Appearing in the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Ontario, May 12-15, 1997.

Surface Registration by Matching Oriented Points

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

Abstract

For registration of 3-D free-form surfaces we have developed a representation which requires no knowledge of the transformation between views. The representation comprises descriptive images associated with oriented points on the surface of an object. Constructed using single point bases, these images are data level shape descriptions that are used for efficient matching of oriented points. Correlation of images is used to establish point correspondences between two views; from these correspondences a rigid transformation that aligns the views is calculated. The transformation is then refined and verified using a modified iterative closest point algorithm. To demonstrate the generality of our approach, we present results from multiple sensing domains.

1. Introduction

Surface registration is the process that aligns 3-D data sets acquired from different view points or at different times. A common application of surface registration is to spatially reconcile multiple views of a scene in order to generate more complete scene descriptions. Another application is to determine the position of the sensor as it traverses a scene by matching scene views taken at different times. An effective surface registration algorithm must be able to handle complex scenes and not make any assumptions about the transformation between views. To this end, we have developed a surface registration algorithm based on matching oriented points; this algorithm makes no assumption about the transformation between views and works on complex scenes containing free-form and polyhedral objects.

Our algorithm proceeds as follows: First, two views of a scene are acquired using a range sensor and converted into triangular surface meshes. Next, point correspondences are established using a new representation for matching points on the surfaces of objects. Next, sets of geometrically consistent point correspondences are used to compute a rigid transformation that aligns the views. Finally, the transformation is refined and verified using a modified iterative closest point algorithm.

Our new representation for matching oriented points, called a *spin-image*, is the fundamental contribution of this paper. The concept of a spin-images developed from ideas in basis geometric hashing [10] and structural indexing [11]. To create a spin-image, a local 2-D basis is computed at an oriented point (3-D points with surface normal) on the surface. The coordinates of the other points on the surface with respect to the basis are then used in a voting procedure to create the descriptive spin-image for the point. Information from the entire surface is used to generate spin-images, instead of a curve or surface patch in the vicinity of the point; thus, spin-images are more discriminating than the curves used in structural indexing. Because bases are computed from single points, our method does not have the combinatorial explosion present in basis geometric hashing as the amount of data is increased. Furthermore, a spin-image computed at any point on a surface is discriminating, so the need for extraction of salient features is eliminated.

Although spin-images can also be used in object recognition [7], the focus of this paper is on 3-D surface registration. Our approach to registration is similar to others in that we use correspondences between low-level surface parameters to calculate rigid transformations that align views using no prior knowledge about the transformation between views. Chua and Jarvis [3] present an algorithm for matching 3-D free-form surfaces by matching points based on principal curvatures. Bergevin et. al. [2] propose a registration algorithm based on matching properties of

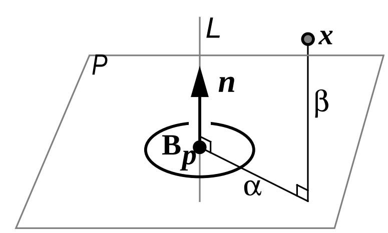

Figure 1. An oriented point basis.

This research was performed as part of the Artisan project in the Field Robotics Center and was supported by the US Department of Energy under contract DE-AC21-92MC29104.

triangles generated from a hierarchical tessellation of an object's surface. Our approach differs from these because the spin-image computed at a point is 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 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 [5] 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.

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 the surface of an object using surface position $\boldsymbol{p}$ and surface normal $\boldsymbol{n}$. We define surface normal as the normal of the best fit plane to the point and its neighbors in the mesh oriented toward the sensor. 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 P through $\boldsymbol{p}$ oriented perpendicularly to $\boldsymbol{n}$ and the line L through $\boldsymbol{p}$ parallel to $\boldsymbol{n}$. The two coordinates of the basis are α, the perpendicular distance to the line L, and β the signed perpendicular distance to the plane 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

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

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.

During recognition, we use coordinate systems defined at specific 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 3-D bases because an oriented point basis can be determined robustly and unambiguously almost everywhere on the surface of an object while a complete 3-D 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 3-D basis at each point, the known stable information (surface normal) is used to compute a 2-D 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.

2.1 Spin-image generation

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

Correspondences are established between oriented points by comparing spin-images. If spin-images are 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 2-D arrays of floating point numbers that are compared through image correlation.

