Shape-Based Interpolation Methods Applied to Medical Imaging

ROBERTO DE ALENCAR LOTUFO
ALEXANDRE XAVIER FALCÃO

Grupo de Computação de Imagens
Departamento de Engenharia de Computação e Automação Industrial
Faculdade de Engenharia Elétrica
C.P. 6101, Unicamp
13081 - Campinas - SP
lotufo@dca.fee.unicamp.br
xavier@dca.fee.unicamp.br

Abstract. Sectional images generated by medical scanners usually have lower interslice resolution than resolution within the slices. Shape-based interpolation is a method of interpolation that can be applied to the segmented 3D volume to create an isotropic data set. It uses a distance transform applied to every slice prior to estimation of intermediate binary slices. Gray-level interpolation has been the classical way of estimating intermediate slices. There is also a combination of these two forms of interpolation, using the average magnitude of the gradients as a normalizing factor of the combination. This paper presents new comparison results of these interpolation methods.

1 Introduction

Three-dimensional data created by tomographic medical imaging devices are usually presented as a sequence of two-dimensional slices. The distance between the slices is typically greater than the distance between the pixels within the slices. An interpolation technique is often applied to convert the data to an isotropic volume with the same resolution in all three dimensions. Interpolation is also necessary for accurate quantitative analysis and/or display of information in such data [1]. The classical method of interpolation [2] uses the gray-level information in the slices to estimate what the gray values would be in intermediate slices.

Raya and Udupa [3] have introduced a technique called shape-based interpolation which can be applied to segmented binary slices. The technique converts the binary slice into a gray-level image, in which the gray value approximates the distance of the pixel to the nearest point on the boundary of the object. Positive values are assigned to pixels inside the object and negative values to pixels outside. The intermediate binary slices are estimated by interpolating the distances and thresholding the result at zero.

Herman et al. [4] reported on a performance analysis of shape-based interpolation, confirming its general superiority to gray-level interpolation. However, they found cases where traditional gray-level interpolation performs better than shape-based interpolation.

Higgins et al. [5] described an extension to the shape-based interpolation algorithm focusing on the problem of extracting the 3-D coronary arterial tree. They used the centroid and the gray-level information on the objects to register each cross-section before applying the interpolation procedure.

Recently, Lotufo et al. [6] introduced a new combination of the gray-level information with the distance measurements used in shape-based interpolation. The resulting algorithm is generally superior to algorithms based either only on gray value or on distance information.

This paper shows new results of the comparison evaluation amoung the classical gray interpolation, the Euclidean shape-based interpolation and their combination. Although shape-based interpolation methods have been studied for a while, we present, at first time, results of the behavior of the algorithms at every slice, whereas previous work usually only report statistical comparison measurements of the total data set.

2 Methods

2.1 Shape-based Interpolation Methods

Shape-based interpolation assigns to each pixel in a slice an estimate of its distance to the nearest boundary point in that slice. The pixels inside an object are assigned positive values while the pixels outside are assigned negative values. The interpolation is then performed on these new distance values and the resultant slices are thresholded at zero.

One way of incorporating the gray-level information into shape-based interpolation is by designing a new distance measurement as a function of the gray information in the slices. If this distance is comparable to the Euclidean Distance (ED) to the boundary, then the two distances can be combined.

In situations where segmentation can be performed by thresholding, we define the *gray-excess* of a pixel as the pixel gray value minus the threshold. Dividing this gray-excess by the average magnitude of the gradients, computed over all the pixels used for between-slice interpolation, results (at least in the neighborhood of the boundary) in a new estimate of the distance to the boundary. We call this new distance *Gray Distance* (GD).

With the two distinct but comparable distance transforms: Euclidean and Gray distances, it is possible to generate three different configurations by using them individually or combined.

- Gray Distance (GD): shape-based interpolation using this distance is equivalent to gray interpolation.
- Euclidean Distance (ED): original shape-based interpolation, but using Euclidean distance (rather than the cityblock distance as in [3]).
- Euclidean-Gray Distance (EGD): is the sum of Euclidean and Gray distances.

