BIOMEDICAL DATA INTERPOLATION FOR 3-D VISUALIZATION

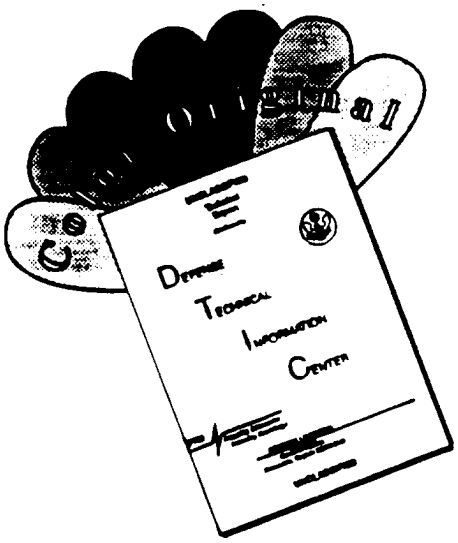
THESIS
Ming-Chung Chen
Major,ROCAF

AFIT/GCS/ENG/95J-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

AFIT/GCS/ENG/95J-01

BIOMEDICAL DATA INTERPOLATION FOR 3-D VISUALIZATION

THESIS
Ming-Chung Chen
Major,ROCAF

AFIT/GCS/ENG/95J-01

Approved for public release; distribution unlimited

AFIT/GCS/ENG/95J-01

BIOMEDICAL DATA INTERPOLATION FOR 3-D VISUALIZATION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Computer Science

Ming-Chung Chen, B.S.
Major,ROCAF

June, 1995

Acknowledgements

I would like to dedicate this work to my parents. As with all of my endevaors, their love and support sustains me. Special thanks go to my wife Li-Huang. Thank you for supporting me through all this time.

I would like to thank my advisor Dr. Steven Rogers. He gave me invaluable support and guidance. His positive outlook and constant encouragement helped me remain motivated through this research. Thanks also go to Dr. Marty Desimio and Dr. David Fulk for their crucial technical guidance.

Finally, I would like to thank Capt. Steve Matechi, Capt. Curtis Martin, and Capt. Terry Wilson for their help and friendship. Without their help, my work would have been much more difficult.

Ming-Chung Chen

Table of Contents

List of Figures

List of Tables

AFIT/GCS/ENG/95J-01

Abstract

Medical imaging devices that produce three-dimensional data usually produce the data in the form of image slices. In such images, the resolution in z direction is lower than in x and y directions. Before extracting and displaying objects in such images, an interpolated 3-D gray-scale volume image can be generated via image interpolation techniques to fill in the missing information.

The subject of this thesis is the applying three different interpolation techniques to generate intermediate slices and comparing their qualities. The three interpolation techniques are linear interpolation, cubic spline interpolation, and Fourier interpolation. We also apply the CT image matching method, developed by Ardeshir Goshtasby, David A. Turner, and Laurens V. Ackerman, which can determine the correspondence between points in two images. Finally, we use the human visual perception model to measure the qualities of interpolation images.

Linear interpolation is shown to be the best of the three interpolation techniques used in this thesis. This research also shows that without the image segmentation or the image matching process poor intermediate images will be generated.

BIOMEDICAL DATA INTERPOLATION FOR 3-D VISUALIZATION

I. INTRODUCTION

1.1 Introduction

In medical imaging, there are many motivations for the development of visualization tools and volume modelling techniques for three-dimensional image (3D) data. 3D medical images provide views of internal organs, bones, tissues, muscles and other body parts otherwise only seen during exploratory or therapeutic surgery. For example, because of the additional insight revealed by 3D images, a surgeon might decide not to perform surgery (26).

Medical imaging devices usually produce data in the form of image slices. In this case, 2D image slices often do not provide enough information for the surgeon to make an accurate assessment of the patient's condition. In order to use 2D image slices to render 3D objects, an interpolated 3D gray-scale volume image can be generated via interpolation. The Computed Tomography (CT) slice matching method and the application of different methods of data interpolation to biomedical slice image data are the subject of this thesis. Linear interpolation, cubic spline interpolation and Fourier interpolation are the three methods used in the research.

When an image is processed, people always like to know the quality of the image. The human visual system is the transformation of the input visual stimulus into a perception. The ability of the human visual system to detect detail in a scene is determined by the contrast characteristics of the scene (3). One way to quantitatively describe some characteristics of the visual system is in terms of the contrast sensitivity of the system for sinusoidal luminance profiles. Since the contrast

sensitivity is a measure of how the human visual system responds to visual stimuli, it will provide better information on the quality of biomedical images. The model developed by Captain Wilson (27) will be used in this thesis.

1.2 Problem Statement

This thesis will compare three interpolation methods : Linear, cubic spline and Fourier interpolation for use in biomedical data interpolation for 3-D visualization.

1.3 Scope and Assumptions

The data set used in this research is Computed Tomography (CT) head data. The data was taken on the General Electric CT scanner and provided courtesy of the North Carolina Memorial Hospital. It consists of 113 slices. Each image and is a 256 x 256 array. The total data volume is z-113 y-256 x-256.

The scope of this thesis will also investigate the use of the matching method algorithms, developed by Ardeshir Goshtasby, David A. Turner, and Laurens V. Ackerman (2), which can determine the correspondence between points on region boundaries. To determine the effectiveness of the matching method and the quality of the interpolation slices, the contrast sensitivity of human visual perception (27) will be used in this research.

In this thesis, we assume the lower slice of two slices is a geometrically deformed version of the upper image with some added noise. The process of determining the corresponding points in two consecutive images is required for interpolation. One intermediate image will be generated between each two consecutive slices. In order to simplify the matching method, we also assume that the positions of the interpolated pixels are the same as on the upper slice.

1.4 Approach/Thesis Organization

The next chapter provides background information describing characteristics of CT imaging. The chapter also includes a discussion on the slice matching method along with different interpolation methods, and an introduction on contrast sensitivity of human visual perception. Chapter III details the algorithms of the slice matching method, the three different interpolation methods, and contrast sensitivity used in this thesis. Chapter IV describes the result of interpolating images on the CT data set. The conclusion and recommendations are discussed in chapter V.

II. BACKGROUND

2.1 Introduction

This chapter will provide an introduction to Computed Tomography (CT) images. Specifically, it will provide the background material necessary to understand the interpolation algorithms, image matching algorithm (2), and the human visual perception model used in this thesis.

2.2 Computed Tomography (CT) Image

Computed Tomography (CT) is an X-ray technique that overcomes the contrast problems encountered with normal radiography. CT produces an image that is essentially a picture of a slice of the body. In the late 1960s, the first extensive research on computed tomography (CT) was performed by Geoffery Hounsfield (20). This laboratory machine took many days to acquire data and more than a couple of hours to reconstruct the image. Now a single slice acquisition may take only one second with dramatically improved image quality.

An engineering drawing of a typical CT scanner (one built by the General Electric Company) is shown in Figure 2.1. The patient lies on the table, and the sliding top moves into the hole of the gantry, which in turn houses the X-ray tube and collimator on one side and the data acquisition unit on the other side. The collimator limits the X-ray beam to the body section under scrutiny. The detector unit contains detectors arranged on an arc of a circle centered at the X-ray source. X-rays travel along straight lines between the source and the detectors. From the strength of the X-ray beam reaching the detector, the total X-ray attenuation along the line between the source and the detector can be estimated (20).

