Shape-based Interpolation

Gabor T. Herman, Jingsheng Zheng, and Carolyn A. Bucholtz
University of Pennsylvania

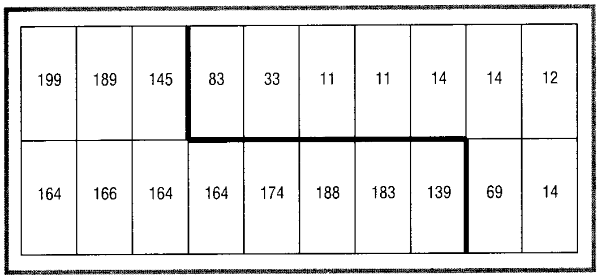
This method estimates the shape of an object from sparsely spaced slices of it. We developed a variant that has superior properties to previously proposed methods.

In tomographic medical imaging (such as computerized tomography and magnetic resonance imaging), we can reconstruct an organ or other object of interest as a series of 2D slices. But the central planes of these slices are typically separated more than the picture elements (pixels) that make up the slices.[1] When we get serial images of a dynamic organ, such as the heart,[2] the situation is even more unsatisfactory because the heart moves considerably between its successive images. For accurate quantitative analysis and display of such data,[3] we need some kind of interpolation between the slices.

The classical way of doing this[4] is to do some type of interpolation of the gray values in the slices to estimate the gray values in the "missing" slices. We then use this extended data set to process the images further. For instance, we can determine the surfaces of specific organs of interest. But you can raise two important objections to this approach:

1. In many applications, the objects of interest are represented in such a way that we can't automatically identify their boundaries in the slices without expert knowledge. Typically, a user (such as a cardiologist) provides the expertise by manually tracing on the video screen.[2] Doing this

0272-17-16/92/0500-0069$0.300 ©1992 IEEE

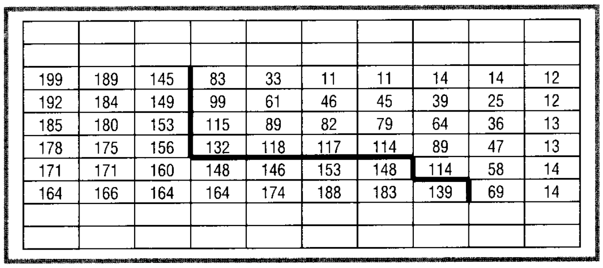


199	189	145	83	33	11	11	14	14	12
164	166	164	164	174	188	183	139	69	14

Figure 1. Values assigned to pixels in two one-dimensional "slices." The heavy line indicates the boundary of the object thresholded at 136. We took these values from matching strings of pixels in two consecutive slices of actual MRI data (namely, the data on which Figure 10 is based).

on the extended data set (which in a cardiac application might contain more than 10 times as many slices as the original data[5]) puts an unacceptable burden on the user.

2. If, as is often the case,[3,4] we use thresholding to identify the objects of interest, a prior linear interpolation for estimating the missing slices leads to an artifact. This artifact arises when the location of a boundary between two nearly uniform regions (one with values above and the other with values below the threshold) shifts considerably between two consecutive measured slices. If we perform a linear interpolation followed by thresholding, then the boundary in all intermediate slices falls in nearly the same place as in one of the measured slices. This gives us an abrupt change in boundary location as we move (through the interpolated slices) from one of the measured slices to the other. We illustrate this abrupt change in Figures 1 and 2. (You can also see this phenomenon in the 3D displays based on this method of interpolation that we show in later figures.)



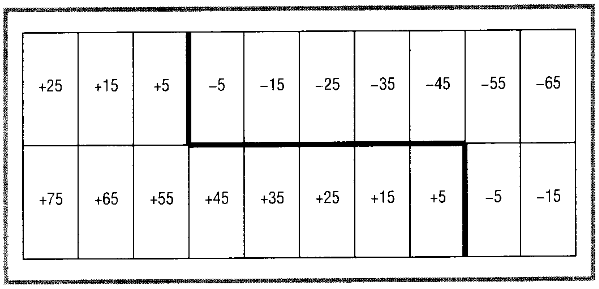
199	189	145	83	33	11	11	14	14	12
192	184	149	99	61	46	45	39	25	12
185	180	153	115	89	82	79	64	36	13
178	175	156	132	118	117	114	89	47	13
171	171	160	148	146	153	148	114	58	14
164	166	164	164	174	188	183	139	69	14

Figure 2. Values assigned to pixels after linear interpolation of the values in Figure 1. The heavy line indicates the boundary of the object thresholded at 136. You can also see this "staircase artifact" in Figure 10a. The location indicated by arrows in that figure corresponds to the location in the MRI slices from which we took the data for Figure 1.

Methods

Shape-based interpolation comes from numerical analysis literature.[6] Raya and Udupa recently adapted this technique to three- and four-dimensional medical image processing.[5] A typical application is in CT, where slice separation within an individual slice is usually much greater than the pixel size. Assuming that a methodology exists to segment an organ of interest in the slice images, we can use shape-based interpolation to estimate the locations of pixels that would be contained in that organ in the (nonexistent) intermediate slices.

We do this shape-based interpolation by converting the segmented slice images into gray-value images. In these images, the grayness is the shortest distance (within the slice) of the pixel from the boundary of the organ (positive values for inside the organ and negative values for outside; see Figure 3). We estimate segmented intermediate slices by interpolating the distance-representing gray-value slices and thresholding at zero (see Figure 4). (A minor technical matter is the handling of interpolated values that are exactly zero. In the experiments reported below, we made our decision based on the average interpolated value of the four edge-sharing pixels and, if this is zero as well, on the average of the four vertex-sharing pixels. If both of these averages turned out to be zero, we made an equal-probability random assignment. Since this rarely occurs, multiple runs on the same data with different seeds for the random-number generator gave us outcomes that are indistinguishable at the numerical accuracy at which we report our results.)



+25	+15	+5	-5	-15	-25	-35	-45	-55	-65
+75	+65	+55	+45	+35	+25	+15	+5	-5	-15

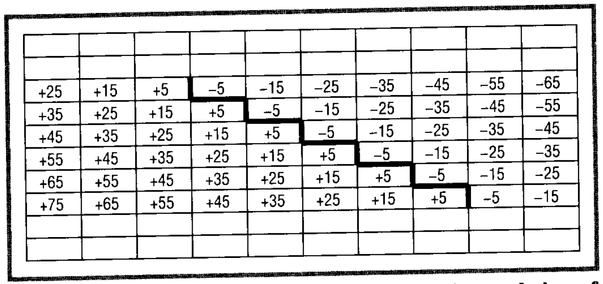
Figure 3. Distance-from-boundary values assigned to pixels in the two one-dimensional "slices" of Figure 1. (We take 10, rather than one, to be the distance between pixel centers so that we can express the distance of a pixel center from a pixel edge by an integer.) The heavy line indicates the boundary of the object thresholded at 0.

Chamfering

The distance function Raya and Udupa[5] used is a version of the city-block distance. You can calculate this distance efficiently, but you get a relatively bad approximation to Euclidean distance.[7,8] For this reason, we decided to investigate whether or not we can see a significant improvement in the performance of shape-based interpolation by using a distance function that approximates Euclidean distance more closely. As Borgefors proposed,[7,8] we studied various chamfer distances. We can calculate these by two consecutive *chamfering* processes. In chamfering, we slide a template (having only a few nonzero elements) across the image and use its nonzero values to update the image as described below. Thus, calculating such distances is not much less efficient than calculating a city-block distance.

