

Multi-Resolution Spin-Images

H. Quynh Dinh
Department of Computer Science
Stevens Institute of Technology
quynh@cs.stevens.edu

Steven Kropac
Department of Computer Science
Stevens Institute of Technology
kropac@lucent.com

Abstract

Johnson and Hebert's spin-images have been applied to the registration of range images and object recognition with much success because they are rotation, scale, and pose invariant. In this paper we address two issues concerning spin-images, namely: (1) comparing uncompressed spin-images across large datasets is costly, and (2) a method to select the appropriate bin size and image width for spin-images is not clearly defined.

Our solution to these issues is a multi-resolution method that generates a pyramid of spin-images by successively decreasing the spin-image size by powers of two. To efficiently correlate surface points, we compare spin-images in a low-to-high resolution manner. Once multi-resolution spin-images are generated for a given object, we have found that the different resolutions can also be used to compare objects that have differing or non-uniform point densities. To select the appropriate bin sizes for comparing such objects, we use the ratio of the average edge lengths of the objects. We also show preliminary results of using the pyramid to converge on the appropriate image width by traversing the pyramid in a low-to-high resolution manner looking for the highest resolution at which the fewest number of highly correlated points are found to match a given feature point.

1. Introduction

Two applications of spin-images and other local shape descriptors are registering range scans of a single object from multiple viewpoints and matching or recognizing objects. In both these applications, pairs of matching points are used to recover the rigid transformation between scans taken of the object from different viewpoints, as done in [8, 12, 18, 20, 19, 21, 17, 28, 30, 33, 34, 36, 35] among many others examples. Two other important applications of corresponding surface points are shape metamorphosis [11, 23, 31, 26] and recognizing objects undergoing non-rigid transformations [3, 24, 4]. In shape metamorphosis, corresponding points between two objects are used to

solve for a rigid transformation and a thin-plate deformation that aligns the objects. Belongie et al. also use the thin-plate spline to compute a measure of shape difference between objects that are similar through a non-rigid deformation [3, 4].

Spin-images have been particularly successful in shape matching because they are rotation, scale, and pose invariant [8, 18, 20, 19, 21, 17, 10, 5]. There are several unresolved issues concerning spin-images, however. One is that finding the point that corresponds to a given spin-image on an object of n points requires a costly search over all n spin-images of the object. In [21], Johnson and Hebert reduce the dimensionality of spin-images and use closest point search to reduce the time required for matching. In this paper, we describe a multi-resolution method to efficiently match points that, unlike dimensionality reduction, is not lossy. We construct a pyramid of spin-images by successively decreasing the spin-image size by powers of 2, and perform point matching in a low-to-high resolution manner that successively prunes the list of potential matches. A second disadvantage of the original spin-images algorithm is that the density of the data points must be uniform across the surfaces being compared. This issue was resolved in [8, 17], but we have found that multi-resolution matching can also effectively compare models with differing or non-uniform sampling densities. A final issue we address is the selection of appropriate values for spin-image parameters – namely the bin size and image width of spin-images. We use the ratio of the average edge lengths of the objects being compared as the bin size ratio, and we show preliminary results for converging to the appropriate image width by traversing the pyramid in a low-to-high resolution manner looking for the highest resolution at which the fewest number of highly correlated points are found to match a given feature point.

In Section 2, we review spin-images and briefly cover other shape descriptors. In Section 3, we describe our multi-resolution spin-image implementation and its use in correlating surface points. In Section 4, we show timing and correlation results of multi-resolution matching.

2. Related Work

Because our work is an extension of the spin-image representation of Johnson and Hebert, we review how spin-images are generated and briefly compare the representation to other shape descriptors.

2.1. Spin-Images

In [18, 20, 19, 21], Johnson and Hebert generate the spin-image for an oriented surface point by spinning the plane containing the normal vector about the normal axis while binning all surface points as they intersect the plane. Throughout the paper, we will call the surface point for which the spin-image is generated the *central point*. The spin-image is indexed by the radius (α) from the central point and the depth (β) from the central point's tangent plane (see Figure 1). Each spin-image pixel is a bin that stores the number of surface points with the given radius (α) and depth (β). Bilinear interpolation is used to spread the contribution of a surface point when the point does not fall exactly on the coordinates of a bin (which is often the case). Note that all points on a cylinder around the central point map to the same α coordinate. In [17], Johnson uses a spherical, rather than cylindrical, parameterization. In either case, the angular dimension around the central point is lost in the spin-image representation. A spin-image is computed for each point on the surface of an object (or each vertex of the mesh representing the object surface).

