

3D Image Interpolation Based on Directional Coherence

Yongmei Wang

Dept. of Information Engineering
The Chinese University of Hong Kong
Shatin, N. T., Hong Kong
ymwang@ie.cuhk.edu.hk

Zhunping Zhang*

Computer Science & Technology
Tsinghua University
Beijing 100084, P. R. China
justinzhang@acm.org

Baining Guo

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

Abstract

Image interpolation is of great importance in biomedical visualization and analysis. In this paper, we present a novel gray-level interpolation method called Directional Coherence Interpolation (DCI). 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. Since the edges of the image and the contents of the objects are well preserved along the MCDs, DCI can incorporate image shape and structure information without the prior requirement of explicit representation of object boundary / surface.

A number of experiments were performed on both synthetic and real medical images to evaluate the proposed approach. The experimental results show that in addition to the substantial improvement of visual effects (qualitative evaluation), the quantitative error measures of DCI are also better than the conventional gray level linear interpolation. Comparing with the shape-based interpolation scheme applied on gray-level images, DCI has much lower computation cost.

1. Introduction

Image interpolation is widely used in computer vision, especially in biomedical image processing, visualization, and analysis [9, 3]. Most 3D biomedical 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 visualization, as the 3D data must be isotropic in order to produce the

correct aspect ratio along each direction when displayed. 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.

The conventional interpolation scheme for 3D volume image is a first-order linear interpolation. In this scheme, unknown values of the “new voxels” between known voxels are linearly interpolated. When the distance between consecutive slices is more than five times the in-plane pixel size, linear interpolation will result in poor approximations and a jagged staircase artifact is produced, which is clearly visible in renderings of these volumes [15].

There are several ways to overcome the shortcoming of the traditional interpolation, including object-based methods, slice matching or correspondence [1, 18], cores [13], and morphology-based algorithms [8]. Among these methods, perhaps the most popular algorithms are object-based methods, where object information extracted from a given scene is used in guiding the interpolation process. Shape-based interpolation [14, 5, 2] is an example of object-based methods. There are basically two types of shape-based interpolation: the algorithm for binary images and the one for gray-level images. 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 structure [14]. Prior requirement of specific structure identification

* Zhunping Zhang participated in this work when he was working as an intern at Microsoft Research Asia.

(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 in [2]. Two complementary stages called lifting and clasp 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). It interpolates missing image data along the smoothed maximal coherence directions (MCD) between image slices instead of the coordinate axes as that in linear interpolation. By this scheme, the object 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.

The remainder of the paper is organized as follows. In section 2, we first give an overview of the proposed method. Then, the theoretical basis and the detailed techniques of our approach is described. Experiments and results are presented in Section 3, followed in Section 4 by conclusions and suggestions for future work.

2. The Proposed Method

2.1 System Overview

The system framework of our proposed approach is shown in Figure 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 gray-level values of “new voxels” between the slice pair are then interpolated along the smoothed 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.

2.2 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 similarities [16]. 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 edge [11]. This idea has been successfully applied in 2D for contour-based image coding [11], and recently in computer graphics for progressive radiance evaluation [4]. In this work, we introduce and utilize the 3D form of this type of coherence, called directional coherence, into our interpolation problem.

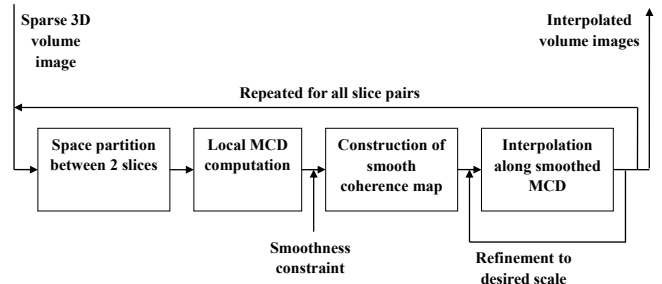


Figure 1: System diagram of our Directional Coherence Interpolation (DCI).