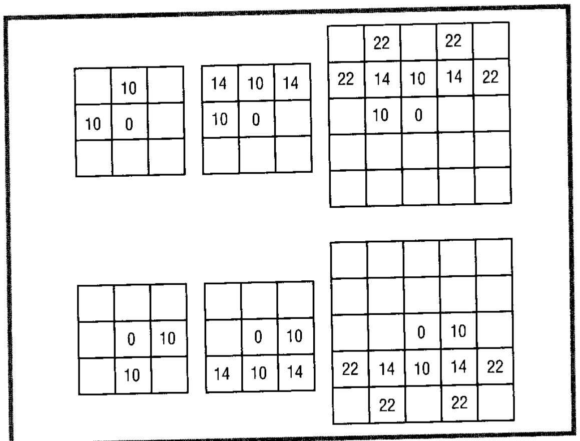
Since Borgefors' original chamfer-distance method[7,8] calculates distances of nonobject pixels from the nearest object pixel, we would have to use this method twice for our application—once for the inside and then for the outside. We found, however, a relatively simple adaptation that lets us calculate the two sets of distances simultaneously. Also, unlike Borgefors, we are interested in the distance from the boundary. That is why we have assigned +5 and −5 to the pixels just inside and just outside the boundary in Figure 3. Borgefors[7,8] and, in fact, Raya and Udupa[5] would have assigned +10 and −10 to these pixels. We now describe precisely how we define and calculate the chamfer distances.

In Figure 5 we show the templates we use in chamfering. Before discussing exactly how to apply them, we have to discuss how to initialize the calculation. We do this first in an example that is appropriate for both the city-block and the near-optimal 3 × 3 template. (Borgefors explained the sense in which templates are "near-optimal."[8,9] We use twice the weights she proposed so that we can deal with distances between pixel centers and pixel edges and yet retain the efficiency of integer arithmetic.)



+25	+15	+5	-5	-15	-25	-35	-45	-55	-65
+35	+25	+15	+5	-5	-15	-25	-35	-45	-55
+45	+35	+25	+15	+5	-5	-15	-25	-35	-45
+55	+45	+35	+25	+15	+5	-5	-15	-25	-35
+65	+55	+45	+35	+25	+15	+5	-5	-15	-25
+75	+65	+55	+45	+35	+25	+15	+5	-5	-15

Figure 4. Values assigned to pixels after linear interpolation of the values in Figure 3. The heavy line indicates the boundary of the object thresholded at 0. As indicated by this figure, there is no "staircase" artifact in Figure 10b at the location in the MRI slices from which we took the data for Figure 1.



	10	
10	0	

14	10	14
10	0	

	22		22	
22	14	10	14	22
	10	0		

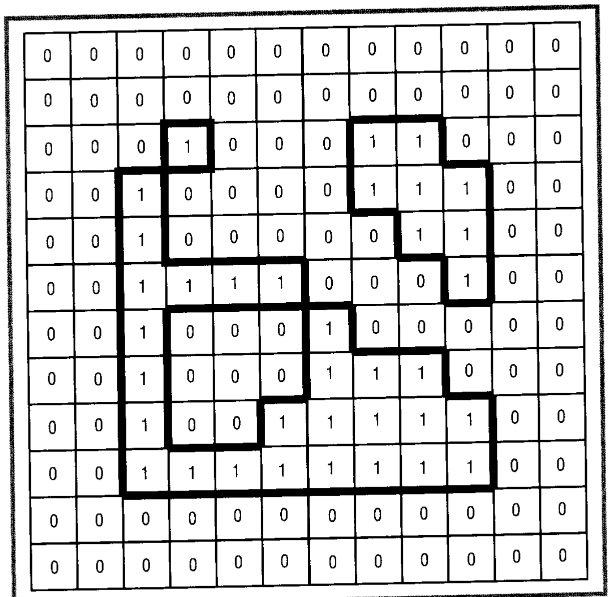
	0	10
	10	

	0	10
14	10	14

		0	10	
22	14	10	14	22
	22		22	

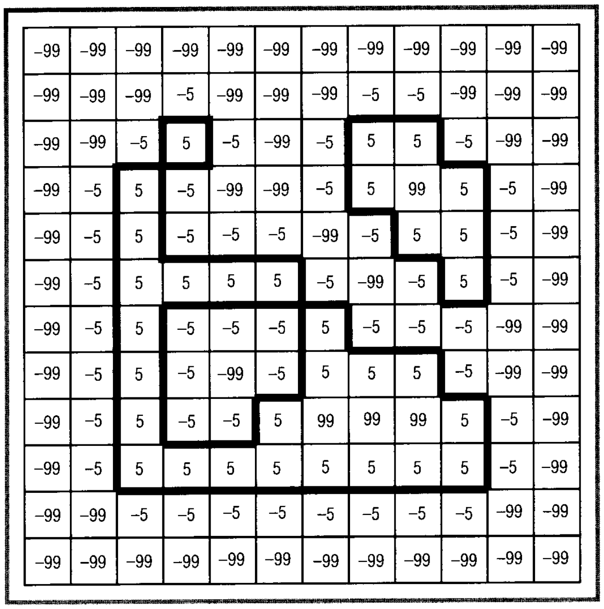
Figure 5. Templates for chamfer distance calculations. On the left, city-block; in the middle, near-optimal 3 × 3, and on the right, near-optimal 5 × 5. The top row shows the template for the first pass, and the bottom row shows the template for the second.

Consider Figure 6. To initialize for the distance calculation, replace all 0's with a very large negative number, and replace all 1's with a very large positive number (represented in our figures by −99 and 99, respectively). But there are exceptions: If a 0 shares an edge with a 1, replace it with −5, and if a 1 shares an edge with a 0, replace it with +5. (As we explain be-



0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	0	0	0
0	0	1	0	0	0	0	1	1	1	0	0
0	0	1	0	0	0	0	0	1	1	0	0
0	0	1	1	1	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	1	1	1	0	0	0
0	0	1	0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 6. A binary array. The boundary is marked by a heavy line.



-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-5	-99	-99	-99	-5	-5	-99	-99	-99
-99	-99	-5	5	-5	-99	-5	5	5	-5	-99	-99
-99	-5	5	-5	-99	-99	-5	5	99	5	-5	-99
-99	-5	5	-5	-5	-5	-99	-5	5	5	-5	-99
-99	-5	5	5	5	5	-5	-99	-5	5	-5	-99
-99	-5	5	-5	-5	-5	5	-5	-5	-5	-99	-99
-99	-5	5	-5	-99	-5	5	5	5	-5	-99	-99
-99	-5	5	-5	-5	5	99	99	99	5	-5	-99
-99	-5	5	5	5	5	5	5	5	5	-5	-99
-99	-99	-5	-5	-5	-5	-5	-5	-5	-5	-99	-99
-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99

Figure 7. The result of initializing Figure 6 for either city-block or near-optimal 3 × 3 distance calculation.

low, we use further exceptions when initializing for the near-optimal 5 × 5 chamfer distance.) Figure 7 demonstrates the result of this initialization. (In our example, all the pixels on the image's border have 0's in them. In fact, in our implemented algorithms we pad the images with 0's before initialization, and we remove these padding pixels from the final result of the chamfer distance calculation.)