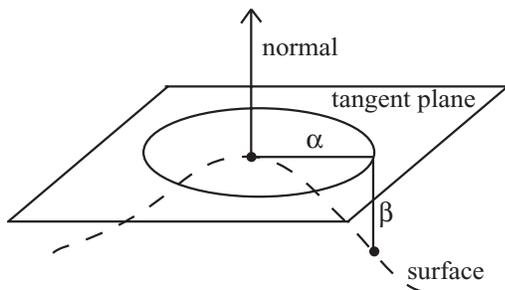


Figure 1. The (α, β) coordinates of a surface point relative to another surface point (central point).

Uniform sampling is required in order for spin-images of corresponding points on two different meshes to match. In the original spin-images implementation, only mesh vertices are aggregated into spin-images [21]. Hence, Johnson and Hebert apply resampling algorithms to ensure that the mesh resolution is uniform. In [8, 17], Carmichael et al. and Johnson avoid resampling, and instead, aggregate uniformly distributed surface points (between mesh vertices) into spin-images. Although this method can be used in conjunction with multi-resolution spin-images, we show in Section 4.2 that the different spin-image resolutions alone have the potential to match surfaces with non-uniform point densities.

To find which points of an object correspond to an input test point (whether on another object or on the same object), the spin-image of the test point is generated. The correlation coefficients of the test point's spin-image to all spin-images of the object provides a measure of how similar the surface points are to the test point. Points with the highest correlation coefficient are most similar to the test point. The correlation coefficient c is given by the following relation where s_1 and s_2 are the two spin-images being compared, and n is the number of pixels in a spin-image.

$$c = \frac{n \sum s_1 s_2 - \sum s_1 \sum s_2}{((n \sum s_1^2 - (\sum s_1)^2)(n \sum s_2^2 - (\sum s_2)^2))^{1/2}} \quad (1)$$

Two key parameters in generating spin-images are the support length (image width in [21]) and bin size of the images. The support length determines the locality of the spin-image. We define the support length as a fraction of the object size (diagonal of the bounding box). If the support length encloses the entire model (factor of 1.0), then the spin-image tallies all surface points of an object. As the support length decreases, surface points far away from the central point are not included in the spin-image. Consequently, a small support length means that only surface points close to the central point are tallied, resulting in a local descriptor. In this way, spin-images can be varied from being a global to a local shape descriptor.

The bin size determines whether neighboring points are binned together. With a small bin size, neighboring surface points will likely fall into separate (α, β) bins; whereas with a large bin size, neighboring surface points will more likely fall into the same bin. Johnson and Hebert set the bin size as a multiple of the surface resolution (computed as the average edge length of the object mesh). When spin-images are correlated, small differences between images are more apparent when the bin size is small; whereas with a large bin size, small differences in bin values do not contribute significantly to the correlation. Bin size, like support length, can be varied to change the spin-image from a more to a less discriminating shape descriptor.

Varying support length and bin size is similar to down-sampling or up-sampling regular images. We exploit this fact to create a pyramid of spin-images with the highest resolution spin-image at the bottom, and lowest resolution at the top (see Figure 2a). We generate two pyramids – one by varying bin size b (Figure 2b), and the other by varying support length l as a fraction of the object size m (Figure 2c). The resulting spin-image size (where the width is the number of α bins, and the height is the number of β bins) is given by the following equations. The number of β bins is about twice the number of α bins because the distance to the central point's tangent plane may be positive or negative, whereas the radius is always positive.

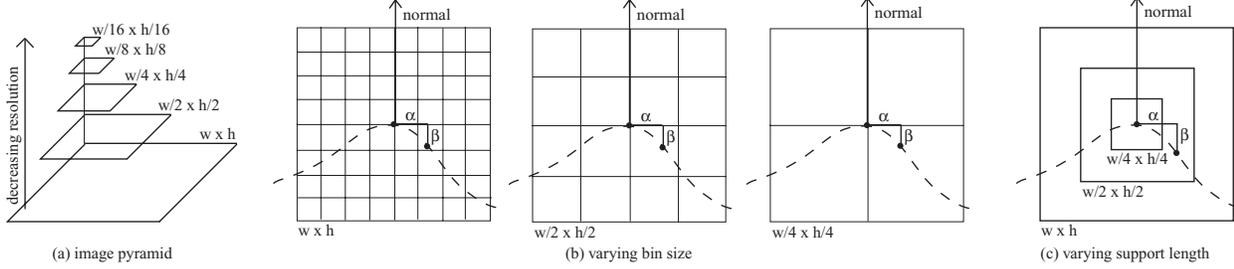


Figure 2. (a) Image pyramid. Multi-resolution spin-images constructed by increasing bin size (b) and reducing support length (c). Note that changing bin size does not eliminate surface points from the resulting spin-image, but reducing support length does.

$$width = \frac{ml}{b} + 1 \quad height = \frac{2ml}{b} + 1 \quad (2)$$

In [21], Johnson and Hebert use principle component analysis (PCA) to compress the spin image, thereby reducing the spin-image dimensions. Once compressed, they compute the L_2 norm of the Eigenvector coefficients (values obtained by projecting the spin-images onto the Eigenvectors) rather than the correlation coefficient between the original spin-images. They also use a closest point search to find the best matching spin-image in the lower dimensional space. Instead of compressing spin-images which is lossy, we retain all the information in a multi-resolution spin-image representation, and we efficiently correlate points in a low-to-high resolution manner.