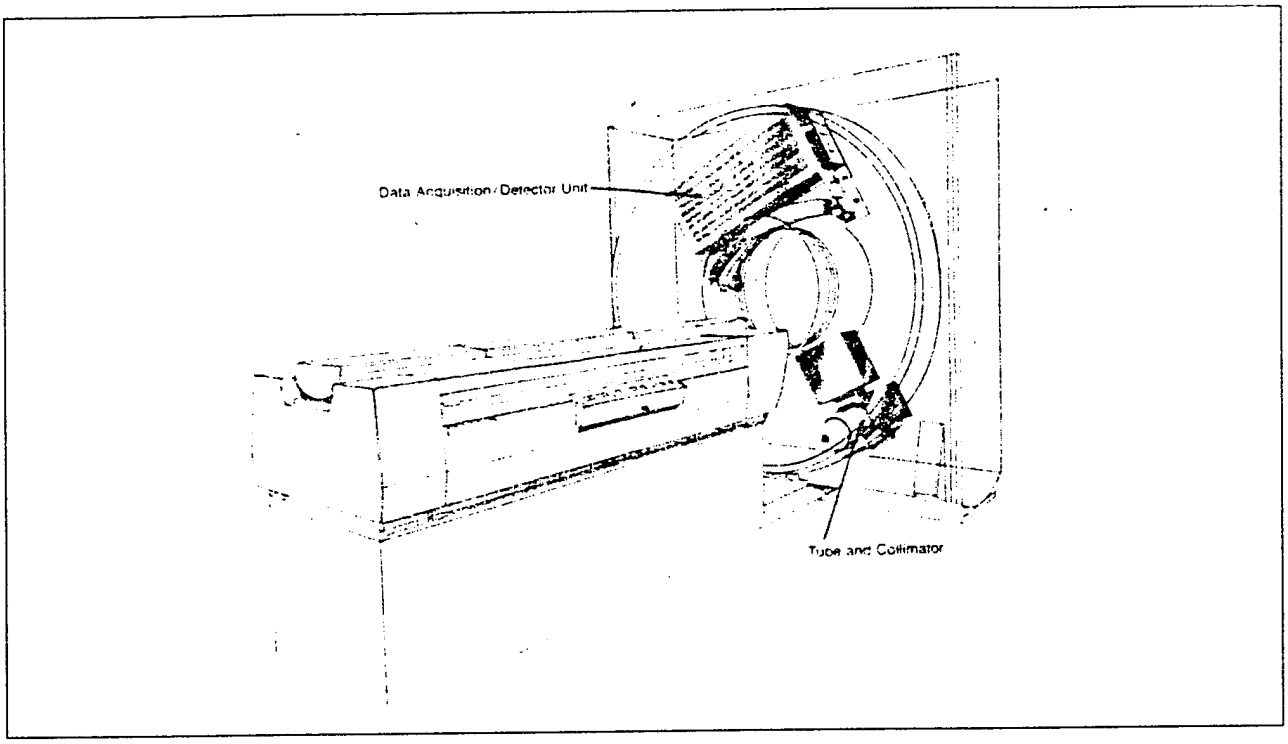


Figure 2.1 Engineering drawing of a typical CT scanner (20).

Figure 2.2 shows the scheme of an elementary CT brain scan. For each scan, the tube and pickup will start at one end and move parallel to the object. Next, the assembly will rotate about 1^0 and again move from the start to the finish of that scan and repeat this procedure until it has scanned 180 positions. There have been several generations of CT scan machines using not only a single detector but using detector arrays, which significantly decrease the required scan time (15).

In order for us to be able to see the CT images, two important procedures have to be executed : Data collection and Image reconstruction. Data collection is specific to scanner type. In data collection and image reconstruction, some of the basic physical, engineering, and mathematical principles of computed tomography

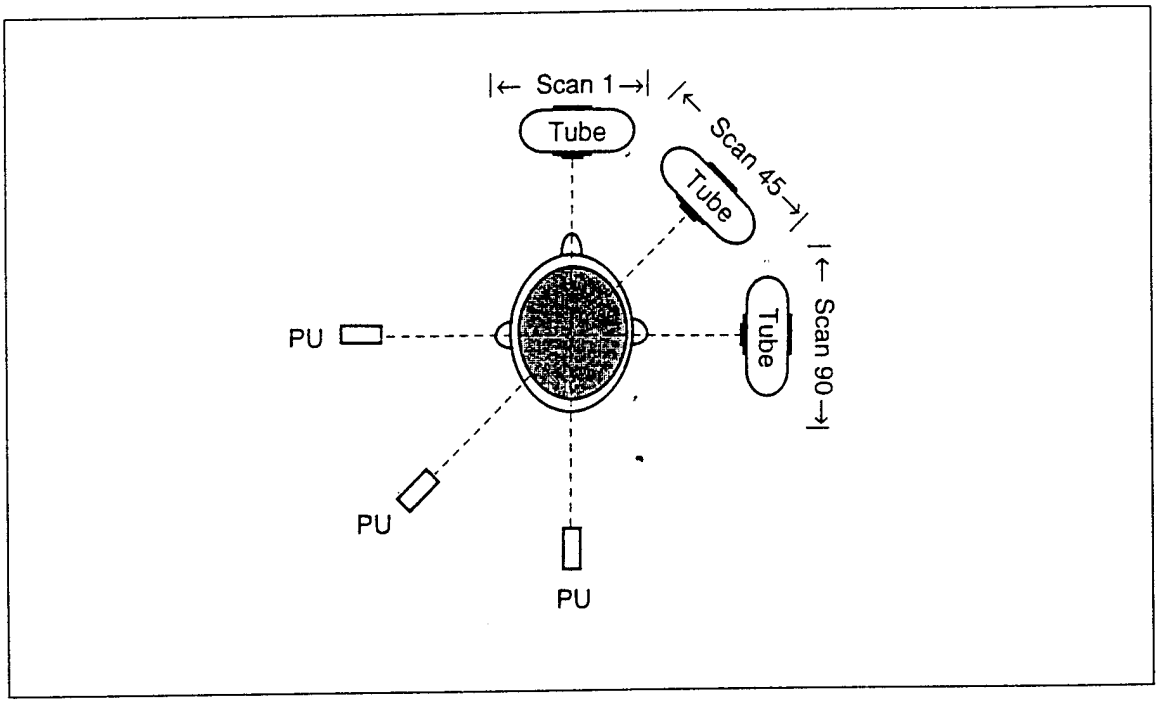


Figure 2.2 The scheme of an elementary CT brain scan. This graphic only shows representative positions for the 0^0, 45^0, and the 90^0 scans.

have been used (20). Both successful sampling and insufficient sampling images will be illustrated in the next chapter.

2.3 Interpolation Methods

Interpolation is the process by which one approximates a discrete function, given on a finite set of samples (11). We can extend the finite set to as an infinite set by using the interpolated data and interpolate again.

Medical imaging devices usually produce the data in the form of 2D image slices. Unfortunately, a 3D image generated by a typical scanner tends to have a slice spacing greater than the spacing of individual samples on a slice. Hence, we use interpolation technique to fill in the missing slices.

2.3.1 Linear Interpolation. The simplest kind of interpolation is linear interpolation. People who use tables of mathematical functions should be familiar with the method of linear interpolation. This due to the fact that they have to use this method to "read between the lines of the table" and to get the value they need (13, 25, 14).

The assumption that a function $f(x)$ is approximately linear, in a certain range, is equivalent to the assumption that the ratio

$$\frac{f(x_1) - f(x_0)}{x_1 - x_0} \tag{2.1}$$

is approximately independent of x_0 and x_1 in that range. This ratio is called the first difference of $f(x)$, relative to x_0 and x_1, and we may designate it by $f[x_0, x_1]$. That is,

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \tag{2.2}$$

It is clear that $f[x_1, x_0] = f[x_0, x_1]$.

Thus the linear approximation may be expressed in the form

$$f[x_0, x] \approx f[x_0, x_1] \tag{2.3}$$

this leads to the interpolation formulas for $x \in [x_0, x_1]$.

$$f(x) \approx f(x_0) + (x - x_0) f[x_0, x_1] \tag{2.4}$$

or

$$f(x) \approx f(x_0) + \frac{x - x_0}{x_1 - x_0}[f(x_1) - f(x_0)] \tag{2.5}$$

or, equivalently, to the formula

$$f(x) \approx \frac{1}{x_1 - x_0}[(x_1 - x)f(x_0) - (x_0 - x)f(x_1)], \tag{2.6}$$

which can also be expressed in the convenient determinant form.

$$f(x) \approx \frac{1}{x_1 - x_0} \begin{vmatrix} f(x_0) & x_0 - x \\ f(x_1) & x_1 - x \end{vmatrix} \tag{2.7}$$

This form is particularly well adapted to machine computation, since its evaluation involves the continuous operation of a cross product followed by a division.

As a numerical example, the linear interpolation of $\sinh x$ for x = 0.23, from tabulated values for $x_0 = 0.20$ and $x_1 = 0.30$, may be arranged as follows :

x_i	$f(x_i)$	$x_i - x$
0.20	0.20134	-0.03
0.30	0.30452	0.07

$$f(0.23) \approx \frac{(0.07)(0.20134) - (-0.03)(0.30452)}{0.10} = 0.23229$$

It is useful to notice that a linear interpolation merely provides a *weighted average* of the two ordinates involved.

2.3.2 Cubic Spline Interpolation. A *spline* is a thin rod of some elastic material equipped with a groove and a set of weights (called ducks or rats) with attached arms designed to fit into the groove. The device is used by architects (particularly naval architects) to draw smooth curves passing through prescribed points. To accomplish this, the spline is forced to pass through the prescribed points

by adjusting the location of the ducks along the rod. The spline was discovered in the mid-1700s by Euler and the Bernoulli brothers (24).