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 Figure 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 Figure 3(a) rather than as the one in Figure 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. We 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.

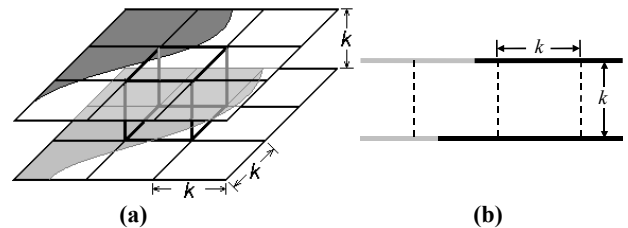


Figure 2: Diagram of our volume space partition scheme between two slices. (a) 3D case; (b) 2D profile with the top and bottom lines representing the two image slices that need to be interpolated. (Note: the dark area in (a) and dark line segments in (b) denote the existence of object / structure.)

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.

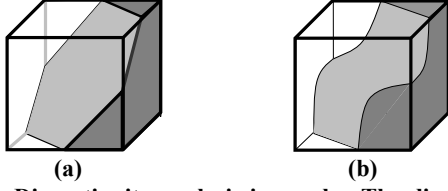


Figure 3: Discontinuity analysis in a cube. The discontinuity in (a) is considered simple whereas that in (b) is not, because of the curvedness.

2.3 Local Maximal 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 Figure 4. The local MCD $\bar{n}(C_k)$ of a $k \times k \times k$ cube can be calculated by minimizing the surface integral

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

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

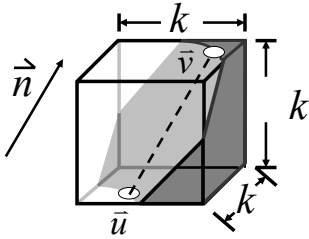


Figure 4: Diagram of Maximal Coherence Direction (MCD) for a small cube C_k .

However, for our interpolation problem here, although the intensity function $f(\bar{u})$ is known on the image slices (shown as the top and bottom facets in Figure 4), we do not have $f(\bar{u})$ 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 \bar{n} as the unknown value $f(\bar{u})$. 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 Figure 5. The corresponding within-slice-points on line segments EI and GL are outside the elementary block $EFGH$. But for the given directions \bar{n} ,

their intensity values will be used during integration due to the unknown $f(\bar{u})$ values on line segments EH and FG , respectively. In the case of 3D cube, as shown in Figure 6, for the given direction \bar{n} , the intensity values on the outside-cube-facets S_1' and S_2' will be used as the $f(\bar{u})$ values for points on the front facet S_1 and the right-side-facet S_2 , respectively. By this mapping method, we then get all the intensity function $f(\bar{u})$ 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 $d(\bar{n}_i)$ in 3D is calculated as

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

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 Figure 5 for 2D) does not exceed certain limit, for example, the cube side length k .

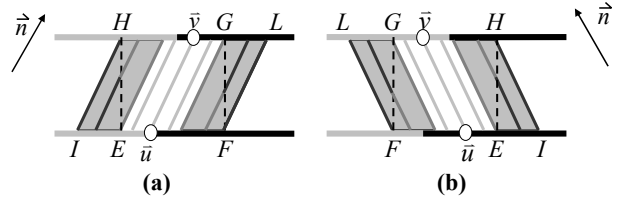


Figure 5: Diagram for 2D block MCD search. (a) and (b) are for different given orientations \bar{n} .

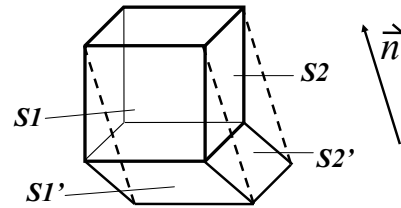


Figure 6: Diagram for image intensity mapping in 3D.

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 Figure 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.

2.4 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

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

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

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

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 [6, 17] and cardiac motion estimation [12].

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.