2.2. Shape Matching

Many shape descriptors have been developed for object matching or recognition. They can be categorized as point-based, patch-based, and global descriptors. Global methods generate a shape signature to compare objects but cannot be used to locally compare surface points for the purpose of point matching. Examples are invariant histograms [16], shape distributions [25], and spherical harmonics [22]. Patch-based methods segment the object surface into patches, and then match patches to template patches either by patch type [7] and/or pose estimation [7, 27, 13]. Objects are often segmented into patches by aggregating similar points using local shape descriptors [7, 27]. Although point correspondences can be found after patches are matched, segmenting models into patches is an additional computation.

Extended Gaussian images and spherical attribute images (SAI) store the shape signature in a spherical domain [14, 15]. Two objects match if a rotation can be found that matches node or point mass values stored on the spheres. The matching nodes provide the point-wise correspondences between two objects. Unfortunately, both SAI and extended Gaussian images are only applicable to objects that are homeomorphic to a sphere. In geometric

hashing [32], a hash table is generated by selecting a pair of surface feature points as the basis and redefining every other feature point with respect to the basis. To perform recognition, two feature points in the input scene are used as the basis to which all other input feature points are defined. These transformed feature points vote for a matching model and basis pair in the database. As noted in [21], spin-images are more discriminating than geometric hashing because all surface points (not just feature points) are incorporated into the spin-image and is not dependent on effective feature extraction.

Algorithms most related to spin-images are local shape descriptors that encode all surface information relative to a surface point in the same manner as spin-images. These include splash, surface signatures, point signatures, point fingerprints, harmonic maps and shape contexts. Splash encodes normal variation along a geodesic circle around a central point [28]. Surface signatures encode the distance and normal variation between a central point and every other feature point in an image [33, 34]. They differ from spin-images in that the image pixels are not bins accumulating surface points, but rather, store the actual distance and average normal variation from the central point. Point signatures record the variation around a central point by computing the distance of a patch boundary around the central point to a plane fitted to the boundary [9]. Point fingerprints are generated by projecting geodesic contours around a central point onto the central point's tangent plane [29, 30]. Point fingerprints do not scale effectively into a global descriptor. In [36], harmonic maps are computed for patches and stored in images to reduce the surface matching problem to one of image matching (like spin-images). Point signatures, point fingerprints, and harmonic maps are not rotationally invariant. Shape contexts can be described as a 2D variant of spin-images. For each point on the silhouette of a shape, the shape context is a log-polar histogram in which each bin stores the number of other points that are some distance and angle from the point of interest [2, 1]. An extension to 3D shape contexts is presented in [12] along with harmonic shape contexts. Although, Frome et al. show that 3D shape contexts have a higher recognition rate than spin-images in

noisy and cluttered scenes, the methods are comparable, and a multi-resolution framework may be applied to either. We believe that a multi-resolution representation can augment many local shape descriptors to speed-up point matching. In this paper, we use spin-images as an effective example.

3. Multi-resolution Spin-Images

To create multi-resolution spin-images, we decrease image size by increasing bin size and reducing support length.

3.1. Increasing Bin Size

We can decrease the image size by increasing the range of radii and depths that a bin covers (increasing bin size). By doing so, we are essentially reducing the radius and depth resolutions (see Figure 2b). Decreasing the resolution in this way does not change the locality of the spin-image which is controlled by the support length. If the high resolution spin-image tallies points of all radii and depth ranges, the low resolution version also does.

Image pyramids are constructed by downsampling a high resolution image. Going up the pyramid, the image dimensions are reduced by half from one level to the next. In typical image pyramids, such as [6], the low resolution images are created by averaging neighboring samples of the high resolution image. For spin-images, we cannot average bin values to create a lower resolution spin-image because the bin values reflect the number of surface points collected at (α, β) coordinates away from the central point. Instead, we must accumulate neighboring values to create a lower-resolution spin-image from a high resolution image. To accurately spread the contribution of a surface point to neighboring bins, Johnson and Hebert bilinearly interpolated a surface point's (α, β) coordinates. As we accumulate the values of neighboring bins in a high resolution spin-image to generate a low resolution image, we must also spread each bin's contribution to neighboring low resolution bins. In 1D, we do so by dividing the value of every other bin equally among the two neighboring bins in the lower-resolution spin-image as shown in Figure 3. This method scales to 2D spin-images where bin values are divided equally among two or four neighboring bins.