Cubic spline functions are now used widely in computer calculations for the interpolation of continuous functions of one variable. The cubic spline function is composed of a set cubic polynomials defined on different intervals. A general cubic polynomial involves four constants. Giving sufficient flexibility in the cubic spline procedure to ensure not only that the interpolant is continuously differentiable on the interval, but that it has a continuous first and second derivative on the interval as well. The cubic spline interpolation is considered interpolation by cubic splines to data, where the cubic polynomial pieces meet at that data point (19, 21, 22). The following algorithm is quoted from (21).

Suppose we have $m + 1$ data points $P_0, ..., P_m$ through which we wish to draw a curve such as that shows in Figure 2.3 (in which m = 6).

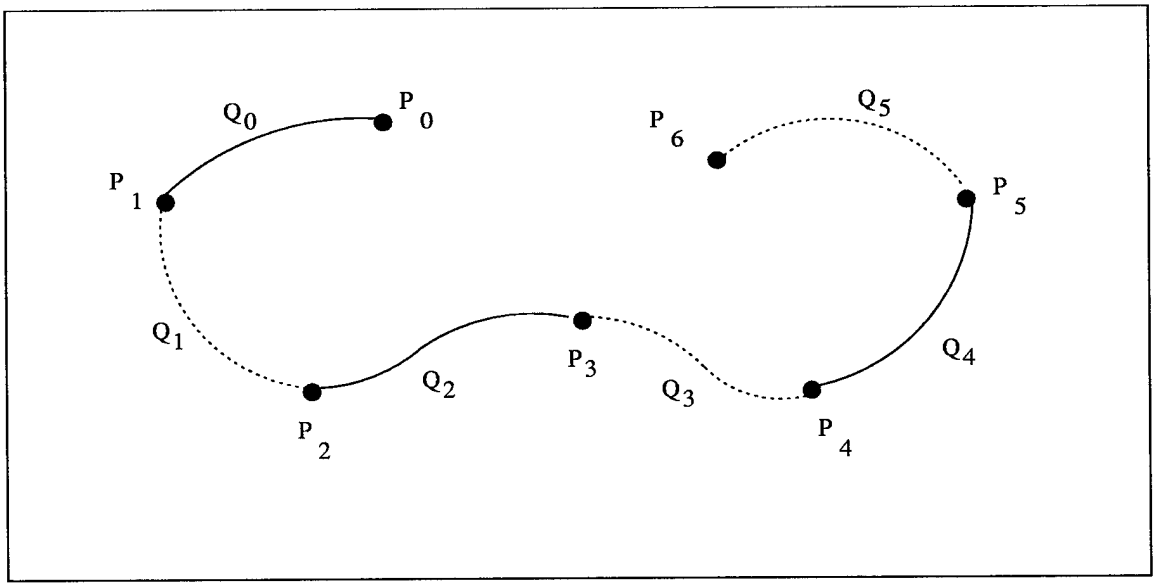


Figure 2.3 An interpolating cubic spline.

Each successive pair of data points is connected by a distinct curve segment, Q_i. The i^{th} segment runs from P_i to P_{i+1}, and we assume that a parameter $\bar{u}$ runs

correspondingly from knot $\bar{u}_i$ to the knot $\bar{u}_{i+1}$ to generate this segment. Each such segment $Q_i(\bar{u})$ is represented parametrically as $(X_i(\bar{u}), Y_i(\bar{u}))$ and the $X_i(\bar{u})$ and $Y_i(\bar{u})$ are determined by the points $P_i = (x_i, y_i)$.

In general, the x-coordinates $X(\bar{u})$ of points on a curve are determined solely by the x-coordinates $x_0, ..., x_m$ of the data points, and similarly $Y(\bar{u})$ is determined solely by the y-coordinates of the data points. Since both $X(\bar{u})$ and $Y(\bar{u})$ are treated in the same way we could discuss only $Y(\bar{u})$; indeed, to obtain curves in three dimensions we can simply define a $Z(\bar{u})$ as well and let $Q_i(u)$ be given by $(X_i(u), Y_i(u), Z_i(u))$. If $X(\bar{u})$, $Y(\bar{u})$, and $Z(\bar{u})$ are independent and determined solely by x, y, and z coordinates, we also can discuss only $Z(\bar{u})$.

For the ease of computation, we might limit our parametric equations to the use of polynomials in defining $X_i(u)$, $Y_i(u)$, and $Z_i(u)$. Then, $Y(\bar{u})$ is shown in Figure 2.4.

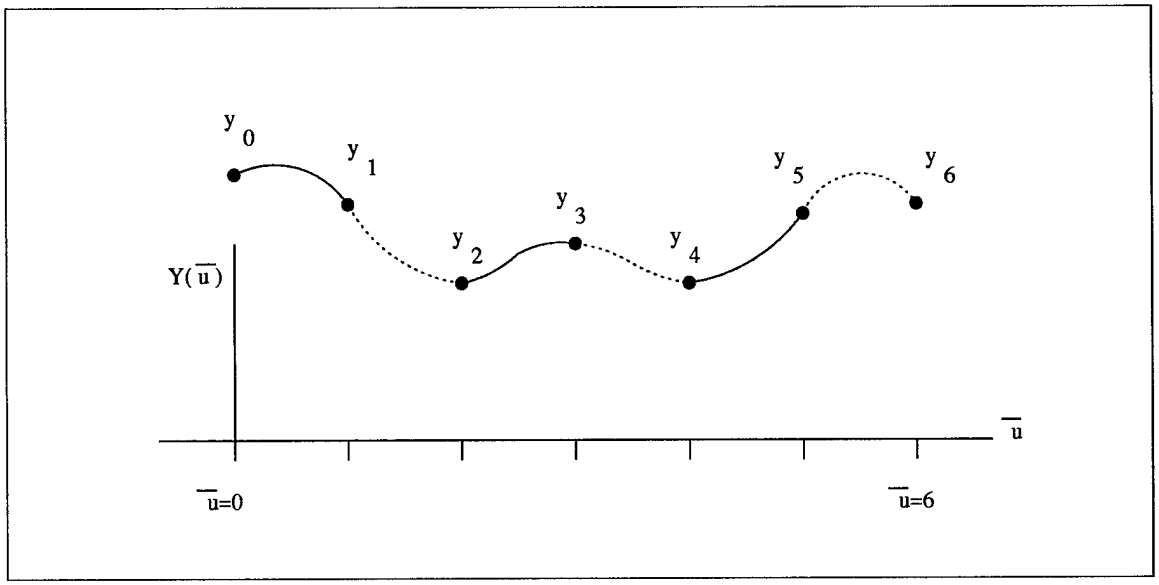


Figure 2.4 $Y(\bar{u})$ for the curve shown in Figure 2.3. In this diagram we choose to use uniform knot spacing, and the knot sequence is (0,1,2,3,4,5,6).

We can also reparameterize each segment Y_i separately by substituting u for $\bar{u}$. This means that $u = \bar{u}_i - i$ for the knot sequence given in Figure 2.3. Each $Y_i(u)$ is a cubic polynomial in the parameter u,

$$Y_i(u) = a_i + b_i u + c_i u^2 + d_i u^3 \tag{2.8}$$

Because Y_i interpolates the data at the end points of the segment we know

$Y_i(0) = y_i = a_i$

and

$Y_i(1) = y_{i+1} = a_i + b_i + c_i + d_i$

Because we have four coefficients to determine, we need two other constraints to completely determine a particular $Y_i(u)$. One easy way to do this is to simply pick, arbitrarily, first derivatives D_i of $Y(u)$ at each knot $\bar{u}_i$, so that

$Y_i^{(1)}(0) = D_i = b_i$

$Y_i^{(1)}(1) = D_{i+1} = b_i + 2c_i + 3d_i.$

These four equations can be solved symbolically to yield

$a_i = y_i$

$b_i = D_i$

$c_i = 3(y_{i+1} - y_i) - 2D_i - D_{i+1}$

$$d_i = 2(y_i - y_{i+1}) + D_i + D_{i+1}$$

Since we use D_i as the derivative at the left end of the i_{th} segment (i.e., as $Y_i^{(1)}(0)$) and at the right end of the $(i-1)^{th}$ segment (as $Y_{i-1}^{(1)}(1)$), $Y(u)$ has a continuous first derivative.

