaces.

The paper is organized as follows: first, we describe how the images used in matching oriented points are generated and compared. We then develop a theoretical model explaining the resistance of matching to scene clutter. Next, we describe how correspondences are grouped to compute plausible transformations from model to scene. We then explain our verification algorithm, which is based on the iterative closest point algorithm. Finally, we show recognition results in cluttered scenes and conclude with a discussion of the algorithm and future work.

2. Spin-images

The fundamental shape element we use to perform object recognition is an oriented point, a three-dimensional point with an associated direction. We define an oriented point O on the surface of an object using surface position p and surface normal n. As shown in Fig. 2, an oriented point defines a 2D basis (p,n) (i.e., local coordinate system) using the tangent plane $\mathcal{P}$ through p oriented perpendicularly

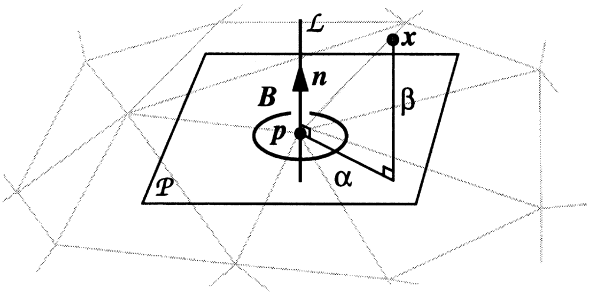


Fig. 2. Oriented point basis.

to n and the line $\mathcal{L}$ through p parallel to 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 $\mathcal{P}$. In other words, an oriented point basis is a cylindrical coordinate system that is missing the polar angle coordinate (because this coordinate cannot be determined using just surface position and normal). Using an oriented point basis, we can define a spin-map S_O as the function that projects 3D points x to the 2D coordinates of a particular basis (p,n) corresponding to oriented point $\mathcal{O}$

$$S_O : R^3 \rightarrow R^2$$

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

Although α cannot be negative, β can be both positive and negative.

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. A consequence of the cylindrical symmetry is that points that lie on a circle that is parallel to $\mathcal{P}$ and centered on $\mathcal{L}$ will have the same coordinates (α,β) with respect to the basis.

During recognition, we use coordinate systems defined at points on the surface of an object to describe the shape of the object independently of the object's pose. We use oriented point bases, which determine only two of the three coordinates of points, instead of complete 3D bases because an oriented point basis can be determined robustly and unambiguously almost everywhere on the surface of an object while a complete 3D basis cannot. An oriented point basis is well defined everywhere on the surface of the object except at surface discontinuities, where first order surface derivatives are undefined, so surface normal cannot be computed. When creating a complete three-dimensional coordinate system, three unambiguous axes must be determined. If surface normal, which can be computed reliably, is chosen as one of the axes, then two axes in the tangent plane of the surface must be determined to complete the basis. An obvious choice for these axes would be the directions of principal curvature on the surface [3] which when combined with the surface normal, create the Darboux frame of the surface. However, determination of the directions of principal curvature of a surface require the computation of second-order surface derivatives, so the resulting axes are very susceptible to noise. Furthermore, as the surface approaches a plane, the directions of principal curvature become undefined. Instead of attempting to calculate a stable 3D basis at each point, the known stable information (surface normal) is used to compute a 2D basis. Although a spin-map is not a rigid transformation, it can still be used to describe (albeit incompletely) the position of a point with respect to other points on the surface of an object. In the next section we will describe how we use this fact to encode the shape of objects in an object centered fashion.

In our recognition system, object models and scene data are represented as polygonal surface meshes. Surface meshes consist of 3D points located on the surface of the object (vertices) connected by edges. Although not necessary for our algorithm, surface meshes also have faces which determine the surface of the object between edges and vertices. We calculate the surface normal at each vertex by computing the eigenvector with the smallest eigenvalue of the inertia matrix of vertex and the other vertices directly connected to it by the edges of the surface mesh [5]. Since the sign of the eigenvector is ambiguous, the surface normals must be oriented to the outside of the object using the following heuristic. First, a vertex is chosen and the orientation of its normal is spread to the normals of all the adjacent vertices. This process is repeated until the normals of all of the vertices are consistently oriented to the inside or outside of the object. Next the orientation (inside/outside) of all of the normals on the surface is determined by calculating the scalar products surface normal at each vertex and the vector from the centroid of the object to the vertex. If the majority of scalar products are positive, the normals are oriented to the outside. Otherwise, the normals have been oriented to the inside, so they are inverted. If the object has multiple connected components, this normal orientation procedure is applied separately to each connected component. To date we have never encountered an object where this heuristic will not generate outside oriented surface normals, although objects can be constructed where it will fail. Given this method for computing surface normal, an oriented point basis can be constructed at each vertex of a surface mesh using the position of the vertex and its surface normal.

2.1. Spin-image generation

Each oriented point $\mathcal{O}$ 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 vertices of a surface mesh $\mathcal{M}$ a set of 2D points is created. In this case, the vertices on the surface of $\mathcal{M}$ are the pre-image of the spin-map, and the resulting 2D points are the image of the spin-map. We will use the term spin-image $I_{O,M}$ to refer to the result of applying the spin-map S_O to the vertices of $\mathcal{M}$. A spin-image is a description of the shape of an object because it is the projection of the relative position of 3D points that lie on the surface of an object (vertices) to a 2D space where some of the 3D metric information is preserved. Fig. 3 shows three 2D point sets that result from applying the spin-map at a point on the surface of a duckie to the vertices on the duckie surface mesh. Since spin-images describe the relative position of points on a rigid object with respect to a particular point on that same object, spin-images are independent of rigid transformations applied to the object and, therefore, are truly object-centered shape descriptions.

Suppose there exists a method for comparison of spin-images. A simple recognition scheme based on matching

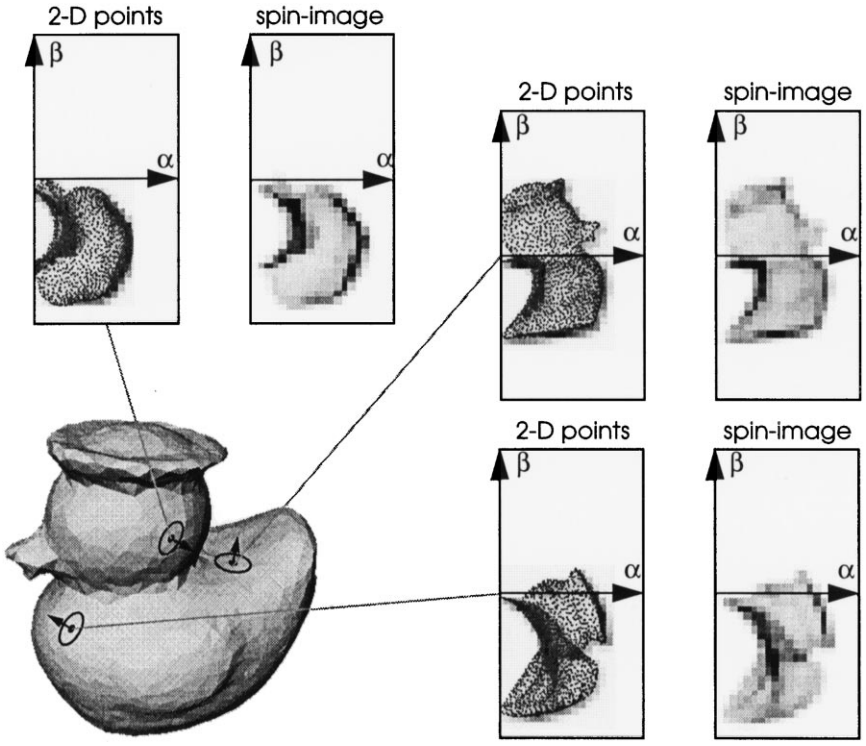


Fig. 3. Spin-images for three oriented points on the surface of a rubber duckie model. The 3D position of vertices in the mesh are mapped into 2D using the spin-map for each oriented point basis. By accumulating 2D points, in discrete bins, spin-images are generated.

spin-images is then possible. Spin-images from vertices in the scene are compared to spin-images from vertices in the model data base; when a good match occurs, a correspondence between the model vertex and the scene vertex is established. With three or more point correspondences a rigid transformation between model and scene can be calculated and verified. The problem is then to find an efficient way to compare spin-images.