The equations for these distance transforms are summarized in Fig. 1.

The effect of the two basic distance transforms used individually and combined is illustrated in Figure 2. For easy visualization, the problem is shown in one dimension, assuming linear interpolation. Fig. 2a shows the gray values of two original slices used to interpolate the three intermediate slices. The threshold value is 136 and the bold lines show the boundaries. Fig. 2b gives the Euclidean distance mapping of the two slices and the interpolated distance values. Fig. 2c shows the Gray distance mapping. This method is equivalent to classical gray-level interpolation. Fig. 2d gives the combination of the two distance mappings (Euclidean-Gray distance).

It can be seen from this example that the original shape-based interpolation gives a straight line connecting the boundaries. The problem is that the straight line is not always the best estimation for the structures to be interpolated. On the other hand, gray interpolation, given by Fig. 2c, resulted in a step too sharp. Euclidean-Gray distance (Fig. 2d) gives an intermediate result of the two previous distances. It is a blend of the methods and the experiments reported here have demonstrated its superiority when linear between-slice interpolation is used.

In [6], it was also introduced a so-called Local Gray Distance, which has been abandoned, as this distance values can sometimes be high, when they are far away from the boundary.

Figure 2 is also useful for illustrating some aspects of the evaluation of interpolation methods. In biomedical imaging applications, the shape complexity of some structures is very high and it is difficult to quantify the way in which a given interpolation technique is better than another. Given the two original slices presented in Fig. 2, it is difficult to assess which method is the best. We can guess that possibly something between shape-based interpolation and gray-level interpolation may be better, but it is hard to say what should be the optimal interpolation for this example.

2.2 Computing the Euclidean distance

In their work on shape-based interpolation, Raya and Udupa [3] used the cityblock distance for their algorithms and Herman et al. [4] used a chamfer distance [7] as an approximation to the Euclidean distance. Raya and Udupa proposed a very fast algorithm which computes the distance only for pixels which are to be used for between-slice interpolation due to a difference in classification from the corresponding pixels in adjacent slices. The same technique cannot be applied to shape-based interpolation with the chamfer distance, as that algorithm requires the processing of all pixels in the slices.

As the number of pixels actually contributing to the between-slice interpolation is normally a small fraction of the total number of pixels, the algorithm reported in [6] uses a fast technique based on a lookup table to help the search for the nearest boundary point.

The table consists of three values per entry: x_{off}, y_{off}, and the distance of the pixel at position (x_{off}, y_{off}) to the origin. The table is built for values of offsets x_{off} and y_{off} within a specified neighborhood and then sorted in increasing order of the distance. To determine the distance mapping at a given pixel in the slice, the offsets, obtained from the table, are added to the coordinates of the current pixel and the pixel at that resulting position is tested. The offsets in the table are used to search sequentially to the nearest border point of the neighborhood specified. This point is found when the first pixel at the offset gives a value different from that of the pixel under the current coordinate. The corresponding distance is then used as the distance value of that pixel. If the table is fully searched without hitting any boundary,

Within slice distance to the boundary	Equation
Gray	$GD = \frac{p(x,y)-threshold}{GR}$
Euclidean	$ED = \sqrt{(x - x_{bound})^2 + (y - y_{bound})^2} - 0.5$
Euclidean-Gray	$EGD = ED + GD$

where :
$p(x, y)$ is the value of pixel (x, y);
x_{bound}, y_{bound} are the coordinates of the nearest boundary pixel;
GR is the average of the magnitudes $|grad(x, y)|$ of the gradients of the pixels (x, y) used during between-slice interpolation.

Figure 1: Equations for the new distance transforms.

it means that there is no boundary within the neighborhood used for building the table. In such cases the largest distance value in the table is used. Normally the parameter to specify the neighborhood is chosen to create a neighborhood whose radius is the maximum interslice distance within the data set.