3.2. Decreasing Support Length (Radius and Depth)

A second way we can reduce image size is by decreasing support length along both the radius and depth dimensions (see Figure 2c). To create a pyramid of spin-images based on decreasing support length, we iteratively halve the spin-image along both dimensions in going from high to lower resolutions and throw away three-quarters of the higher resolution spin-image at each iteration. In other words, we clamp the lower resolution spin-image at half the high resolution radius and half the high resolution depth in going

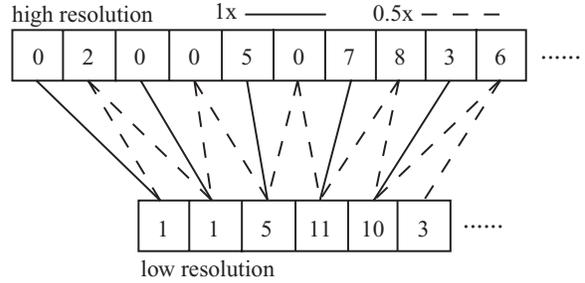


Figure 3. 1D example of creating a lower resolution spin-image from a high resolution spin-image. Solid and dashed lines indicate how bin values in boxes are distributed.

down one resolution. In practice, we do not store separate images for each level of the pyramid to save on memory and database space. Instead, we compute the correlation coefficient on a subset of the image when correlating at smaller support length.

3.3. Multi-resolution Matching

In a spin-image pyramid, there are now more images to correlate when comparing two points. However, we actually reduce the time required to find the best matching surface points to a given test point by computing the correlation coefficient for most points on low resolution spin-images and fewer points at higher resolutions. Comparing low resolution spin-images is faster because the time to compute the correlation coefficient between two spin-images grows linearly with size. In general, two points' spin-images are similar at low resolution and become increasingly discriminative at higher resolutions. Hence, points that are uncorrelated at low resolution will not be highly correlated at higher resolutions. For spin-image pyramids based on support length, this invariance is true because lower resolution images are a subset of the high resolution spin-image. The image difference measured by the correlation coefficient will only increase as more bins are included in the calculation as we go down the pyramid.

For pyramids based on increasing bin size, this invariance is less obvious. We speculate that the correlation coefficient is greater for two low resolution spin-images than for their higher resolution counterparts despite the n factor. The $\sum s_1$ and $\sum s_2$ values in Equation 2.1 remain the same since the total value of all the bins remains the same from one resolution to the next, whereas all sum of squared values and the $\sum s_1 s_2$ term is greater for low resolution images since they aggregate bins of high resolution images. Hence, points that are not highly correlated at low resolution (where the correlation coefficient is higher) will not be highly correlated at high resolutions. We exploit this invariant to iteratively cull potential matches as we correlate points in a low-to-high resolution manner for both types of spin-image pyramids.

The pruning of potential matching candidates from one resolution to the next is based on a threshold correlation coefficient below which the surface point is no longer considered a potential match. At each level in the pyramid, we pass only points with correlation coefficients of 0.95 or above (out of a max of 1.0) to the next higher resolution. At the highest resolution, there are only a few number of candidate points remaining for which the correlation coefficient must be computed. An example of how quickly points are culled in this process are shown in Table 2. In practice, we do not traverse all resolutions in the pyramid. Instead, we have found that computing the correlation coefficient and pruning candidate matches at every other pyramid level is a good balance between the increased number of spin images for correlation per surface point and the reduction in the number of candidate points. The speed-up we obtain is 2x or more to find the best matching surface points using multi-resolution spin-images compared to single resolution matching at the highest resolution. Table 1 summarizes the results of self-correlation (where the test point is a surface point on the object) on several models.

4. Results

In the following sections, we show the speed-ups we achieve using multi-resolution spin-images. Because we want to focus on the efficiency and accuracy of spin-image pyramids in matching points, we demonstrate examples of correlating a selected surface point on an object with all other surface points of the same object (self-correlation). We show that the appropriate bin size is imperative to correlating surface points of models that do not have the same point density or do not have uniform sampling. Finally, we describe the potential for using spin-image pyramids to determine an appropriate support length for matching.