Variations in viewing direction, sensing modalities and surface mesh construction all combine to prevent the position and connectivity of vertices in the scene and vertices on the model from being the same. Any algorithm that assumes that vertices will have similar positions between model and scene cannot be robust because this assumption cannot be guaranteed. However, the global shape of an object represented by two different surface meshes will be the same. Therefore, when comparing spin-images, the goal is to compare global shape conveyed by the two images while not being distracted by local variations.

To encode global shape while reducing the effects of local variations in vertex position, we originally approached the problem of matching spin-images with a method based on geometric hashing [14]. Our initial algorithm used a hash table to encode geometric constraints between points; indices to oriented point bases were stored in the hash table in the bins determined by spin-map coordinates of other points on the object. Point matching proceeded by choosing a point in the scene, computing bin locations from spin-map coordinates of the other points in the scene, and voting for model points with indices in the computed bins. The model point with the highest vote was chosen as the point corresponding to the current scene point. By placing the indices in discrete bins of a hash table, the effect of the exact position of individual points on matching was reduced.

Because all of the points on the surface of the object and not just a small number of feature points are used, the density of indices in the table will be very high. We discovered that this high density of indices was defeating the purpose of using a hash table. It turns out that a stack of 2D arrays (one for each spin-image) that encode the density of points in each spin-image, results in more efficient storage and comparison of spin-images than possible with a hash table. By encoding the density of points in the spin-image, 2D arrays reduce the effects of local variations in vertex positions while still describing global shape of the object.

To create the 2D array representation of a spin-image, the procedure described in pseudo-code in Fig. 4 and pictorially in Fig. 5 is invoked. First an oriented point O on the surface of an object is selected. Then for each point x on the surface of the object, the spin-map coordinates with respect to O are computed Eq. (1), the bin that the coordinates index is determined Eq. (3), and then the 2D array is updated by incrementing the surrounding bins in the table. A simple way to update the table for each point x would be to increment the bin to which the point is spin-mapped by one. However, in order to spread the position of the point in the 2D array to account for noise in the data, the contribution of the point is

```
CreateSpinImage(oriented_point O, spin_image SI, surface_mesh M)
   for all points x on M
      (α,β) = SpinMapCoordinates(O,x)          // (1)
      (i,j) = SpinImageBin(α,β)                // (3)
      (a,b) = BilinearWeights(α,β)             // (4)
      SI(i,j) = SI(i,j) + (1-a)*(1-b)
      SI(i+1,j) = SI(i+1,j) + (a)*(1-b)
      SI(i,j+1) = SI(i,j+1) + (1-a)*b
      SI(i+1,j+1) = SI(i+1,j+1) + a*b
```

Fig. 4. Pseudo-code for the generation of a 2D array representation of a spin-image.

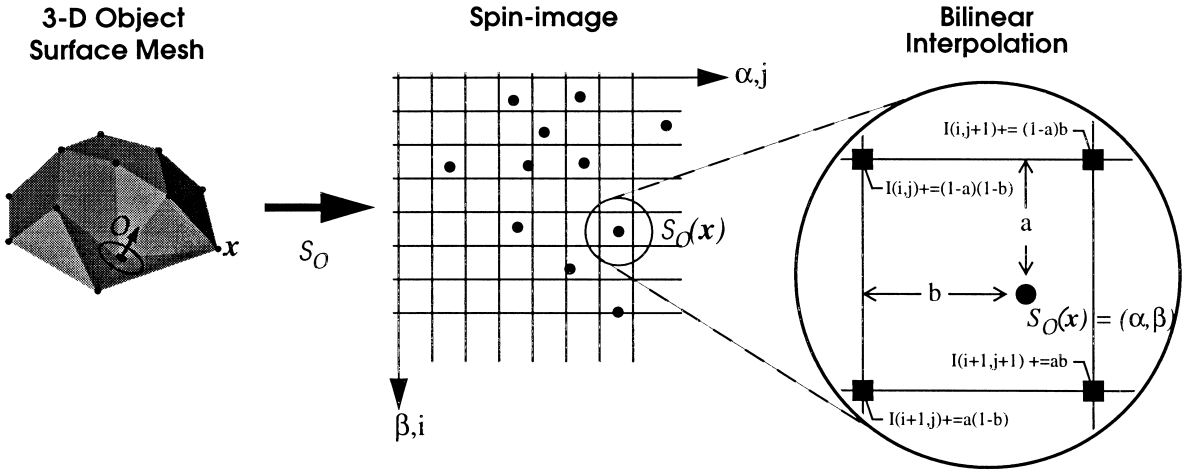


Fig. 5. The addition of a point to the 2D array representation of a spin-image.

bilinearly interpolated to the four surrounding bins in the 2D array. This bilinear interpolation of the contribution of a point will spread the location of the point in the 2D array, making the array less sensitive to the position of the point. Once all of the points on the surface of the object have been processed, a 2D array representation of the spin-image is generated. As shown in Fig. 6, the spin-image generation process can be visualized as a sheet spinning around the oriented point basis, accumulating points as it sweeps space.

Fig. 3 shows the 2D array representations of three spin-images generated for three oriented points on the surface of a duckie. The darker the pixel, the higher the number of points projected into that particular bin. Using spin-images to match points opens up the entire field of image based matching, giving us powerful comparison tools such as image correlation. For the rest of this paper we will also use the term spin-image to refer to the 2D array generated from the spin-mapping of points on a 3D object. 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.

Our system finds corresponding points by comparing spin-images. For the spin-images of two corresponding points on surface mesh representation of the same object to be the similar, the sampling of vertices over the surfaces of the instances should be uniform. This is a much weaker constraint than requiring the positions of vertices to be the same for the two instances. If the vertex samplings are uniform, then on average, each corresponding bin of the spin-images will have the same number of vertices projected into it, making the spin-images similar. In general, objects imaged with the same sensor will have a uniform sampling of vertices. If a mesh is non-uniformly sampled, we use a

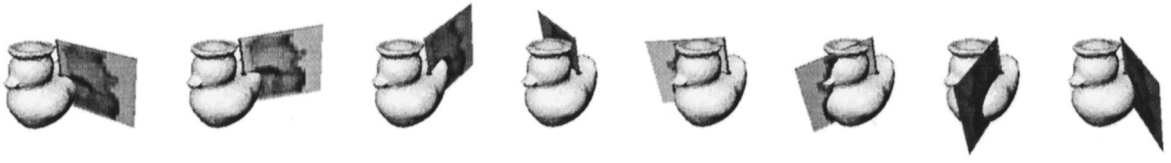


Fig. 6. Eight frames of an animation that motivate the name spin-image. The spin-image generation process can be visualized as a sheet spinning around the oriented point basis, accumulating points as it sweeps space.

mesh simplification algorithm [9] to add and remove points from the surface meshes until the sampling is uniform.

Before a spin-image can be generated, the size of the spin-images must be determined. In order to ensure that spin-images are large enough to store contributions from all points on the object while not being larger than necessary, the maximum size of the object in oriented point coordinates is computed. To determine the maximum sizes, oriented point bases are constructed at every oriented point on the object. Then, for each oriented point basis, the spin-map coordinates of all the other points on the object are computed. The maximum α and $|\beta|$ encountered for all of the oriented point bases are the maximum sizes $\alpha_{\max}, \beta_{\max}$ of the object in oriented point coordinates. As discussed in Section 4, the maximum sizes can be decreased in order to limit the effects of clutter and occlusion.

The size of the bins in the spin-images has to be determined as well. Bin size b is very important because it determines the storage size of the spin-image and has an effect on the descriptiveness of the spin-images. The bin size is set as a multiple of the resolution of the surface mesh in order to reduce the effects of object scale and resolution on this decision. We define mesh resolution as the average of the edge lengths in the mesh. Setting bin size based on mesh resolution is feasible because mesh resolution is related to the size of shape features on an object and the density of points in the surface mesh. We have found that setting the bin size to be two to four times the mesh resolution sufficiently blurs the position of individual points in the spin-images while still adequately describing global shape.

Once the bin size and maximum size of the object in spin-map coordinates have been calculated, the sizes of the spin-image $(i_{\max}, j_{\max})$ can be calculated. The sizes of the spin-image are

$$i_{\max} = \frac{2\beta_{\max}}{b} + 1 \quad j_{\max} = \frac{\alpha_{\max}}{b} + 1. \tag{2}$$

Because the distance to the tangent plane of an oriented point can be both positive and negative, the size of the spin-image in the β direction is twice $\beta_{\max}$. The equations relating spin-map coordinates and spin-image bin (i,j) are

$$i = \left\lfloor \frac{\beta_{\max} - \beta}{b} \right\rfloor \quad j = \left\lfloor \frac{\alpha}{b} \right\rfloor, \tag{3}$$

where $[f]$ is the floor operator which rounds f down to the nearest integer. Given the bin size, the bilinear weights used to increment the bins in the spin-image can be calculated

$$a = \alpha - ib \quad b = \beta - jb. \tag{4}$$

Given the size and number of bins in the spin-images and properly oriented surface normals, a spin-image can be generated for any point in the surface mesh representing an object.