To create the 2-D array representation of a spin-image

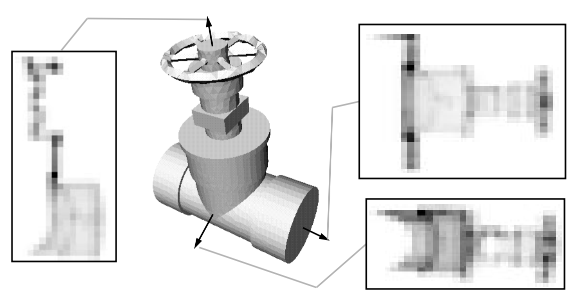

Figure 2. Some example spin-images for a CAD model of a valve

for the oriented point O on the surface of an object 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 bin B that the coordinates index in the 2-D array of floats is determined by discretizing (α,β). Finally, the array is updated by incrementing the bins surrounding B in the table. In general, the bin size is set to two times the resolution of the surface mesh (measured as the median of the edge lengths in the mesh). Figure 2 shows some spin-images for a CAD object.

In order to spread the position of the point in the 2-D array 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 surrounding bins in the 2-D array. This bilinear interpolation of the contribution of a point will spread the location of the point in the 2-D array, making the array less sensitive to the position of the point.

The idea of spin-images evolved from concepts used in geometric hashing. Our initial idea was to use all of the points on the surface of an object in a geometric hashing algorithm. Unfortunately, constructing coordinate systems from all tuples of points would lead to a combinatoric explosion in the indexing [10] (given the large number of points on the surface). Furthermore, coordinate systems constructed from tuples of points are very sensitive to the position of points sensed on the surface. Instead, we decided to encode the position of points with respect to a 2-D basis defined with one oriented point, in order to reduce the combinatoric explosion and position sensitivity. After matching points using a hash table, we determined that it would be just as effective and much more efficient to simply store an image that described the location of other points with respect to the oriented point instead of performing lookup in a hash table. From this, the concept of a spin-image was born. Using images to match points opens up the entire field of image-based matching, giving us powerful comparison tools such as image correlation.

Our system finds corresponding points by comparing spin-images. For the spin-images of two corresponding points on different views of the same scene to be similar, the resolution of the surface meshes (i.e., density of points over the surfaces) must be similar. This is a weaker constraint than requiring the positions of points to be the same for the two views. If the resolutions of the meshes are the same, then on average, each corresponding bin of matching spin-images will have the same number of points projected into them, making them correlated. We measure surface mesh resolution as the median of edge lengths in the mesh.

In general, objects imaged with the same sensor will have the same resolution. In cases where the density of points is different, we use a mesh simplification algorithm [7] to add and remove points from the surface meshes until the densities are the same. In most cases, the requirement of equal resolutions can be weakened, if the density of points in one view is linearly related to the density in the other Matching between meshes of different resolutions (10:1) is demonstrated with elevation maps in Figure 8.

A parameterized model of the effects of scene clutter on spin-image generation has shown that, if present, scene clutter is localized in spin-images and the effect of clutter increases smoothly as clutter is increased. Using this model, the effects of clutter and self occlusion can be controlled by setting two parameters that determine which points contribute to spin-image generation. The first parameter sets the maximum distance between the oriented point basis and a point in the mesh contributing to the spin-image. In general this threshold is set to one half the extent of overlap between the two meshes. The second parameter sets the threshold on the maximum angle between the oriented point basis surface normal and the surface normal of points in the scene. This threshold prevents points that will be self-occluded from contributing to the spin-image and is usually set to 90˚.

2.2 Establishing point correspondences

Spin-images generated from different views of a scene 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. A standard way of comparing linearly related images is the normalized linear correlation coefficient. The correlation coefficient can be used to rank point correspondences, so that correct and incorrect correspondences can be differentiated.

The linear correlation coefficient provides a simple and established 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). To handle clutter and occlusion robustly, spin-images are compared only in the bins where both of the images have data. In other words, the data used to compute the linear correlation coefficient is taken only from the region of overlap between two spin-images. In this case, knowledge of the spin-image generation process is used to eliminate outliers in the correlation computation.

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 relative 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

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

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 Equation 2, λ weights the variance against the expected value of the correlation coefficient. In practice, λ is set to three.

3. Global registration

The similarity measure provides a way to rank correspondences so that only reasonable correspondences are established. Suppose we wish to register two surface meshes: a model and scene. First, spin-images are generated for all points on a model surface mesh and then these images are stored in a spin-image stack. Next, 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 (2) for each image pair are calculated and inserted in a histogram. 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 fraction of scene points (~1/10 total scene points) that adequately cover the scene surface. The end result is a set of model/scene point correspondences which can be filtered and grouped in order to compute transformation from model to scene.