2.5 Interpolation with Refinement

Given the smoothed MCD for each cube, the value of a point within this cube is interpolated linearly from the values of points with known values along this MCD.

Refinement scheme is applied to interpolate the images to the desired scale. 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 Section 2 is repeated for all the slice pairs.

3. Experimental Results

For all the experiments, we have applied the traditional linear interpolation (LI) as a direct comparison. 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 to the computation expense, we have compared DCI with the simple linear interpolation, as well as the state-of-the-art shape-based interpolation (SBI).

3.1 Synthetic Images

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

The experiment in Figure 7 shows the 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 well preserved structure information.

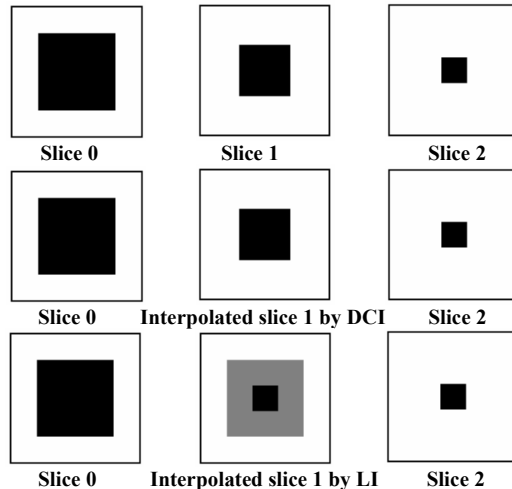


Figure 7: Comparison for DCI and linear interpolation (LI) for the synthetic image. Top row: three consecutive slices taken from a 3D synthetic image containing a frustum; Middle row: corresponding slices with the middle one interpolated by DCI; Bottom row: corresponding slices with the middle one interpolated by LI.

Figure 8 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 object’s cross sections do not overlap on the slices that need to be interpolated (Slice 0 and Slice 2 in Figure 8), the local

MCDs over the non-overlap areas are not able to reasonably reflect the 3D object’s 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 image or structure information is included.

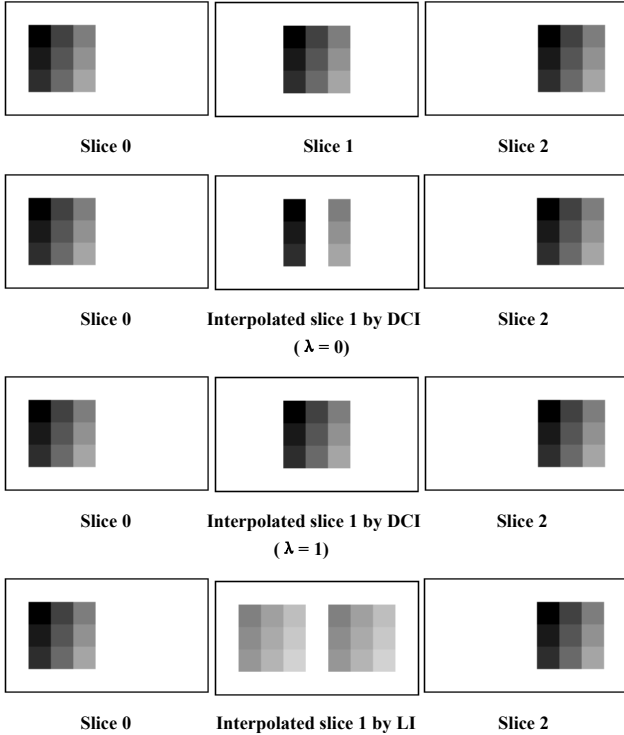


Figure 8: Effects of smoothness constraint on the synthetic image. Top row: three consecutive slices taken from a 3D synthetic image; 2nd row: corresponding slices with the middle one interpolated by DCI with $\lambda = 0$ (no smoothness constraint); 3rd row: corresponding slices with the middle one interpolated by DCI with $\lambda = 1$; Bottom row: corresponding slices with the middle one interpolated by LI.