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 Fig. 7 the vertex positions and connectivity of two models of a duckie are different, yet the spin-images from corresponding points are similar. A standard way of comparing images that exhibit a linear relationship between corresponding pixels 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 data used to compute the correlation coefficient is taken only from the region of overlap between two spin images. As shown in Fig. 11, clutter in the scene can create pixels in the scene spin-image that have data where the model spin-image has no data. Similarly, occlusion in the scene can create scene pixels that have no data where the

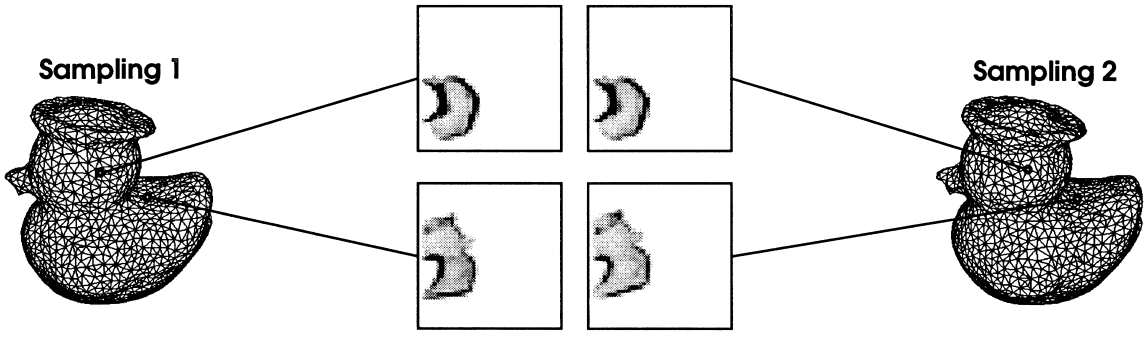


Fig. 7. Spin-images generated from two different samplings of the duckie model. Although the surface meshes are different, the spin-images generated from corresponding points are similar, even though the coordinates and surface normals of the points are not exactly the same.

model spin-image has data. In this case, knowledge of the effect of clutter and occlusion on the spin-image generation process is used to eliminate outlier pixels from the correlation coefficient computation.

Since the linear correlation coefficient is a function of the number of pixels used to compute it, the amount of overlap between spin-images will have an effect on the correlation coefficients obtained. The more pixels used to compute a correlation coefficient, the more confidence there is in its value. We have found that matching of spin-images based on the magnitude of correlation coefficient as well as the confidence in the correlation coefficient results in a greater number of corented point correspondences. The confidence in the correlation coefficient can be measured by its variance. We use the similarity measure C to combine correlation coefficient R and its variance into a single function. When comparing two spin-images P and Q where N is the number of overlapping pixels used in the computation of R, the similarity measure C is defined as

$$C(P,Q)(\text{atanh}(R(P,Q)))^2 - \lambda\left(\frac{1}{N-3}\right). \quad (5)$$

This similarity measure 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 ([4] Chapter 12) performed by the hyperbolic arctangent function, transforms the correlation coefficient into a distribution that has better statistical properties. In particular, under the transformation of variables, the variance of the transformed correlation coefficient becomes $1/(N-3)$ which is a simple function of the number of pixels used to compute R. In Eq. (5), λ weights the variance against the expected value of the correlation coefficient. In practice λ is set to 3.

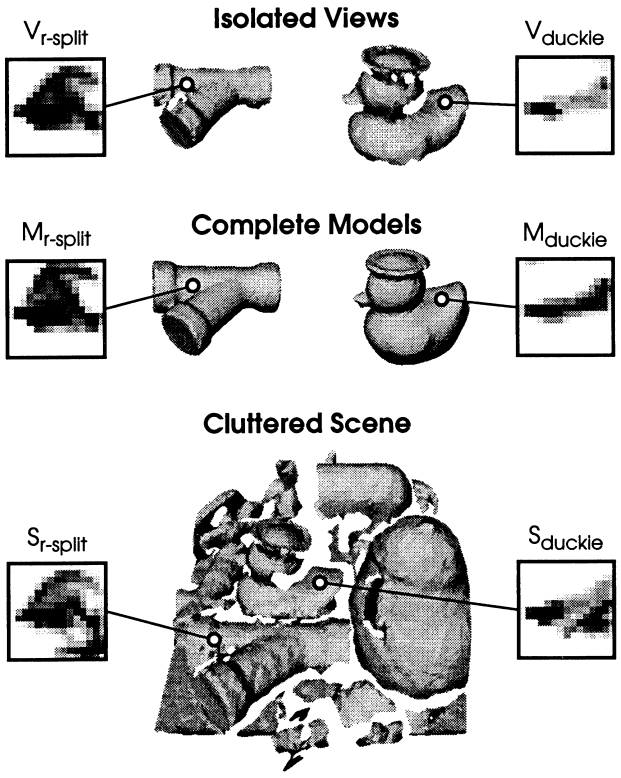
4. Limiting the effect of clutter and occlusion

In real scenes, clutter and occlusion are omnipresent. Any object recognition system designed for the real world must somehow deal with clutter and occlusion. Unlike in our system, 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. The effect of clutter and occlusion on recognition can be limited by setting two thresholds that determine which points contribute to spin-image generation. The first threshold sets the maximum distance D_s between the oriented point basis and a point in the mesh contributing to the spin-image. In general this distance threshold is set to the size of the model (maximum distance between points on the model). The second threshold A_s sets the maximum angle between the oriented point basis surface normal and the surface normal of other points on the surface. This threshold prevents points that will be self-occluded from contributing to the spin-image without specifying a viewing direction. When matching a complete model to a partial scene A_s is set to 60°, otherwise it is set to 180°.

Fig. 8 shows an experimental validation that spin-image matching is resistant to clutter and occlusion while still being effective for recognition. First a spin-image generated from a complete model of a duckie is compared to an isolated range view of the duckie; a high correlation exists. However, when the duckie model spin-image is compared to a spin-image from an isolated view of an rsplit pipe, the correlation is low. This indicates that the duckie will match a view of itself better than a view of a different model, so spin-images can differentiate between isolated objects. This relationship is upheld when the duckie model spin-image is compared to spin-images from a cluttered scene. The duckie model spin-image matches the spin-image from the duckie surface in the cluttered scene much better than the rsplit pipe surface in the cluttered scene. This suggests that spin-images can be used for differentiation between objects and hence recognition in cluttered scenes. Similar reasoning using complete model of the rsplit pipe results in the same conclusions.

In order to analyze the effects of clutter, we have developed a simple model of the occurrence of clutter in spin-images by modeling objects as spheres (details given in [12]). Suppose a spin-image is being created for a point on the object to be recognized, then the appropriate object sphere is the largest sphere contained entirely within the object that also intersects the point. Similarly, suppose there exists a cluttering object in the scene. The appropriate size for the spherical model of the clutter object is determined as follows. Select a point on the clutter object. Find the largest sphere that touches the point and is still contained within the clutter object. Take the minimum diameter of these spheres over all points on the clutter object and you have the appropriate diameter for the spherical model of the clutter object. These spherical definitions place a lower bound on the space taken up by the object and an upper bound on the space taken up by the clutter and are sufficient for determining the worst case effect of clutter on spin-images.

Our model of the effect of clutter on spin-image generation is shown (in 2D) in Fig. 9. Suppose we would like to determine the spin-image for an oriented point p on an object modeled as a sphere as defined above. A point on the clutter object can be projected into the spin-image of the oriented point if: The clutter point is within the bounds of the spin-image D_s, the clutter object does not intersect the object, and the angle between the surface normal of the clutter point and the oriented point is less than the angle threshold A_s. These three constraints describe a connected region in space, that corresponds to a connected region in the spin-image, where clutter might occur. It follows that clutter can only effect a limited number of the pixels in a spin image, and the effect of clutter can be limited by setting the thresholds stated above. A similar model shows that the effect of occlusion is also limited to connected regions in



Model-View Correlation

	M_{rsplit}	M_{duckie}
V_{rsplit}	0.814	0.478
V_{duckie}	0.498	0.885

Model-Scene Correlation

	M_{rsplit}	M_{duckie}
S_{rsplit}	0.809	0.426
S_{duckie}	0.545	0.746

Fig. 8. Experimental validation the of robustness of spin-image matching to clutter. Comparing a complete model to an isolated view of the model results in high correlation. Comparison of the model to an isolated view of a different model results in low correlation. When the objects models are imaged in a cluttered scene, this relationship still holds.

spin-images. The exact number of pixels affected by clutter (N_c) in a particular spin-image can be computed by checking which pixels in the spinimage satisfy the clutter constraints stated above. Fig. 10 contains a plot showing the percentage of clutter for a spin-image as a function of spin-image size D_s for different values of the angle threshold A_s. From the plot it can be seen that choosing low values for D_s and A_s will produce spin images that contain no clutter.

