

International Journal of Image and Graphics
© World Scientific Publishing Company

Directional Coherence Interpolation For Three-Dimensional Gray-Level Images

Yongmei Michelle Wang

*Department of Diagnostic Radiology, Yale University, P. O. Box 208042
New Haven, CT 06520-8042, USA
wang@noodle.med.yale.edu*

Jingdan Zhang, Zhunping Zhang

*Computer Science and Technology, Tsinghua University
Beijing 100084, P. R. China*

Baining Guo

*Microsoft Research Asia, 5F Sigma Center
Beijing 100080, P. R. China
bainguo@microsoft.com*

Received (15 February 2003)

Revised (30 July 2003)

Accepted (30 August 2003)

A novel three-dimensional gray-level interpolation method called Directional Coherence Interpolation (DCI) is presented in the paper. The principal advantage of the proposed approach is that it leads to significantly higher visual quality in 3D rendering when compared with traditional image interpolation methods. The basis of DCI is a form of directional image-space coherence. DCI interpolates the missing image data along the maximum coherence directions (MCD), which are estimated from the local image intensity yet constrained by a generic smoothness term. In order to further improve both the algorithm efficiency and robustness, we also propose to apply a pyramidal search strategy for MCD estimation. This coarse-to-fine scheme requires less computation time by starting with the reduced amount of data and propagating searching results to finer resolutions. DCI can incorporate image shape and structure information without the prior requirement of explicit representation of object boundary / surface. Extensive experiments were performed on both synthetic and real medical images to evaluate the proposed approaches. The experimental results show that the proposed methods are able to handle general object interpolation, while achieve both accuracy and efficiency in interpolation compared with existing techniques.

Keywords: Interpolation; Directional coherence; Shape; Smoothness; Multiresolution; Visualization.

1. Introduction

Image interpolation is widely used in computer vision, especially in biomedical image processing, visualization, and analysis¹⁵. Most three-dimensional (3D) biomed-

ical volume images are sampled anisotropically, with the distance between consecutive slices significantly greater than the in-plane pixel size. Either prior to display and measurement or during these manipulations, the volume image must be transformed in order to compensate for this anisotropy. This is done by creating a number of new slices between two known slices using image interpolation. Image interpolation is usually required for proper 3D image visualization and analysis, as the 3D data must be isotropic in order to produce the correct aspect ratio along each direction²⁴. Other situations where multidimensional image interpolation is needed include: changing the orientation of the discretization grid (resectioning); combining image information about the same object from multiple modalities with different resolution (multi-modality registration); and changing grid systems, such as from polar to rectangular. Therefore, in most cases, 3D biomedical volume images need to be interpolated to isotropic dimensions and potentially transformed in orientation in order to achieve the desired quality for visualization and/or quantitative analysis in 3D.

The conventional interpolation scheme for 3D volume image is a first-order linear interpolation. In this scheme, unknown gray values of the “new voxels” between known voxels are linearly interpolated¹⁷. When the distance between consecutive slices is more than five times the in-plane pixel size, or when the location of a boundary between two uniform regions shifts considerably between two adjacent slices, linear interpolation will result in poor approximations and a jagged staircase artifact is produced, which is clearly visible in renderings of these volumes²⁴. Higher order functions may be exploited^{15,21} to reduce such artifacts with increased computation time.

Object-based methods have been introduced to overcome the shortcoming of the traditional interpolation, where object information extracted from a given scene is used in guiding the interpolation process. Among these methods, in addition to slice matching or correspondence^{6,34}, cores²², and morphology-based algorithms¹⁴, shape-based interpolations^{1,7,10,11,23} are also popular examples. Shape information has been widely used in biomedical image processing and analysis, such as for interpolation (referred above), segmentation^{26,31}, and registration^{30,32}. There are basically two types of shape-based interpolation: the algorithm for binary images^{10,23} and the one for gray-level images^{1,7}. In the former scheme, the objects of interest in each image are first segmented to generate a set of sliced binary image. Distance transforms are then performed to compute the closest distance to the desired edge of the structure (both inside and outside) for every voxel in the volume image and interpolate these distance values to estimate the location of the structure²³. Prior requirement of specific structure identification (segmentation) before interpolation limits the usefulness of the shape-based interpolation for binary images. To circumvent the segmenting process, a generalization of the binary shape-based method to gray-level data has been proposed in⁷. Two complementary stages called lifting and clasping are added at the beginning and the end so that the binary shape-based method can be applied to create an (n+1)-D binary interpolated

image. Although this shape-based method is able to deal with gray-level images, distance transform of a high dimensional array is extremely time consuming.

In order to pursue a high visual quality gray-level interpolation with modest computation cost, we develop a new method called Directional Coherence Interpolation (DCI) ³³. It interpolates missing image data along the smoothed maximum coherence directions (MCD) between image slices instead of the coordinate axes as that in linear interpolation. By this scheme, the object shape and structure information is well preserved without explicit representation of object boundary / surface (segmentation). Therefore, the interpolated image can lead to dramatic improvement for 3D volume rendering in visualization. In addition, in order to further reduce the computation time and improve the interpolation accuracy, we use a multiresolution scheme for MCD search ³⁵.

Multiresolution (or pyramidal) processing is a powerful idea ^{2,3,25}, whose birth is motivated by human perception. The basic principle is to first build an image pyramid. The procedure of image processing or analysis is then performed from the coarsest level to the finest level that corresponds to the image itself. The key advantages of this scheme are computation speed and processing quality. There are numerous applications of this concept in the literature, such as for medical image analysis ^{4,5,26,28,30}, texture synthesis ⁹, and wavelet transform ¹⁸, etc. As mentioned above, in this work, we apply the similar principle for searching Maximum Coherence Direction in 3D image interpolation to improve both computation efficiency and interpolation quality.

The remainder of the paper is organized as follows. In section 2, we give an overview of the proposed methodology. The theoretical basis, the algorithm and implementation details of our approaches are described in Sections 3, 4, 5, and 6. Experiments and results are presented in Section 7, followed in Section 8 by conclusions and suggestions for future work.

2. System Overview

The system framework of our proposed Single Resolution Directional Coherence Interpolation (SR-DCI) is shown in Fig. 1. It consists of the following stages. We first partition the space between two slices taken from the volume image into small cubes. Then for each cube, we calculate its local Maximum Coherence Direction (MCD) by minimizing an integral function called directional discrepancy. The incorporation of a smoothness constraint gives us a smooth directional coherence map that is also consistent with the local MCDs. The MCD for each voxel is derived by point-wise MCD computation from the cube-based MCDs. The gray-level values of “new voxels” between the slice pair are then interpolated along the point-wise MCDs. Refinement scheme is applied to interpolate the images to the desired scales. A high-resolution 3D gray-level image with satisfactory object structure is then generated by repeating this whole process for all slice pairs of the volume image.

The difference between our Multiresolution Directional Coherence Interpolation

4 *Y. M. Wang, J. Zhang, Z. Zhang, and B. Guo*

(MR-DCI) and the SR-DCI is: MR-DCI performs cube-based MCD search (consisting of the gray procedure boxes inside the dash square in Fig. 1) hierarchically on the built image pyramid. In this way, the search process is faster and the derived MCDs are more robust.

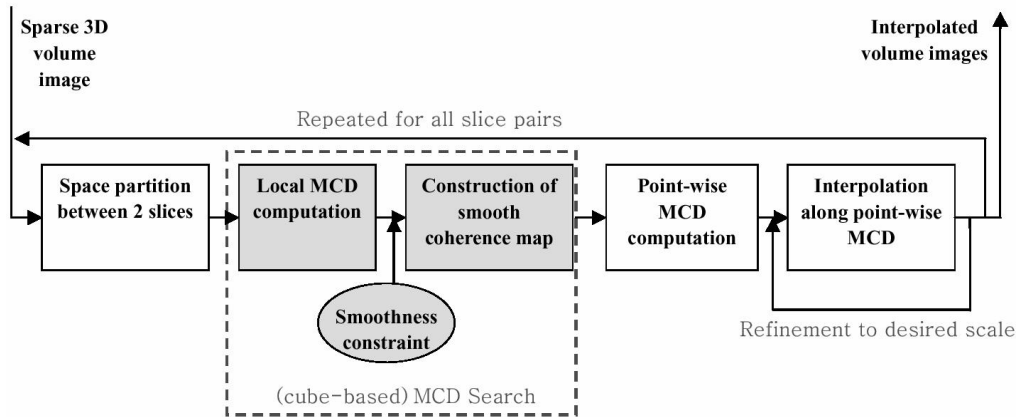


Fig. 1. System diagram of our Single Resolution Directional Coherence Interpolation (SR-DCI).

3. Single Resolution Directional Coherence Interpolation (SR-DCI)

3.1. Discontinuity and Directional Coherence

The basis of our proposed approach is a form of image-space coherence called directional coherence. Coherence is the degree to which parts of a scene or its projection exhibit local similarities²⁷. Since image data change abruptly across edge, we usually think that a discontinuity edge (in 2D) or surface patch (in 3D) is the break of coherence. However, discontinuity does not break all forms of coherence. In fact, image data are typically coherent along the direction of the discontinuity edge even if they change abruptly across the edge¹⁹. This idea has been successfully applied in 2D for contour-based image coding¹⁹, and recently in computer graphics for progressive radiance evaluation⁸. In this work, we introduce and utilize the 3D form of this type of coherence, called directional coherence, into our interpolation problem.

Given a 3D image, we first divide the volume space between two consecutive slices with spacing k into small cubes with size $k \times k \times k$, as shown in Fig. 2(a), so that most cubes are crossed by no more than one discontinuity surface patch. Moreover, the surface patch is expected to have small curve as the example in Fig. 3(a) rather than as the one in Fig. 3(b). A simple way to capture a discontinuity

surface within a cube is to build a mathematical model for the surface. Since the cube is sufficiently small, the surface patch can be regarded as flat, as shown in Fig. 3(a). We then model the surface patch behavior by locating its end points on the cube boundary facets. This is essentially the scheme we take, although the basic idea is modified in several ways in order to solve our specific problem — image interpolation.