It is possible to arrange for successive segments to match the second as well as the first derivatives at joints, using only cubic polynomials. Suppose that we want to interpolate the $(m+1)$ points $P_0, ..., P_m$ by such a curve. Each of the m segments $Y_0(u), ..., Y_{m-1}(u)$ is a cubic polynomial determined by four coefficients. Hence we have $4m$ unknown values to determine. At each of the $(m-1)$ interior knots $\bar{u}_1, ..., \bar{u}_{m-1}$ (where two segments meet) we have four conditions :

$$Y_{i-1}(1) = y_i$$

$$Y_{i-1}^{(1)}(1) = Y_i^{(1)}(0)$$

$$Y_i(0) = y_i$$

$$Y_{i-1}^{(2)}(1) = Y_i^{(2)}(0).$$

Since we also require that

$$Y_0(0) = y_0$$

$$Y_{m-1}(1) = y_m$$

we have a total of $4(m-1) + 2 = 4m - 2$ conditions from which to determine our $4m$ unknowns. A common choice to complete the system is simply to require

that the second derivatives at the endpoints $\bar{U}_0$ and $\bar{U}_m$ both be zero. This is called a natural cubic spline and it was used in this thesis.

2.3.3 Fourier Interpolation. Polynomials are usually the most convenient functions for the approximation and interpolation of a continuous function when the approximation interval is finite. Polynomials are also adapted to the approximation and interpolation of periodic functions over relatively short ranges. When a function $f(x)$ is periodic and is to be approximated over one or more complete periods, it is desirable to make use of periodic coordinate functions, have monically related to $f(x)$, in constructing an approximation or interpolation. The most convenient set of such functions is the composite of all sines and cosines which possess integer multiples of that period. The Fourier interpolation applies this principle.

Many common operations on images are accomplished in the frequency domain. There are several reasons for this. One reason is that, for certain manipulations, it is far more efficient this way, particularly in certain kinds of filtering (e.g. convolution, low-pass spatial filter, high-pass spatial filter, etc). To simplify and make the processing easy to understand is another reason. The Fourier transform is an efficient method to transform a digital signal from the spatial domain to the frequency domain and it can simplify the manipulations of image processing. For example it can be used to compute the distribution energy for different frequency components of an image (4). We will use the Fourier interpolation in frequency domain.

The Fourier interpolation system can be expounded by a general system for increasing the sampling rate. Figure 2.5 shows a system to obtain samples $x_i[n]$ from original $x[n]$ using discrete time processing. For L-fold interpolation, first we create $x_e[n]$ by inserting (L-1) zeros samples of $x[n]$, and then we use a lowpass filter to "fill in" the zeros of $x_e[n]$ (16).

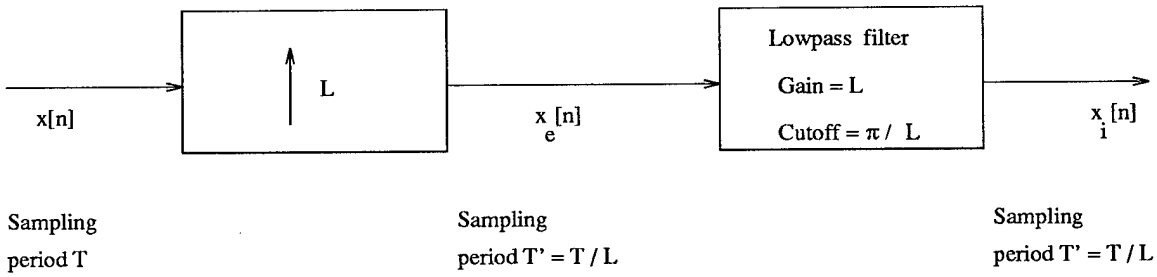


Figure 2.5 General system for sampling rate increase by L.

The system on the left is called an expander (7). Its output is

$$x_e[n] = \begin{cases} x[n/L], & \text{n } = 0, \pm L, \pm 2L, ..., \\ 0 & \text{otherwise,} \end{cases} \tag{2.9}$$

or, equivalently,

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL]. \tag{2.10}$$

The system on the right is a lowpass discrete-time filter with cutoff frequency π/L and gain L. The lowpass filter passes on lower frequency components of an image, while attenuating or rejecting the higher frequency components.

The Fourier transform of $x_e[n]$ in Figure 2.5 can be expressed as

$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} (\sum_{k=-\infty}^{\infty} x[k]\delta[n - kL]e^{-j\omega n})$$

$$= \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega Lk}$$

$$= X(e^{j\omega L}) \tag{2.11}$$

Thus the Fourier transform of the output of the expander is a frequency-scaled version of the Fourier transform of the input, i.e., Ω is the continuous-time frequency variable. ω is replaced by ωL so that ω is now normalized by

$$\omega = \Omega T'. \tag{2.12}$$

This effect is illustrated in Fig.2.6 with $L = 2$. Figure 2.6(a) shows a bandlimited continuous-time Fourier transform, and Fig.2.6(b) shows the discrete-time Fourier transform of the sequence $x[n] = x_c(nT)$, where $\pi/T = \Omega_N$. Figure 2.6(c) shows $X_e(e^{j\omega})$ according to Eq. (2.11), with $L = 2$, and Fig.2.6(e) shows the Fourier transform of the desired signal $x_i[n]$. We see that $X_i(e^{j\omega})$ can be obtained from $X_e(e^{j\omega})$ by correcting the amplitude scale from $1/T$ to $1/T'$ and by removing all the frequency-scaled images of $X_c(j\Omega)$ except at integer multiples of 2π. As shown in Figure 2.6(d) this requires a lowpass filter with a gain of 2 and cutoff frequency $\pi/2$. In general, the required gain would be L, since $L(1/T) = [1/(T/L)] = 1/T'$, and the cutoff frequency would be π/L.

This example shows that if the sampling without aliasing then the system of Figure 2.5 does give an output that fills in the missing samples. The system is therefore called an *interpolator*, and the operation of unsampling is considered to be synonymous with *interpolation*.

2.4 Image Matching Method

In the previous section we discussed three different interpolation methods employed to fill the missing slices. Before these methods can be successfully employed, there is an important problem to be addressed. That is the question of searching for the corresponding pixels in adjacent slices. Because a tissue may bend, shrink, expand, or disappear from one slice to the next, pixels belonging to a tissue in one slice do not necessarily connect to pixels directly beneath them in the next slice. A