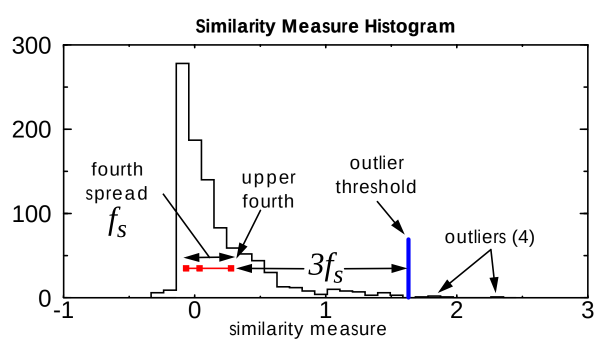

Figure 3. Similarity measure histogram

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 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 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 outliers. 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. Figure 3 shows a similarity measure histogram with detected outliers.

3.1 Grouping correspondences

Multiple correspondences are established between model and scene to account for sensor noise, scene symmetry and inherent symmetry in the spin-image representation. In order to compute plausible transformations from model to scene, it is necessary to filter the correspondences and then group them into sets that are geometrically consistent First, to eliminate correspondences between points that are outside of the area of overlap between the two views, correspondences with similarity measure that falls below a threshold are removed from consideration.

Two oriented point correspondences $[s_1,m_1]$ and $[s_2,m_2]$ are geometrically consistent if their spin-map coordinates (1) are within some distance threshold D_{gc}.

$$\left\|S_{m_1}(m_2) - S_{s_1}(s_2)\right\| < D_{gc} \qquad \left\|S_{m_s}(m_1) - S_{s_2}(s_1)\right\| < D_{gc} \tag{3}$$

Using distance between spin-map coordinates is a compact way to combine the constraints on position and normals using a single threshold. The threshold D_{gc} is set to two times the resolution of the model surface mesh.

With a list of N correspondences remaining after filtering, there exist a combinatoric number of possible sets of correspondences which can be used to compute a transformation between model and scene. To avoid this combinatoric explosion, geometric consistency and the similarity measure are used to partition the remaining correspondences into sets which can be used to compute plausible transformations. First the correspondences are sorted based on similarity measure. A set of correspondences is then created by iteratively adding correspondences in similarity measure rank order to the current set that are geometrically consistent with the current set. The end result is a disjoint partition of the correspondences where, in general, correspondences with high similarity measure are grouped together.

A plausible transformation T from model to scene is calculated from each geometrically consistent group $\{[m_i,s_i]\}$ of correspondences by minimizing [6]

$$E_T = \sum \|s_i - T(m_i)\|^2 . \quad (4)$$

The transformations and associated correspondence groups are then input into a verification procedure.

3.2 Iterative closest point verification

The purpose of verification is to find the best transformation between model and 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][13] that can handle partially overlapping point sets and arbitrary transformations between model and scene because it is initialized with a transformation generated from independently determined point correspondences. This is in contrast to traditional ICP [1] which requires an initial guess of the transformation between model and scene and assumes that one point set is a proper subset of the other.

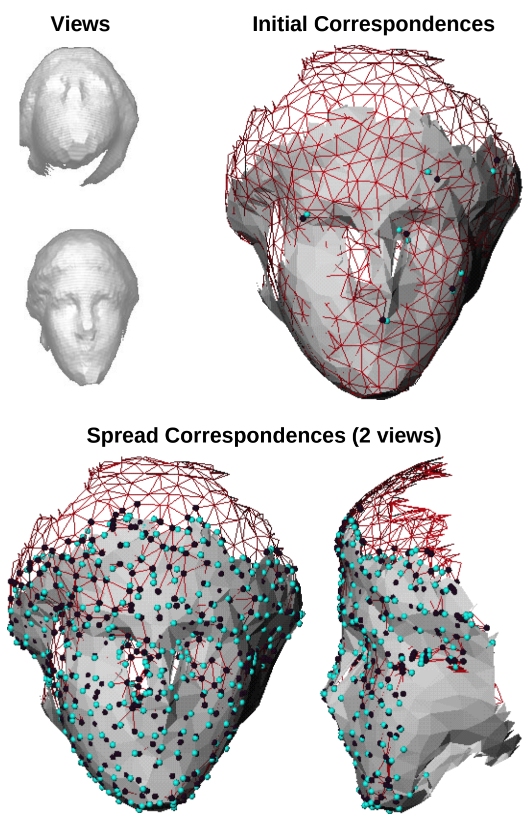

Figure 4. Registration of two views of a plastic model of the head of venus (taken with a structured light range sensor) demonstrating the spread of correspondences during verification.