4.1. Multi-resolution Matching

Table 1 (plotted in Figure 4) shows that performing low-to-high resolution matching of a test point to all surface points on an object is more efficient than matching based only on the highest resolution spin-image in each pyramid. Table 2 shows the number and percentage of surface points that are culled away after each iteration in performing low-to-high resolution matching on the Stanford Bunny model with both types of pyramids. The first row in Table 2 shows the number of surface points above 0.95 correlation when using only the highest resolution spin-image for matching (single resolution matching). Nearly the same number of matching points are identified when using bin size pyramids and support length pyramids. Note that we do not get the exact same matches because the correlation threshold used at lower resolutions is not strictly equivalent to that used at higher resolutions even if the threshold value is the same. This is due to the non-linearity of Equation 2.1. Fig-

ure 5 shows the surface points (in red) at each iteration in low-to-high resolution matching that pass the threshold for the Stanford Bunny and for a cluttered scene. As expected, the number of remaining potential matches decreases as we go to higher resolutions. At each iteration, the correlation coefficient is computed only for those points that pass the threshold in the previous iteration. In all tables and figures, the bin size is a multiple of the average edge length, the support length is given as a percentage of the model size (diagonal of bounding box), and the max correlation is 1.0.

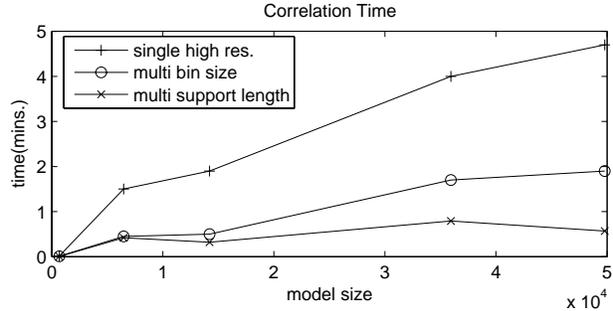


Figure 4. Correlation Times vs. Model Size

model	num. of points	single res.	multi-res. (bin size)	multi-res. (support)
Buddha	49794	4.7	1.9	0.57
Bunny	35947	4.0	1.7	0.79
Octopus	14210	1.9	0.5	0.32
Fandisk	6475	1.5	0.45	0.42
Distr. cap	685	0.014	0.008	0.006

Table 1. Single vs. Multi Resolution Correlation Times (mins.)

bin size	support	corr. thresh.	verts > thresh.	% culled
4	100%	0.95	7	99.98
64	100%	0.99	5144	85.69
16	100%	0.95	1024	97.15
4	100%	0.95	6	99.98
4	3.125%	0.95	14138	60.67
4	6.25%	0.95	6297	82.48
4	100%	0.95	5	99.98

Table 2. Stanford Bunny Correlation Results

We have also simultaneously varied bin size and support length such that the resulting spin-image resolution is kept fixed. The purpose of doing so is to keep the correlation time constant while optimizing on the bin size and support length. Figure 6 shows the percentage of vertices of the Stanford Bunny with correlation above 0.95 plotted against the various combinations of support lengths and bin sizes used to compute correlation. For each line plot, the combinations result in a fixed image size (4x7 or 7x12). As expected, a large bin size is not discriminating, and many

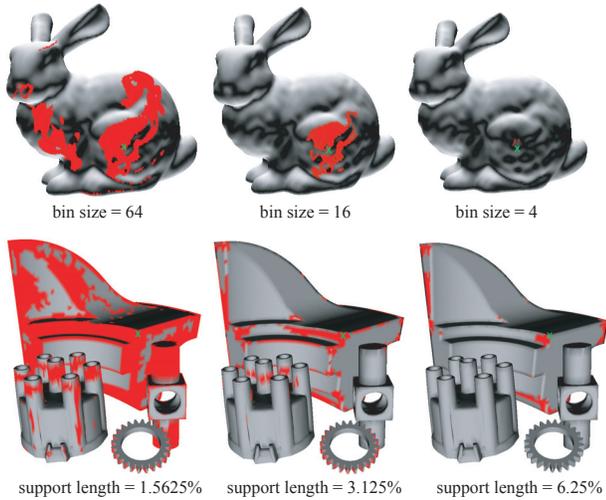


Figure 5. Top: low-to-high resolution correlation of Stanford Bunny to a selected point (\mathbf{x}) on the model. Bottom: low-to-high resolution correlation of a cluttered scene to a selected point (\mathbf{x}) on the Fandisk model. Only points with correlation coefficients higher than the threshold (light regions) are used in the correlation calculation at the next higher resolution.

vertices are highly correlated even though support length is at 100%. At the other extreme, a very small support length is also not discriminating. The optimal support length and bin size for a fixed resolution spin-image falls in the middle, though there is a slight bias towards smaller bin sizes. The higher resolution spin-images (7x12) are more discriminating than the lower resolution spin-images (4x7).