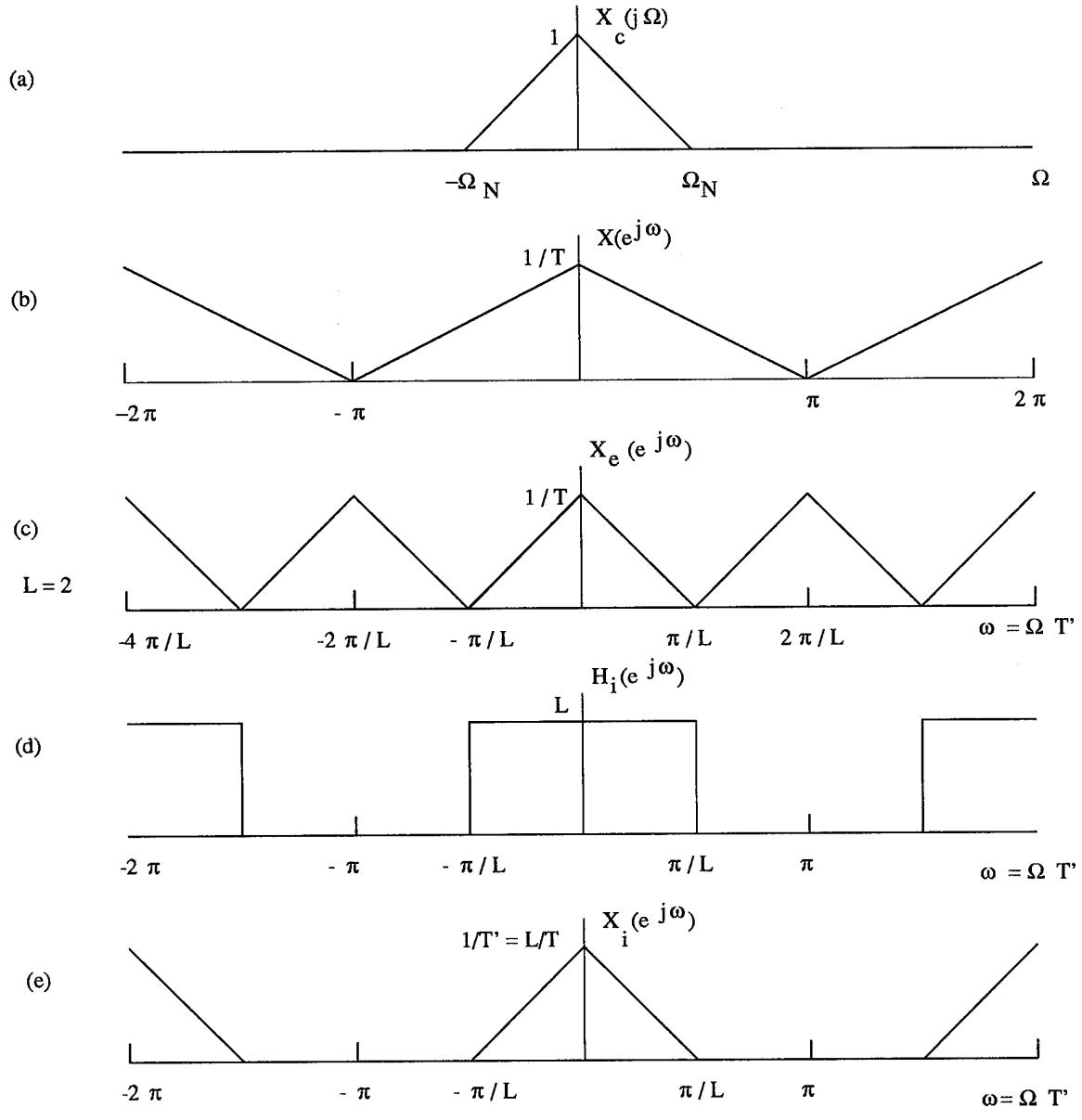


Figure 2.6 Frequency domain illustration of interpolation. (a) shows a bandlimited continuous-time Fourier transform. (b) shows for $L = 2$ the discrete-time Fourier transform. (c) shows the Fourier transform of $x_e[n]$. (d) shows the lowpass filter. (e) shows the Fourier transform of the desired signal $x_i[n]$.

typical image matching method (10) selects a set of feature points from each image, determines the correspondence between the feature points, and uses the feature correspondences to determine a transformation function that can map one image into another. Selection of the same feature points in two images is a very difficult task. This step is usually assisted by a human operator who selects a set of feature points in one image and determines the corresponding feature points in the other image by a cross-correlation template-matching process (1, 9). In this section we are going to describe the CT slices matching algorithm, developed by Ardeshir Goshtasby, David A. Turner, and Laurens V. Ackerman (2). The feature point selection and correspondence is achieved without segmentation (17) of the images. It uses all pixels with gradient magnitudes above a threshold value as feature points.

2.4.1 Synopsis. In this algorithm, the term "matching" means the process that determines the correspondences. Match and mismatch, respectively, refer to two points that are correctly selected as corresponding points, and two points that are mistaken by selected as corresponding points. Figure 2.7 shows two consecutive slices and the generated intermediate slices to obtain isotropic volume data. The upper slice is called the *reference image* and the lower slice is called the *target image*. The intermediate slices are generated by establishing correspondence between points in the reference and target images and determining the intersection of lines that connect the corresponding points with the intermediate slices. In this illustration, if tissue point A and tissue point B correspond to each other, the intersections of line AB with the intermediate slices trace the tissue in 3-D from A to B. The intensity of the point of intersection of line AB with an intermediate slice is determined by the linear interpolation of intensities of points A and B. Similarly, if we use cubic spline interpolation or Fourier interpolation, the only difference is the pixels between points A and B will fit different functions.

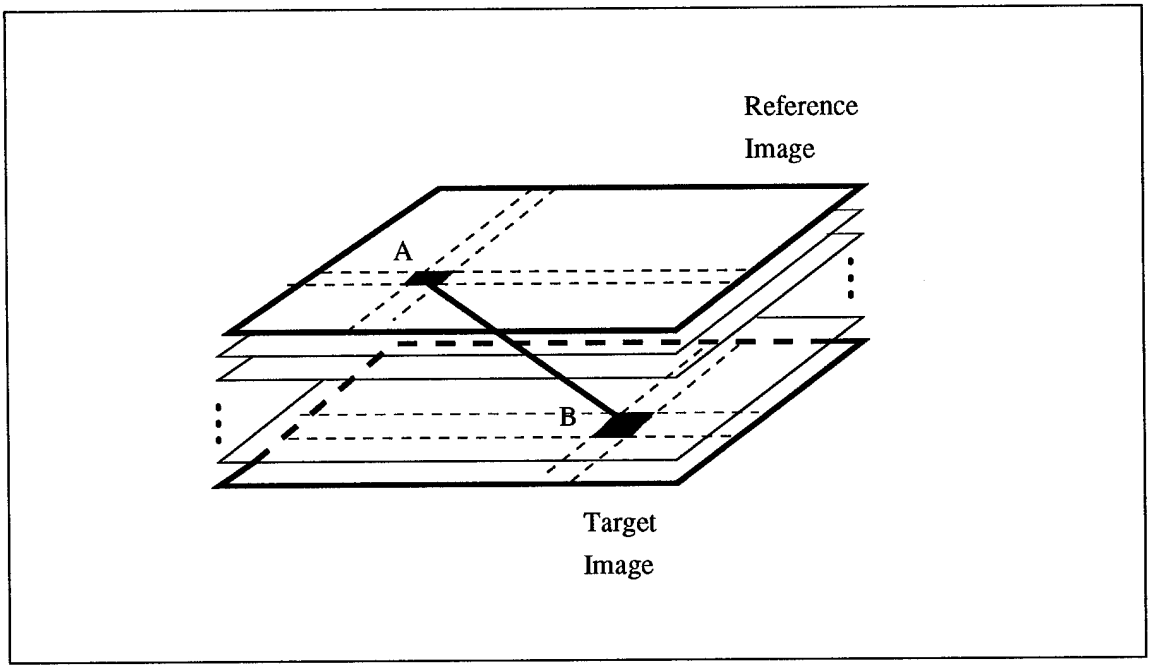


Figure 2.7 Two consecutive slices and the generated intermediate slices to obtain isotropic volume data.

There are some special properties of tomographic images which allow us to establish correspondence between points in the images without any operator assistance. Tomographic images have the following properties in common:

1. Image slices do not have scaling differences, and have very small (due to patient motion) or no translational and rotational differences.
2. The correspondence to a point in the reference image lies in a small area centered at the same coordinates in the target image.
3. There is continuity in the correspondences. That is, neighboring points in the target image map to neighboring points in the reference image. A point in the target image may map to several points in the reference image due to shrinkage of a tissue, or several points in the target image may map to one point in reference image due to expansion of a tissue; however, continuity is preserved in the correspondences.

4. The geometric difference between consecutive slices is local. Therefore, a single global transformation function cannot map the images to each other, especially when dynamic images such as those of a beating heart are used in the matching. Mapping should take place between local areas in the images.

2.4.2 The Matching Process. If we imagine that a target image is drawn on an elastic membrane which can be stretched or shrunk in any direction, then the job of the image matching process is to deform the target image to overlay with the reference image. To determine the deformation, for each point in the reference image a correspondence is searched for in a small neighborhood beneath it in the target image. The size of the neighborhood can be determined by the maximum expected displacement between corresponding points in the images. The displacement between corresponding points in the images is referred as *disparity* between the points.

Correspondence is established between points in two images using the intensities and gradients of the points. The *gradient* is the most commonly used method of differentiation in image processing. There is a vector cost function defined for matching a point (x, y) in the target image to a point (x', y') in the reference image.

$$\begin{aligned} C(x,y,x',y') &= u_1[I(x,y) - I'(x',y')]i \\ &+u_2[D(x,y) - D'(x',y')]j \\ &+u_3[\theta(x,y) - \theta(x',y')]k \\ &+u_4\sqrt{(x-x')^2 + (y-y')^2}l \end{aligned}$$

$I(x,y)$, $D(x,y)$, and $\theta(x,y)$ are, respectively, the intensity, gradient magnitude, and gradient direction of point (x,y) in the target image; and $I'(x',y')$, $D'(x',y')$, and $\theta'(x',y')$ are,respectively, the intensity, gradient magnitude, and gradient direction of point (x',y') in the reference image.

Since the gradient direction of two points cannot differ by more than π, in the cost function if $[\theta(x,y) - \theta'(x',y')] > \pi$, it is replaced by $2\pi - [\theta(x,y) - \theta'(x',y')]$

and if $[\theta(x,y) - \theta'(x',y')] < -\pi$, it is replaced by $2\pi + [\theta(x,y) - \theta'(x',y')]$. The cost function is a four-dimensional vector. Its magnitude is used to measure the dissimilarity between points (x,y) and (x',y'). This dissimilarity measure depends on the intensities and gradients of points being matched, and on the disparity between them. The fourth term in the cost function is used to favor the correspondences that have smaller disparities over the ones that have larger disparities. It is likely that a point in the reference image connects to a point in its immediate neighborhood in the target image rather than to a point further away.