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. 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 two times the resolution of the meshes. This threshold allows for noise but prevents establishment of correspondences in regions where the data sets do not overlap.

Our algorithm grows patches of correspondences between the model and the scene from the initial correspondences and a cascade effect occurs. If the match is good, a large number of points will be brought into correspondence; if the match is bad, the number of correspondences will remain close to the number in the original match. Therefore, a good measure of the validity of the match is the number of correspondences after verification. Since a large number of correspondences are used to compute the final transformation, the alignment will be more accurate than that computed through matching of spin-images.

Figure 4 illustrates how initial correspondences, established by matching spin-images, are spread over the surfaces of two range views (taken with a OGIS structured light range camera) 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.

4. Range Image Registration Results

An important task in interior modeling is the automatic registration of range images. By registering and merging range images, more complete scene descriptions are generated. Figure 5 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. Figure 5 shows the intensity channel of the range scanner for the three scene positions and the resulting 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.

Figure 6 shows the results of aligning three views of a stack of graphite bricks imaged with a structured light range sensor. These views also vary by approximately 30 degrees of rotation about the vertical axis. The top views of the registered points in both cases clearly show 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. This result demonstrates the effectiveness of our algorithm in registering scenes composed of polyhedral objects.

A common result in the biomedical imaging literature is

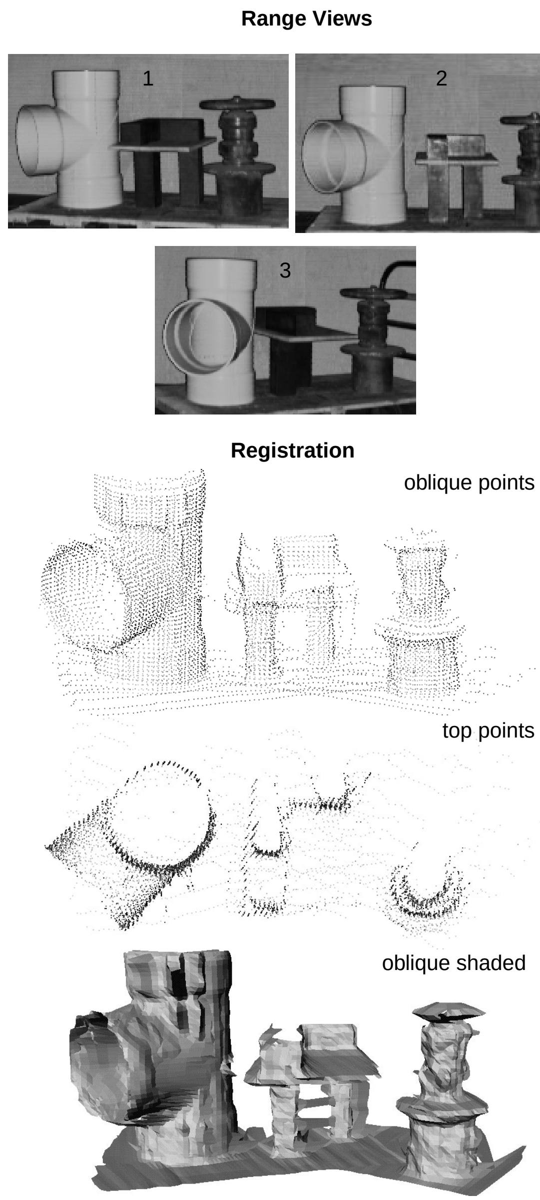

Figure 5. Registration of laser rangefinder images of industrial objects. The range views differ by roughly 30 degrees, so registering the views provides a more complete description of the scene.