3.2 Real Images

We have applied our method 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 λ for the smooth term is determined from experiments.

3.2.1 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 Levoy [7] is employed to visualize the 3D images due to its efficiency as well as good quality in visualization.

The rendering effects with our method and with linear interpolation are shown in Figures 10, 11, and 12. 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 rendering results from linear interpolation have jagged staircase artifacts. The more slices are dropped, the severer the artifacts are ((d), (e), and (f) in Figures 10, 11, and 12). 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. Therefore, 3D rendering results with DCI are substantially improved. When the slice spacing is either 2 or 4, all the rendered images from DCI interpolated data ((a) and (b) in Figures 10, 11, and 12) exhibit similar visualization effect as the ones from the original volume data ((g) in Figures 10, 11, and 12). Note that even for slice spacing being 8, the interpolated CT head (Figure 11(c)) and MR knee (Figure 12(c)) with DCI still show quite good visual quality. Our interpolated MR brain (Figure 10(c)) is not very promising due to the original data’s low resolution along the third dimension. However, when compared with that with linear interpolation (Figure 10(f)), the improvement is still dramatic.

3.2.2 Quantitative Evaluation --- Error Measures

The mapping from acquired data to opacity performs the essential task of surface classification in volume rendering. As pointed in [10], high opacity indicates the presence of region boundary surfaces. A diagram of the opacity is shown in Figure 9. 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.

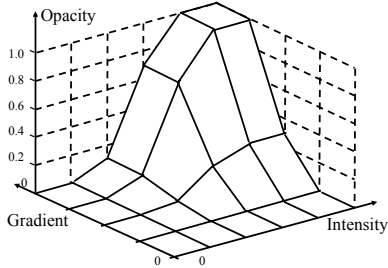


Figure 9: Diagram of an opacity function.

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

- Square root of mean squared difference in intensity over the whole volume, E_a .
- Square root of weighted mean squared difference in intensity over the whole volume, E_b .
- Square root of weighted mean squared difference in image gradient over the whole volume, E_c .
- Square root of weighted mean squared difference in opacity over the whole volume, E_d .

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 all the error measures from our DCI are better than those from LI method, although the improvement is limited. Please note that the smaller error values over the two methods are highlighted in these tables.

Data	Slice Spacing	DCI	LI
MR Brain	2	2.27	2.48
	4	4.08	4.62
	8	6.78	7.17
CT Head	2	1.56	1.60
	4	4.33	4.50
	8	7.58	8.19
MR Knee	2	1.91	2.07
	4	4.67	4.83
	8	8.02	8.76

Table 1: Comparison of error measure E_a for our DCI and linear interpolation (LI).

Data	Slice Spacing	DCI	LI
MR Brain	2	1.98	2.18
	4	3.69	4.17
	8	6.20	6.40
CT Head	2	1.46	1.49
	4	4.10	4.18
	8	6.97	7.40
MR Knee	4	1.76	1.91
	4	4.43	4.50
	8	7.51	8.01

Table 2: Comparison of error measure E_b for our DCI and linear interpolation (LI).

Data	Slice Spacing	DCI	LI
MR Brain	2	1.26	1.43
	4	2.39	2.75
	8	3.40	3.67
CT Head	2	1.03	1.15
	4	2.43	2.68
	8	3.49	3.91
MR Knee	2	1.17	1.31
	4	2.66	2.86
	8	3.80	4.15

Table 3: Comparison of error measure E_c for our DCI and linear interpolation (LI).

Data	Slice Spacing	DCI	LI
MR Brain	2	0.135	0.136
	4	0.191	0.192
	8	0.230	0.233
CT Head	2	0.093	0.108
	4	0.101	0.178
	8	0.200	0.215
MR Knee	2	0.133	0.136
	4	0.198	0.201
	8	0.238	0.248

Table 4: Comparison of error measure E_d for our DCI and linear interpolation (LI).