The parameters u_1, u_2, u_3, and u_4 are weights that specify the relative importance of the intensity, gradient magnitude, gradient direction, and disparity in determination of the correspondences. When the difference in intensities of images to be matched is small, intensities can be used to match the images; therefore u_1 should be given a higher value than when intensities in the images differ considerably. The gradients in an image depend not only on the absolute intensities, but on the spatial arrangement of the intensities as well. If it is known that the images have similar intensity variations, then u_2 and u_3 should be given higher values than when the images have spatial differences. For example, if one image is a smoother version of another image, then u_2 and u_3 should be given smaller values than when both images have the same spatial resolution. For slices that have similar intensities and gradients, preference should be given to the one that has a smaller disparity value. This is because it is more likely that a tissue point in one image connects to a tissue point in its immediate neighborhood in the next image, than for it to bend and connect to a tissue point far away.

To determine the point (x,y) in the target image that corresponds to point (x',y') in the reference image, a small search window is selected, centered at (x',y') in the target image. Suppose the window contains n pixels with gradient magnitudes above a given value: $(x_i,y_i), i = 1,...,n$. The cost function is then used to determine the dissimilarity of each of the n points in the window with point (x',y') in the

reference image. The point in the window that produces the least cost magnitude is selected as the point corresponding to (x', y'). That is, if $C(x_m, y_m, x', y') = min\{C(x_i, y_i, x', y') : i = 1, ..., n\}$, then point (x_m, y_m) in the target image is selected as the point corresponding to point (x', y') in the reference image.

The specific steps of the matching method are described as following.

Input : Target image intensities $I(x, y)$; reference image intensities $I'(x', y')$; gradient magnitude threshold value t_g; size of search window w; and weight factors u_1, u_2, u_3, and u_4.

Output : x and y components of the corresponding point in target image and its intensity.

- Determine the gradient magnitudes and gradient directions of the target and reference images : $D(x, y)$, $\theta(x, y)$, $D'(x', y')$, $\theta'(x', y')$.
- For each point (x', y') in the reference image with gradient magnitude $D'(x', y') > t_g$, determine the corresponding point (x, y) in the target image. Take a window centered at (x', y') in the target image, and for each point (x, y) in the window so that $D(x, y) > t_g$, compute the cost magnitude $C(x, y, x', y')$. Determine the point (x, y) in the window which has the minimum cost magnitude. Then, take that point in the target image as the point corresponding to (x', y') in the reference image.
- For points with gradient magnitudes smaller than or equal to t_g, we estimate the intensity difference between reference point (x', y') and each point in the sampling window in the target image, then choose the point which has the minimum intensity difference as the corresponding point.

2.5 Contrast Sensitivity

Human visual behaviors can be attributed to our visual anatomy. Roughly speaking, modelling visual perception increases in complexity as one follows the pro-

gression of information from the retina through optic nerve, the optical chiasm, the laternal geniculate nuclei, and the cerebral cortex (5). Figure 2.8 shows the information flow of neural image formation. Scene information in the form of light intensity comes to us as a function of position in the scene $\bar{p}$ of time t and of the radiation's wavelength λ. Refraction by our cornea, intraocular fluids, and lens focuses some of this information on our retina position, time, and wavelength. Receptor cells at the back of the retina sense these intensities and, through a complex network of interconnecting cells, encode the image into neural signals to be carried by the optic nerve (8). Contrast sensitivity is a measure of how the human visual system detects detail in a scene, and as such, it is a valuable tool to measure the quality of interpolation image.

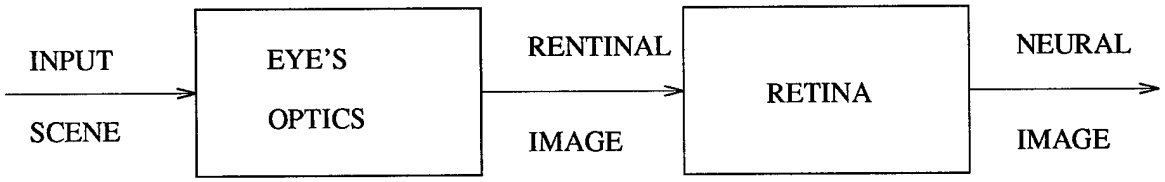


Figure 2.8 Generic model of neural image formation.

Contrast is a measure of the difference in brightness of image or scene. It is also the difference between the brightness of an object and its background. For example, advertisement notices with white background and black letters have a high contrast, whereas a gray background and slightly lighter gray letters have low contrast. How well an individual can detect the relative differences in contrast sensitivity (which is often measured by how a person responds to contrast at threshold) is defined as one over contrast (23, 18).

Contrast sensitivity is measured as a function of spatial frequency (23, 18). Spatial frequency is usually measured in terms of cycles per degree of visual angle. There are several methods to measure an individual's contrast sensitivity. One method determines contrasts using the number of bars per unit distance at varying

spatial frequencies. That is, it defines a cycle as a black bar plus a white bar (the unit of the grating that repeats itself). Figure 2.9 shows two gratings of different spatial frequencies. As we move a grating farther away, or make the bars narrower, the spatial frequency of the grating increases. This method measures the threshold contrast for seeing the bars and the threshold can be converted into contrast sensitivity (contrast sensitivity = 1/threshold contrast). Another method is to present a subject with a series of sinusoidally varying gratings and each with different spatial frequencies and contrasts to measure the minimum contrast needed for an individual to detect the orientation of the grating. Figure 2.10 shows a high contrast vertically oriented test grating. To determine the contrast for a particular grating the modulation contrast (or Michelson contrast) is generally computed (23, 18). The Michelson contrast formula is defined as:

$$C = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \tag{2.13}$$

where L_{max} and L_{min} are the maximum and minimum luminance values, respectively, in the gratings. For gray-scale images, L_{max} and L_{min} are the maximum and minimum values of the sinusoid.

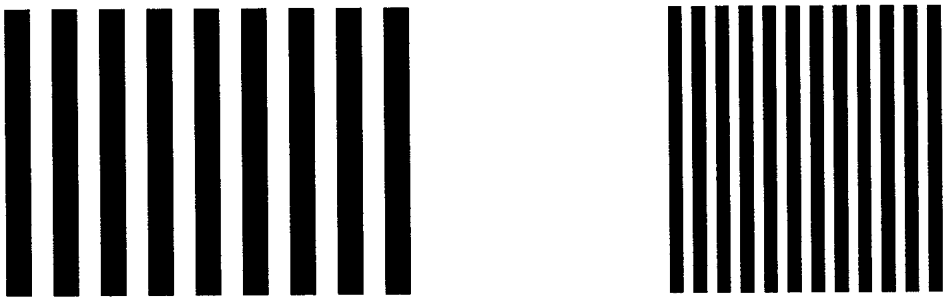


Figure 2.9 Example of different acuity gratings.

An experiment to measure humans' contrast sensitivity is presented by McCormick in his book *Human Factors Engineering* (23). The image quality measure

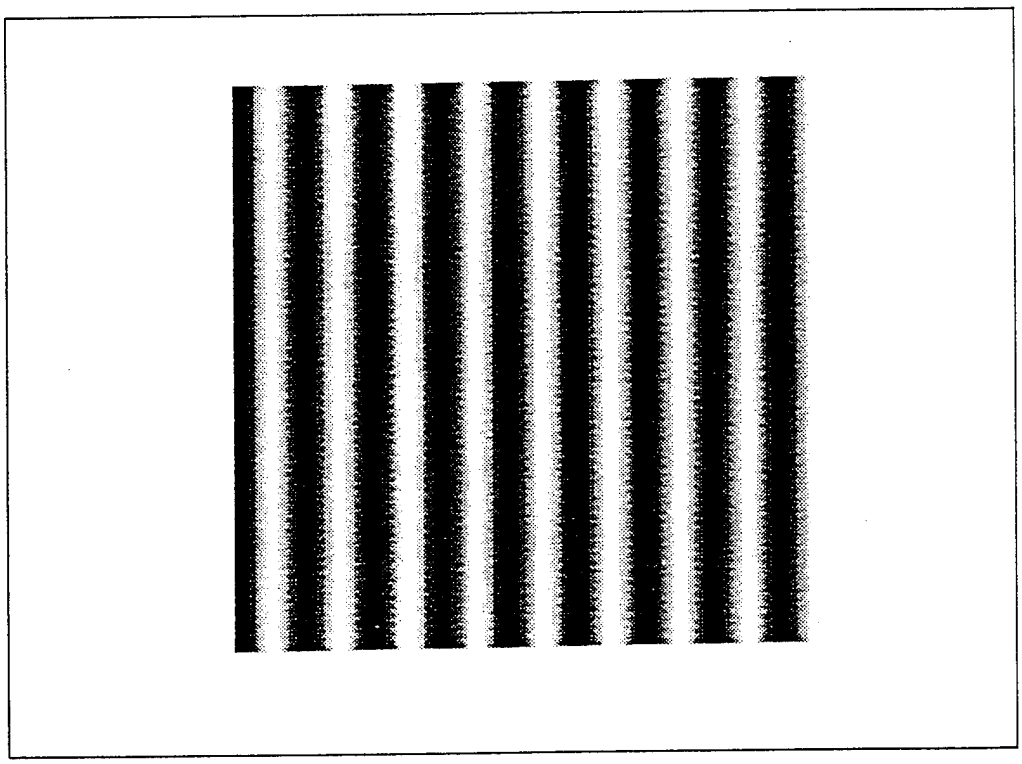


Figure 2.10 Example of a low frequency high contrast sinusoidally varying test grating.