Given the number of corrupted pixels in a spin-image is bounded, it is possible to predict a lower bound on the correlation coefficient, and therefore determine the effect of clutter and occlusion on the matching of spin-images. Suppose there exists a model surface mesh and a scene surface mesh that contains an exact copy of the model surface mesh. In this case, spin-images of corresponding points in the model and scene will be exactly the same except for the pixels in the scene that are corrupted by clutter and occlusion. Let the subset of the scene spin-image (with N total pixels) that is corrupted by clutter be I_C (with N_C pixels) and the portion that is not corrupted be I_N. Let the scene spin-image have pixels x_i^s, the model spin-image have pixels x_i^m and the corruption at each scene pixels be δ_i. Then the relationship between model and scene pixels is

$$\text{if } x_i^s \epsilon I_N \text{ then } x_i^s = x_i^m, \text{ and if } x_i^s \epsilon I_C \text{ then } x_i^s = x_i^m + \delta_i. \quad (6)$$

Clutter and occlusion manifest as extra and missing points in the scene where the number of these points is bounded. Therefore, based on the spin-image generation process, the total change of any pixel in a scene spin-image that is corrupted by δ_i is bounded $|\delta_i| \leq \delta$. Furthermore, the model and scene pixel values can be normalized on [0,1] with no effect on the correlation coefficient computed. Substitution of Eq. (6) into the definition of correlation coefficient

$$\rho = \left(\frac{1}{N}\sum x_i^m x_i^s - \frac{1}{N^2}\sum x_i^m \sum x_i^s\right) / \left(\left(\frac{1}{N}\sum (x_i^m)^2 - \left(\frac{1}{N}\sum x_i^m\right)^2\right) \times \left(\frac{1}{N}\sum (x_i^s)^2 - \left(\frac{1}{N}\sum x_i^s\right)^2\right)\right)^{\frac{1}{2}}, \quad (7)$$

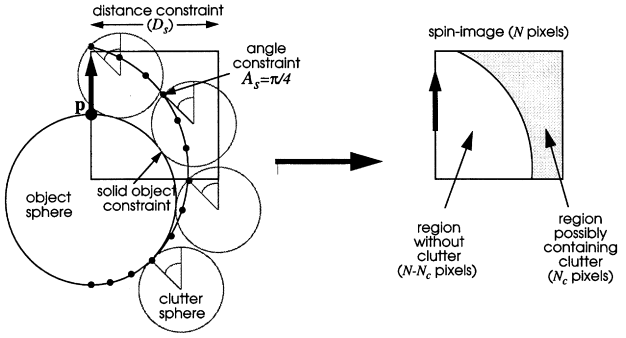


Fig. 9. Theoretical clutter model. Because solid objects cannot intersect, and the support of spin-images is limited, the corruption of pixels due to clutter is limited to connected regions in spin-images.

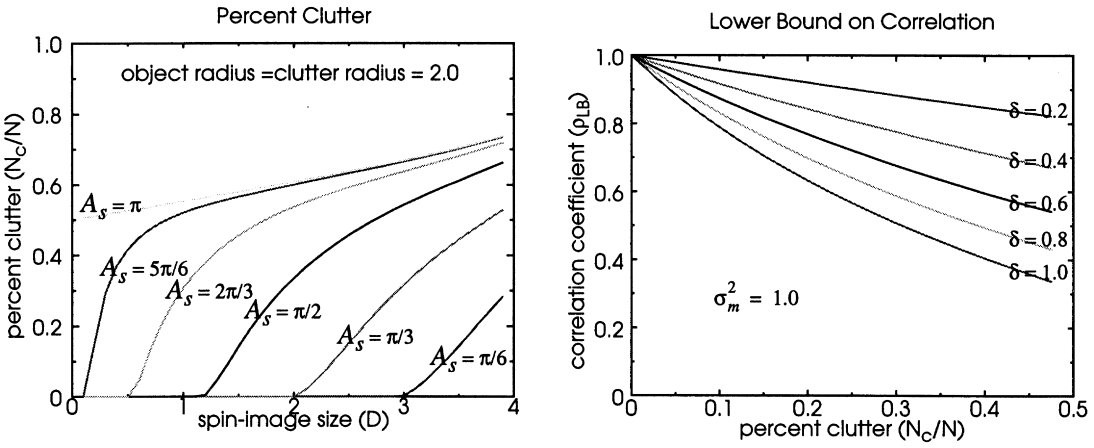


Fig. 10. Clutter model plots. The effect of spin-image size and surface normal angle threshold on the percent of clutter in a scene spin-image is shown on the left and the worst case effect of clutter correlation coefficient is shown on the right.

and using the normalization of pixel values and the bound on corruption, results in a lower bound on the correlation coefficient when comparing model and scene spin-images

$$\rho_{LB} = \left(\sigma_{\mathrm{m}}^2 - \frac{N_{\mathrm{C}}}{N}\delta\right) \Big/ \left(\sigma_{\mathrm{m}}\sqrt{\sigma_{\mathrm{m}}^2 + \frac{N_{\mathrm{C}}}{N}(\delta+\delta^2)}\right). \tag{8}$$

σ_{m}^2 is the variance of the pixels in the model spin-image (derivation given in Ref. [12]). Fig. 10 shows plots of ρ_{LB} versus the percent of corrupted scene pixels for increasing values of δ. From the plot it is clear that as the percentage of clutter increases, the worst case dffect on correlation coefficient decreases gradually. This important result shows that our algorithm for matching spin-images does not fail catastrophically with the addition of clutter. Instead, the performance of our matching algorithm will gradually degrade as clutter and occlusion increase in the scene. Fig. 11 validates the lower bound on correlation coefficient in the presence of clutter and occlusion in a real scene. The correlation coefficient of the two corresponding images is 0.746, and the predicted lower bound on correlation coefficient computed using Eq. (8) is 0.668.

5. Generating point correspondences

The similarity measure Eq. (5) 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 spinimage is then correlated with all of the images in the model spin-image stack and the similarity measures Eq. (5) 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 with 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 in the scene, this number can vary between one tenth and one half of the points in the scene. The end result is a list of 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.

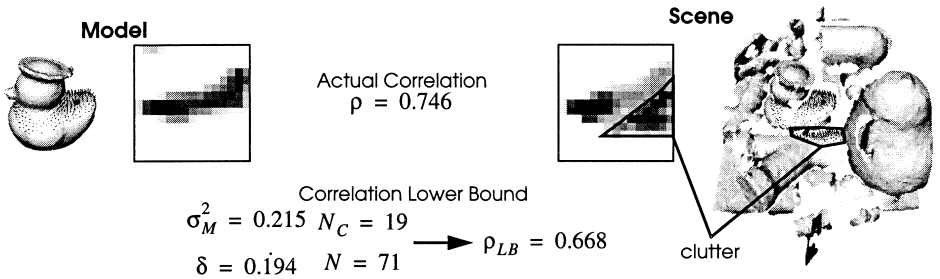


Fig. 11. Experimental verification of lower bound on correlation coefficient in the presence of clutter. 19 of 71 pixels in the scene spin-image are corrupted by an amount 8 less than 0.194. The correlation coefficient for the two spin-images of 0.746 is well above the lower bound of 0.668 predicted by the clutter model.