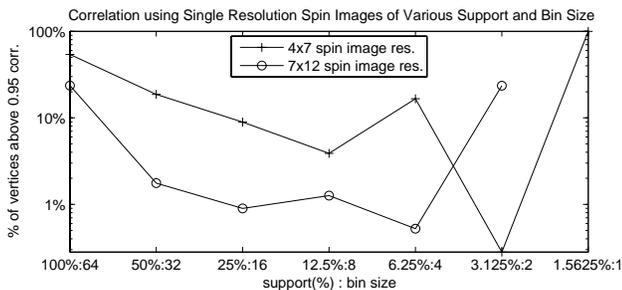


Figure 6. Correlation vs. Support Length and Bin Size

In [21], Johnson and Hebert show that compressing spin-images using PCA and closest point search significantly reduces the time to find best matching points. In order to compute the most representative subspace (set of Eigenvectors), all the objects in the database must be included in the computation. Matching is then assumed to be on sample points from an object that is in the database. This assumption is valid for recognition applications in which we are trying to identify objects in a scene given a database of possible objects. Even if the surface point is from an object that is not in the database, matching using the com-

pressed spin-images will likely identify good matches, but the match is not guaranteed to be as accurate as an exhaustive search on the original, uncompressed spin-images, and the space spanned by the compressed images may not be representative of new spin-images (that were not included in the computation of the Eigenvectors).

Although multi-resolution spin-images may not be as efficient as compressed spin-images for matching large numbers of objects, multi-resolution spin-images can be generated for any new object that enters the database. Low-to-high resolution matching will identify the same set of matching points as exhaustive single resolution matching and does not depend on selecting an appropriate level of compression to retain the ability to discriminate between similar points. Additionally, multi-resolution and compression are not mutually exclusive. Compression can be applied to spin-image pyramids for more efficient matching.

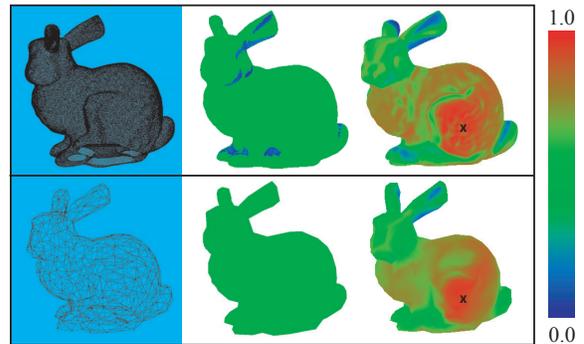


Figure 7. Left: high (top) and low (bottom) resolution models of the Stanford Bunny. Top: selected point (\mathbf{x}) on low resolution model is correlated with the high resolution model. Bottom: selected point (\mathbf{x}) on high resolution model is correlated with the low resolution model. Center: correlation using highest resolution spin-images. Right: correlation using spin-images with corrected bin sizes of 9.1 to 1. Correlation color coding is shown at the right.

4.2. Matching Surfaces of Differing or Non-uniform Densities

Once multi-resolution spin-images are generated and stored for a given object, the different resolutions can be used to compare objects that have differing point densities. For example, the simplified version of an object will have a much lower point density than the high resolution version of the object. If spin-images of the same bin size are used to compare points of the two objects, the match will fail as seen in Figures 7 and 8 even when normalized by the total number of surface points. We can, however, compute appropriate bin sizes at which the point densities will be equalized based on the average edge length of the object (if the object is not represented as a mesh, it is possible to use the average distance to neighboring points within a given radius). The ratio of the larger average edge length

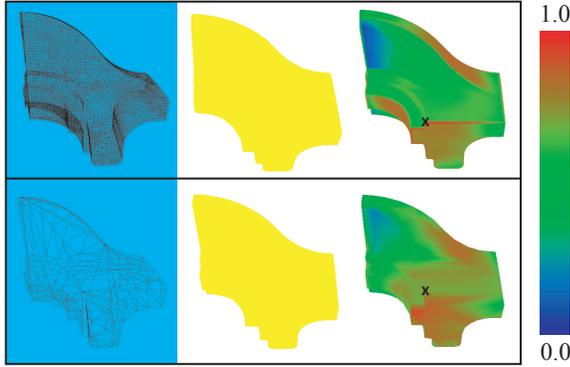


Figure 8. Left: high (top) and low (bottom) resolution models of the fan disk. The low resolution model has non-uniform point density. Top: selected point (\mathbf{x}) on low resolution model is correlated with the high resolution model. Bottom: selected point (\mathbf{x}) on high resolution model is correlated with the low resolution model. Center: correlation using highest resolution spin-images. Right: correlation using spin-images with corrected bin sizes of 6.8 to 1.

over the smaller average edge length is used as the bin size for the higher resolution mesh, while a bin size of 1 is used for the lower resolution mesh (recall that bin size is multiplied with the average edge length). For example, the average edge length of the low resolution Stanford Bunny model is 0.052, while the high resolution model has an average edge length of 0.0057 giving a ratio of 9.1 to 1. We have found that even though a simplified mesh tends to have non-uniform sampling (fewer surface points at planar regions than in regions of high curvature), bin sizes based on edge length remains effective as shown in Figure 8 where the simplified fan disk has non-uniform sampling. To more robustly handle non-uniform sampling, interpolative sampling methods such as [8, 17] should be used in generating the multi-resolution spin-images.