The maximum coherence direction discussed below forms an integrated approach to extract discontinuity from image data, and then provides us the important image shape and structure information for interpolation.

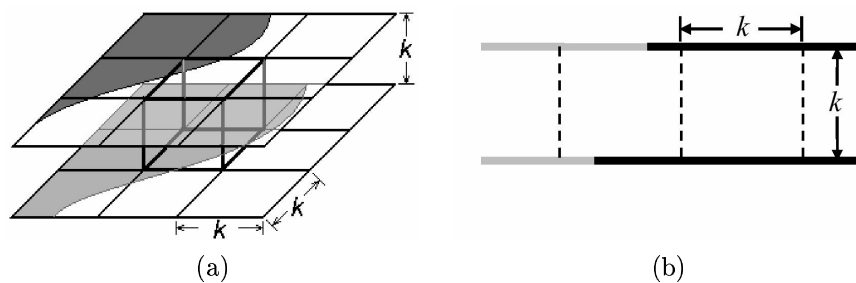


Fig. 2. Diagram of our volume space partition scheme between two slices. (a) 3D case; (b) 2D profile with the top and bottom lines representing the two image slices that need to be interpolated. (Note: The dark area in (a) and dark line segments in (b) denote the existence of object / structure.)

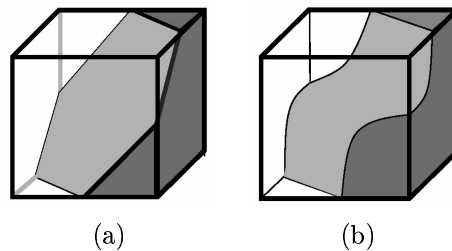


Fig. 3. Discontinuity analysis in a cube. The discontinuity in (a) is considered simple whereas that in (b) is not, because of the curvedness.)

3.2. Local Maximum Coherence Direction (MCD)

In this work, we assume that the divided cubes are small enough so that the discontinuity characteristics of each cube can be approximated by a 3D unit vector — Maximum Coherence Direction (MCD), as shown in Fig. 4. The local MCD $\vec{n}(C_k)$ of a $k \times k \times k$ cube can be calculated by minimizing the surface integral

$$d(\vec{n}) = \frac{1}{A} \int_S (f(\vec{u} + t(\vec{u})\vec{n}) - f(\vec{u}))^2 ds \quad (1)$$

where S is the boundary surface of cube C_k and A is a normalization constant. For a given direction \vec{n} and a point \vec{u} on S , the scalar $t(\vec{u})$ is chosen such that the parametric line $\vec{v}(t) = \vec{u} + t(\vec{u})\vec{n}$ intersects the boundary surface S of C_k at \vec{u} and $\vec{v}(t) = \vec{u} + t(\vec{u})\vec{n}$ (Fig. 4). Once the intensity function $f(\vec{u})$ is known on S through surface evaluation, the *directional discrepancy* is then a well-defined function of the direction \vec{n} .

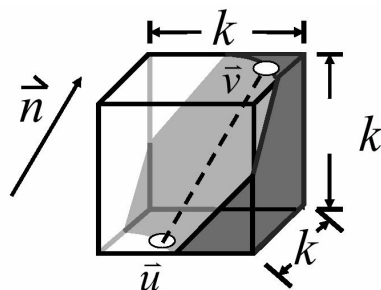


Fig. 4. System diagram of our Single Resolution Directional Coherence Interpolation (SR-DCI).

However, for our interpolation problem here, although the intensity function $f(\vec{u})$ is known on the image slices (shown as the top and bottom facets in Fig. 4), we do not have $f(\vec{u})$ value for points on the other four facets. In order to solve this problem, we use an image data mapping technique. The idea of this mapping is to simply take the projected corresponding intensity values of the closest within-slice-points along the given direction \vec{n} as the unknown value $f(\vec{u})$. The basis of this scheme is that the contents and coherence of the image should be well preserved along the MCD. A 2D analogy of this technique is shown in Fig. 5. The corresponding within-slice-points on line segments EI and GL are outside the elementary block EFGH. But for the given directions \vec{n} , their intensity values will be used during integration due to the unknown $f(\vec{u})$ values on line segments EH and FG, respectively. In the case of 3D cube, as shown in Fig. 6, for the given direction \vec{n} , the intensity values on the outside-cube-facets and will be used as the $f(\vec{u})$ values for points on the front facet and the right-side-facet, respectively. By this mapping method, we then get all the intensity function $f(\vec{u})$ over the whole cube boundary surface S .

In implementation, a discrete exhaustive search scheme is utilized to find the local MCD. The discrete directional discrepancy $d(\vec{n}_i)$ in 3D is calculated as

$$d_i = d(\vec{n}_i) = \frac{1}{6k^2} \sum_{\vec{w} \in W} [f(\vec{w} + t(\vec{w})\vec{n}_i) - f(\vec{w})]^2$$

where W is the set of all points on the cube facets. Also, in order to minimize the error caused by the data mapping technique described above, we limit the direction search range such that the length of the projected line segments (EI and GL in Fig. 5 for 2D) does not exceed certain limit, for example, the cube side length k .

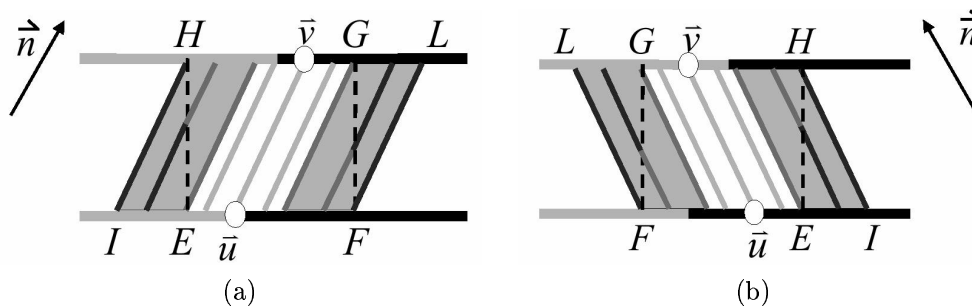


Fig. 5. Diagram for image intensity mapping in 2D. (a) and (b) are for different given orientations \vec{n} .)

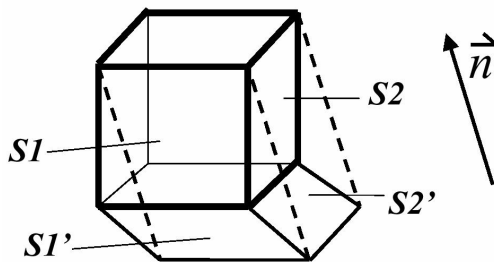


Fig. 6. Diagram for image intensity mapping in 3D.

Please note that with the above partition strategy, the cube side length is set as the slice spacing k so that both the top and bottom facets of the cube contain real intensity information (see Fig. 2(a)). In fact, there is a trade-off between the cube size and the algorithm performance. Theoretically, the cube should be small enough so that the MCD can better approximate the discontinuity characteristics of this cube. On the other hand, if the cube size is too small, for example, when it equals a voxel size, the MCD will not be able to incorporate any local directional coherence

during the estimation. The resulting interpolation algorithm is then equivalent to the linear interpolation, and it is no longer DCI. Our experiments also show that using the slice spacing k as the cube side length is a reasonable choice.

3.3. Directional Coherence Map with Smoothness Constraint

Given the local MCDs computed above, we now seek to construct a smooth directional coherence map that describes the coherence direction over the full space between the two slices. A natural complementary constraint can now be added to the above estimation procedure. For sufficient small cubes, we assume that adjacent cubes between the two slices tend to have similar direction of coherence due to the smooth variation of the object / structure surfaces represented in the image volume. Thus, the coherence map construction is an optimization process by compromising between adhering to the local MCD and the constraint that the MCDs for adjacent cubes between the two slices change smoothly. The problem can be defined as

$$\vec{n}^* = \operatorname{argmin}_{\vec{n}} F = \operatorname{argmin}_{\vec{n}} [d(\vec{n}) + \lambda \cdot s(\vec{n})] \quad (2)$$

where \vec{n}^* is the argument that minimizes $F \forall \vec{n}$, λ is a weighting coefficient, and $d(\vec{n})$ is the directional discrepancy in Eq. (1). We define the smoothness term as

$$s(\vec{n}) = \sum_{j=1}^P \|\vec{n} - \vec{n}_j\| \quad (3)$$

where P is the number of neighboring local MCDs used in the constraint. The degree or scale of smoothing can be controlled by using different number of MCDs of the neighboring cubes. The motivation or goal of the use of this smoothness term is similar to that in boundary finding^{12,31} and cardiac motion estimation²⁰.

For implementation, we employ a direct iterative method to solve for the optimal coherence map by allowing the energy term to relax over iterations. The initial MCDs used in the optimization are the local MCDs computed according to Eq. (1). During iterations, for each cube, the MCD is updated by evaluating the objective function in Eq. (2). The final converged coherence map gives us a set of smoothed MCDs for the corresponding cubes between the two image slices.

4. Multiresolution Directional Coherence Interpolation (MR-DCI)

4.1. Motivation

In general, the single resolution algorithm described above works well. However, SR-DCI has following drawbacks: i) An iterative process is employed to calculate the smoothed MCDs. In every iteration, we must exhaustively search each cubes all possible directions to determine the value of Eq.(1). Because the direction range for each cube must be large enough to obtain the correct MCDs, the exhaustive search

process is quite time-consuming. ii) When the distance between two existing slices is large, the cubes size also becomes large, the discontinuity characteristics of each cube can not be well approximated by the estimated MCD, which will therefore affect the final interpolation quality.