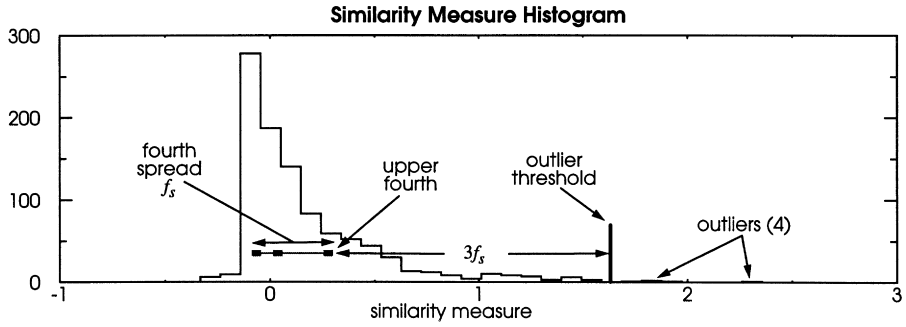


Fig. 12. Similarity measure histogram.

Second, if multiple outliers exist, then multiple model spin-images are similar to a single scene spin-image, so multiple model/scene point correspondences should be considered in the matching process. We use a standard method for automatic detection of outliers in a histogram ([4] 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 outlier's. Another useful property of the transformation of variables (hyperbolic arctangent) used in the similarity measure is magnification of the correlation coefficients that are close to unity, making the detection of outliers in the similarity measure histogram easier. Fig. 12 shows a similarity measure histogram with detected outliers.

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.

During matching, some points selected from scene clutter may be incorrectly matched to model points. However, given the numerous correspondences, it is possible to filter out bad correspondences based on properties of the correspondences taken as a group. This integral approach is robust because it does not require reasoning about specific point matches to decide which correspondences are the best. This approach is in contrast to hypothesize and test and alignment paradigms of recognition [6] where the minimal number of correspondences required to match model to scene are proposed and then verified through some other means.

The first method of correspondence filtering uses similarity measure to remove unlikely correspondences. All correspondences with similarity measures that are less than some fraction of the maximum similarity measure of all of the correspondences are eliminated. In practice, this fraction is set to one half, The second method for filtering out unlikely correspondences uses geometric consistency which is a measure of the likelihood that two correspondances can be grouped together to calculate a transformation of model to scene. If a correspondence is not geometrically consistent with other correspondences, then it cannot be grouped with other correspondences to calculate a transformation, and it should be eliminated. The geometric consistency of correspondences $C_1 = [s_1,m_1]$ and $C_2 = [s_2,m_2]$ is measured by comparing the spin-map coordinates Eq. (1) of corresponding points.

$$d_{\mathrm{gc}}(C_1, C_2) = 2\frac{||S_{\mathrm{m}_2}(m_1) - S_{\mathrm{s}_2}(s_1)||}{||S_{\mathrm{m}_2}(m_1) - S_{\mathrm{s}_2}(s_1)||}$$

$$D_{\mathrm{gc}} = \max(d_{\mathrm{gc}}(C_1, C_2), d_{\mathrm{gc}}(C_2, C_1)). \quad (9)$$

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 distance D_{gc}. When D_{gc} is small (i.e. C_1 and C_2 are geometrically consistent), the scene and model points in C_1 are the same distance apart and have the same angle between surface normals as the scene and model points in C_2.

To filter out correspondences based on geometric consistency, correspondences that are not geometrically consistent with at least one quarter of the correspondences in the list are eliminated. The end result after filtering on similarity measure and geometric consistency is a list of point correspondences $L = \{C_1,\ldots,C_{\mathrm{n}}\}$ that are the most likely to be correct. In practice this number is between 20 and 50 correspondences. Fig. 1 shows three of the best correspondences and matching spin-images between a Mr Potato Head model and a scene containing it. The next step is to group these correspondences into sets that can be used to compute transformations.

6. Grouping correspondences

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. At least two oriented point correspondences are needed to calculate a transformation if position and normals are used. To avoid combinatoric explosion, geometric consistency is used to determine a groups of correspondences

from which plausible transformations can be computed. Since many correspondences are grouped together and used to compute a transformation, the resulting transformation is more robust than one computed from a few correspondences.

We group correspondences based on a grouping criterion W_{gc} that is the geometric consistency distance between two correspondences Eq. (9) augmented by a weight that promotes grouping of correspondences that are far apart.

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

$$W_{gc}(C_1, C_2) = \max(w_{gc}(C_1, C_2), w_{gc}(C_2, C_1)). \quad (10)$$

W_{gc} will be small when two correspondences are geometrically consistent and far apart. Since correspondences that are not geometrically consistent will produce transformations of high error, geometric consistency is a necessary condition for grouping. Correspondences whose points are close together generate transformations that are susceptible to noise in point position [3], so correspondences with points that are spread apart are more desirable. The grouping criterion between a correspondence C and a group of correspondences $\{C_i,\ldots,C_n\}$ is

$$W_{gc}(C, \{C_1, \ldots, C_n\}) = \max_i (W_{gc}(C, C_i)) \quad (11)$$

In other words, the grouping criterion between a correspondence and a group of correspondences will be small if the grouping criterion of the correspondence and all of the correspondences in the group is small.

Given a list of most likely correspondences $L = \{C_1,\ldots,C_n\}$ the grouping procedure that is applied for each correspondence in the list is as follows: Select a seed correspondence C_i in L and initialize a group $G_i - \{C_i\}$. 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. Continue adding the correspondence with minimum grouping criterion until no more correspondences can be added to G_i. The grouping procedure is performed for each correspondence in L, and the end result is n clusters, one for each correspondence in L. This clustering algorithm allows a correspondence to appear in multiple clusters, a condition which is necessary to handle model symmetry. For example, the duckie model in Fig. 3 has a plane of symmetry resulting in two feasible transformations. Correspondences along the plane of symmetry contribute to two distinct transformations.

A plausible transformation T from model to scene is calculated from each group $\{[m_i,s_i]\}$ of correspondences by minimizing

$$E_T = \sum ||s_i - T(m_i)||^2 \quad (12)$$

Instead of using points and normals in the minimization of Eq. (12), we use only the position of the oriented points. This allows us to use a well defined algorithm for finding the best rigid transformation that aligns two point sets [8]. The transformations and associated correspondence groups are then input into a verification procedure.

7. Verification

The purpose of verification is to find the best transformation of model to scene by eliminating transformations that are inconsistent when all of the scene data is compared to all of the model data. Our verification algorithm is a formulation of the iterative closest point algorithm [1,20] that can handle partially overlapping point sets and arbitrary transformations. During verification, point correspondences are spread over the surfaces of the scene and the model from the initial correspondences established by matching of spin-images. If many correspondences are established through spreading, a match between model and scene is validated.

Verification starts with an initial set of point correspondences from which the transformation of model to scene is computed This transformation is then applied to the model surface mesh. Next, correspondences are spread from each initial correspondence as follows: for each scene point in an initial correspondence, the scene points directly connected to it by edges in the scene surface mesh are turned into correspondences (with their closest model points) if the distance between scene and closest model points is less than a threshold D_V. This process is recursively applied to each of the correspondences just added until no more correspondences can be created. To speed up the search for closest model points, a k–D tree is generated (off-line) for points in the model surface mesh. The threshold D_V in the verification stage (that sets the maximum distance by which two points can differ and still be brought into correspondence) is set automatically to twice the resolution of the meshes. This threshold allows for noise but prevents establishment of correspondences in regions where the data sets do not overlap.

This verification algorithm spreads correspondences between the model and the scene from the initial correspondences and a cascade effect occurs. If the initial transformation is correct, a large number of points will be brought into correspondence; if the transformation is bad, the number of correspondences will remain close to the original number. Therefore, our measure of the validity of the match is the number of correspondences after verification which is related to the total aligned surface area between the model and scene by the transformation. If the number of correspondences after verification is greater than one fourth to one third the total number of points on the model, the transformation is considered valid.

To refine the transformation after it has been verified and accepted, a face-based iterative closest point registration (ICP) algorithm [16] is applied to the model (with faces) and the scene points from the spread correspondences. The final transformation computed from the ICP algorithm is more accurate than the transformation computed through

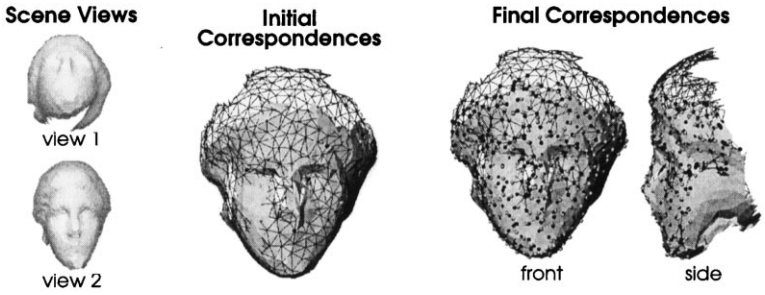


Fig. 13. Transformation verification of two views of a plastic model of the head of Venus. From left to right are shown: two camera images of the views to be registered, the registration of the views (view 1 shaded, view 2 wireframe) before verification with initial correspondences shown as spheres, and a frontal and side view of the registration after verification with established correspondences. Correspondences are not established outside the region of overlap between the two views.

matching of spin-images because a large number of scene points are registered to the faces of the model. Considered as a ICP region algorithm, our recognition algorithm solves two of the biggest problems with ICP: first, the need for an initial guess at the transformation between model and scene is provided by the transformation determined by matching spin-images. Second, registering data sets when one is not a proper subset of the other is handled by the controlled spreading of correspondences from initial correspondences.