4.3. Selecting the Appropriate Support Length

One ambiguity of spin images as described in [18, 20, 19, 21] is that the appropriate bin size and image width for comparing objects is not apparent. Johnson and Hebert suggest values that they found effective. As we have shown in the previous section, we can use the ratio of average edge lengths to select appropriate bin sizes for comparing models. In this section we show that the spin-image pyramid based on support length has the potential to automatically identify a suitable support length. We stress that this approach has not been thoroughly verified.

In preliminary experiments, we have found that the spin-image pyramid can be used to converge to an appropriate support length by tallying the number of surface points that are highly correlated to an input test point. As we go from the top of the pyramid to the bottom, the number of surface points that are highly correlated to the test point decreases.

We speculate that the level at which there are highly correlated points but at which the number of such points is small is the appropriate level for matching. The heuristic we use is to find the the lowest pyramid level (largest support length) at which there remain highly correlated points (below this level there are none). With this heuristic, it is necessary to define a threshold describing *high* correlation. Throughout this paper, we consistently use thresholds of 0.95 and 0.99.

An example of using spin-image pyramids to converge to an appropriate support length is shown in Figure 9 where the tip of the nose of one face (face on the left) is correlated with the surface points of another face. In this example, we have already selected the bin size according to edge lengths. At the maximum support length (100%), none of the surface points are highly correlated to the test point. As we go from left to right – low resolution (small support length) to high resolution – the number of points with a high correlation coefficient decreases, but the points around the tip of the nose remain highly correlated throughout. From this progression, we automatically identify 12.5% as the appropriate support length and simultaneously identify the surface point that best matches the test point.

5. Conclusions

We have described a multi-resolution representation for spin-images to efficiently compare uncompressed spin-images in a low-to-high resolution manner. We have also described ways for selecting bin size and support length. In future work, we intend to conduct a more thorough analysis on determining appropriate support length, as well as explore ways of extending spin-images so that they can handle matching of points between deformed objects.

References

- [1] S. Belongie and J. Malik. Matching with shape contexts. *Proc. of Int. Conf. Computer Vision (ICCV)*, 2001.
- [2] S. Belongie and J. Malik. Shape contexts enable efficient retrieval of similar shapes. *Proc. of Conf. Computer Vision and Pattern Recognition (CVPR)*, 2001.
- [3] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. *Proc. of Int. Conf. Computer Vision (ICCV)*, 2001.
- [4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE TPAMI*, 24(24):509–522, 2002.
- [5] N. Brusco, M. Andreetto, A. Giorgi, and G. Cortelazzo. 3d registration by textured spin-images. *Proc. of Int. Conf. on 3D Digital Imaging and Modeling*, pages 262–269, 2005.
- [6] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Tr. on Comm.*, 31(4):532–540, 1983.
- [7] R. Campbell and P. Flynn. Recognition of free-form objects in dense range data using local features. *Proc. of Int. Conf. on Pattern Recognition (ICPR)*, 3:607–610, 2002.

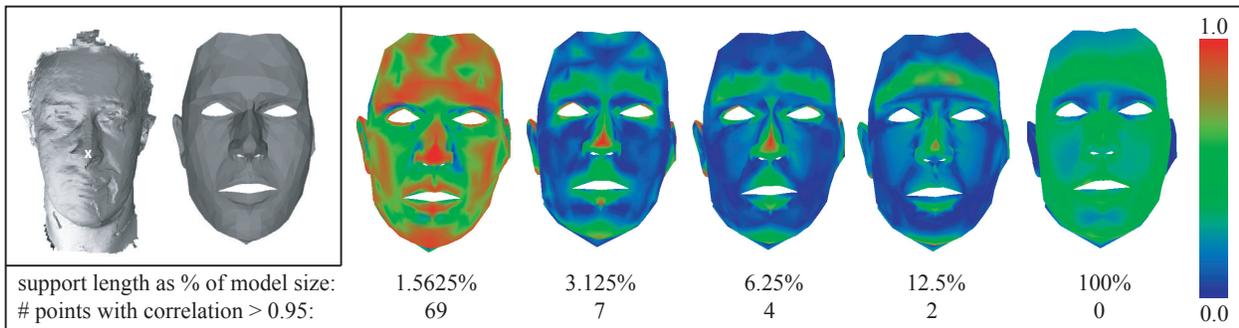


Figure 9. Correlating the selected point (\mathbf{x}) on one face model (left) to the surface points of the other face model. Left to right: the number of points that are highly correlated to the selected point decreases as the support length increases. Right: at the maximum support length none of the points are found to be highly correlated to the selected point. Correlation color coding shown at the right.