These problems can be solved by using a multiresolution pyramid which represents large scale structures more compactly by a few pixels in a certain lower resolution pyramid level. The two key benefits of this technique are: First, it decreases the computation cost because most iterations are spent in processing reduced versions of the image data. Second, it usually also improves the algorithm accuracy, since the pyramid has a smoothing effect on the criterion that often minimizes the likelihood of being trapped by local optima^{3,25}.

4.2. Pyramidal Search Scheme for MCDs

A Gaussian pyramid is first constructed from each existing slice image by using a low-pass down-sampling operation³. The pyramid image, $G_i(I)$, at level i , for image slice I , is given by:

$$G_i(I) = 2 \downarrow [G_{i-1}(I) \otimes g] \quad (4)$$

where $2 \downarrow [\cdot]$ is the $2 \times$ down-sampling operation; g is a two dimensional Gaussian kernel; and $G_0(I) = I$, the original image slice. In this way, image resolution decreases by a factor of $\frac{1}{2}$ after each operation in Eq. (4). In implementation, we empirically utilize 3 levels of Gaussian pyramid.

After constructing an image pyramid, we start applying the MCD search at the coarsest resolution on a small version of the image data. Upon convergence, the solution is propagated to the next finer resolution where it is used as the starting condition. For each slice pair at each pyramid level G_i , the smoothed MCDs are calculated in a way similar as that for the single resolution case with the following modifications: At each level, the initial MCD value and the direction search range for every cube are determined based on the respective resolution level. Specifically, at the coarsest level (Level 2 in Fig. 7), the initial MCD of the cube is set to be vertical to the cubes top and bottom facets. For the other fine levels (Level 1 and Level 0 in Fig. 7), the initial MCD of the cube is obtained through bilinear interpolation of the MCDs that are already calculated in the previous lower resolutions (corresponding Level 2 and Level 1, respectively). A hidden condition during the search is that the length of the cubes at all levels is the same. As to the direction search range, at the coarsest level, the range must be large enough to cover all the possible directions; whereas at finer levels, this range can be gradually decreased since the major aim of the fine-scale search is to refine the MCDs estimated from the previous coarser level. This type of coarse-to-fine iteration strategy is proceeded until the finest level of the pyramid (Level 0 in Fig. 7) that corresponds to the image itself is reached.

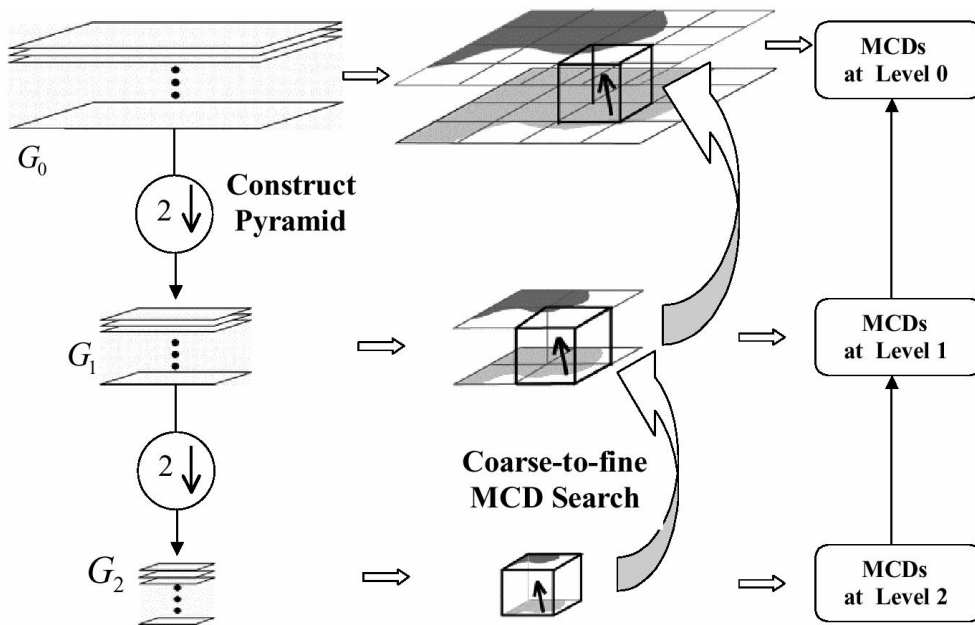


Fig. 7. Coarse-to-fine flow diagram of 3-level MR-DCI. (The highlighted cube is an elementary cube at each search level, with the arrow inside representing its MCD.)

5. Point-wise MCD Computation

At this stage of the process, the MCDs are cube-based no matter the algorithm is SR-DCI or MR-DCI. That is, all points within one cube only have one MCD. In order to assign a MCD for every point and ensure the neighboring points MCDs vary smoothly, we employ a bilinear interpolation of MCDs. As shown in Fig. 8, O_1 , O_2 , O_3 , and O_4 represent the center points of 4 nearby cubes. Their MCDs are defined as the respective cubes estimated ones from Section 3 (SR-DCI) or Section 4 (MR-DCI). The point-wise MCD for an arbitrary point, P , within these 4 cubes, is calculated through bilinear interpolation of the 4 nearby center points MCDs.

6. Gray-level Interpolation with Refinement

Given the point-wise smooth MCD for each voxel, the gray-level value of the “new voxel” is then interpolated linearly from the values of points with known values along its individual MCD. Refinement scheme is applied to interpolate the images to the desired scale (Fig. 1 in Section 2). That is, the original image is first interpolated to double the resolution. Then with the double-resolution image as input, we repeat the same procedure until the desired scale is achieved.

The volume image is thoroughly interpolated when the whole process in Sections 3 through 6 is repeated for all the slice pairs. This leads to an isotropic 3D image

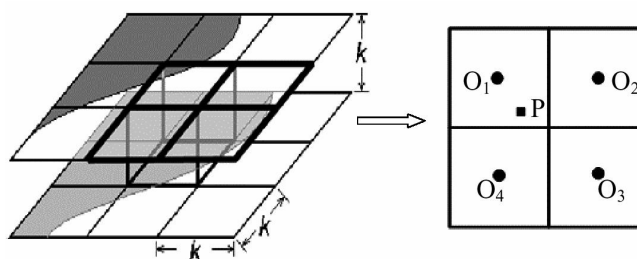


Fig. 8. Diagram of point-wise MCD computation.

with well preserved shape and structure.

7. Experimental Results

We have compared the proposed Directional Coherence Interpolation (DCI) - including Single Resolution DCI (SR-DCI) and Multiresolution DCI (MR-DCI) - with the traditional linear interpolation (LI) and the shape-based interpolation applicable to gray level images (SBI), based on our own implementation. In LI, the unknown intensities of “new voxels” between existing voxels are simply linearly interpolated between the intensities of the existing voxels along the coordinate axes. As mentioned in the introduction, shape-based interpolation (SBI) can be divided into two categories: one for binary images and the other for gray-level images. Since DCI is a gray-level interpolation, a reasonable and meaningful comparison with SBI shall be the one applicable to gray level images⁷. The comparisons have been performed on both synthetic images - for some simple tests, and real medical images - with results on visual quality, quantitative error measures, computation cost, as well as the sensitivity to noise experiments.

7.1. Synthetic Images

We have designed several synthetic images to provide some tests so that we can clearly understand how certain aspects of the object features are affected by the interpolation methods.

7.1.1. Comparison of DCI (SR-DCI) and LI

The experiment in Fig. 9 shows a simple comparison of DCI and LI. The results show that DCI gives more reasonable interpolated slice, since the interpolation is along the smoothed maximum coherence direction with the objects shape and structure well preserved.

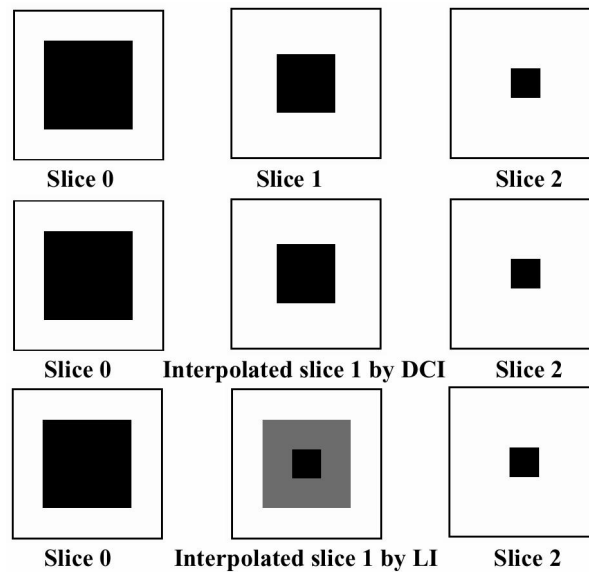


Fig. 9. Comparison of DCI and linear interpolation (LI) on synthetic image. Top row: three consecutive slices taken from a 3D synthetic image containing a frustum; Middle row: corresponding slices with the middle one interpolated by DCI (Note: SR-DCI is used here); Bottom row: corresponding slices with the middle one interpolated by LI.

7.1.2. *Effect of Smoothness Constraint*

Fig. 10 demonstrates the effect of the smoothness constraint, as well as the comparison between DCI and LI. From the results, we can see that without the smoothness constraint, DCI is not able to interpolate the missing slice (slice 1) correctly. This is because when the objects cross sections do not overlap on the slices that need to be interpolated (Slice 0 and Slice 2 in Fig. 10), the local MCDs over the non-overlap areas are not able to reasonably reflect the 3D objects shape and structure. However, the smoothed MCDs can capture the right object feature information. Also, linear interpolation generates a false slice since the interpolation is purely along the third dimension and no other shape or structure information is included.