Let's look more closely at the two chamfering processes. The first uses the top template in Figure 5, and the second uses the bottom template.

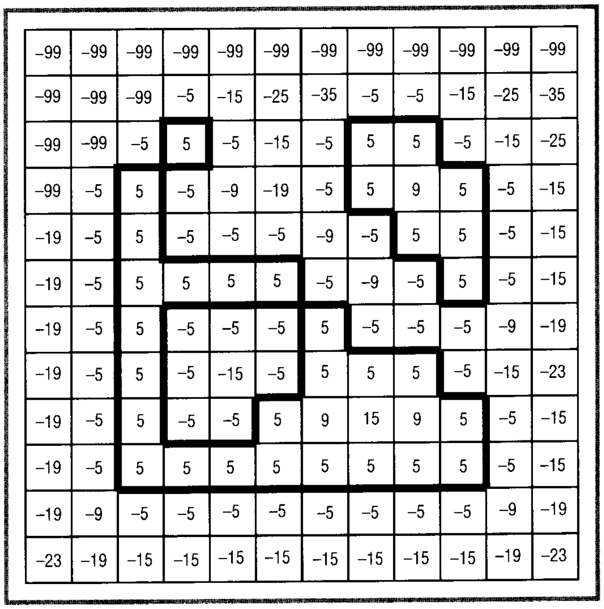
The first chamfering updates the pixels row by row from top to bottom with a left-to-right ordering within the rows. Each pixel is updated as follows. A pixel assigned the value +5 or −5 during initialization is left unaltered. Otherwise, we place the template over the image in such a way that the central (0) entry in the template is over the pixel in question. What we do next depends on the sign of the content of the pixel to be updated. If it is positive (that is, the pixel in question is inside the object), then for each image-pixel covered by a nonempty template-pixel, we add together the contents of the image-pixel and the covering template-pixel. We replace the value assigned to the pixel we're updating with the smallest of these sums. (Carefully combining initialization and chamfering insures that the new value is still positive.) If the current content of the pixel to be updated is negative (that is, the pixel in question is outside the object), then for each image-pixel covered by a nonempty template-pixel, we subtract the content of the covering template-pixel from the content of the image-pixel. We replace the value assigned to the pixel we're updating with the largest of these differences. (The new value is still negative.) We show the result of this first chamfering for the near-optimal 3 × 3 distance in Figure 8.

We apply the second chamfering to the result of the first. We update the pixels (one after the other) row by row from the bottom to the top with a right-to-left ordering within the rows. The details of this process are identical to those in the first chamfering except that now we use the bottom template in Figure 5. The result of this second chamfering for the near-optimal 3 × 3 distance is shown in Figure 9.

In Figurę 9, we see that the value −9 is assigned to pixels outside the object but sharing a vertex (but not the edges at that vertex) with the object. The value +9 is assigned to pixels inside the object that similarly share a vertex with the outside of the object. For the near-optimal 5 × 5 chamfer templates (Figure 5, right) to achieve their desired effect, we must assign these values (9) explicitly to the appropriate pixels during initialization. (See the "Initializing for 5 × 5 chamfer distance" sidebar for a detailed explanation.)

Interpolating between slices

One method of interpolating between slices, illustrated in Figure 4, is linear interpolation. However, Raya and Udupa[5] reported that cubic spline interpolation gives superior results. For this reason, we implemented both linear interpolation and interpolation using a Catmull-Rom spline.[10] In medical imaging, the latter method has become known as modified cubic spline interpolation.[11,12] This method estimates the values between two sample points (x_1 and x_2, say) by using the values at those two sample points and the values at the two



-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-5	-15	-25	-35	-5	-5	-15	-25	-35
-99	-99	-5	5	-5	-15	-5	5	5	-5	-15	-25
-99	-5	5	-5	-9	-19	-5	5	9	5	-5	-15
-19	-5	5	-5	-5	-5	-9	-5	5	5	-5	-15
-19	-5	5	5	5	5	-5	-9	-5	5	-5	-15
-19	-5	5	-5	-5	-5	5	-5	-5	-5	-9	-19
-19	-5	5	-5	-15	-5	5	5	5	-5	-15	-23
-19	-5	5	-5	-5	5	9	15	9	5	-5	-15
-19	-5	5	5	5	5	5	5	5	5	-5	-15
-19	-9	-5	-5	-5	-5	-5	-5	-5	-5	-9	-19
-23	-19	-15	-15	-15	-15	-15	-15	-15	-15	-19	-23

Figure 8. The result of applying the first chamfering for the near-optimal 3 × 3 distance calculation to Figure 7. (See Figure 5, the middle top template.)

sample points on either side of them (x_0 and x_3, say). A cubic polynomial is used. It is uniquely determined by the conditions that its values at x_1 and x_2 are exactly the sampled values, its derivative at x_1 is the difference between the values at x_2 and at x_0 divided by $x_2 - x_0$. We determine the derivative at x_2 similarly. The resulting interpolating function is continuous, has a continuous first derivative, and depends only locally on the sampled values. On the other hand, it is not defined between the first and the second sample and between the penultimate and last sample.

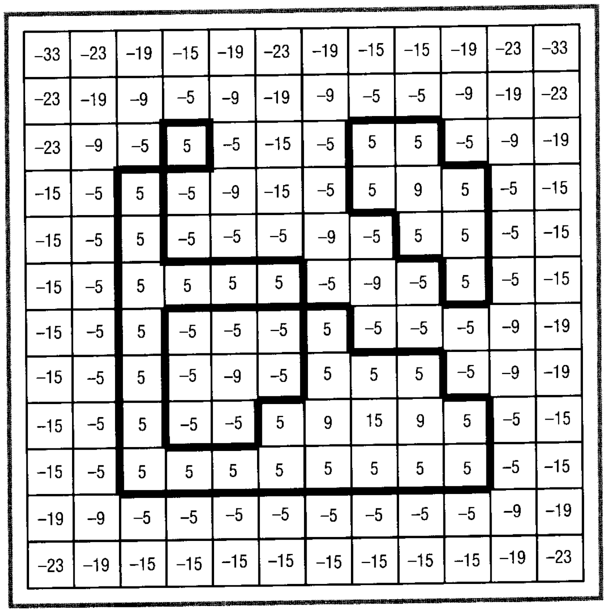
Methods compared

This completes our discussion of the various methodologies we selected for comparison. Taking into account both linear and modified cubic spline between-slice interpolation, we compared the performance of six methods:

1. Interpolation of gray values followed by thresholding.
2. Shape-based interpolation as implemented by Raya and Udupa[5].
3. Shape-based interpolation with a city-block distance (Figure 5, left) in which pixels with an edge on the boundary are initialized to be half a pixel-step (that is, ± 5) from the boundary, as illustrated in Figure 7. Raya and Udupa[5] considered such pixels to be a pixel step from the boundary.
4. Shape-based interpolation with a city-block distance (Figure 5, left) in which pixels with either an edge or a vertex on the boundary are initially assigned their (approximate) Euclidean distances from the boundary (that is, ±5 and ±9).
5. A near-optimal chamfer distance based on 3 × 3 templates (Figure 5, middle) with initial assignments to boundary pixels as illustrated in Figure 7.
6. A near-optimal chamfer distance based on 5 × 5 templates (Figure 5, right) with initial assignments to pixels using both ±5 and ±9.