3.2.3 Computation Cost

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 images [2]. We have applied the

three methods, LI, DCI, and SBI based on our own implementation, on the CT head data with slice spacing 4. The three methods' approximate execution times on a PC (Pentium III 866Mhz 256MB RAM) are respectively: LI 1.76 *seconds* (0.03 *minutes*); DCI 171.14 *seconds* (2.85 *minutes*); SBI 10846.57 *seconds* (180.78 *minutes*). We can see although DCI takes longer time than the simple linear interpolation, but it is much faster than the shape-based scheme applied on gray-level images [2]. Moreover, both the qualitative and quantitative evaluations of SBI for the mentioned CT head data are better than those of linear interpolation, but not as good as those of our DCI.

4. Conclusions and Future Directions

A novel directional coherence based interpolation method is presented in this paper. The theoretical basis, the schemes to estimate the directional coherence map, and the technique to interpolate the image along the smoothed MCDs are described. A number of experiments were performed on both synthetic and real medical images to evaluate our Directional Coherence Interpolation. It is shown that DCI preserves the shape structure effectively for visualization. Comparing with the traditional linear interpolation, DCI improves the interpolating quality substantially with modest computation cost. Comparing with the shape-based interpolation, our preliminary results show that the proposed method has significantly lower computation complexity.

As to the future study, we will perform further validation for DCI, including sensitivity to noise tests, more qualitative and quantitative comparisons with the shape-based method on both synthetic and real images. We are also 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 directional coherence to solve this dynamic interpolation problem. Finally, since good visualization is the major advantage of DCI, we will also 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 used to isolate regions of interest, producing a better visualization. Therefore, DCI has many potential applications.

Acknowledgements

We are grateful to the anonymous reviewers for their constructive critique and detailed comments. This work is 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).

References

- [1] A. Goshtasby, D. A. Turner and L. V. Ackerman, "Matching of tomographic slices for interpolation", *IEEE Trans. on Medical Imaging*, vol. 11, no. 4, pp. 507-516, 1992.
- [2] G. J. Grevera and J. K. Udupa, "Shape-based interpolation of multidimensional gray-level images," *IEEE Trans. on Medical Imaging*, vol 15, no. 6, pp. 881-892, 1996.
- [3] G. J. Grevera and J. K. Udupa, "An objective comparison of 3-D image interpolation methods," *IEEE Trans. on Medical Imaging*, vol. 17, no. 4, pp. 642-652, 1998.
- [4] B. Guo, "Progressive radiance evaluation using directional coherence maps", *Proc. Siggraph 98*, Orlando, FL, pp. 255-266, July 1998.
- [5] G. T. Herman, J. Zheng and C. A. Bucholtz, "Shape-based interpolation", *IEEE Computer Graphics and Applications*, vol. 12, no. 3, pp. 69-79, 1992.
- [6] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models", *Intl. J. Computer Vision*, vol. 1, no. 4, pp.312-331, 1988.
- [7] P. Lacroute and M. Levoy, "Fast volume rendering using a shear-warp factorization of the viewing transformation", *Proc. SIGGRAPH'94*, Orlando, Florida, pp. 451-458, July 1994.
- [8] T. Y. Lee and W. H. Wang, "Morphology-based three-dimensional interpolation", *IEEE Trans. on Medical Imaging*, vol. 19, no. 7, pp. 711-721, 2000.
- [9] T. M. Lehmann, Claudia Gonner and K. Spitzer, "Survey: Interpolation methods in medical image processing", *IEEE Trans. on Medical Imaging*, vol. 18, no. 11, pp. 1049-1075, 1999.
- [10] M. Levoy, "Display of surfaces from volume data", *IEEE Computer Graphics and Applications*, vol. 8, no., 3, pp. 29-37, 1988.
- [11] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710-732, 1992.
- [12] J. C. Mceachen and J. S. Duncan, "Shape-based tracking of left ventricular wall motion", *IEEE Trans. on Medical Imaging*, vol. 16, no. 3, pp. 270-283, 1997.
- [13] D. T. Puff, D. Eberly and S. M. Pizer, "Object-based interpolation via cores", *Proc. SPIE: Medical Imaging '94, Image Processing*, vol. 2167, pp. 104-115, 1994.
- [14] S.P. Raya and J. K. Udupa, "Shape-based interpolation of multidimensional objects", *IEEE Trans. on Medical Imaging*, vol. 9, no. 1, pp. 32-42, 1990.
- [15] R. A. Robb, *Biomedical Imaging, Visualization, and Analysis*, Wiley-Liss, Inc., 2000.
- [16] L. Sutherland, R. Sproull and R. Schumacker, "A characterization of ten hidden-surface algorithms", *ACM Computing Surveys*, vol. 6, no. 1, pp.387-442, March 1974.
- [17] Y. Wang and L.H. Staib, "Boundary finding with prior shape and smoothness models", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 738-743, 2000.
- [18] W. L. Williams, "Optical flow interpolation of serial slice images", *Proc. SPIE Medical Imaging '93, Image Processing*, vol. 1898, pp.93-104, 1993.