model, used in this thesis, developed by Captain Wilson, chose the sensitivity response of 95 percentile curve. The curve was represented by a one-dimensional vector and it represents the frequency response in cycles per degree of spatial resolution. In order to measure the quality of a two dimensional image, the sensitivity response curve was also modified to a two dimensional sensitivity response (27). This conversion relies on the assumption that the response of the human visual system is the same both horizontally and vertically (6, 12). The response can also be extended to other orientations by using a linear combination of the components of the horizontal and vertical response (6, 12). Since the distance between human's eyes and an object will change the spatial frequencies observed by the viewer (27). This model assumed that the viewer is looking at an image on a computer screen 24 inches away.

In this thesis, we use the two dimensional model and calculate the visual perceptual difference between the estimated intermediate image and the known intermediate image to measure the image quality with different interpolation methods. The visual perception difference function is as following :

$$perceptualdiff = \sum_i \sum_j C_{i,j} \times energydiff_{i,j} \tag{2.14}$$

C is the two dimensional contrast sensitivity response weight matrix.
Energydiff is the difference between the energy normalized Fourier coefficients of the first 20 frequencies.

2.6 Conclusion

This chapter introduced the Computed Tomography image and discussed three interpolation algorithms. It also presented the image matching method developed by Ardeshir Goshtasby, David A. Turner, and Laurens (2). There are several matching techniques, but it is very difficult to design a single method that can match all types of images. The method used in this thesis was designed specifically for matching

tomographic images, using their special properties. We also introduced the application of contrast sensitivity of human visual perception as a method to measure the quality of the interpolated image. The model was developed by Captain Wilson (27). The next chapter will specifically describes the processes of three interpolation methods and the process to find the appropriate values of the four parameters (u_1, u_2, u_3, and u_4) in the matching cost function.

III. APPROACH

3.1 Introduction

In the previous chapter, the formation of CT images was introduced and we discussed three interpolation algorithms along with the image matching method (2). We also discussed how contrast sensitivity plays an important role in the human visual system and how it can be used to measure that system is responding to visual stimuli. We can calculate image visual perceptual difference and compare image qualities for a variety of interpolation methods. This chapter will describe how to display the 2D CT image provided by the North Carolina Memorial Hospital, the implementation of the three interpolation methods used in this thesis, and the process of finding the appropriate parameter values (u_1,u_2,u_3, and u_4) in the matching cost function. For this thesis, the program design and data processing were developed using the MATLAB interactive software package on SUN Workstations.

3.2 Two-Dimensional CT image

The CT head data set provided by the North Carolina Memorial Hospital has 113 slices. Slices are stored consecutively as a 256 x 256 array with dimensions of z-113 y-256 x-256 in z-y-x order. The format is 16-bit integers, that is, two consecutive bytes make up one binary integer. In order to display the 2D CT image with MATLAB, we convert the binary data set to an ASCII file and store it as 113 matrices of size 256 x 256. We also normalized all image intensities to 0-255.

The CT image with successful reconstruction, conversion, and transformation can be displayed as Figure 3.1. Figure 3.2 shows the CT image which was recon-

structed from too few views or reconstructed with too few photons during the actual measurement.