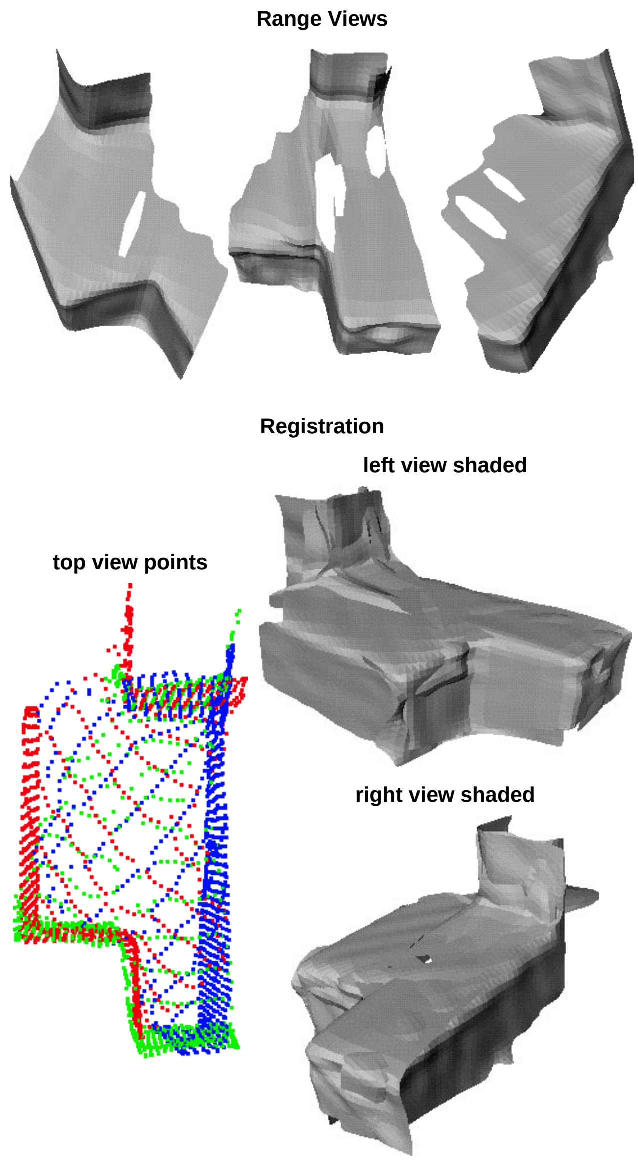

Figure 6. Registration of structured light range images of a stack of graphite bricks. The views differ by roughly 35 degrees. This result shows the ability of our algorithm to register views of polyhedral objects.

the registration of two skull data sets generated from volumetric medical scanners [5]. Figure 7 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 and then simplifying the meshes[8]. 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 Figure 7 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 Figure 7, the translation is $[0.019\ -0.069\ -0.013]^T$ mm and the fixed rotation angles are $[-0.06\ -0.03\ 0.018]^T$ degrees. This corresponds to a translation error magnitude of 0.072 mm, which is less than 2% of the resolution of the surface meshes (5.9mm) 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.

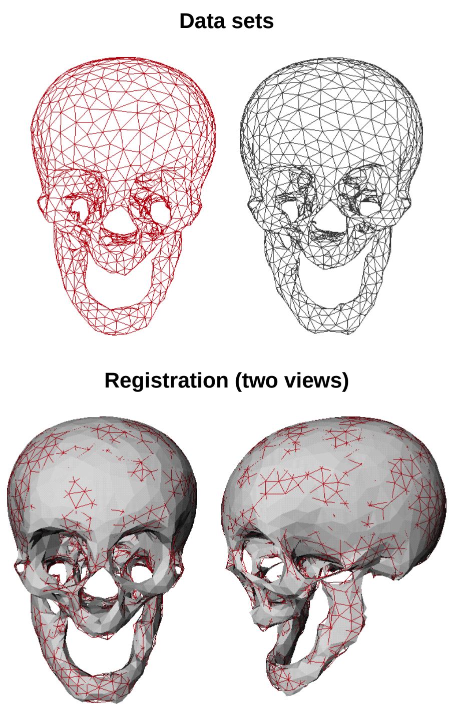

Figure 7. Registration of two skull data sets generated from CT. The accuracy of the registration (0.072mm) demonstrates the usefulness of our algorithm in medical registration procedures where the transformation between data sets is unknown or highly uncertain.

Figure 8 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 on-board 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 Figure 8 was generated by subsampling (10:1) a digital terrain map (real elevation data collected from the Lockheed Martin facility in Denver, Colorado) and the scene data set was generated from the cropping the same digital elevation map without subsampling the data. This result shows that our algorithm works even when the resolutions of the meshes being compared are substantially different.

5. Discussion

Spin-images offer a method for performing 3-D 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 oriented 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 3-D surface registration.

The computational complexity of our algorithm is much less than that attributed to methods of basis geometric hashing because our algorithm does not construct coordinates frames from multiple points. 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 spin-image stack 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 spin-image stack 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+SMlogM) 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(SMlogM)). Since logM 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. For the results shown in this paper M~1000, S~100 and I~100.