Methodology of comparison

We illustrate the methodology of comparison on a particular example. We extracted matching 256 × 256 regions from each of 64 contiguous CT slices of a patient's head. We segmented all the slices for bone by thresholding at the same value. (CT value 200, on a scale in which water has value 0 and air has value −1,024. We show 3D displays based on this data set in Figure 11.) For n from 4 to 61, we applied the six methods, combined with linear between-slice interpolation, to slices $n - 1$ and $n + 1$ to estimate the segmented nth slice. (As will become clearer below, this range of n is appropriate for experiments evaluating the modified cubic spline interpolation. We could have extended it for evaluating linear between-slice interpolation, but we chose not to so that we can also compare the two between-slice interpolation methodologies.) We found, in particular, that the third method misclassified 88,426 pixels, and the fifth method misclassified 87,361 pixels. Considering that there are more than 3,800,000 pixels in the 58 slices, the number of pixels that the estimators misclassified is small—less than 2.5 percent. This is to be expected, since, except near the edges of the object, both estimators should always be correct.



−33	−23	−19	−15	−19	−23	−19	−15	−15	−19	−23	−33
−23	−19	−9	−5	−9	−19	−9	−5	−5	−9	−19	−23
−23	−9	−5	5	−5	−15	−5	5	5	−5	−9	−19
−15	−5	5	−5	−9	−15	−5	5	9	5	−5	−15
−15	−5	5	−5	−5	−5	−9	−5	5	5	−5	−15
−15	−5	5	5	5	5	−5	−9	−5	5	−5	−15
−15	−5	5	−5	−5	−5	5	−5	−5	−5	−9	−19
−15	−5	5	−5	−9	−5	5	5	5	−5	−9	−19
−15	−5	5	−5	−5	5	9	15	9	5	−5	−15
−15	−5	5	5	5	5	5	5	5	5	−5	−15
−19	−9	−5	−5	−5	−5	−5	−5	−5	−5	−9	−19
−23	−19	−15	−15	−15	−15	−15	−15	−15	−15	−19	−23

Figure 9. The result of applying the second chamfering for the near-optimal 3 × 3 distance calculation to Figure 8. (See Figure 5, the middle bottom template.)

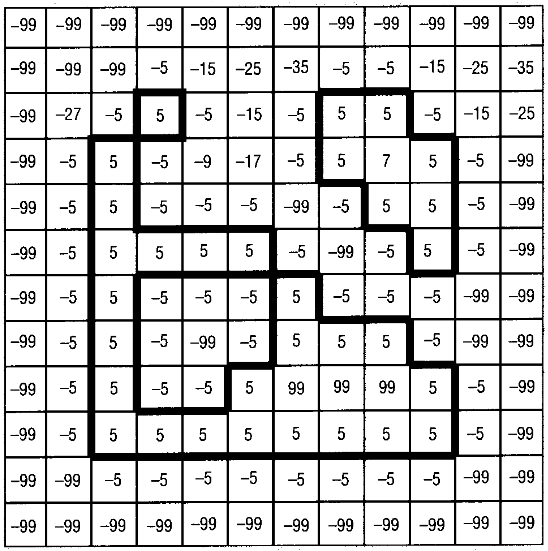
We want to attach a measure of significance, in the statistical sense, to the observed difference between the two methods. The estimates provided by the third and fifth methods differed for a total of 11,045 pixels in the 58 estimated slices. If the two methods were equally accurate, we would expect each to be correct about half the time. However, the chamfer distance provided a match of the actual segmented value 6,055 times rather than the expected 5,522.5 times. This provides us with the level of statistical significance to reject the null hypothesis that "there is no difference in performance because of the choice of the distance function" in favor of the alternative: "The method based on the near-optimal 3 × 3 chamfer distance performs better." We calculate this level of significance as follows.

Let Q be the total number of pixels that two methods classify differently. Let q be the number of such pixels that have been correctly classified by the interpolation method that has the greater accuracy for this particular data set. If we further assume that the decision by a classifier at any pixel where the two classifiers differ is independent of the decision at any other such pixel, then the null hypothesis that the methods perform equally well implies that q is a random sample from a binomial distribution with total number of items Q and equal probabilities assigned to the two classes. The probability of randomly selecting an element from this distribution

Initializing for 5 x 5 chamfer distance

You cannot simply use the initialization of Figure 7 for the 5 x 5 chamfer distance. Suppose that we attempt to do that and apply the template on the top-right of Figure 5 to the image in Figure 7. After a few steps, we arrive at the situation indicated in the figure here. The pixel that we reached is the one with 7 in it. We get 7 by adding 22 to –15, which is the content of the pixel two up and one to the right. The problem is that this –15 will be changed during the second chamfering into a –9, so we obtained the value 7 based on the fallacious information that the pixel is 22 away from a pixel with value –15. In fact, the correct distance is 9, rather than 7, which is indeed the value that is explicitly assigned during the correct initialization phase for the 5 x 5 chamfer distance. This value is not altered during the first chamfering because correct initialization also assigns the value –9 to the pixel containing –15 in the figure in this sidebar, and –9 + 22 = 13 is greater than 9.

Proper initialization insures that the final distance assignments in the whole image are correct. This means that, for any pixel, we can get to it from a boundary pixel (value 5) by a series of chamfer steps so that the final value assigned to the pixel is the sum of the value assigned to the starting pixel and the values assigned to the steps. The absolute value of the distance assigned to the pixel is as small as possible, consistent with the condition of the previous sentence.



–99	–99	–99	–99	–99	–99	–99	–99	–99	–99	–99	–99
–99	–99	–99	–5	–15	–25	–35	–5	–5	–15	–25	–35
–99	–27	–5	5	–5	–15	–5	5	5	–5	–15	–25
–99	–5	5	–5	–9	–17	–5	5	7	5	–5	–99
–99	–5	5	–5	–5	–5	–99	–5	5	5	–5	–99
–99	–5	5	5	5	5	–5	–99	–5	5	–5	–99
–99	–5	5	–5	–5	–5	5	–5	–5	–5	–99	–99
–99	–5	5	–5	–99	–5	5	5	5	–5	–99	–99
–99	–5	5	–5	–5	5	99	99	99	5	–5	–99
–99	–5	5	5	5	5	5	5	5	5	–5	–99
–99	–99	–5	–5	–5	–5	–5	–5	–5	–5	–99	–99
–99	–99	–99	–99	–99	–99	–99	–99	–99	–99	–99	–99

Figure 13 Partial result of applying to figure 7 the first chamfering precess for the near-optimal 5 × 5 distance calculation. (see Figure 5, right top template.

with value q or higher is the level of significance for rejecting the null hypothesis. (For definitions of the statistical terminology, see for example Mould's book.[13] The weak point of this methodology is the assumption that the classifiers' decisions are independent. Since this isn't likely to be the case, the actual statistical significance of the reported differences is smaller than they would be if the independence assumption was always satisfied. In this article we leave open the problem of statistically rigorously handling the dependence of decisions by these classifiers.)

Above, we compared interpolation methods based on accuracy on a pixel-by-pixel basis. One of the major applications of these methods is the estimation of volumes.[2,14] For this reason, we also compared the methods to see how well they estimate the total volume of the object in the intermediate slices.

In all these methods, we assumed that the actual segmented slices represent the "truth." In some cases, the estimate based on interpolation might be more correct from the point of view of the underlying biological structure than what is provided by segmenting the actual measured slice. This is unfortunate, but not relevant to our aim, which is estimating what we would obtain in a missing slice if we measured and then segmented it.