With this lookup table method, the definition of the distance can be precomputed once during the table-building process. In the implementation of the algorithm, the 2D Euclidean distance is given by:

$$D(x_{off}, y_{off}) = \sqrt{x_{off}^2 + y_{off}^2} - 0.5$$

The subtraction of half pixel distance gives the distance from the center of the pixel to the boundary, as suggested in [4]. Interpolation now is as fast as the Raya-Udupa algorithm.

2.3 Computing the average magnitude of the gradients GR

The Gray distance requires the determination of the average magnitude of the gradients computed over the pixels used for between-slice interpolation. The estimation of the local gradient is made by the Sobel edge operator applied within the slice. The equations of the operator are given by:

$$\begin{aligned} G_x(x,y) &= \frac{p_{(x-1,y-1)} + 2p_{(x,y-1)} + p_{(x+1,y-1)}}{8} - \\ &- \frac{p_{(x-1,y+1)} + 2p_{(x,y+1)} + p_{(x+1,y+1)}}{8} \\ G_y(x,y) &= \frac{p_{(x-1,y-1)} + 2p_{(x-1,y)} + p_{(x-1,y+1)}}{8} - \\ &- \frac{p_{(x+1,y-1)} + 2p_{(x+1,y)} + p_{(x+1,y+1)}}{8} \\ |G(x,y)| &= \sqrt{G_x^2(x,y) + G_y^2(x,y)} \end{aligned}$$

In regions where the boundary is near an homogeneous area, the local gradient can be very small or even null. To avoid such situations, which result in large values for the Gray distance, a parameter is introduced that specifies the minimum allowed gradient value used in determining Gray distance.

The average magnitude GR is estimated as the average of the magnitudes $|G(x, y)|$ of the gradients of the pixels (x, y) used during between-slice interpolation.

3 Experiments

The data sets used for comparing the methods were a set of CT slices of a dry skull, a set of CT slices of a pacient head and a set of MRI slices of a brain specimen. Their description is given in Table 1. The CT data have been thresholded at 68 for the Skull-A, -200, -100 and 200 HU for Skull-B and the MRI data at 100 and 136. We used linear interpolation and four-point cubic spline interpolation [8]. Tables 2 and 3 show the results of applying linear between-slice interpolation to slices n-1 and n+1 and of comparing the outcome to the n-th slice and of applying modified cubic spline to slices n-3, n-1, n+1 and n+3 and comparing the outcome to the n-th slice.

One of 3D image analysis objectives is to estimate the volume of a specified object. While the accuracy of the volume computation is very much dependent on the segmentation process, the interpolation method applied to the segmented data also affect accuracy. We computed two comparison measurements in our experiments, the number of individual errors (Table 2) and the global percentage error on the volume estimates (Table 3). For each column, a bullet marks the overall best performance and a dagger marks the best performance either for linear or cubic spline between-slice interpolation. The equations of the individual error and volume error

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

a) original slices: upper and lower

	0.5	**-0.5**	**-1.5**	**-2.5**	**-3.5**	**-4.5**	**-5.5**	
	1.7	0.7	-0.2	-1.2	-2.2	-3.2	-4.2	
	3.0	2.0	1.0	0.0	-1.0	-2.0	-3.0	
	4.2	3.2	2.2	1.2	0.2	-0.7	-1.7	
	5.5	**4.5**	**3.5**	**2.5**	**1.5**	**0.5**	**-0.5**	

b) original shape-based interpolation (Euclidean distance)

	0.3	**-1.7**	**-4.3**	**-16**	**-9.6**	**-4.2**	**-3.8**	
	0.5	-1.1	-2.8	-10	-6.3	-3.1	-3.4	
	0.7	-0.4	-1.4	-4.7	-3.0	-2.1	-2.9	
	0.8	0.2	0.1	1.0	0.3	-1.0	-2.5	
	1.0	**0.9**	**1.6**	**6.7**	**3.6**	**0.1**	**-2.1**	

c) gray interpolation (Gray distance)