- [8] O. Carmichael, D. Huber, and M. Hebert. Large data sets and confusing scenes in 3-d surface matching and recognition. *Proc. of Int. Conf. on 3-D Digital Imaging and Modeling (3-DIM)*, pages 358–367, 1999.
- [9] C. Chua and R. Jarvis. Point signatures: A new representation for 3-d object recognition. *Int. Journal Computer Vision*, 25(1):63–85, 1997.
- [10] P. Claes, D. Vandermeulen, L. Gool, and P. Suetens. Automatic, robust and accurate 3d modelling based on variational implicit surfaces. *Katholieke Universiteit Leuven, Center for Processing Speech and Images Technical Report*, 2004.
- [11] D. Cohen-Or, D. Levin, and A. Solomovici. Three dimensional distance field metamorphosis. *ACM Trans. on Graphics*, 17(2):116–141, 1998.
- [12] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. *Proc. of European Conf. on Computer Vision (ECCV)*, 3:224–237, 2004.
- [13] A. Gruen and D. Akca. Least squares 3d surface and curve matching. *PandRS*, 59(3):151–174, May 2005.
- [14] M. Hebert, K. Ikeuchi, and H. Delingette. A spherical representation for recognition of free-form surfaces. *IEEE TPAMI*, 17(7):681–690, 1995.
- [15] B. Horn. Extended gaussian images. *Proc. of the IEEE*, 72(12):1671–1686, 1984.
- [16] K. Ikeuchi, T. Shakunaga, M. Wheeler, and T. Yamazaki. Invariant histograms and deformable template matching for sar target recognition. *Proc. of Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 100–105, 1996.
- [17] A. Johnson. Surface landmark selection and matching in natural terrain. *Proc. of Conf. Computer Vision and Pattern Recognition (CVPR)*, 2:413–420, 2000.
- [18] A. Johnson and M. Hebert. Surface registration by matching oriented points. *Proc. of Int. Conf. Recent Advances in 3-D Digital Imaging Modeling*, pages 12–15, 1997.
- [19] A. Johnson and M. Hebert. Efficient multiple model recognition in cluttered 3-d scenes. *Proc. of Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 671–677, 1998.
- [20] A. Johnson and M. Hebert. Recognizing objects by matching oriented points. *Proc. of Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 684–689, 1998.
- [21] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE TPAMI*, 21(5):433–449, 1999.
- [22] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. *Eurogr. Sym. Geometry Processing*, 2003.
- [23] A. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. *SIGGRAPH*, pages 343–350, 1999.
- [24] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. *Proc. of Conf. Computer Vision and Pattern Recognition*, pages 723–730, 2001.
- [25] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Tr. Graphics*, 21(4):807–832, 2002.
- [26] E. Praun, W. Sweldons, and P. Schröder. Consistent mesh parameterizations. *SIGGRAPH*, pages 179–184, 2001.
- [27] I. Stamos and M. Leordeanu. Automated feature-based range registration of urban scenes of large scale. *Proc. of Conf. Computer Vision and Pattern Recognition*, 2:555–561, 2003.
- [28] F. Stein and G. Medioni. Structural indexing: efficient 3-d object recognition. *IEEE TPAMI*, 14(2):125–145, 1992.
- [29] Y. Sun and M. Abidi. Surface matching by 3d point’s fingerprint. *Proc. of 8th Int. Conf. Computer Vision (ICCV)*, pages 263–269, 2001.
- [30] Y. Sun, J. Paik, A. Koschan, D. Page, and M. Abidi. Point fingerprint: a new 3-d object representation scheme. *IEEE Tr. Sys., Man, and Cybernetics*, 33(4):712–717, 2003.
- [31] G. Turk and J. O’Brien. Shape transformation using variational implicit functions. *SIGGRAPH*, pages 335–342, 1999.
- [32] H. Wolfson. Geometric hashing: an overview. *IEEE Computational Science and Engineering*, pages 10–21, 1997.
- [33] S. Yamany and A. Farag. Free-form surface registration using surface signatures. *Proc. of 7th Int. Conf. Computer Vision (ICCV)*, pages 1098–1104, 1999.
- [34] S. Yamany, A. Farag, and A. El-Bialy. Free-form object recognition and registration using surface signatures. *Proc. of Int. Conf. on Image Processing*, 2:457–461, 1999.
- [35] S. Yiyong, J. Paik, A. Koschan, D. Page, and M. Abidi. Point fingerprint: A new 3-d object representation scheme. *IEEE Trans. Sys., Man, and Cybernetics*, 33(4):712–717, 2003.
- [36] D. Zhang and M. Hebert. Harmonic maps and their applications in surface matching. *Proc. of Conf. Computer Vision and Pattern Recognition*, 1999.