Results

First, let's look at the results for linear between-slice interpolation. To compare the methods, we used three different data sets of CT slices of human heads and a data set of MRI slices of a brain specimen. Our clinical protocol[15] for 3D imaging of skulls from CT scans uses a threshold of 200. For imaging skin, we considered thresholding either at value -200 or at value –100 reasonable. We therefore performed our experiment for each of the three CT scans of heads at each of these three threshold values. For one of the scans, we also performed the experiment at three higher threshold values (561, 661, 761). By studying the resulting 3D displays of the brain specimen,[14] we see that numbers in the range 106 to 136 provide us with acceptable thresholds for segmenting the brain in the MRI scan. However, the specimen was wrapped in a cloth whose MRI value is near the higher end of this range. We therefore performed our comparison on the MRI data with threshold 136. We did a second comparison at threshold 100, to see how well the methods perform when we consider both the brain and the cloth wrapping to be part of the object of interest. All together, we carried out 14 comparison experiments.

In Table 1 we show the total number of incorrectly estimated values for each of the 14 data sets and each of the six interpolation methods.

First, we confirm the results Raya and Udupa reported.[5] For skulls from CT and for brains from MRI, their shape-based interpolation method is superior to gray-value interpolation followed by thresholding (see specifically columns I.h,

Table 1.

	I.l	I.m	I.h	J.l	J.m	J.h	K.l	K.m	K.h	K.hl	K.hm	K.hh	M.l	M.h
1	•90	128	237	51	77	115	103	123	326	270	256	•241	57	117
2	134	163	216	54	72	100	119	133	271	274	278	275	46	100
3	96	124	185	44	61	88	99	112	244	247	251	248	44	98
4	94	122	182	43	60	88	98	111	242	245	250	246	•44	98
5	94	†122	182	•43	†60	87	†98	†111	•242	245	•249	246	44	•98
6	94	122	•182	43	60	•87	98	111	242	•244	249	246	44	98

Number (in thousands) of incorrectly classified pixels in the intermediate slices (not using the first three and the last three) with linear between-slice interpolation. Rows refer to method number. I, J and K refer to the three CT data sets of heads thresholded at –200 (l), –100 (m), +200 (h) +561 (hl), +661 (hm) and +761 (hh). M refers to the MRI data set of brain thresholded at 100 (l) and at 136 (h). A dagger before a number indicates that the method in question was overall most accurate for that data set. A bullet before a number indicates that the method was most accurate among those using linear between-slice interpolation, but not as accurate as the most accurate method using modified cubic spline between-slice interpolation.

J.h, K.h, M.l, and M.h). These were the only types of data sets Raya and Udupa investigated. Interestingly, in all but one of our other experiments, gray-value interpolation followed by thresholding was actually more accurate than Raya and Udupa's method.[5]

The other four methods behaved similarly from the point of view of accuracy and, with the exception of two experiments discussed below, they outperformed both Raya and Udupa's shape-based method and gray-value interpolation followed by thresholding. In all but one of the remaining twelve experiments, we achieved the best accuracy either by Method 5 or 6. Although the differences in accuracy between the four methods are relatively small, they can nevertheless be statistically significant, as we discussed at the end of the last section. The example given there refers to the column J.h in Table 1 and compares Methods 3 and 5. In that example, the mean of the binomial distribution in question is 5,522.5 and its standard deviation is less than 55. Hence, the null hypothesis that the methods are equally accurate implies that our observation is a sample from a binomial distribution that is more than 10 standard deviations from the mean.

We may therefore conclude that Method 5 is more accurate than Method 3 (on this data set) with a significance level of better than 0.00001. This is typical of comparisons of Method 5 with either of Methods 3 or 4 in the 11 experiments. On the other hand, Methods 5 and 6 do not differ significantly: sometimes one is more accurate, sometimes the other. Since Method 5 is computationally less expensive, it is the method of choice among the six. (The underlying cost of obtaining such data from CT and MRI is in the hundreds of dollars. Compared with that, the cost of applying any of our interpolation methods is insignificant. We therefore only comment on relative computational speed of various methods if their accuracy appears to be the same.)

In our exceptional instances, namely I.l and K.hh, Method 1 is more accurate than Method 5 at a significance level better than 0.00001. Thus, in these two experiments out of the 14, shape-based interpolation failed in comparison to the classical method.[4] In the case of I.l, the reason appears to be that (in the part of the head that was scanned) the outer boundary (which becomes the important factor at this low threshold) does not change much from slice to slice. Thus the disadvantage of gray-value interpolation described in Figures 1 through 4 does not exist in this case. The reason in the case of K.hh is that, at this clinically unreasonably high threshold, only the hard outer part of the skull is in the thresholded object. Thus the object consists of two thin parallel curving layers, which confound the shape-based interpolation method.

Rather interestingly, our calculations of accuracy of overall volumes gave us much less clear-cut results (see Table 2). First of all, since errors are likely to occur near the boundaries of objects, the percentage error has much to do with the ratio of the area of the boundary to the volume of the object. Thus the error tends to be small for the outer surface of the heads (l and m in Table 2), large for the skull (h in Table 2), and between these two for the brain. So these differences result from the object's being imaged rather than from the interpolation method used. We also note that, in accordance

Table 2.

	I.l	I.m	I.h	J.l	J.m	J.h	K.l	K.m	K.h	K.hl	K.hm	K.hh	M.l	M.h
1	0.3	•0.3	8.8	0.7	0.9	10.0	•0.2	•0.4	9.2	•3.2	•1.9	•8.0	1.4	6.5
2	0.2	0.4	1.0	•0.1	•0.4	8.8	0.4	0.4	1.9	5.9	8.1	10.3	1.4	1.9
3	0.2	0.4	1.0	0.5	0.8	8.5	0.4	0.4	1.9	6.0	8.2	10.3	1.4	1.5
4	0.1	0.4	†1.0	0.5	0.8	8.4	0.4	0.4	•1.9	6.0	8.1	10.2	1.4	1.5
5	0.1	0.3	1.0	0.5	0.8	•8.3	0.4	0.4	1.9	5.9	8.0	10.2	•1.4	1.5
6	•0.1	0.3	1.0	0.5	0.8	8.3	0.4	0.4	1.9	5.9	8.1	10.2	1.4	†1.5

Percentage error (absolute value) in the estimates of the number of object pixels (volume) in the intermediate slices as compared to the actual number of object pixels when using linear between-slice interpolation. See Table 1 for the explanation of layout and notation.

Table 3.

	I.l	I.m	I.h	J.l	J.m	J.h	K.l	K.m	K.h	K.hl	K.hm	K.hh	M.l	M.h
1	†84	126	199	48	77	111	•98	123	304	265	253	†235	57	116
2	134	168	210	50	69	94	119	135	267	263	266	262	45	99
3	98	134	179	43	62	86	102	119	243	239	242	238	44	†98
4	98	136	178	43	62	85	103	120	243	237	240	237	44	98
5	97	133	177	42	61	†85	102	118	†241	236	239	236	44	98
6	97	•133	†177	†42	•61	85	102	•118	241	†236	†239	236	†44	98

Number (in thousands) of incorrectly classified pixels in the intermediate slices (not using the first three and the last three) with modified cubic spline between-slice interpolation. See Table 1 for the explanation of layout and notation. (Here, a bullet before a number indicates that the method was most accurate only among those using modified cubic spline between-slice interpolation.)

with the previously stated results on accuracy of individual pixel classification, Methods 3 through 6 are similar in performance, with Methods 5 and 6 extremely similar. However, contrary to the results on the accuracy of individual pixel classification, the performance of classical Method 1 is the best (among these methods using linear between-slice interpolation) for nearly half the cases.