Fig. 13 illustrates how initial correspondences, established by matching spin-images, are spread over the surfaces of two range views of a plastic model of the head of.the goddess Venus. The correspondences are established only in the regions where the two surface meshes overlap, thus preventing a poor registration caused by correspondences being established between non-overlapping regions.

8. Results

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 command such as ‘turn off the valve’ [10].

Fig. 14 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. Both the models and scene data were processed by removing long edges associated with step discontinuities, applying a ‘smoothing without shrinking’ filter [18], 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 [9]. 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. These results clearly show the ability of our algorithm to recognize complete 3D models in partial 3D scenes.

Another important task in interior modeling is the automatic registration of range images. By registering and

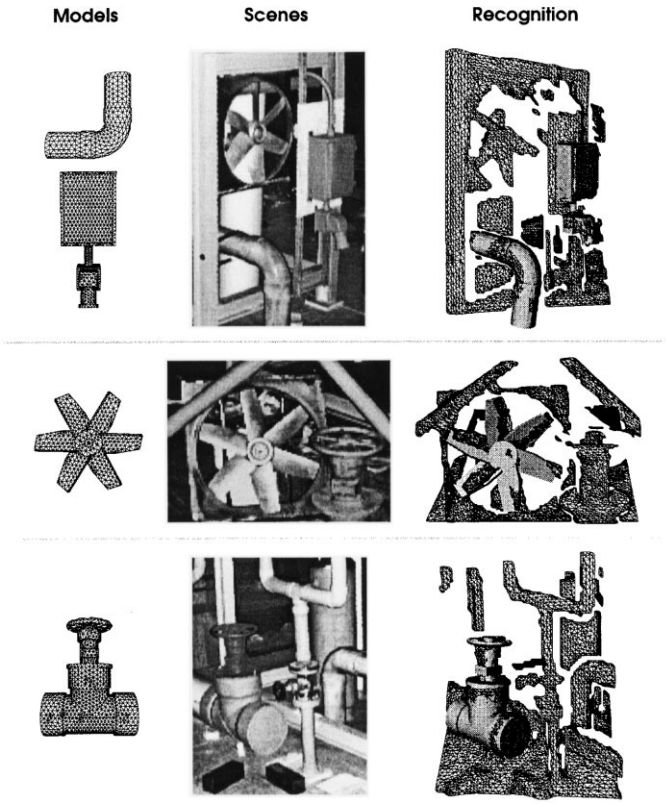


Fig. 14. The recognition of industrial objects in complex scenes. On the left are shown wireframe models which were created from CAD drawings using finite element software for surface tessellation. In the middle are shown the intensity images acquired when a scanning laser rangefinder imaged the scene. On the right are shown the recognized models (shaded) superimposed on the scene data (wireframe). These results demonstrate the recognition of complicated symmetric objects in 3D scene data containing extreme clutter and occlusions. All of the scene data points are used in the recognition and no object/background segmentation is performed.

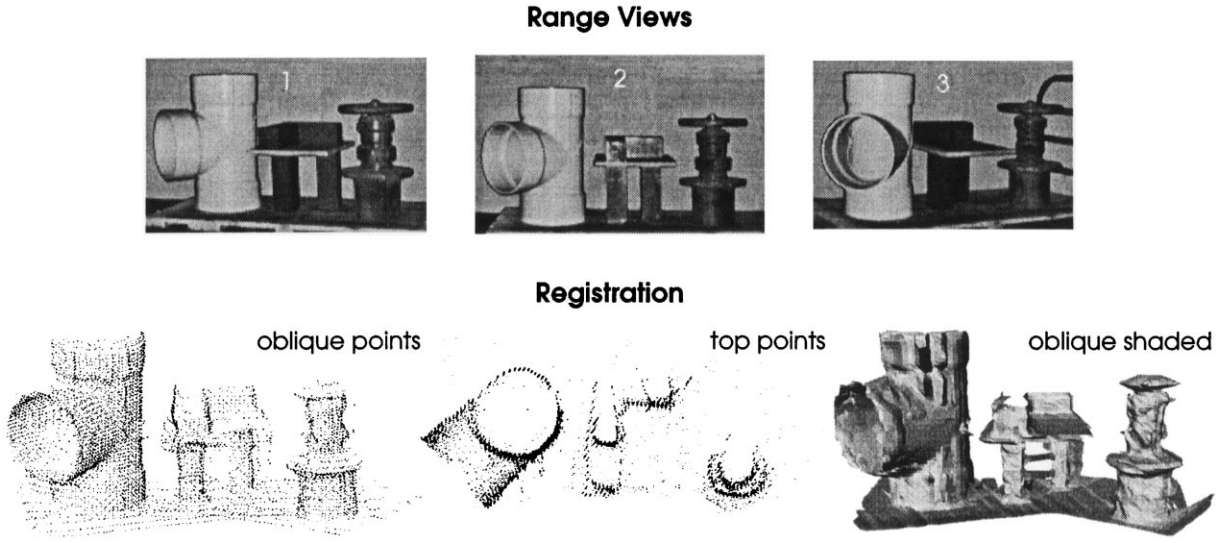


Fig. 15. Registration of laser rangefinder images of industrial objects. The range views differ by roughly 30°, so registering the views provides a more complete description of the scene.

merging range images, more complete scene descriptions are generated. Our algorithm provides a technique for determining the transformation between range views when it is unknown or highly uncertain. Fig. 15 shows a scene composed of a PVC pipe joint, four graphite bricks, a piece of plywood and a steel valve placed on a mobile cart. This scene was imaged in three different positions by moving the cart and taking a range image with a Perceptron 5000 laser rangefinder at each position. The position of the cart varied each time by approximately 30 degrees of rotation about the vertical axis. Fig. 15 shows the intensity channel of the range scanner for the three scene positions, a shaded view of the resulting registrations and an oblique and top view of the points in the three scenes after registration. The top view of the registered points clearly shows that a more complete scene description than is available from one range image is generated and that the registration is correct up to the resolution of the points.

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 with 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 grouped. Fig. 16 demonstrates the simultaneous recognition of two models of freeform shape. The femurand pelvis model were acquired using a computerized tomography (CT) scan of the bones, followed by surface mesh generation from contours. The scene was acquired using a K^2T structured light range camera. Before recognition, the scene data is processed to remove long edges and small surface patches and then smoothed. 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. This result also demonstrates the recognition of objects sensed with different sensing modalities.

A common result in the biomedical imaging literature is the registration of two skull data sets generated from volumetric medical scanners [7]. Fig. 17 shows the registration of two different surface meshes of a skull created from the same CT data set. The surface meshes were created by adding small amounts of Gaussian noise to the points in the original surface mesh generated from the CT scans

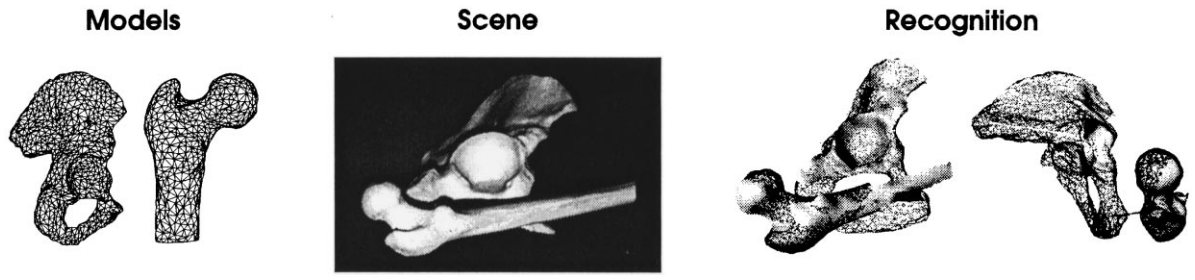


Fig. 16. The simultaneous recognition of the models of a femur and a pelvis bone in a range image. The femur and pelvis are recognized even in the presence of extreme occlusion (only 20% of the pelvis surface is visible) and clutter. From left to right are shown the pelvis and femur models acquired with CT, the scene image from a structured light range camera, and then two views of the recognition results where the scene data is shaded and the models are in wireframe. This result demonstrates simultaneous recognition of free-form objects acquired using different sensing modalities.