7.1.3. *Branching Case for LI, DCI (SR-DCI) and SBI on Synthetic Image Pair*

The experiment in Fig. 11 shows the case for object split. The left and right columns are the original slice pair (Slice 0 and Slice 1). The three middle slices at each row are respectively the interpolated ones by LI, DCI and SBI, respectively. Similarly, the linear interpolation leads to unreasonable results. Both our DCI (SR-DCI is used here) and the shape-based method are able to mimic the object split through the interpolated slices. However, after we add noise to the original two input slices, the

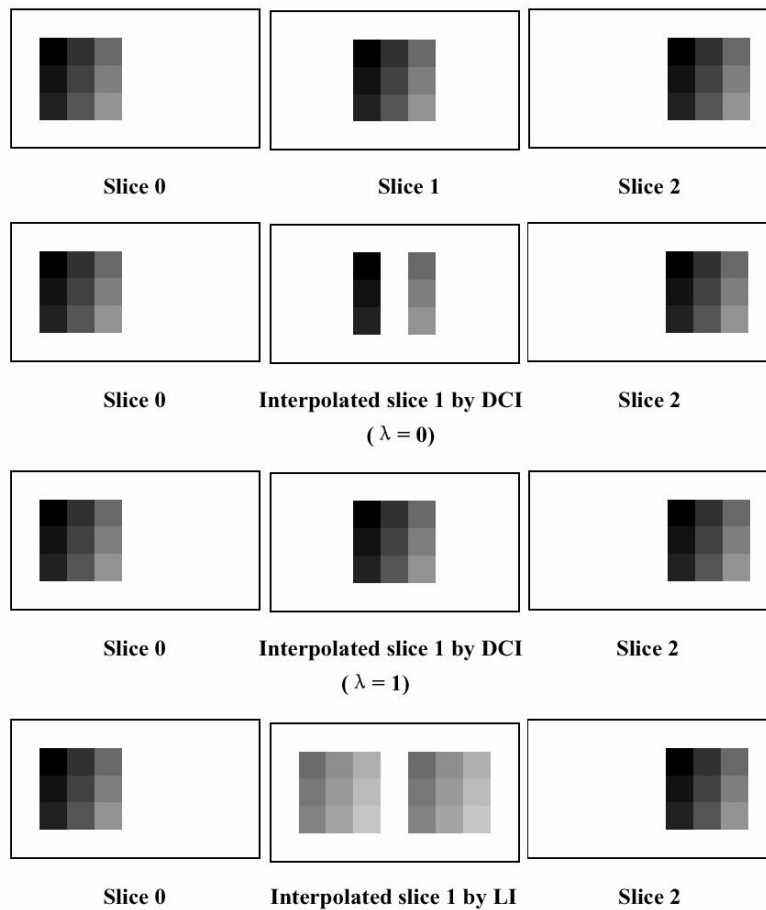


Fig. 10. Effects of smoothness constraint on synthetic image. Top row: three consecutive slices taken from a 3D synthetic image; 2nd row: corresponding slices with the middle one interpolated by DCI (Note: SR-DCI is used here) with $\lambda = 0$ (no smoothness constraint); 3rd row: corresponding slices with the middle one interpolated by DCI (SR-DCI) with $\lambda = 1$; Bottom row: corresponding slices with the middle one interpolated by LI.

interpolated slices from SBI are not reasonable any more (Fig. 12). This is because SBI can be disrupted by noise and might connect the wrong pieces, whereas our DCI still performs well and it is not quite sensitive to noise. The detailed sensitivity to noise experiments about the three methods will be demonstrated in Section 7.2.4.

7.1.4. Robustness of MR-DCI on Synthetic Image Pair

The experiment in Fig. 13 demonstrates the robustness of MR-DCI. The left and right columns are the original slice pair (Slice 1 is a rotated, translated and scaled

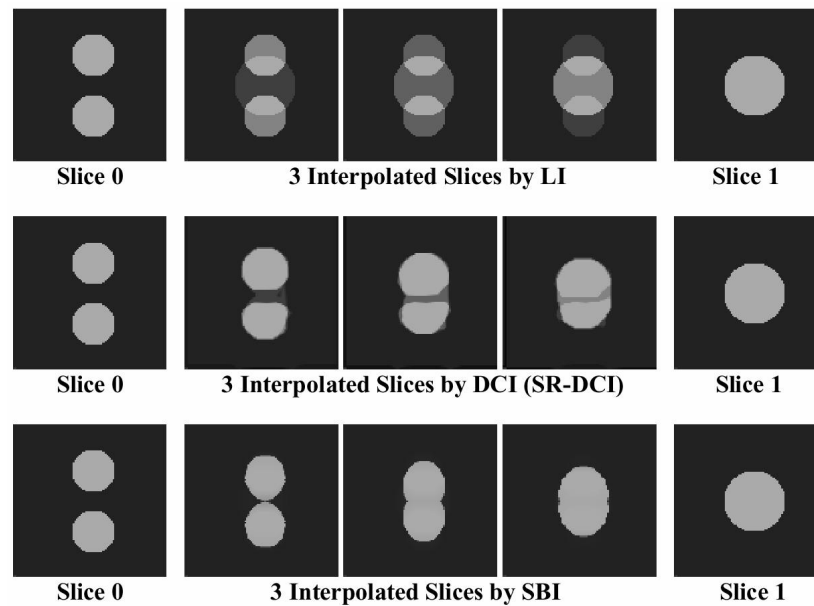


Fig. 11. Branching case for LI, DCI (SR-DCI is used here) and SBI on synthetic image pair (Slice 0 and Slice 1).

version of Slice 0). Different rows of the three middle columns correspond to the interpolated slices by the four methods. We can see that the object structure / shape is well preserved in MR-DCI by using a coarse-to-fine search strategy. Due to the large difference between the two existing slices, MCD search at single resolution is not as robust as the pyramidal search of MCD. The interpolated slices by SBI and LI are also not reasonable.

7.2. Real Images

We have applied our methods (SR-DCI and MR-DCI) to real medical volume images for evaluation. The data sets used in our experiments are MR scan of a human brain $256 \times 256 \times 167$, CT scan of a human head $256 \times 256 \times 225$ and MR scan of a human knee $256 \times 256 \times 110$. The test images are generated by dropping the slices between every 2, 4, and 8 slices for all the original images. The dropped slices from the original volume image are used as the ground truth for comparison. In this way, we have a total of nine test data sets that are sparse in the third dimension and need to be interpolated. They are: slice spacing 2, 4, and 8 for MR brain, CT head, and MR knee, respectively. We first interpolate 1, 3, and 7 slices for the corresponding generated data so that the interpolated volume images are recovered to the same size as the original ones. Then, both qualitative and quantitative validations are performed on the interpolated volume images. For all experiments, the coefficient

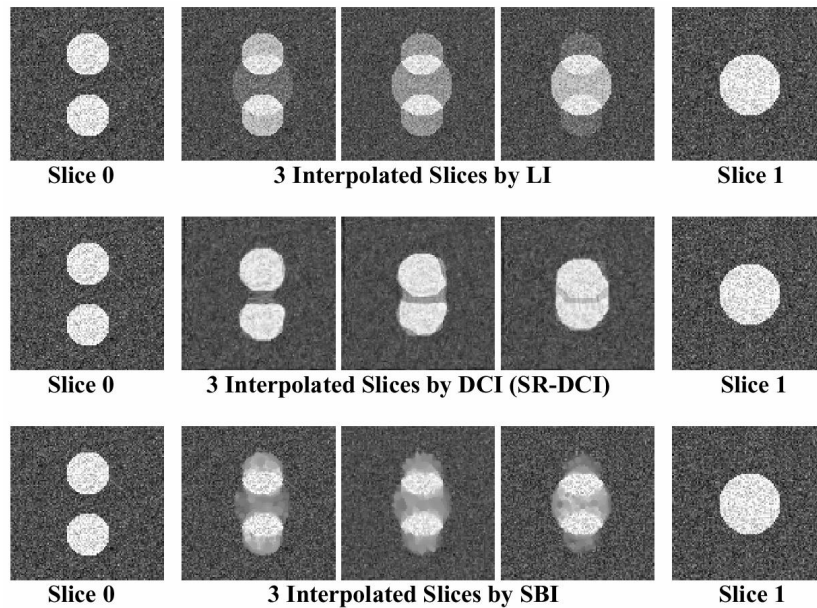


Fig. 12. Branching case for LI, DCI (SR-DCI is used here) and SBI on synthetic image pair with noise added (Slice 0 and Slice 1).

λ for the smooth term is determined empirically.

7.2.1. Qualitative Evaluation — Visual Effects

An important advantage of our DCI method is that it leads to good 3D rendering effects in visualization. Visualization of 3D biomedical volume images has been divided into two different techniques: surface rendering and volume rendering. Surface rendering techniques characteristically require the extraction of contours (edges) that define the surface of the structure to be visualized. Volume rendering techniques provide direct visualization of the volume images without the need for prior surface or object segmentation, preserving the values and context of the original image data. In our experiments, the volume rendering package *Volpack* developed by Lacroute and Levoy¹³ is employed to visualize the 3D images due to its efficiency as well as good quality in visualization.

The rendering effects with the four different methods (LI, MR-DCI, SR-DCI and SBI) are shown in Figs. 14, 15, and 16. The respective rendered images from the original true data are also available in the corresponding figures for comparison. From the results, we can see that the rendering results from MR-DCI and SR-DCI are similar (with MR-DCI a little better smoother), but they are substantially improved compared with those from LI, where there are jagged staircase artifacts.