	0.8	**-2.2**	**-5.8**	**-18.5**	**-13.1**	**-8.7**	**-9.3**	
	2.2	-0.4	-3.0	-11.2	-8.5	-6.3	-7.6	
	3.7	1.6	-0.4	-4.7	-4.0	-4.1	-5.9	
	5.0	3.4	2.3	2.2	0.5	-1.7	-4.2	
	6.5	**5.4**	**5.1**	**9.2**	**5.1**	**0.6**	**-2.6**	

d) Euclidean-Gray shape-based interpolation

Figure 2: Illustrative one-dimensional example from [4]. Threshold = 136.

object	description	imager	pixel size	slice spacing	dimensions
Skull A	skull specimen	GE8800 CT	0.8 mm	3.0 mm	256 x 256 x 66
Skull B	patient skull	GE9800 CT	0.4 mm	1.5 mm	256 x 256 x 66
Brain	brain specimen	GESigma-MR	3.0 mm	3.0 mm	256 x 256 x 31

Table 1: Summary of the data sets used in the experiments

are given by:

$$ie = \sum_{i=1}^{n} |x_i - \hat{x_i}|$$

where:
ie is the individual error,
x_i is the real value of the $i-th$ voxel,
$\hat{x_i}$ is the estimated value of the $i-th$ voxel,
n is the total number of voxels.

$$\begin{aligned} \hat{v} &= \sum_{x,y,z} \hat{f}(x,y,z) \\ v &= \sum_{x,y,z} f(x,y,z) \\ ve &= \frac{\hat{v} - v}{v} \end{aligned}$$

where:
$\hat{v}$ is the estimated total volume,
v is the real total volume,
ve is the volume error,
$\hat{f}(x,y,z)$ is the estimated volume of the (x,y,z) voxel,
$f(x,y,z)$ is the real volume of the (x,y,z) voxel.

Although the performance on individual errors of the Euclidean shape-based interpolation is comparable with the results from Euclidean-Gray method, for volume estimation, the Euclidean-Gray method gave the best performance. We can note a tendency of overestimation of the volume for Euclidean method and an underestimation of the volume for the Gray method, or vice-versa. In this sense, as the Euclidean-Gray combines both methods, it was expected that its performance on volume estimates should be the best.

To better visualize the nature of the combination of Gray and Euclidean interpolations, we plot several graphics (Figures 3 to 6) showing the individual and volume errors at each slice for each data set. This information gives a better insight to understand the advantage of the Euclidean-Gray method. For volume estimation, the Euclidean-Gray method is always between the Gray and the Euclidean shape-based interpolation.

4 Computer graphic display of interpolated objects

We illustrate, in Figures 8 to 7 the effect on three-dimensional display of the interpolation methods. The images were produced using the visualization toolbox of Khoros [9,10]. For all images, the same programs were used for detecting the surfaces of binary objects and for displaying the detected surfaces using a particular variant of the so-called z-buffer gradient shading method. As in [4], we chose to use this shading method to reflect more accurately the shape of the surface of the interpolated binary object.

The Figure 7 is a zoom on the square of the figures 8 to 10, which are based on the dry skull specimen. Generally, the Figures confirm the behavior of the methods according to the one-dimensional case illustrated in Figure 2: the Euclidean shape-based interpolation tries to connect the contours by flat surfaces while the combined shape-based interpolation gives a blend of the Gray-level interpolation and the Euclidean shape-based interpolation.

Confirming the statistics results of Tables 2 and 3, the combination of the Gray and Euclidean shape-based interpolation gave the best performance for three-dimensional display of the interpolated objects.

5 Conclusions and Final Comments

We have presented techniques for interpolating 3D objects based on voxel representation. The techniques are the classical gray-level interpolation, the Euclidean shape-based interpolation and a combination of both introduced in 6. New results of the comparison evaluation amoung these methods are given in graphical form giving clues for further investigation in this area.

We intend to study and classify in detail the situation where the classical gray interpolation and the Euclidean shape-based interpolation perform best. Once identified those situations we can be able to use a weighting function based on this knowledge to better combine both interpolation methods.

6 Acknowledgments

Roberto A. Lotufo is supported by a CNPq research grant n. 500355-924-EE and Alexandre X. Falcão is supportted by a CNPq doctorate scholarship.