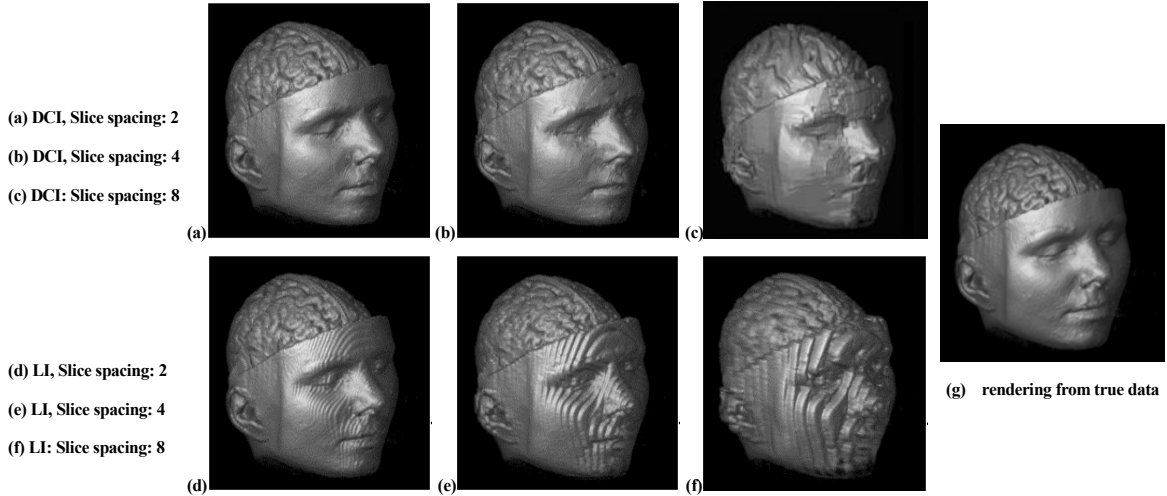


Figure 10: Comparison of 3D volume rendering for MR brain with different interpolation methods. (a), (b), (c): results from images interpolated by our DCI; (d), (e), (f): results from images interpolated by LI; (g): result from the true volume image.