Although this last result is surprising enough to warrant further investigation, it does not by itself indicate a definite error in our experiments. A classifier that is more accurate on individuals might easily be less accurate on volumes. The following is a good example. Consider a method (call it Method I) that has 80 percent probability of estimating a bone pixel correctly and 95 percent probability of estimating a "not-bone" pixel correctly. Consider Method II, for which the corresponding probabilities are 90 and 99 percent. Clearly, Method II is better than Method I. Indeed, if we have an object of 200 bone pixels and 800 not-bone pixels, the expected number of errors for Method I will be 80 and for Method II, only 28. However, the expected estimate of the volume (number of bone pixels) by Method I will be 200 (perfect!). By Method II, the volume estimate will be 188. Whether a phenomenon of this type is present in our interpolators (and what we should do about it if it is) will form part of our future work on this topic.

Table 4.

	I.l	I.m	I.h	J.l	J.m	J.h	K.l	K.m	K.h	K.hl	K.hm	K.hh	M.l	M.h
1	0.2	0.4	6.5	0.7	1.2	10.7	0.2	0.4	7.6	4.6	0.9	3.8	1.7	6.3
2	†0.0	0.1	•1.2	0.3	0.1	†0.3	0.1	0.1	†0.6	0.2	0.6	1.7	†1.2	2.6
3	0.0	†0.0	1.2	0.1	0.2	0.7	0.1	0.1	0.7	0.2	0.6	1.5	1.2	2.2
4	0.0	0.1	1.3	†0.1	†0.1	1.0	†0.1	†0.0	0.8	0.3	†0.4	†1.3	1.3	2.3
5	0.0	0.0	1.2	0.1	0.2	0.6	0.1	0.1	0.7	0.1	0.7	1.7	1.2	•2.2
6	0.0	0.0	1.2	0.1	0.2	0.6	0.1	0.1	0.7	†0.1	0.7	1.7	1.2	2.2

Percentage error (absolute value) in the estimates of the number of object pixels (volume) in the intermediate slices as compared to the actual number of object pixels when using modified cubic spline between-slice interpolation. See Table 1 for the explanation of layout and notation. (Here, a bullet before a number indicates that the method was most accurate only among those using modified cubic spline between-slice interpolation.)

Modified cubic spline between-slice method

We repeated the experiments described in the previous section for the modified cubic spline between-slice interpolation method. In estimating slice n, we use slices $n - 3$, $n - 1$, $n + 1$, and $n + 3$. (This mimics the standard situation in which the measured slices are equally spaced.) In Table 3 we show the total number of incorrectly estimated values for each of the 14 data sets and each of the six interpolation methods. In Table 4 we also show the percentage errors in volume estimation.

First let's discuss the effect of using modified cubic spline as opposed to linear between-slice interpolation. We found that for correctly estimating pixel values for all but four data sets (I.m, J.m, K.l, and K.m), the best result using modified cubic spline interpolation is better than the best result using linear between-slice interpolation. However, the difference between the best results produced by the two between-slice interpolation methods is less than 10 percent for all 14 data sets. For volume estimation, for all but two data sets (I.h and M.h), the best result using modified cubic spline interpolation is better than the best result using linear between-slice interpolation. We also note that the error in volume estimate is less than 3 percent when estimated by any of the shape-based interpolation methods combined with modified cubic spline between-slice interpolation. The error can be as much as 8 percent for some data sets (J.h and K.hh), even if you select the most advantageous method using linear between-slice interpolation. Thus, modified cubic spline between-slice interpolation seems more advantageous than linear between-slice interpolation.

Comparing the six methods described at the end of the "Interpolation methods" section, all with modified cubic spline between-slice interpolation, we again find just about no difference in performance between Method 5 (near-optimal 3 x 3 chamfer distance) and Method 6 (near-optimal 5 x 5 chamfer distance). Comparing them with the other four methods to check their accuracy in classifying individual pixels, we find the better of them to be superior to the best of the other four in all but four cases. In two of these cases (I.l and K.l, cases of slowly varying outer boundary), the classical Method 1 is more accurate at a significance level of more than 0.00001. In the third case (K.hh, a case of thin parallel curving layers), the classical Method 1 is again more accurate but only at a significance level of 0.00023. In the fourth case (M.h), Method 3 was slightly more accurate than Method 5, but the level of significance is worse than 0.07 (not significant). Comparing Method 5 with Method 1 in the 11 experiments where Method 5 is superior and with Method 2 in all 14 experiments, we find that the superiority of Method 5 in accurately classifying individual pixels is significant at a level better than 0.00001. This is typically, but not always, so when we compare Method 5 to Methods 3 and 4.

The results regarding volume estimation (Table 4) are again much less clear cut, except that they clearly indicate classical Method 1's inferiority for volume estimation as opposed to any of the shape-based interpolation techniques. The percentage error in volume provided by gray-value interpolation followed by thresholding is never superior and is sometimes as much as 10 times worse than the percentage error in volume for the same data set provided by any of the shape-based interpolation methods. The shape-based methods' volume-error percentages are similar. From our results, we can't deduce which volume estimation is superior using any one of the shape-based interpolation techniques as opposed to any other.

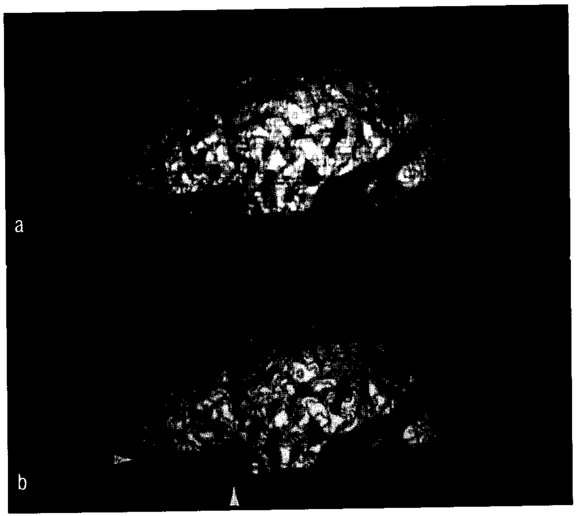


Figure 10. 3D displays of the surfaces of objects produced by interpolating MRI slices of a brain. (a) We used linear gray-value interpolation followed by thresholding at 136. (b) We performed thresholding of the original slices at 136, then used our shape-based interpolation method based on the near-optimal 3 × 3 chamfer distance with modified cubic spline between-slice interpolation. The surfaces are displayed with the same display methodology. Note the essential differences in the appearance of the superior part of the brain and also in the region indicated by the arrows in 10a; we took the numbers for Figure 1 from this region.