7 References

1. J. Udupa and G. Herman, *3D Imaging in Medicine*, Boca Raton, FL: CRC Press, 1991.

2. G. Herman and C. Coin, "The use of three-dimensional computer display in the study of disk disease", *J. Comput. Assist. Tomogr.* Vol. 4, pp. 564–567, 1980.

3. S. Raya and J. Udupa, "Shape-based interpolation of multidimensional objects", *IEEE Trans. Med. Imag.*, Vol. 9, pp. 32–42, 1990.

	Skull A - CT	Skull B - CT			Brain - MRI	
thr	68	-200	-100	200	100	136
linear interpolation						
gray	58.2	51.2	77.5	114	57.3	119
euclidean	53.9	45.1	62.6	88.3†	44.2†	97.3•
gray-euclidean	50.7†	40.7†	60.2†	89.1	45.4	103
cubic spline interpolation						
gray	56.8	48.6	77.9	111	57.2	117
euclidean	50.0•	41.6	59.9	84.0•	43.1•	97.6†
gray-euclidean	50.4	39.6•	59.9•	89.5	45.3	103

Table 2: Individual errors: (x 1000). A bullet marks the overall best performance and a dagger marks the best performance either for linear or cubic spline between-slice interpolation

	Skull A - CT	Skull B - CT			Brain - MRI	
thr	68	-200	-100	200	100	136
linear interpolation						
gray	0.89•	-0.7	-0.9	9.7	-1.9	-8.3
euclidean	-6.4	0.9	1.3	-3.4	2.4	4.3
gray-euclidean	-6.4	0.01•	0.1•	-0.4•	0.09•	-3.3†
cubic spline interpolation						
gray	5.7	-0.7	-1.2	10.4	-2.2	-7.9
euclidean	-0.97†	0.4	0.5	-1.9†	1.1	1.6•
gray-euclidean	2.3	-0.2†	-0.3†	4.0	-0.5†	-3.6

Table 3: Volume errors: (absolute %). A bullet marks the overall best performance and a dagger marks the best performance either for linear or cubic spline between-slice interpolation

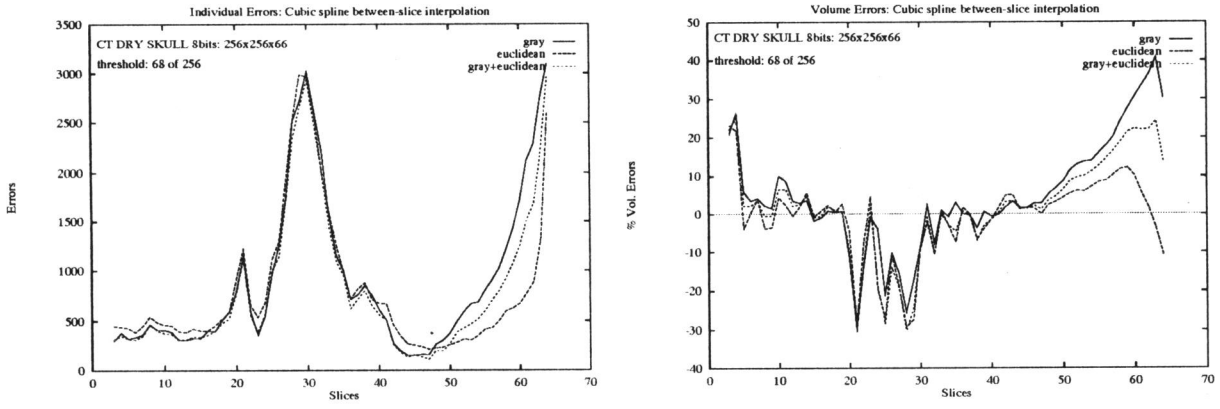

Figure 3: Skull A (dry skull) - Individual and Volume Errors, cubic spline between-slice interpolation using threshold 68.