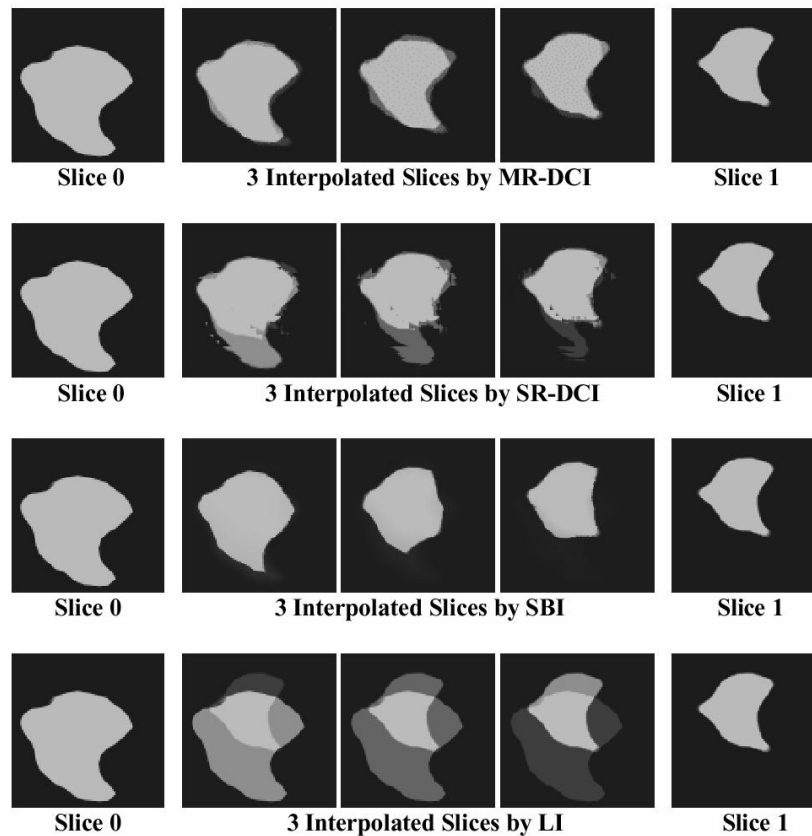


Fig. 13. Robustness of MR-DCI on synthetic image pair (Comparison of MR-DCI, SR-DCI, SBI and LI).

SBI gives overall better rendering quality than LI, but not as good as our two DCI methods (MR-DCI and SR-DCI). By LI, the more slices are dropped, the severer the artifacts are. This is because linear interpolation purely interpolates missing data along the coordinate axes without any structure or object surface information, whereas DCI interpolates the values of “new voxels” along the maximum coherence direction, where the image shape and structure surface are well preserved. When the slice spacing is either 2 or 4, all the rendered images from DCI interpolated data exhibit similar visualization effect as the ones from the original volume data. Note that even for slice spacing being 8, the interpolated CT head and MR knee with DCI still show quite good visual quality. Our interpolated MR brain is not very promising due to the original data’s low resolution along the third dimension. However, when compared with that by linear interpolation, the improvement is still dramatic. Also note that the overall visual effects for MR knee by all four methods



3D rendering from true data

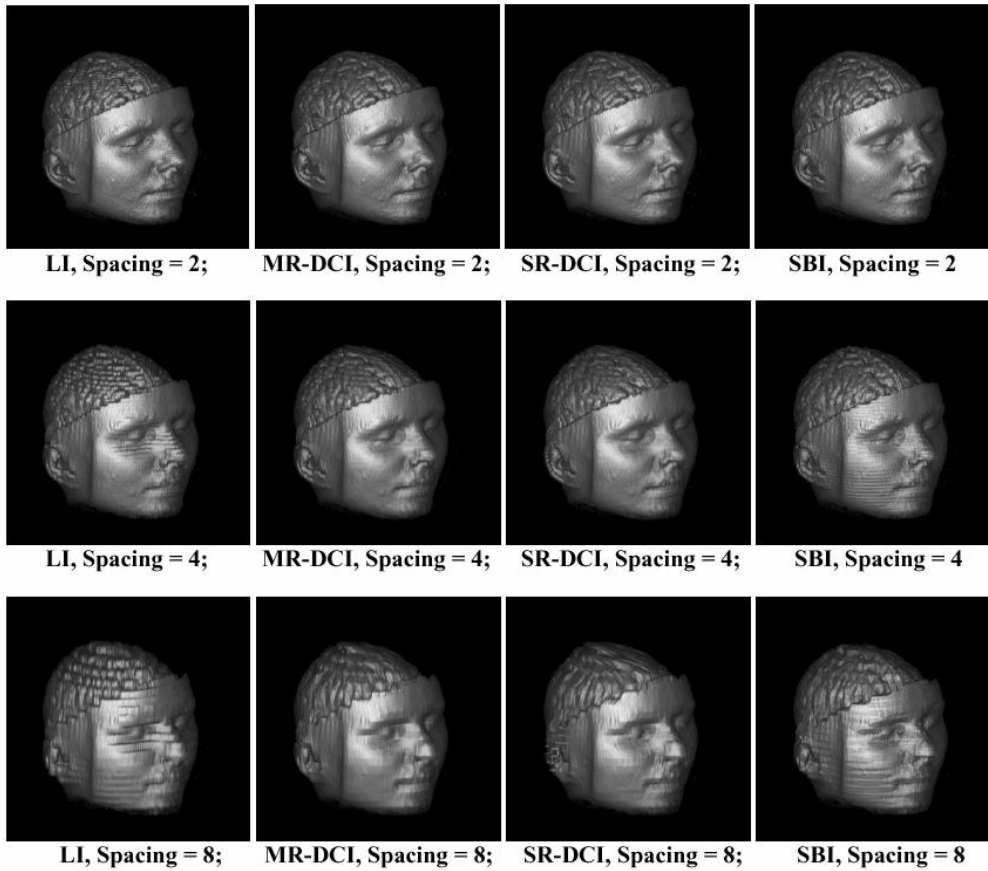


Fig. 14. Comparison of 3D volume rendering for MR brain with four different interpolation methods: LI, MR-DCI, SR-DCI and SBI.

are better than those for MR brain. This is because the structure and shape for knee is much simpler and the surface is much smoother compared with those for brain.

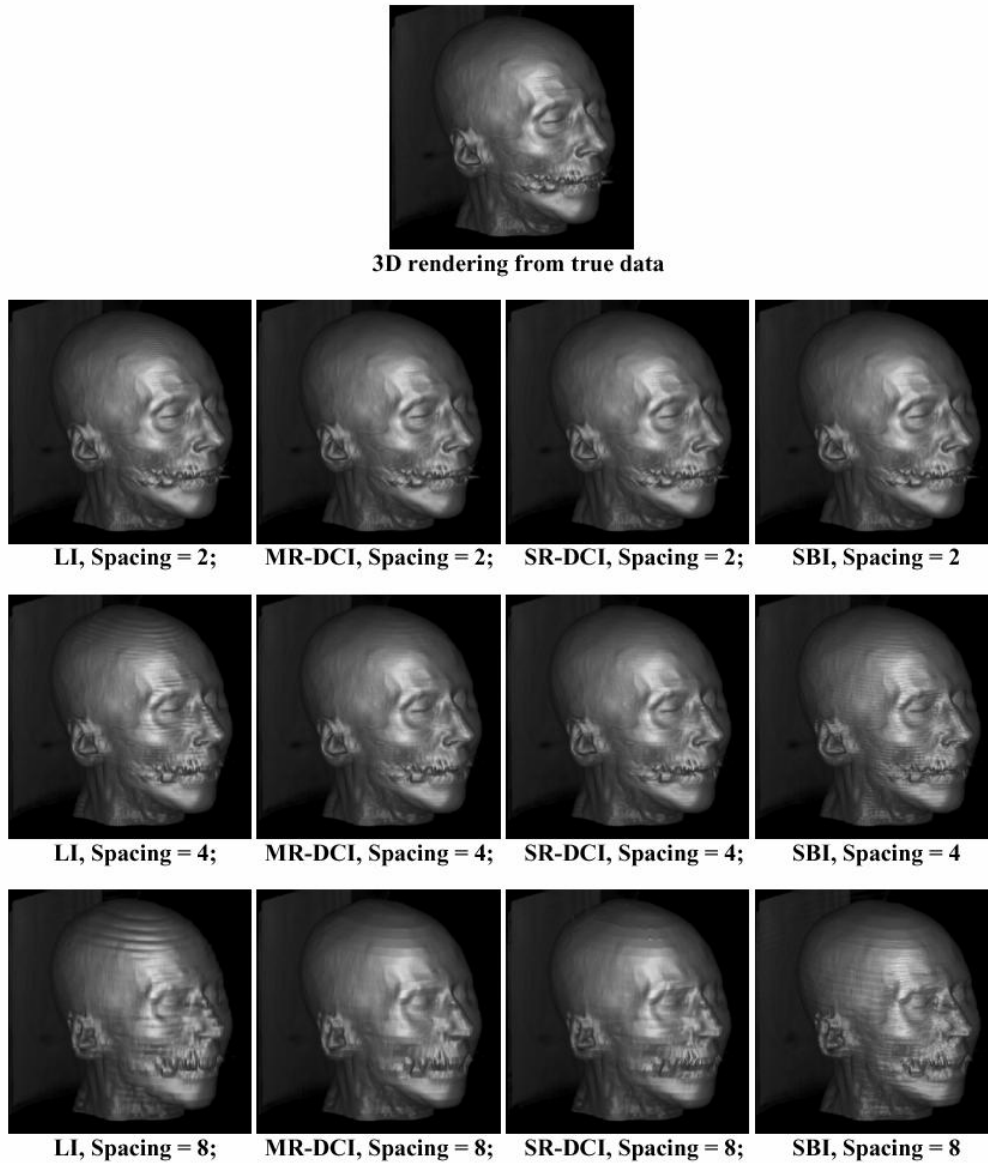


Fig. 15. Comparison of 3D volume rendering for CT head with four different interpolation methods: LI, MR-DCI, SR-DCI and SBI.