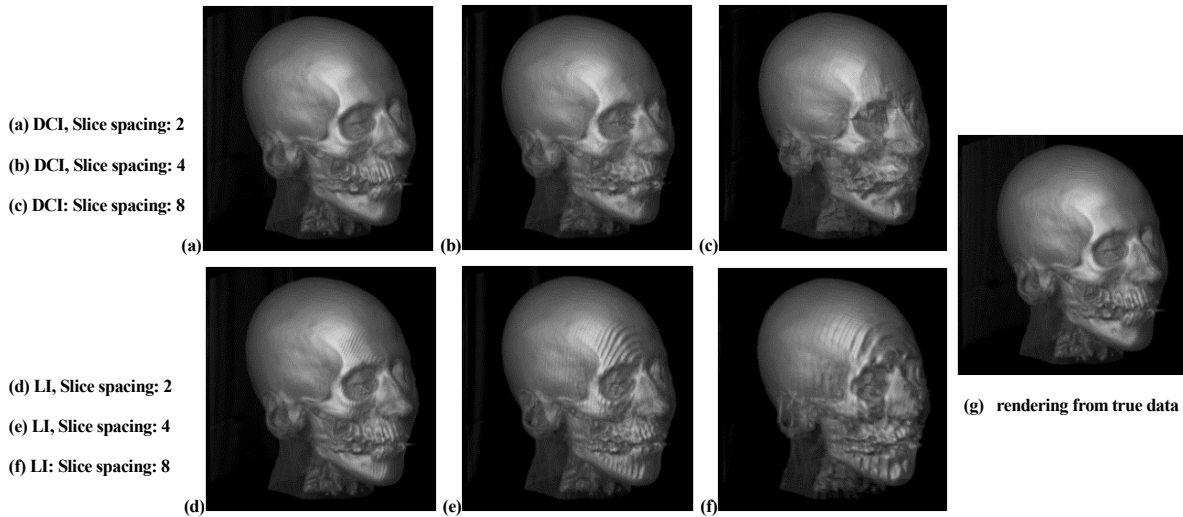


Figure 11: Comparison of 3D volume rendering for CT head with different interpolation methods. (a), (b), (c): results from images interpolated by our DCI; (d), (e), (f): results from images interpolated by LI; (g): result from the true volume image.

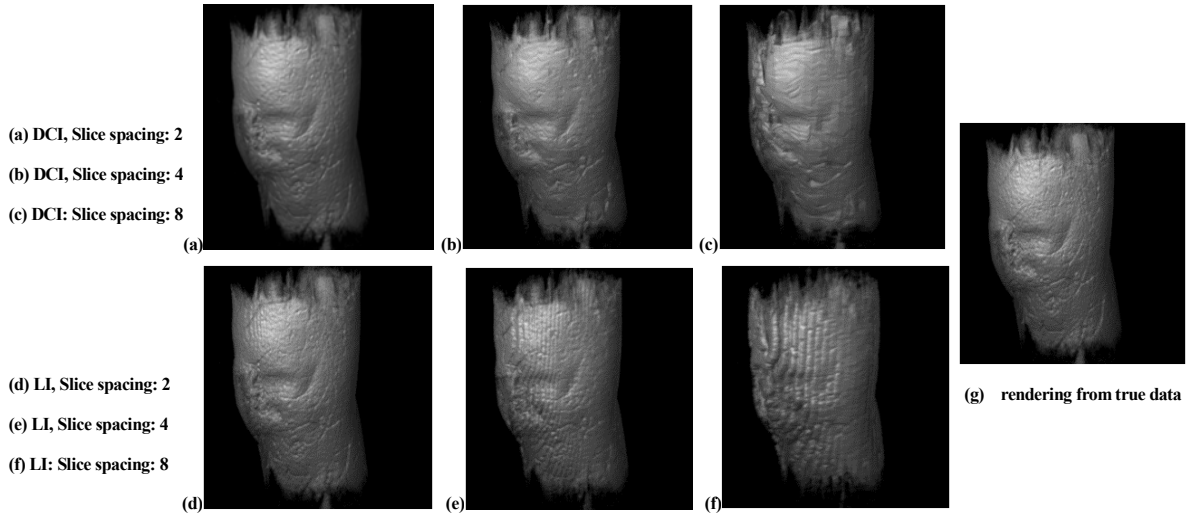


Figure 12: Comparison of 3D volume rendering for MR knee with different interpolation methods. (a), (b), (c): results from images interpolated by our DCI; (d), (e), (f): results from images interpolated by LI; (g): result from the true volume image.