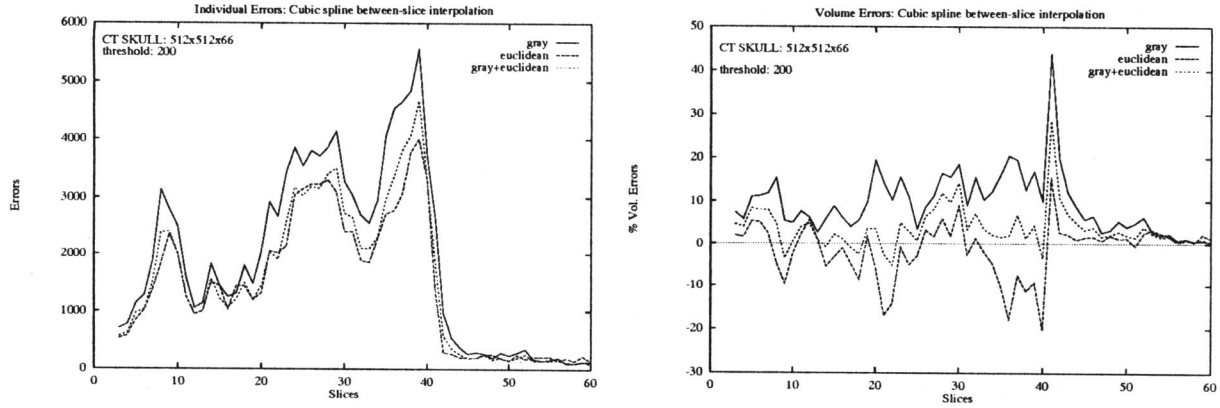

Figure 4: Skull B (bone) - Individual and Volume Erros, cubic spline between-slice interpolation using threshold 200.

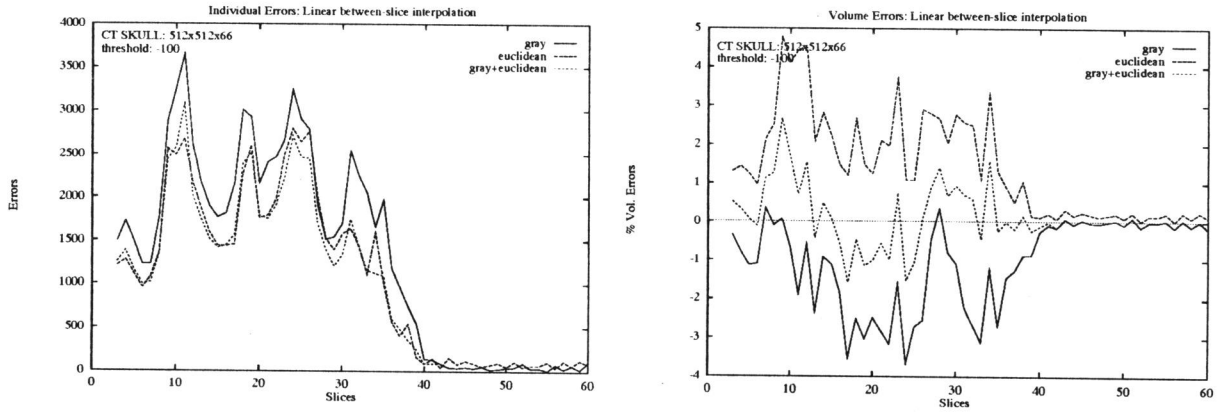

Figure 5: Skull B (skin) - Individual and Volume Errors, linear between-slice interpolation using threshold -100.