7.2.2. Quantitative Evaluation — Error Measures

The mapping from acquired data to opacity performs the essential task of surface classification in volume rendering. As pointed in ¹⁶, high opacity indicates the

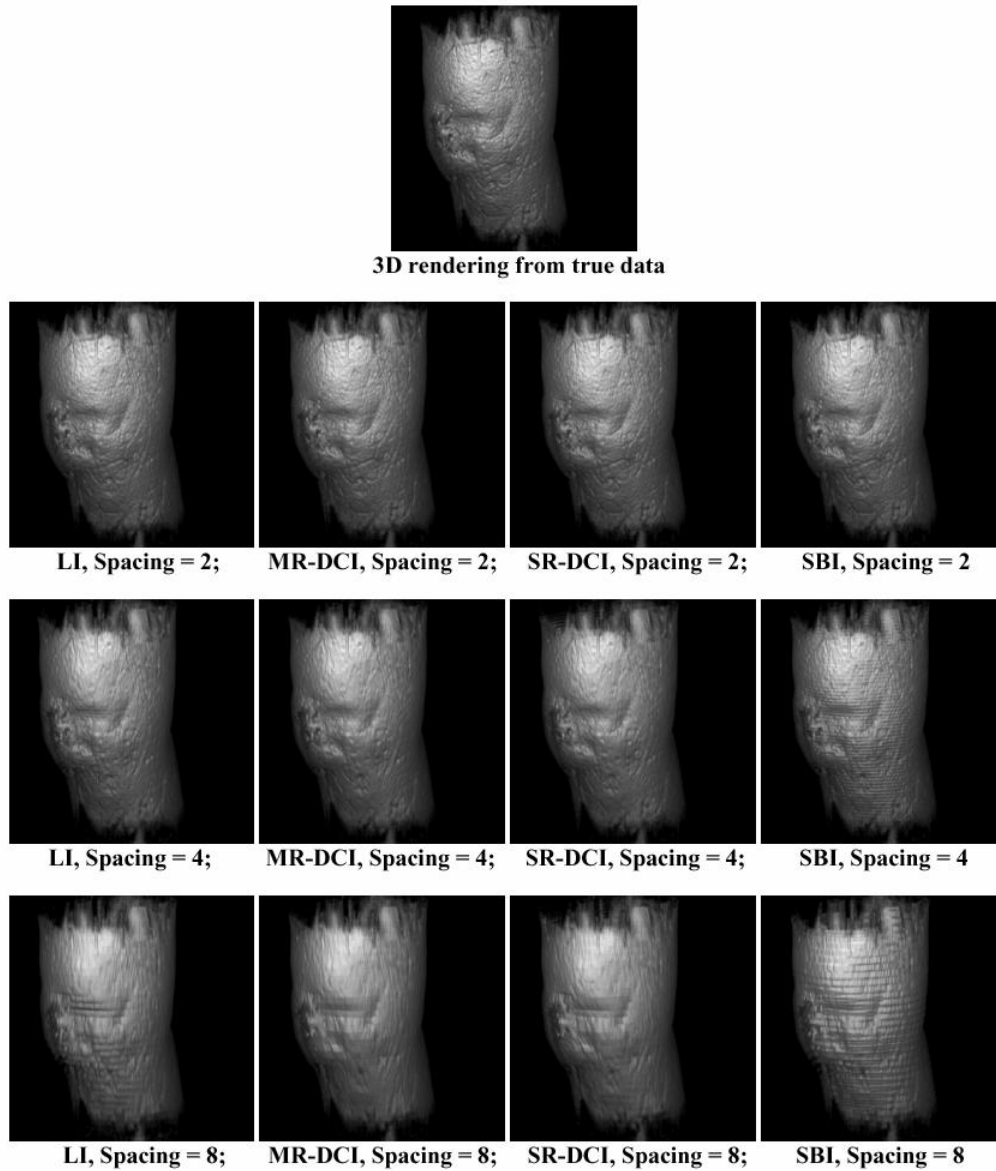


Fig. 16. Comparison of 3D volume rendering for MR knee with four different interpolation methods: LI, MR-DCI, SR-DCI and SBI.

presence of region boundary surfaces. A diagram of the opacity is shown in Fig. 17. Since we do not have the segmentation information, we incorporated opacity into our error measure criteria so that the performance of the interpolation on the

structures region boundary surfaces can be evaluated.

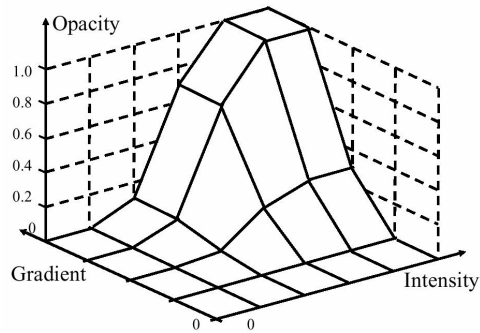


Fig. 17. Diagram of an opacity function.

To evaluate the methodology quantitatively, we use the following error measures computed between the original true data and the interpolated data:

a) Square root of mean squared difference in intensity over the whole volume, E_a . That is,

$$E_a = \sqrt{\text{Mean squared difference in Intensity}}$$

b) Square root of weighted mean squared difference in intensity over the whole volume, E_b . That is,

$$E_b = \sqrt{\text{Opacity} \cdot \text{Mean squared difference in Intensity}}$$

c) Square root of weighted mean squared difference in image gradient over the whole volume, E_c . That is,

$$E_c = \sqrt{\text{Opacity} \cdot \text{Mean squared difference in Gradient}}$$

d) Square root of weighted mean squared difference in opacity over the whole volume, E_d . That is,

$$E_d = \sqrt{\text{Opacity} \cdot \text{Mean squared difference in Opacity}}$$

Please note that at each voxel, the weighting coefficient in E_b , E_c , and E_d is the corresponding opacity calculated at this voxel. In this way, the errors on or near region boundary surfaces will contribute more to the error measures E_b , E_c , and E_d . Thus, E_b , E_c , and E_d are equivalent to surface error measures in intensity, image gradient, and opacity, respectively, whereas E_a is a volume error measure in intensity.

The mentioned error measures for the above medical data are shown in Tables 1, 2, 3, and 4, from which we can see that: The error measures from our MR-DCI are the usually smallest among the four methods (LI, MR-DCI, SR-DCI and

SBI), although the improvement in some cases is limited; SR-DCI is the second accurate algorithm among the four; LI and SBI have the similar error measures. These conclusions can also be seen obviously by averaging all the errors over the 3 different data and the 3 different slice spacing (i.e., averaging of 9 cases), as shown in Table 6 and Fig. 18.

Table 1. Comparison of error measure E_a for LI, MR-DCI, SR-DCI and SBI.

Dataset	Slice Spacing	LI	MR-DCI	SR-DCI	SBI
MR	2	2.46	2.31	2.31	2.39
Brain	4	4.22	4.01	4.09	4.19
Data	8	6.43	6.12	6.41	6.24
CT	2	0.72	0.83	0.92	1.08
Head	4	2.69	2.22	2.38	2.86
Data	8	6.33	5.23	5.28	5.98
MR	2	1.30	1.31	1.33	1.46
Knee	4	2.76	2.62	2.68	2.93
Data	8	4.89	4.43	4.60	4.84

Table 2. Comparison of error measure E_b for LI, MR-DCI, SR-DCI and SBI.

Dataset	Slice Spacing	LI	MR-DCI	SR-DCI	SBI
MR	2	2.17	2.03	2.04	2.14
Brain	4	3.81	3.64	3.71	3.87
Data	8	5.68	5.48	5.70	5.73
CT	2	0.64	0.69	0.78	1.03
Head	4	2.56	2.08	2.22	2.76
Data	8	5.89	4.93	4.93	5.70
MR	2	1.12	1.13	1.16	1.30
Knee	4	2.53	2.40	2.46	2.74
Data	8	4.56	4.15	4.28	4.57

7.2.3. Computation Cost

The four methods approximate execution times on a PC (Pentium III 866Mhz 256MB RAM) are listed in Table 5. The averaging times (over the 9 cases: 3 different data and 3 different slice spacing) are respectively: LI 0.041 *minutes*; MR-DCI 1.615 *minutes*; SR-DCI: 6.300 *minutes* SBI 134.6 *minutes*, as shown in the last row of Table 6. We can see although our DCI methods (MR-DCI and SR-DCI) take longer time than the simple linear interpolation, but they are much faster than the

Table 3. Comparison of error measure E_c for LI, MR-DCI, SR-DCI and SBI.

Dataset	Slice Spacing	LI	MR-DCI	SR-DCI	SBI
MR	2	1.37	1.29	1.27	1.37
Brain	4	2.52	2.39	2.38	2.54
Data	8	3.40	3.21	3.30	3.45
CT	2	0.43	0.49	0.55	0.70
Head	4	1.61	1.30	1.39	1.74
Data	8	2.98	2.40	2.46	2.88
MR	2	0.76	0.78	0.79	0.90
Knee	4	1.68	1.59	1.64	1.80
Data	8	2.72	2.44	2.59	2.72

Table 4. Comparison of error measure E_d for LI, MR-DCI, SR-DCI and SBI.

Dataset	Slice Spacing	LI	MR-DCI	SR-DCI	SBI
MR	2	0.139	0.138	0.138	0.143
Brain	4	0.193	0.192	0.194	0.204
Data	8	0.232	0.232	0.235	0.244
CT	2	0.066	0.065	0.069	0.076
Head	4	0.122	0.108	0.113	0.128
Data	8	0.164	0.151	0.154	0.170
MR	2	0.112	0.113	0.114	0.122
Knee	4	0.167	0.167	0.169	0.176
Data	8	0.207	0.205	0.207	0.215

Table 5. Comparison of computation time (in *minutes*) for LI, MR-DCI, SR-DCI and SBI.

Dataset	Slice Spacing	LI	MR-DCI	SR-DCI	SBI
MR	2	0.041	1.795	9.630	168.9
Brain	4	0.040	1.432	4.762	99.09
Data	8	0.040	1.374	3.766	64.47
CT	2	0.057	2.804	14.96	333.0
Head	4	0.055	2.220	7.463	186.8
Data	8	0.054	2.237	5.874	113.4
MR	2	0.029	1.042	5.272	126.2
Knee	4	0.028	0.829	2.724	73.24
Data	8	0.027	0.800	2.226	46.46

shape-based scheme applied on gray-level images. Also, our MR-DCI is faster than SR-DCI due to the use of the multiresolution MCD search strategy described in Section 4.