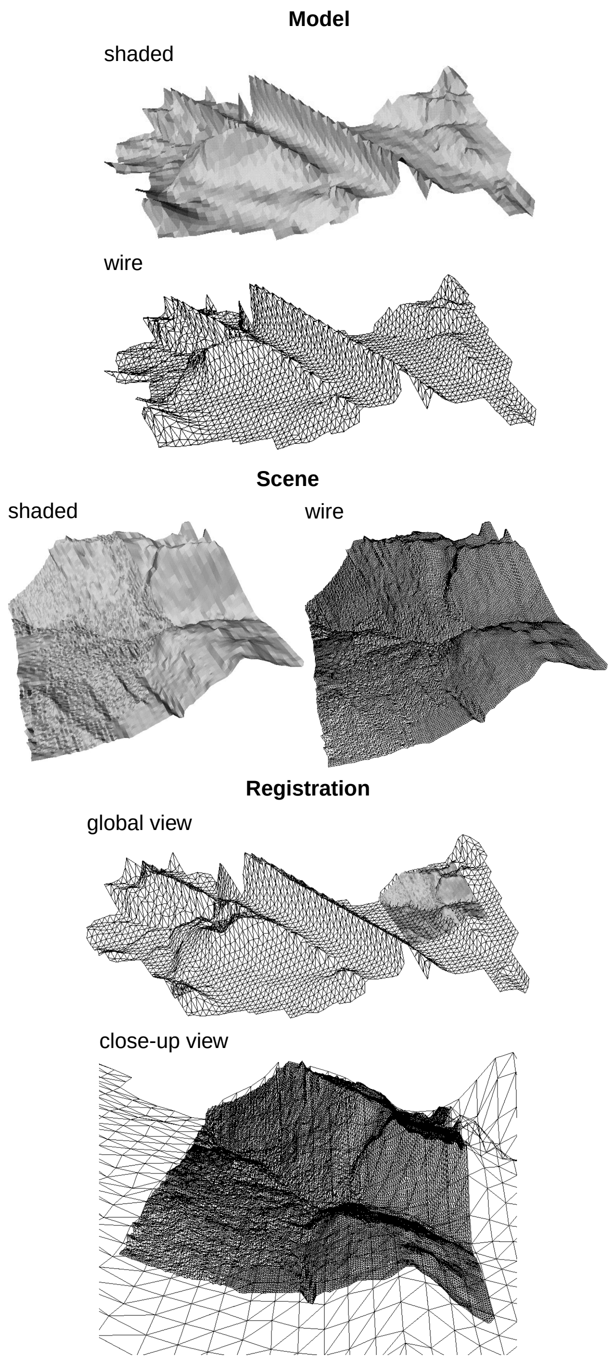

Figure 8. Registration of a fine local elevation map to a coarse global elevation map. This result demonstrates the ability of our algorithm to match meshes of different resolution.

6. Conclusion

We have presented a new representation for surface registration that is based on indexing of spin-images generated using oriented points on the surface of an object. The effectiveness of our new representation comes from its ability to combine the descriptiveness of global shape properties with the view invariance of local features.

Currently, we are developing analytical models to describe the effects of clutter and resolution on spin-image generation and comparison. Faster and more robust statistical methods for comparison and storage of spin-images are also being investigated.

References

[1] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.

[2] R. Bergevin, D. Laurendeau and D. Poussart, "Registering range views of multipart objects," *Computer and Vision Image Understanding*, vol. 61, no. 1, pp. 1-16, 1995.

[3] C. Chua and R. Jarvis, "3-D free-form surface registration and object recognition," *IJCV*, vol. 17, pp. 77-99, 1996.

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

[5] A. Guéziec and N. Ayache, "Smoothing and matching of 3-D space curves," *IJCV*, vol. 12, no. 1, pp. 79-104, 1994.

[6] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Optical Soc. Amer.*, vol. 4, no. 4, pp. 629-642, 1987.

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

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

[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. Stein and G. Medioni, "Structural Indexing: efficient 3-D object recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2): 125-145, 1992.

[12] J. Thirion, "New feature points based on geometric invariants for 3D image registration," *Int'l J. Computer Vision*, vol. 18, no. 2, pp. 121-137, 1996.

[13] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int'l J. Computer Vision*, vol. 13, no. 2, pp. 119-152, 1994.

Acknowledgments

We would like to thank André Guéziec for providing the skull data set and Karun Shimoga for use of his structured light range sensor.