Displaying interpolated objects

Since, based on our experiments, shape-based interpolation based on the near-optimal 3 x 3 chamfer distance with modified cubic spline between-slice interpolation appears to be the method of choice, we illustrate (in Figures 10 and 11) its performance for 3D display (as compared to the classical method of linear gray-value interpolation followed by thresholding[4]). We produced the images using Raya's Softvu package.[16] For all the images, we used Softvu to detect the surface of an (interpolated) binary object[17] and display the detected surface using a variant[18] of the image-space shading method.[19] Although some methods of rendering the detected surfaces are superior to image-space shading[20-22] (both in the perceived and actual quality of their approximations of the underlying biological surfaces' appearance), we chose to use image-space shading because it more accurately reflects the interpolated binary object's surface shape; we focus on this object in comparing the interpolation methods. Rendering methods that attempt to approximate the appearance of the underlying biological structure introduce visual clues in addition to those in the surface of the interpolated binary object. These methods thus tend to obscure the differences in the appearance of surfaces obtained with different interpolation methods.

We based Figure 10 on the MRI data set referred to as M.h in Tables 1 through 4. We used the same data set to get the numbers for Figure 1 (and consequently for Figures 2 through 4). The original data set contained 31 slices with a slice-spacing of 3.0 mm with pixel size 0.7813 mm by 0.7813 mm. After interpolation, based in both cases on thresholding at 136, we have 108 (binary-valued) slices with slice-spacing 0.7813 mm. The first of these "interpolated" slices coincided with the second original slice. (This is because of the nature of the modified cubic spline interpolation explained above.) We can see the most noticeable differences between the two images (and in the

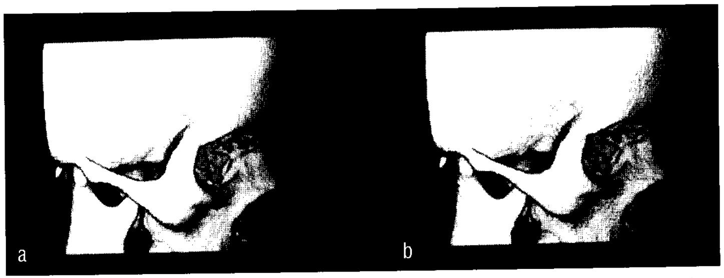


Figure 11. 3D displays of the surfaces of objects produced by interpolating CT slices of a patient's head. (a) We used linear gray-value interpolation followed by thresholding at 200. (b) We performed thresholding of the original slices at 200, then used our shape-based interpolation method. The surfaces are displayed with the same display methodology.

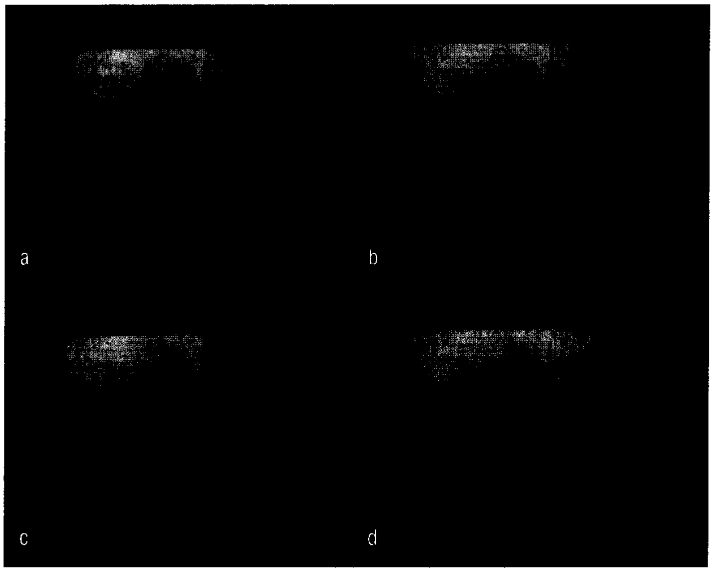


Figure 12. 3D displays of the surfaces of objects produced by interpolating CT slices of a patient's head. We used the same data as in Figure 11, but we removed every second slice to increase the difficulty of the interpolation. (a) We used linear gray-value interpolation followed by thresholding at 200. (b) We performed thresholding of the original slices at 200, then used Raya and Udupa's shape-based interpolation method. (c) We performed thresholding of the original slices at 200, then used our shape-based interpolation method. (d) We performed thresholding of the original slices at 200, then used our shape-based interpolation. All surfaces are displayed with the same methodology.

underlying interpolated 3D objects) in the superior part of the brain and in the part indicated by arrows: Gray-value interpolation followed by thresholding results in the staircase-like artifacts explained in Figure 2. For reasons explained in Figure 4, shape-based interpolation eliminated these artifacts. All versions of shape-based interpolation reduce these artifacts significantly. For this data set, the differences between 3D displays of objects produced by the various versions of shape-based interpolation are subtle.

We based Figure 11 on the CT data set referred to as J.h in Tables 1 through 4. We used the same data as an illustration in the "Methodology of comparison" section. The original data set contained 64 slices with a slice-spacing of 1.5 mm with pixel size 0.4004 mm by 0.4004 mm. After interpolation, based in both cases on thresholding at 200, we have 229 (binary-valued) slices with slice-spacing 0.4004 mm. The first of these "interpolated" slices coincides with the second original slice. We can see the most noticeable differences between the two images (and in the underlying interpolated 3D objects) where the slice-to-slice changes are the most rapid, such as the lower end of the cheekbone and the region above the nose between the tops of the eye sockets. All versions of shape-based interpolation reduce these artifacts significantly. For this data set also, the differences between 3D displays of objects produced by the various versions of shape-based interpolation are subtle.

To illustrate clearly the effect of the choice of interpolation method on 3D display, we have to make the interpolation task more difficult. We do this by removing some of the slices from the data set. We did this with the CT data set used to produce Figure 11. By removing every second slice (and so doubling the distance between the slices to be interpolated to 3.0 mm), we demonstrate in Figure 12 the improvements as we move from linear gray-value interpolation to Raya and Udupa's shape-based interpolation.[5] We then show the improvements in shape-based interpolation based on the near-optimal 3 x 3 chamfer distance—first with linear and then with modified cubic spline between-slice interpolation.

Discussion

We proposed and evaluated some extensions of Raya and Udupa's shape-based interpolation.[5] In particular, we suggested that pixels that share a boundary edge (one inside and the other outside the object) should both be considered to be at a distance from the boundary that is half the distance between adjacent pixel centers. Using such an initialization for distance calculations, we developed a generalization of the chamfer distance calculation Borgefors proposed.[7,8] This generalization lets us simultaneously calculate distances within the object and its background by two consecutive chamfering processes.

We evaluated the performance of a number of variants of our method. For between-slice interpolation we confirmed the previously reported[5] superiority of cubic spline-based interpolation over linear interpolation. Based on our experiments, we strongly recommend that this be combined with the within-slice distance calculation method based on the near-optimal 3 x 3 chamfer distance (Figure 5, middle). The accuracy of this method is as good as that of the method based on the near-optimal 5 x 5 chamfer distance (Figure 5, right), but it requires fewer calculations. From the point of view of classifying individual pixels in the missing slices, we can say with statistically significant confidence that, under typical medical circumstances, a shape-based interpolation method based on this distance is better than the classical method of gray-value interpolation followed by thresholding[4] or any of the shape-based interpolation methods using the city-block distance (Figure 5, left) that we tried. As for volume estimation, the performance of our method appears to be as good as any of the other methods we tested, and it is superior to the classical method,[4] even if the latter is combined with the superior modified cubic spline between-slice interpolation.

There is, however, a computational price to be paid for the gain in accuracy. Raya and Udupa^5 describe a very efficient implementation of their approach, but some of their ideas don't carry over to the chamfer approach. While the efficiency of the double chamfering used in our approach is acceptable, it's not as efficient as the implementation Raya and Udupa described.