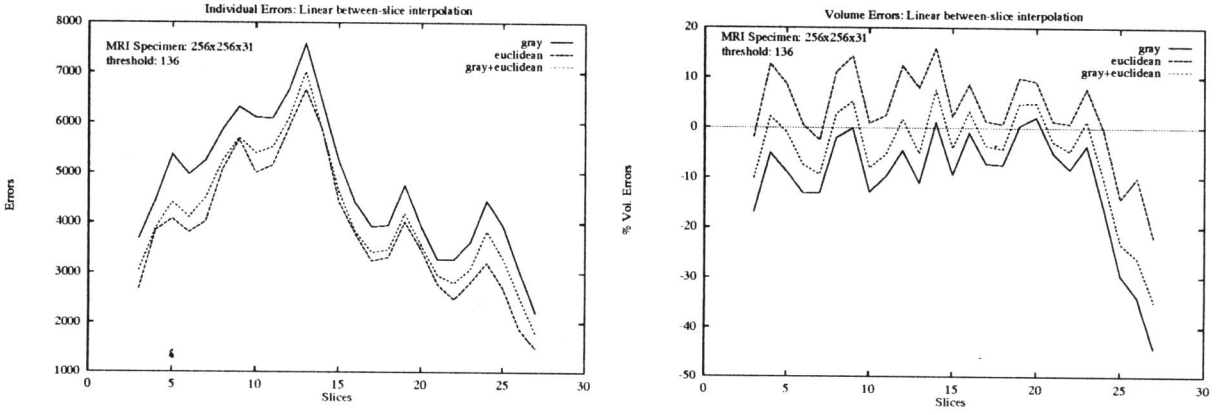

Figure 6: Brain - Individual and Volume Errors, linear between-slice interpolation using threshold 136.

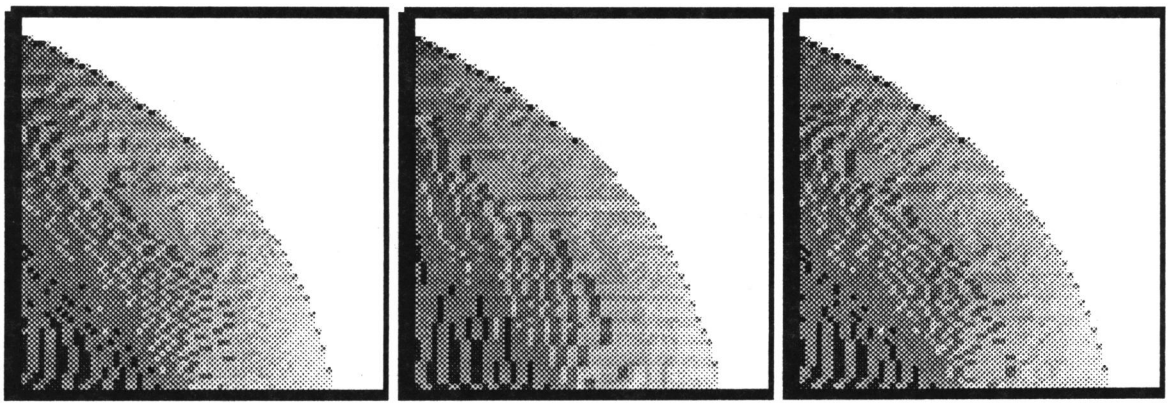

Figure 7: Skull A, zoom of the linear between-slice interpolation using: left is Gray distance, center is Euclidean distance and right is Euclidean-Gray distance.

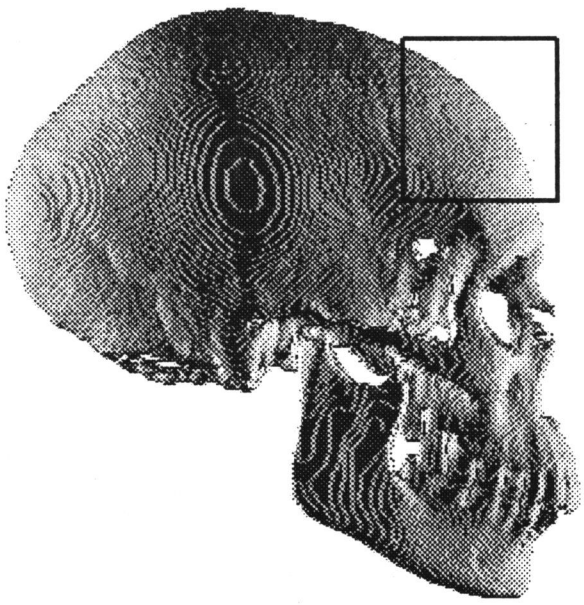

Figure 8: Skull A, linear between-slice interpolation using Gray distance.