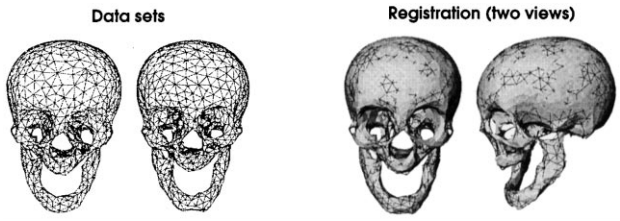


Fig. 17. Registration of two skull data sets generated from CT. The accuracy of the registration (0.169 mm) demonstrates the usefulness of our algorithm in medical registration procedures where the transformation between data sets is unknown or highly uncertain.

and then simplifying the meshes. Different random noise values were added to the original surface mesh for each of the meshes shown, resulting in surface meshes with different points and connectivity after simplification. A close look at the two wireframe data sets in Fig. 17 shows that the two surface meshes are completely different while still approximating the shape of the skull. This skull data set is especially difficult to register because the inner and outer surface of the cranium are extracted from the CT data, increasing the complexity of the model and possibly introducing ambiguities when registering the data sets. Since the two data sets are already co-registered, any non-identity transformation calculated for the registration will represent the registration error. For the result shown in Fig. 17, the translation is $[0.019\ -0.069\ -0.013]^{T}$ mm and the fixed rotation angles are $[-0.06\ -0.03\ 0.018]$ degrees. This corresponds to a translation error magnitude of 0.072 mm, which is less than 2% of the resolution of the surface meshes (5.9 mm) being registered and an angular error magnitude of 0.07 degrees. The accuracy of the registration demonstrates the usefulness of our algorithm in medical registration procedures where the transformation between data sets is unknown or highly uncertain.

Fig. 18 demonstrates the registration of a very fine elevation map to a coarse digital elevation map. The ability to register a local elevation map (perhaps generated by onboard sensors) to a coarse global elevation map is very useful in outdoor mobile robot navigation. It can be used to initially register a robot to a global map and also correct for subsequent drift in global position as the robot traverses the terrain. The model data set in Fig. 18 was generated by subsampling (10:1) a digital terrain map[1] and the scene data set was generated from the cropping the same digital elevation map without subsampling the data. This result shows the usefulness of our algorithm in mobile robot navigation and shows that the algorithm works even when the resolutions of the meshes being compared are substantially different.

In order to accurately convey the dependence of our algorithm on thresholds and free variables, we have provided a table of values for each of the results shown. In Table 1, the first free variable, bin size, is the size of the bins used to create spin images for the model (Section 2.1) expressed in units of model resolution. The next free variable, A_s is the threshold on maximum difference in angle between surface normals, used when creating spin images (Section 4). N_{sp} is the number of scene points selected randomly from the scene (Section 3)expressed as a percent of the total number of scene points. T_{sim} is the threshold on

[1] Real elevation data collected from the Lockheed Martin facility in Denver, Colorado.

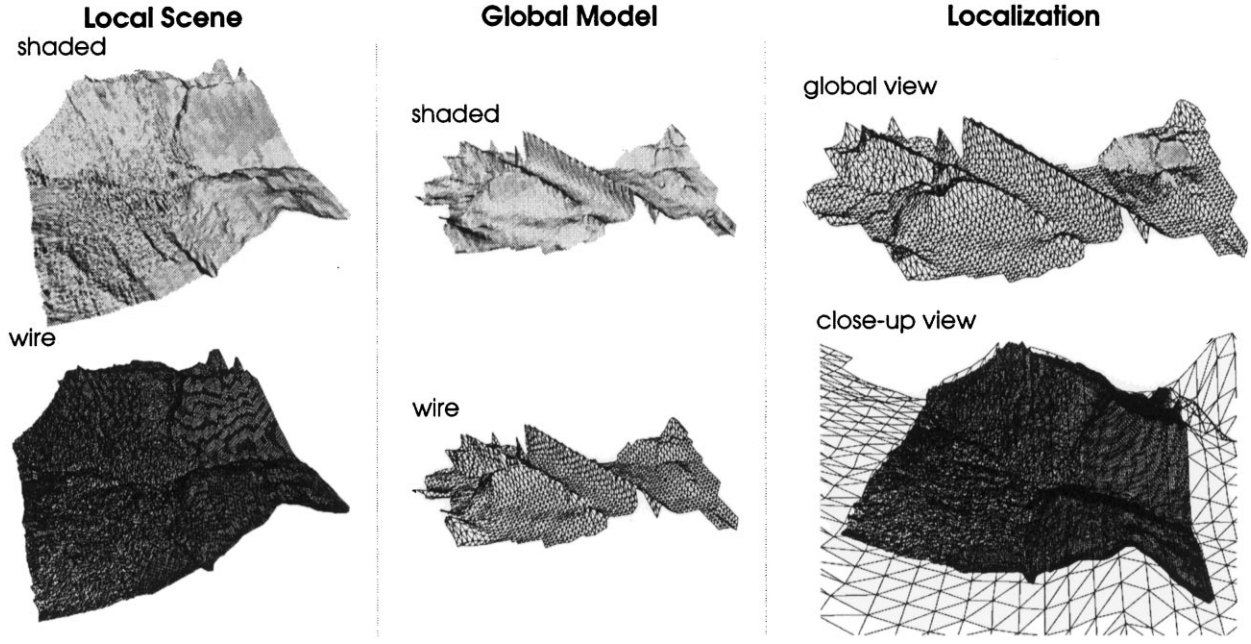


Fig. 18. Registration of a fine local elevation map to a coarse global elevation map. This result shows the usefulness of our algorithm in mobile robot navigation and shows that the algorithm works even when the resolutions of the meshes being compared are substantially different.

Table 1
Thresholds and free variables for recognition

Result	Bin size	A_s	N_{sp}	T_{sim}	D_{gc}	D_V
Control box (Fig. 14)	2	60	20%	50%	0.25	2
Elbow joint (Fig. 14)	2	60	20%	50%	0.25	2
Valve (Fig. 14)	2	60	20%	50%	0.25	2
Fan (Fig. 14)	2	60	20%	50%	0.25	2
Multi-view 1 (Fig. 15)	4	180	20%	50%	0.25	2
Multi-view 2 (Fig. 15)	4	180	20%	50%	0.25	2
Femur and pelvis (Fig. 16)	3	60	20%	50%	0.25	2
Skull (Fig. 17)	4	180	20%	50%	0.25	2
Elevation map (Fig. 18)	1	180	5%	50%	0.25	0.25

similarity measure, expressed as a percentage of the maximum similarity measure of all correspondences generated, used to remove unlikely correspondences (Section 5). D_{gc} is the geometric consistency threshold, expressed in units of mesh resolution, used to remove correspondences that are not geometrically consistent with other correspondences (Section 5). D_V is the maximum search distance in the verification algorithm (Section 7) also expressed in units of mesh resolution.

The parameters shown in the table are all basically the same. Some of the results involve matching a complete model do a partial scene, so the angle threshold is set to 60°. Otherwise the angle threshold is unconstrained (180°). The elevation map parameters are different from the other results, because a very dense scene is being matched to a coarse model. In this case, less scene points need to be selected, so N_{sp} is lower (5%). Since the model resolution is much larger than the scene resolution, the bin size is set to 1 in order to have more of the variation in the scene represented in the spin-images. D_V is lowered in order to decrease the spreading of correspondences between model and scene. In this case, sufficient correspondences for validation will be established with a low verification threshold because of the large ratio of scene points to model points.

9. Discussion

Spin-images offer a method for performing 3D object recognition and registration using an image based representation. This has allowed the application of powerful image-based tools, like image correlation, to the problem of matching spin-images and their associated oriented points. The underlying assumption in our algorithm is that spin-images from corresponding points will be similar enough to be matched. This is the same assumption that is used in template matching and correlation based stereo: two pervasive methodologies in computer vision. Through the use of spin-images, we are attempting to bring the success of image based techniques to 3D object recognition.