Table 6. Average (over the 3 different dataset and the 3 different slice spacing) of all measurements.

Measure	LI	MR-DCI	SR-DCI	SBI
E_a	3.53	3.23	3.33	3.55
E_b	3.22	2.95	3.03	3.32
E_c	1.94	1.77	1.82	2.01
E_d	0.16	0.15	0.16	0.16
Computation Time (<i>minutes</i>)	0.041	1.615	6.300	134.6

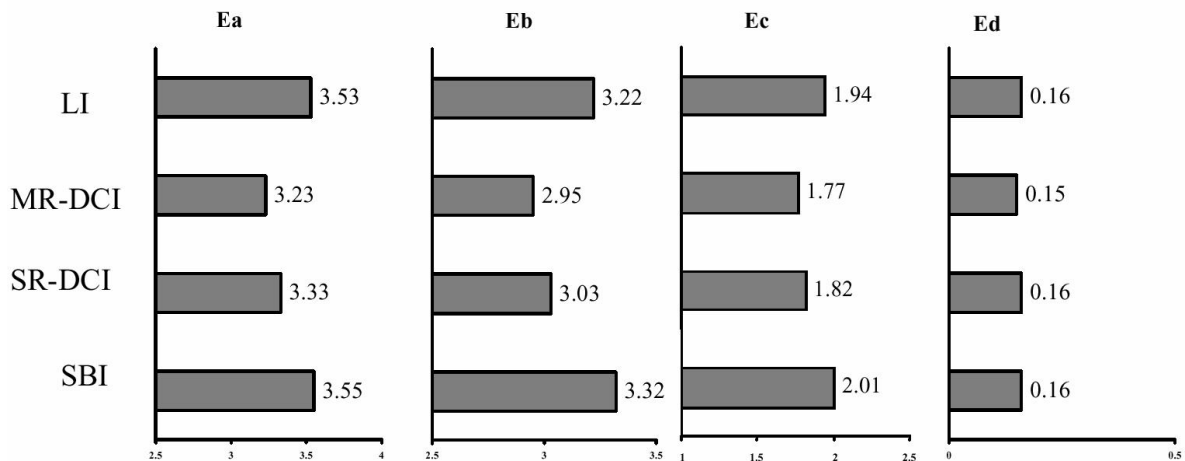


Fig. 18. Average error bar of the four measurements for four different methods: LI, MR-DCI, SR-DCI, and SBI.

7.2.4. Sensitivity to Noise Experiments

We have also performed some sensitivity to noise experiments for the CT head with slice spacing 4. The 3D rendering results with increasing noise level are shown Fig. 19. Generally, it might be difficult to tell which method is more robust purely from these renderings, although the corresponding rendering results from DCI (Note: SR-DCI is used here; In fact, MR-DCI has similar or even better performance than SR-DCI) seem a little better than the ones from LI and SBI. But from the error lines in Fig. 20 (E_a , E_b , E_c , and E_d with increasing noise level), we can see that our DCI is more robust to noise. The SBI method is shown to be more sensitive to noise. This is because in SBI method, the interpolation is based on distance interpolation. The distance transform affected by noise at certain points can be propagated to a lot other points. Therefore, SBI is quite easily disrupted by noise.

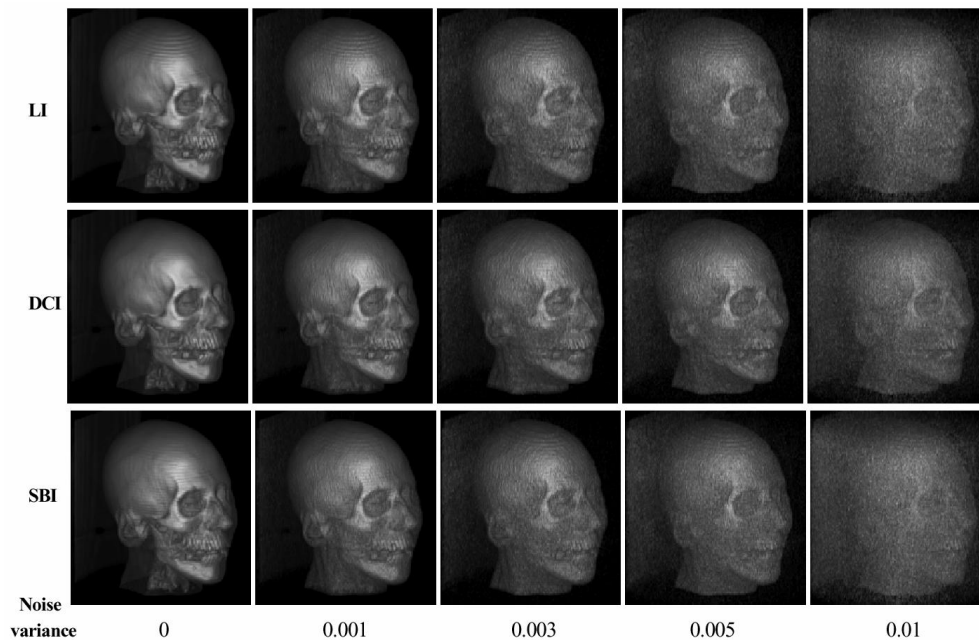


Fig. 19. Rendering results with increasing noise level for CT head (slice spacing = 4) by using different methods: LI, DCI (SR-DCI is used here) and SBI. (Note: The noise added is White Gaussian noise with zero mean.)

8. Conclusions and Future Directions

Novel directional coherence based interpolation methods are presented in this paper. The theoretical basis, the schemes to estimate the directional coherence map, and the technique to interpolate the image along the point-wise smooth MCDs are described. Pyramidal MCD search is also proposed to further enhance the method. A number of experiments were performed on both synthetic and real medical images to evaluate our Directional Coherence Interpolation (DCI). It is shown that DCI preserves the shape structure effectively for visualization and analysis without prior requirement of segmentation. Comparing with the traditional linear interpolation, DCI improves the interpolating quality substantially with modest computation cost. Comparing with the shape-based interpolation, DCI has significantly lower computation complexity yet leads to better interpolation results. In addition, Multiresolution DCI (MR-DCI) is shown to be able to further decrease computation time and augment interpolation accuracy.

As to the future study, we are interested in applying DCI to the dynamic objects. We want to interpolate in the time dimension as well, since, very often, the sampled time instances are either not sufficient to depict a smooth motion of the object or the sampling itself may be non-uniform. We will utilize the spatio-temporal

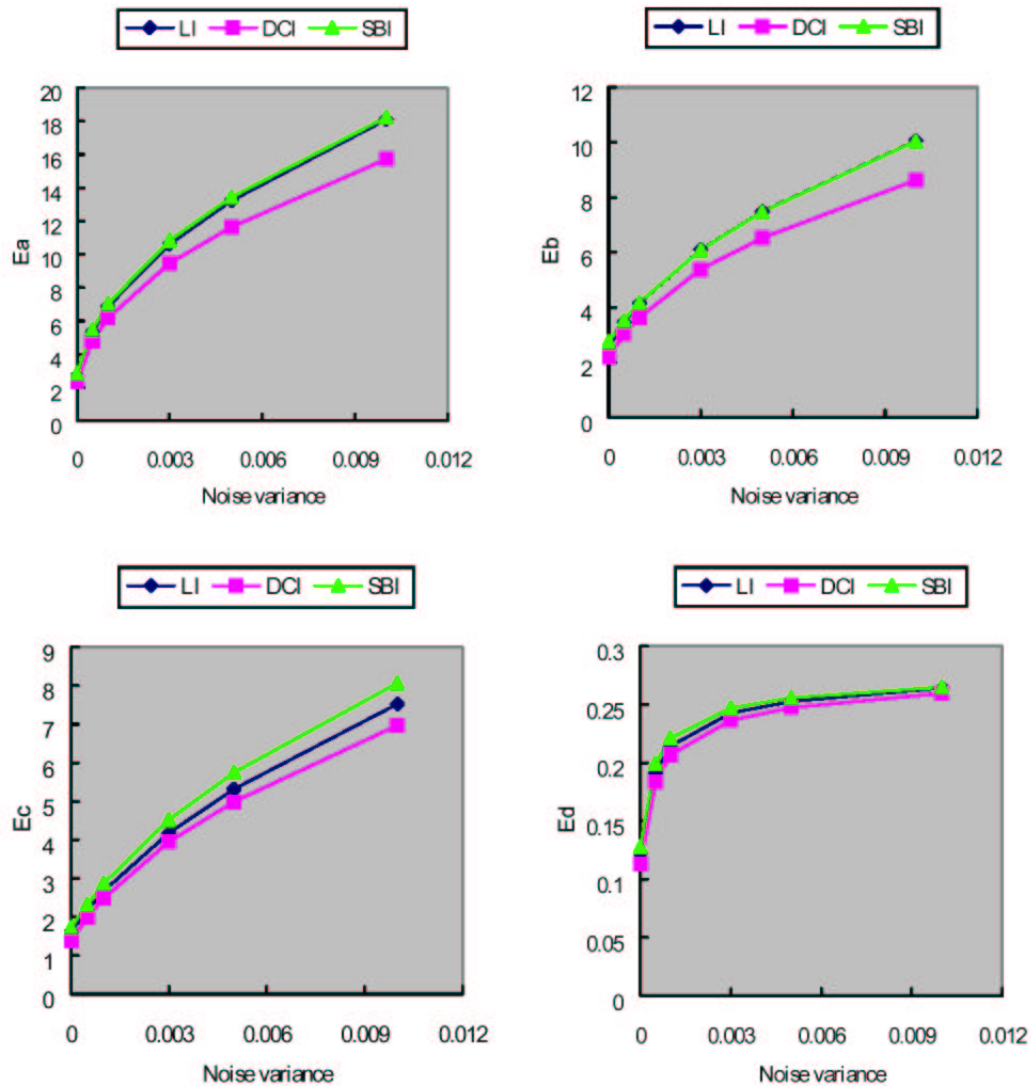


Fig. 20. Error measures with increasing noise level for CT head (slice spacing = 4) by using different methods: LI, DCI (SR-DCI is used here) and SBI.

directional coherence to solve this dynamic interpolation problem. Since good visualization is the major advantage of DCI, we also would like work on combining volume rendering with interpolation to develop a directional coherence based 3D volumetric segmentation and visualization algorithm. Initial visualization, made without the benefit of object surface, would be used to guide scene analysis and segmentation, which would then be utilized to isolate regions of interest, producing

a better visualization.