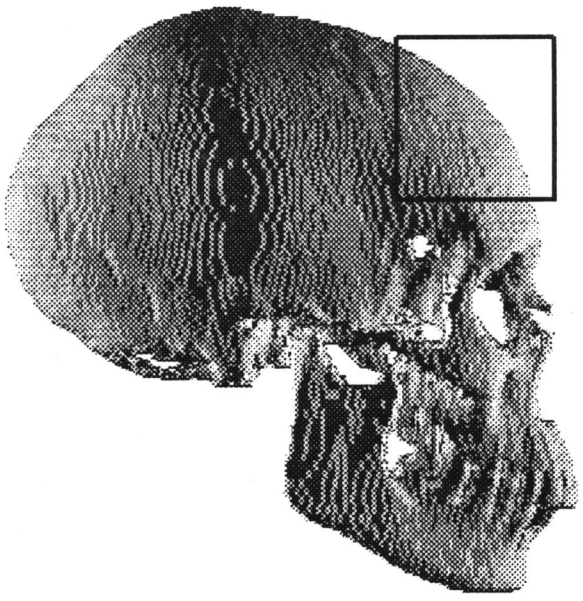

Figure 9: Skull A, linear between-slice interpolation using Euclidean distance.

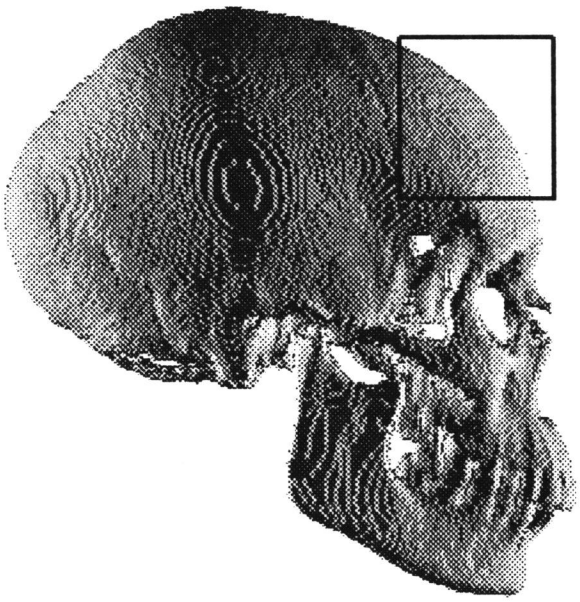

Figure 10: Skull A, linear between-slice interpolation using Euclidean-Gray distance.

4. G. Herman, J. Zheng and C. Bucholtz, "Shape-based interpolation", *Computer Graphics & Applications*, Vol. 12(3), pp. 69–79, 1992.

5. W. Higgins, C. Morice and E. Ritman, "New shape-based interpolation technique for three-dimensional images", *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, (Albuquerque, NM), pp. 1841–1844, 1990.

6. R. Lotufo, G. Herman and J. Udupa, "Combining Gray-level into Shape-based Interpolation", *Visualization in Biomedical Computing 1992 (VCB'92)*, (Chapel Hill, North Carolina, USA), 13-16 Outubro 1992.

7. G. Borgefors, "Distance transformations in digital images", *Comput. Vision Graph. Image Proc.*, Vol. 34, pp. 344–371, 1986.

8. J. Foley, A. Van Dam, S.K. Freiner and J.F.Hughes, "Fundamentals of Interactive Computer Graphics", 2nd edition, Addison Wesley Publishing Company, 1990.

9. R. Lotufo, A. Falcão, "Extending the Khoros to support 3D-Imaging in Biomedical Computing", Boston Visualization, Khoros workshop, Boston, 19-23 Outubro 1992.

10. A. Falcão, "Volume Visualization Applied to Medicine", MSc Thesis, FEE-Unicamp, Feb. 1993 *in Portuguese.*