For two spin-images to be similar, the shape of the scene must be similar to that of the model. In the case of complete and continuous object surfaces (and hence continuous spin-images) the spin images crested for corresponding points in the model and the scene will be exactly the same. However, when the objects are represented as discrete points connected in a surface mesh, the spin-images from corresponding points will not be the same because the points on the two objects will usually be in different positions. Fortunately, the way spin-images are, generated minimizes the effects of discretization. During spin-mapping, points are bilinearly assigned to the four nearest bins in the spin-image, thus smearing their position in space and making the exact positions of the points less relevant. Since the surface normal at a point is determined from the best fit plane in a neighborhood of a point, the calculated surface normals of corresponding oriented points will be similar (if the local shape does not have large curvature.) Since the connectivity of the surface mesh is only used when determining the nearest neighbors of a point for the surface normal calculation, it will also have little effect on the appearance of the spin-images.

Recognizing objects by establishing point correspondences between a model and a scene data set is not a new idea. Previous methods have relied on the extraction of a small number of feature points between which correspondences can be established [6]. Many times these features are located at the positions that are most susceptible to noise on the surface of the object (e.g. edges, corners). Our method is novel in that it considers all of the points on the surface of an object for establishment of point correspondences, so it does not have to rely on a possibly unreliable feature extraction method. Furthermore, correspondences are established between stable locations on the surface of the object making, the computed transformation from model to scene more accurate. Our approach can be considered an integral approach in that it uses all of the points available, and lets the matching algorithm decide, at run-time, which points are the best for recognition.

Because our algorithm does not construct coordinates frames from multiple points, its computational complexity is much less than that attributed to methods of basis geometric hashing. Let S be the number of points selected from the scene, M the number of model points and I the size of the spin-images. The time to generate the model hash table

Table 2
Recognition timing statistics

Result	Model points (M)	Total no. scene	No. scene point selected (S)	Spin image size (I)	Stack size in bytes	Stack create time	Match time	Verify time
Control box (Fig. 14)	2182	4042	808	200	1.7M	72s	48s	10s
Elbow joint (Fig. 14)	637	4042	808	200	0.5M	10s	42s	3s
Valve (Fig. 14)	2433	4273	854	200	1.9M	79s	60s	8s
Fan (Fig. 14)	1054	5207	1041	200	0.8M	18s	39s	5s
Multi-view 1 (Fig. 15)	2716	2349	469	200	2.2M	101s	27s	22s
Multi-view 2 (Fig. 15)	2369	2349	469	200	1.9M	78s	23s	23s
Femur and pelvis (Fig. 16)	2313	2792	558	200	1.9M	41s	40s	21s
Skull (Fig. 17)	1541	1532	306	200	1.2M	59s	7s	2s
Elevation map (Fig. 18)	2216	22447	1122	200	1.8M	75s	538s	44s

(which can be done off-line) is $O(M^2)$ because a spin-image is generated for every point on the model and each spin-image requires the spin-mapping of every point in the model. The size of the model table is O(MI) because there is one spin-image for every model point. The establishment of correspondences between the model stack and the scene is $O(SMI + SM\log M)$ because each scene point spin-image must be pixelwise multiplied with all of the model spin-images ($O(SMI)$), and the M similarity measures of the correspondences must be sorted ($O(SM\log M)$). Since $\log M$ is always much less than I, the establishment of correspondences can be reduced to $O(SMI)$. The iterative closest point verification algorithm is worst case O(MlogM) for each iteration of the algorithm. This assumes that all of the model points are brought into correspondence with scene points. The computational complexity of the algorithm is not prohibitive as is born out in the recognition times (real-time) shown in Table 2 for the results shown in Figs 14–18. In Table 2, stack create time is the off-line time taken to create the spin-image stack of the model, match time is the time used to create and match all scene spin-images to the model stack and generate transformations, and verify time is the time taken to verify the transformations generated including the refinement of the transformations using our iterative closest point algorithm. The sum of the match time and verify times encompasses all computations done on-line for recognition. All times are real wall clock times on a Silicon Graphics O2 with a 174 MHz R10000 processor. At this point, it should be stressed that the times shown constitute the total time needed to match surfaces; no additional time is needed to preprocess the surfaces before matching. The unusually long time taken in matching of elevation maps is due to the exceptionally large number of scene points (22447) that must be spin-mapped for each scene spin-image.

10. Conclusion

We have presented a recognition algorithm that is based on matching spin-images generated using oriented points on the surface of an object. Although spin-images are global representations, they are robust to clutter and occlusions. We have demonstrated our algorithm's ability to handle clutter and occlusions through recognition of objects in complex cluttered scenes.

In the future, we will extend the 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 all of the spin-images for a model lie on a 2D manifold in the high-dimensional spin-image space. It is likely that, through the use of eigen-space compression, this manifold can be projected onto a much lower dimensional manifold. Rapid determination of which in the scene could then be determined by projection of scene spin-images onto a manifold generated for each model in the library. This eigen-spin-image approach is similar to parametric appearance based matching.

Acknowledgements

This research was performed as part of the Artisan Project in the Robotics Institute and was funded by the Department of Energy under contract DE-AC21-92MC2910. We would like to thank all members of the Artisan Project for providing a robotic system to test our recognition techniques in realistic settings. In particular, we would also like to thank Jim Osborn, David Simon, Fabio Cozman and Paul Heckbert for many enlightening conversations on robotics, registration, statistics and low-level representations. We would also like to thank André Guéziec for providing us with the skull data set and Karun Shimoga for the use of his structured light range sensor.

References

[1] P. Besl, N. McKay, A method for registration of 3-D shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239–256.

[2] R. Bergevin, D. Laurendeau, D. Poussart, Registering range views of multipart objects, Comput. Vision Image Understanding 61 (1) (1995) 1–16.
[3] C. Chua, R. Jarvis, 3-D free-form surface registration and object recognition, Int. J. Comput. Vision 17 (1) (1996) 77–99.
[4] J. Devore, Probability and Statistics for Engineering and Sciences, Brooks/Cole, Belmont, CA, 1987.
[5] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley-Interscience, New York, 1973.
[6] W.E.L. Grimson, Object Recognition by Computer: The Role of Geometric Constraints, MIT Press, Cambridge, MA, 1990.
[7] A. Guéziec, N. Ayache, Smoothing and matching of 3-D space curves, Int. J. Comput. Vision 12 (1) (1994) 79–104.
[8] B. Hon, Closed-form solution of absolute orientation using unit quaternions, J. Optic. Soc. Am. 4 (4) (1987) 629–642.
[9] A. Johnson, M. Hebert, Control of mesh resolution for 3-D computer vision. Tech. Report CMURI-T,R-96-20, The Robotics Institute, Carnegie Mellon University, 1996.
[10] A. Johnson, R. Hoffman, J. Osborn, M. Hebert, A system for semi-automatically modeling of complex environments. Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling (3DIM ’97), Ottawa, Canada, 1997, pp. 213–220.
[11] A. Johnson, M. Hebert, Object recognition by matching oriented points. Computer Vision and Pattern Recognition (CVPR ’97), San Juan, PR, 1997, pp. 684–689.
[12] A. Johnson, Spin-images: a representation for 3-D surface matching. Ph.D. Dissertation, The Robotics Institute, Carnegie Mellon University, August 1997.
[13] K. lkeuchi, T. Shakunaga, M. Wheeler, T. Yamazaki, Invariant histograms and deformable template matching for SAR target recognition. Computer Vision and Pattern Recognition (CVPR ’96), , San Francisco, CA, 1996, pp. 100–105.
[14] Y. Lamdan, H. Wolfson, Geometric hashing: a general and efficient model-based recognition scheme. Second Int. Conf. Computer Vision (ICCV ’88), Tampa, FL, 1988, pp. 238–249.
[15] F. Pipitone, W. Adams, Tripod operators for recognizing objects in range images; rapid rejection of library objects. IEEE Robotics and Automation (R and A ’92), Nice, France, 1992, pp. 1596–1601.
[16] D. Simon, M. Hebert, T. Kanade, Real-time 3-D pose estimation using a high-speed range sensor. IEEE Robotics and Automation (R and A ’94), Pittsburgh, PA, 1994, pp. 2235–2241.
[17] F. Stein, G. Medioni, Structural indexing: efficient 3-D object recognition, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 125–145.
[18] G. Taubin, A signal processing approach to fair surface design. ACM Computer Graphics (SIGGRAPH ’95), New York, 1995, pp. 351–358.
[19] J. Thirion, New feature points based on geometric invariants for 3D image registration, Int. J. Comput. Vision 18 (2) (1996) 121–137.
[20] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, Int. J. Comput. Vision 13 (2) (1994) 119–152.