Future researchers might want to investigate the one aspect in which shape-based interpolation clearly fails as compared to the classical method of gray-value interpolation and thresholding. In the classical method, it makes good sense to cover the object of interest with slices and use additional slices at the ends. In an additional slice, there is no object (that is, if thresholded at the desired value, it becomes blank). Nevertheless, the classical method can use the gray-value information in it to estimate gray values in missing slices between it and the rest of the object. When you perform thresholding on these newly estimated slices, they might well be found to contain part of the object. As opposed to this, shape-based interpolation as we described it does not let us go beyond the last slice in which we identified some part of the object.

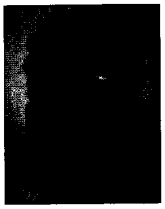
Nevertheless, we have demonstrated that shape-based interpolation using a near-optimal 3 x 3 distance and modified cubic spline between-slice interpolation clearly has superior properties to previously proposed methods for estimating object locations in missing slices in tomographic radiology. ❑

Acknowledgments

We are grateful to Etemad Hossein for writing the first programs in our laboratory for comparing Methods 3 and 5 (with linear interpolation) and to Jayaram K. Udupa and Robert M. Lewitt for useful discussions and comments on earlier versions. Our research is supported by NIH grants HL28438 and CA50851 and by NSF grant IRI-9013341. Carolyn A. Bucholtz's visit to Philadelphia was made possible by a CSIRO Overseas Study Award.

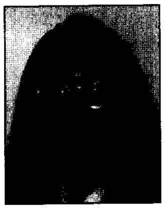
References

1. G. Herman, "Special Issue on Computerized Tomography," *Proc. IEEE,* Vol. 71, No. 3, Mar. 1983, pp. 291-435.
2. J. Raichlen, et al., "Dynamic 3D Reconstruction of the Left Ventricle from 2D Echocardiograms," *J. Amer. Coll. Card.,* Vol. 8, No. 2, Aug. 1986, pp. 364-370.
3. J. Udupa and G. Herman, eds., *3D Imaging in Medicine,* CRC Press, Boca Raton, Fla., 1991.
4. G. Herman and C. Coin, "The Use of 3D Computer Display in the Study of Disk Disease," *J. Computer Assisted Tomography,* Vol. 4, No. 4, Aug. 1980, pp. 564-567.
5. S. Raya and J. Udupa, "Shape-based Interpolation of Multidimensional Objects," *IEEE Trans. Medical Imaging,* Vol. 9, No. 1, Mar. 1990, pp. 33-42.
6. D. Levin, "Multidimensional Reconstruction by Set-valued Approximation," *IMA J. Numerical Analysis,* Vol. 6, 1986, pp. 173-184.
7. G. Borgefors, "Distance Transformations in Arbitrary Dimensions," *Computer Vision, Graphics and Image Processing,* Vol. 27, NO. 3, Sept. 1984, pp. 321-345.
8. G. Borgefors, "Distance Transformations in Digital Images," *Computer Vision, Graphics and Image Processsing,* Vol. 34, No. 3, Sept. 1986, pp. 344-371.
9. G. Borgefors, "Another Comment on "A Note on 'Distance Transformations in Digital Images'," *Computer Vision, Graphics and Image Processsing,* Vol. 54, No. 2, Sept. 1991, pp. 301-306.
10. E. Catmull and R. Rom, "A Class of Local Interpolating Splines," in *Computer Aided Geometric Design,* R. Barnhill and R. Risenfeld, eds., Academic Press, San Francisco, 1974, pp. 317-326.
11. B. Oppenheim, "Reconstruction Tomography from Incomplete Projections," in *Reconstruction Tomography in Diagnostic Radiology and Nuclear Medicine,* M. Ter-Pogossian et al., eds., University Park Press, Baltimore, 1977, pp. 155-183.
12. G. Herman, S. Rowland, and M.-M. Yau, "A Comparative Study of the Use of Linear and Modified Cubic Spline Interpolation for Image Reconstruction," *IEEE Trans. Nuclear Science,* Vol. 26, No. 2, Apr. 1979, pp. 2879-2894.
13. R. Mould, *Introduction to Medical Statistics,* Adam Hilger, Bristol, England, 1989.
14. M. Kohn et al., "Analysis of Brain and Cerebrospinal Fluid Volumes with MR Imaging, Part 1: Methods, Reliability, and Validation," *Radiology,* Vol. 178, No. 1, Jan. 1991, pp. 115-122.
15. G. Herman, "3D Imaging on a CT or MR Scanner," *J. Computer Assisted Tomography,* Vol. 12, No. 3, May-June 1988, pp. 450-458.
16. S. Raya, "Softvu—A Software Package for Multidimensional Medical Image Analysis," *Proc. SPIE,* Vol. 1232, 1990, pp. 162-166.
17. D. Gordon and J. Udupa, "Fast Surface Tracking in 3D Binary Images," *Computer Vision, Graphics and Image Processsing,* Vol. 45, No. 2, Feb. 1989, pp. 196-241.
18. S. Raya, J. Udupa, and W. Barrett, "A PC-based 3D Imaging System: Algorithms, Software and Hardware Considerations," *Computerized Medical Imaging and Graphics,* Vol. 14, No. 5, Sept. 1990, pp. 353-370.
19. D. Gordon and R. Reynolds, "Image Space Shading of 3D Objects," *Computer Vision, Graphics and Image Processsing,* Vol. 29, No. 3, Mar. 1985, pp. 361-376.
20. L.S. Chen et al., "Surface Shading in the Cuberille Environment," *IEEE CG&A,* Vol. 5, No. 12, Dec. 1985, pp. 33-43.
21. U. Tiede et al., "Investigation of Medical 3D-Rendering Algorithms," *IEEE CG&A,* Vol. 10, No. 2, Mar. 1990, pp. 41-53.
22. J. Udupa, H.M. Hung, and K.S. Chuang, "Surface and Volume Rendering in 3D Imaging: A Comparison," *J. Digital Imaging,* Vol. 4, No. 3, Aug. 1991, pp. 159-168.



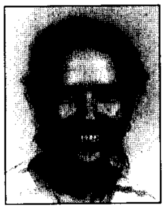
Gabor T. Herman is a professor in the medical imaging section of the University of Pennsylvania's department of radiology. Before that he worked at Brighton College of Technology, England, IBM UK, and in the computer science department at the State University of New York at Buffalo.

Herman received his MS and PhD degrees in mathematics from the University of London, England, and his MS degree in engineering science from the University of California.



Jingsheng Zheng is a postdoctoral fellow in the medical imaging processing group of the University of Pennsylvania's Department of Radiology. Her research interests include signal processing, image processing, 3D visualization, and image reconstruction.

Zheng received her BS in electrical engineering from Tsinghua University, China, and her MS and PhD in bioengineering from the University of Pennsylvania.



Carolyn A. Bucholtz is a database/software manager for the Australian National Genomic Information Service, based at Sydney University. From 1968-1991, she worked in CSIRO's Division of Biomolecular Engineering as an experimental scientist. Part of this work concerned 3D imaging.

Bucholtz received her BSc in biochemistry from the University of Sydney and her BA in computing from Macquarie University in Sydney.

Contact Herman at the University of Pennsylvania, Blockley Hall, Fourth Floor, 418 Service Drive, Philadelphia, PA 19104-6021.