Acknowledgements

This work was carried out while the first author was an Assistant Professor in the Department of Information Engineering at the Chinese University of Hong Kong, supported in part by the Research Grants Council of the Hong Kong SAR, under RGC Earmarked Grant (Project No. CUHK 4195/01E), and under Direct Grant for Research (Project Code: 2050258). The second and the third authors participated in this work when they were working as interns at Microsoft Research Asia.

References

1. W. A. Barrett and R. R. Stringham, "Shape-based interpolation of grayscale serial slice images", in *SPIE Proc.* (vol. 1898, 1993), pp. 105-115.
2. P. Brigger, F. Muller, K. Illgner and M. Unser, "Centered Pyramids", *IEEE Trans. on Image Processing* **8**(9), 1254-1264 (1999).
3. P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics", *ACM Trans. Graphics* **2**(4), 217-236 (1983).
4. T. F. Cootes, C. J. Taylor and A. Lanitis, "Active shape models: Evaluation of a multi-resolution method for improving image search", in *Proc. British Machine Vision Conference* (1994), pp. 327-336.
5. J. C. Gee, D. R. Haynor, M. Reivich and R. Bajcsy, "Finite element approach to warping of brain images", in *SPIE Medical Imaging 1994: Image Processing* (M. H. Loew, ed., 2167, Washington, 1994).
6. A. Goshtasby, D. A. Turner and L. V. Ackerman, "Matching of tomographic slices for interpolation", *IEEE Trans. on Medical Imaging* **11**(4), 507-516 (1992).
7. G. J. Grevera and J. K. Udupa, "Shape-based interpolation of multidimensional gray-level images", *IEEE Trans. on Medical Imaging* **15**(6), 881-892 (1996).
8. B. Guo, "Progressive radiance evaluation using directional coherence maps", in *Proc. Siggraph 98* (Orlando, FL, July 1998), pp. 255-266.
9. D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis", in *Proc. SIGGRAPH'95* (1995), pp. 229-238.
10. G. T. Herman, J. Zheng and C. A. Bucholtz, "Shape-based interpolation", *IEEE Computer Graphics and Applications* **12**(3), 69-79 (1992).
11. W. E. Higgins, C. Morice and E. L. Ritman, "Shape-based interpolation of tree-like structures in three-dimensional images", *IEEE Trans. on Medical Imaging* **12**(3), 439-450 (1993).
12. M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models", *Intl. J. Computer Vision* **1**(4), 312-331 (1988).
13. P. Lacroute and M. Levoy, "Fast volume rendering using a shear-warp factorization of the viewing transformation", in *Proc. SIGGRAPH'94* (Orlando, Florida, July 1994), pp. 451-458.
14. T. Y. Lee and W. H. Wang, "Morphology-based three-dimensional interpolation", *IEEE Trans. on Medical Imaging* **19**(7), 711-721 (2000).
15. T. M. Lehmann, C. Gonner and K. Spitzer, "Survey: Interpolation methods in medical image processing", *IEEE Trans. on Medical Imaging* **18**(11), 1049-1075 (1999).
16. M. Levoy, "Display of surfaces from volume data", *IEEE Computer Graphics and Applications* **8**(3), 29-37 (1988).

17. C. C. Liang, C. T. Chen and W. C. Lin, "Intensity interpolation for reconstructing 3-D medical images from serial cross-sections", in *Proc. IEEE Eng. Med. Bio. Soc. 10th Int. Conf.* (New Orleans, LA, CH2566-8, Nov. 1988), pp. 1389-1390.
18. S. G. Mallat, "A theory of multiresolution singla decomposition: the wavelet representation", *IEEE Trans. on Pattern Analysis and Machine Intelligence* **11**, 674-693 (1989).
19. S. Mallat and S. Zhong, "Characterization of signals from multiscale edges", *IEEE Trans. on Pattern Analysis and Machine Intelligence* **14(7)**, 710-732 (1992).
20. J. C. Mceachen and J. S. Duncan, "Shape-based tracking of left ventricular wall motion", *IEEE Trans. on Medical Imaging* **16(3)**, 270-283, 1997.
21. E. H. W. Meijering, W. J. Niessen, and M. A. Viergever, "Piecewise polynomial kernels for image interpolation: A generalization of cubic convolution", in *Proc. International Conf. on Image Processing* (vol. 3, 1999), pp. 647-651.
22. D. T. Puff, D. Eberly and S. M. Pizer, "Object-based interpolation via cores", in *Proc. SPIE: Medical Imaging'94, Image Processing* (vol. 2167, 1994), pp. 104-115.
23. S. P. Raya and J. K. Udupa, "Shape-based interpolation of multidimensional objects", *IEEE Trans. on Medical Imaging* **9(1)**, 32-42 (1990).
24. R. A. Robb, *Biomedical Imaging, Visualization, and Analysis* (Wiley-Liss, Inc., 2000).
25. A. Rosenfeld, *Multiresolution Image Processing* (Berlin, Germany: Springer-Verlag, 1984).
26. D. Shen, E. H. Herskovits, and C. Davatzikos, "An adaptive focus statistical shape model for segmentation and shape modeling of 3-D brain structures", *IEEE Trans. on Medical Imaging* **20(4)**, 257-270 (2001).
27. L. Sutherland, R. Sproull and R. Schumacker, "A characterization of ten hidden-surface algorithms", *ACM Computing Surveys* **6(1)**, 387-442 (1974).
28. P. Thevenaz, M. Unser, "Optimization of mutual information for multiresolution image registration", *IEEE Trans. on Image Processing* **9(12)**, 2083-2099 (2000).
29. B. C. Vemuri, A. Radisavljevic and C. M. Leonard, "Multi-resolution 3D stochastic shape models for image segmentation", in *13th International Conf. on Information Processing in Medical Imaging* (June 1993), pp. 62-76.
30. Y. Wang, B. S. Peterson, and L. H. Staib, "Shape-based 3D surface correspondence using geodesics and local geometry", in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (vol. II, June 2000), pp. 644-651.
31. Y. Wang and L. H. Staib, "Boundary finding with prior shape and smoothness models", *IEEE Trans. on Pattern Analysis and Machine Intelligence* **22(7)**, 738-743 (2000).
32. Y. Wang and L. H. Staib, "Physical model based non-rigid registration incorporating statistical shape information", *Medical Image Analysis* **4(1)**, 7-20 (2000).
33. Y. Wang, Z. Zhang and B. Guo, "3D image interpolation based on directional coherence", in *proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis* (Hawaii, December 2001), pp. 195-202.
34. W. L. Williams, "Optical flow interpolation of serial slice images", in *Proc. SPIE Medical Imaging'93, Image Processing* (vol. 1898, 1993), pp.93-104.
35. J. Zhang, Y. Wang and B. Guo, "Pyramidal search of maximum coherence direction for biomedical image interpolation", in *International Symposium on Biomedical Imaging* (Washington D.C., July 2002), pp. 887-890.

Photo and Bibliography

Yongmei Michelle Wang received her M.S., M.Phil., and Ph.D. degrees in Electrical Engineering from Yale University in 1994, 1996, and 1999, respectively. From June 1999 to July 2000, she was a postdoctoral fellow in the Image Processing and Analysis group and the Child Study Center at Yale University. After that, she visited Microsoft Research Asia for several months. In January 2001, she joined the faculty of the Department of Information Engineering at the Chinese University of Hong Kong, where she worked as an Assistant Professor till July 2002. Dr. Wang is currently affiliated with the Department of Diagnostic Radiology at Yale University. Her research interests include structural and functional medical image analysis, biosignal processing, computer vision and pattern recognition.

Jingdan Zhang received his B.E. and M.S. degree in Computer Science from Tsinghua University, China, in 2000 and 2003, respectively. He is currently an intern at Microsoft Research Asia. His main interests are computer graphics and related applications.

Zhunping Zhang received his B.E. degree in Computer Science from Tsinghua University, China, in 2002. From April 2001 to June 2002 he was an intern at Microsoft Research Asia. He is currently an assistant researcher at Microsoft Research Asia.

Baining Guo is a senior researcher and the research manager of the Internet Graphics group at Microsoft Research Asia (formerly Microsoft Research China). Before joining Microsoft, Baining was a senior staff researcher in the Microcomputer Research Labs at Intel Corporation in Santa Clara, California. He was a key contributor to Intel's next-generation graphics architecture. He was also managing Intel's graphics research and development in Russia. Before moving to the Silicon Valley, Baining worked at University of Colorado, University of Toronto, and York University. He was also a visiting professor at Ecole Nationale Supérieure Des Telecommunications and Princeton University. Baining has published a few dozen of refereed papers, including a number of papers in Siggraph. He holds a dozen of granted and pending patents. In his Ph.D. thesis, Baining developed mechanical CAD systems based on implicit surfaces. He was a pioneer in that area and received NATO fellowship from the National Science Foundation. Baining received his Ph.D. and M.S. from Cornell University and his B.S. from Beijing University. His main research interests are in modeling and rendering. His current projects include networked games, appearance modeling, natural phenomena, and texture synthesis.