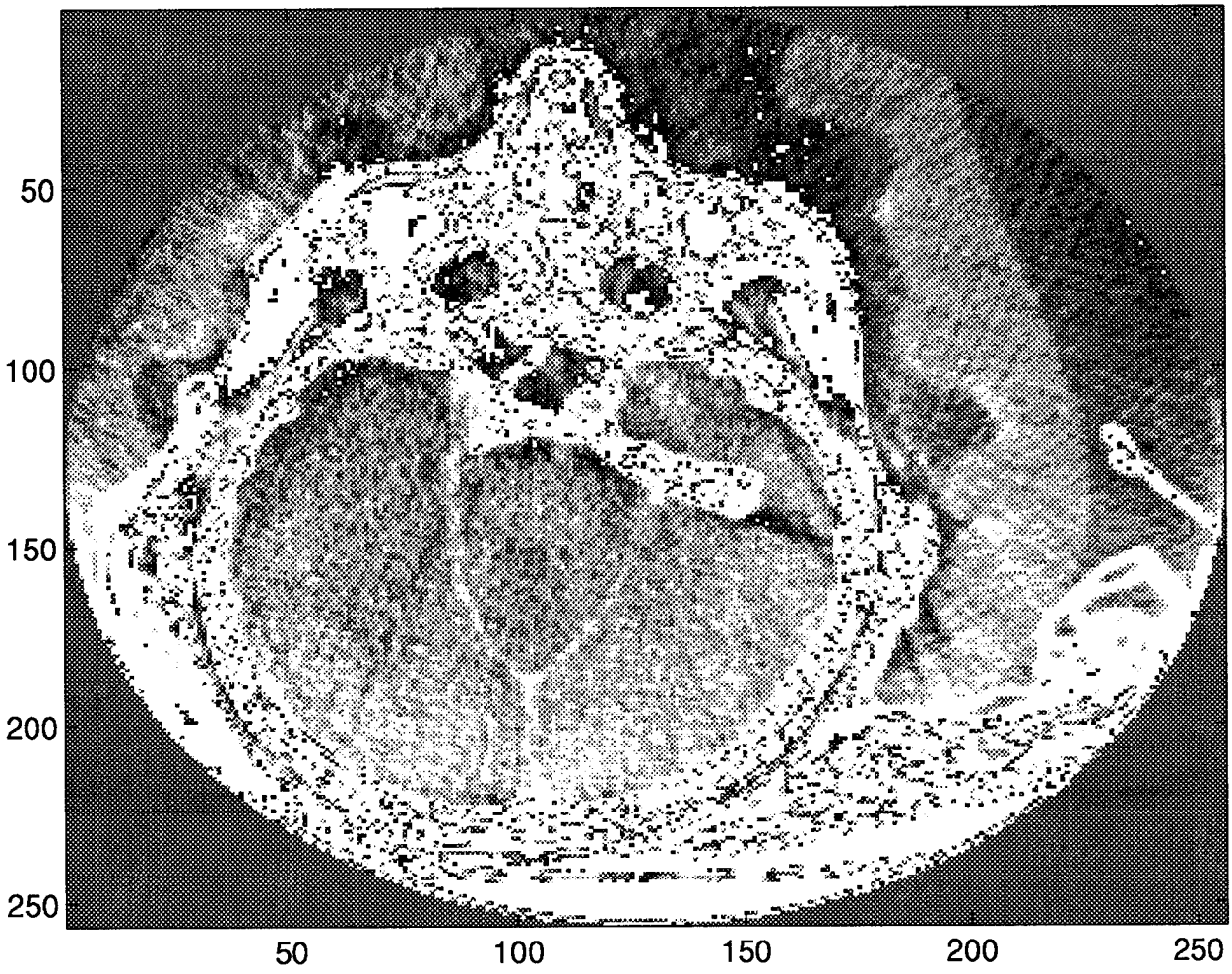


Figure 3.1 2D CT image with successful reconstruction and conversion

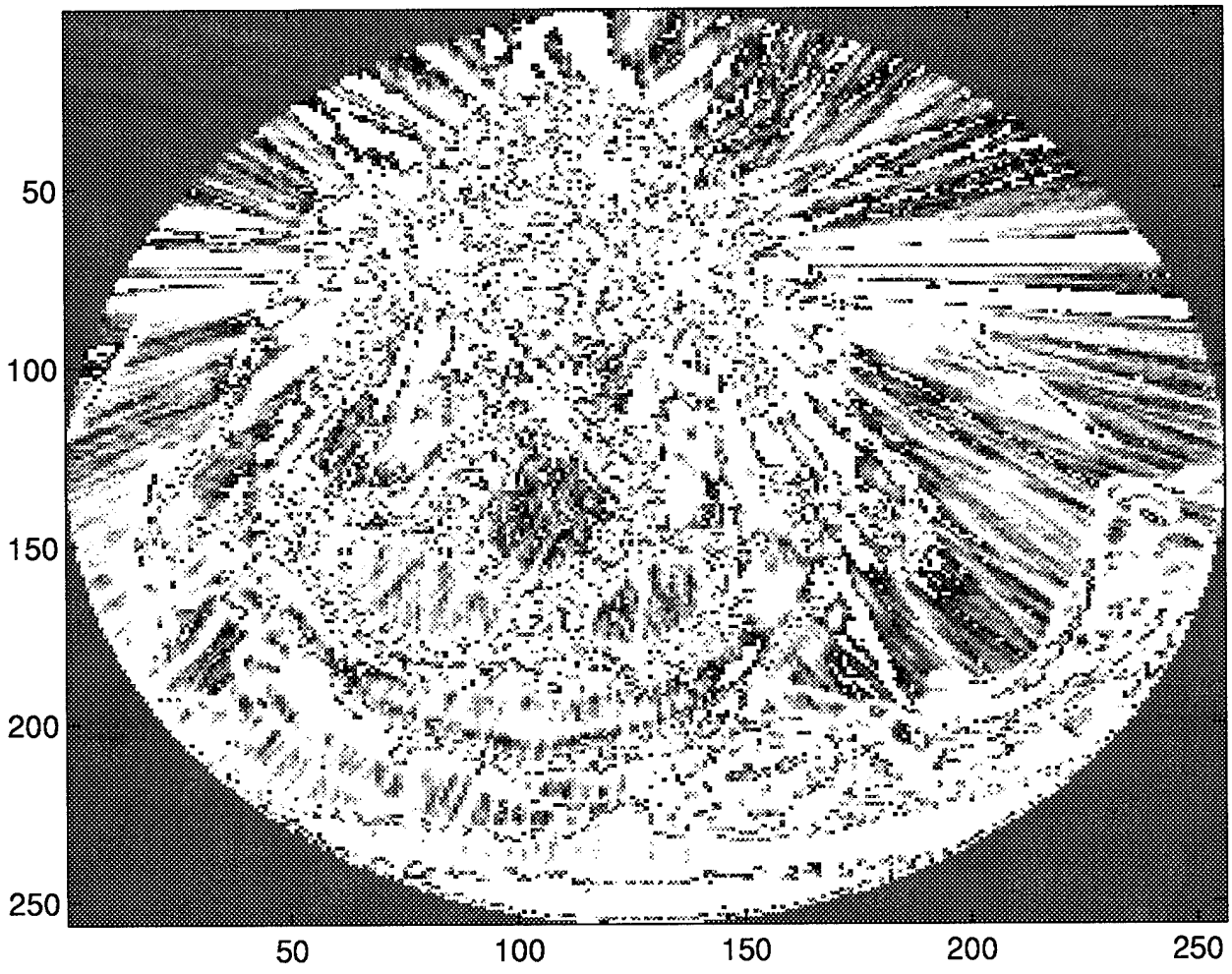


Figure 3.2 Illustration of 2D CT image from reconstruction with too few views or reconstruction with too few photons during the actual measurement.

3.3 Implementation of Interpolation

As we know, medical imaging devices usually produce 2D image slices and the interslice distance is larger than the interpixel distance. In order to make a 3D image, we need more intermediate slices along the z-axis. This section describes the manner in which we pick a set of numbers as the series of matching point intensities of consecutive images and process it with three interpolation methods. The consecutive images are organized as shown Figure 2.7.

We assume the following 1 x 8 vector is a series of matching points of eight consecutive images and for the convenience of continuing the discussion, we will call the vector a matching vector. In the image interpolation process, there are 256 x 256 matching vectors.

example vector = [6.95 9.95 11.94 8.95 19.91 21.90 20.90 17.91]

The example vector is depicted in Figure 3.3.

3.3.1 Linear Interpolation. The MATLAB one-dimensional linear interpolation function was designed as the algorithm described in chapter II. The input is the matching vector and the elements we want to interpolate between each two consecutive images. The output is a linearly interpolated row with monotonicity. Figure 3.4 shows one intermediate point added between each two consecutive points of the matching vector. Figure 3.5 shows four intermediate points added between each two consecutive points of the matching vector. From these two figures, it is obvious that the relationship between known points and intermediate points is linear.

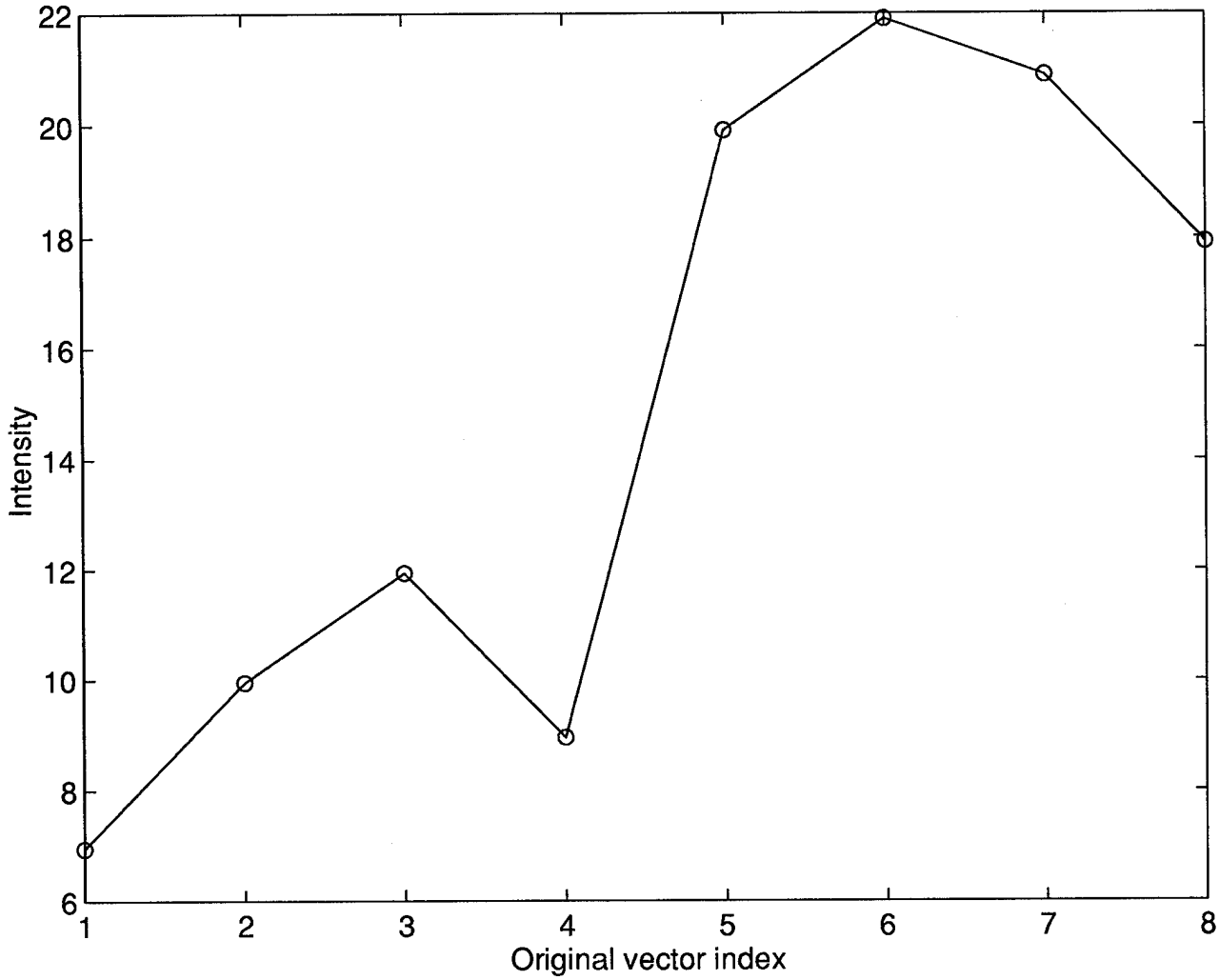


Figure 3.3 Example vector = [6.95 9.95 11.94 8.95 19.91 21.90 20.90 17.91]

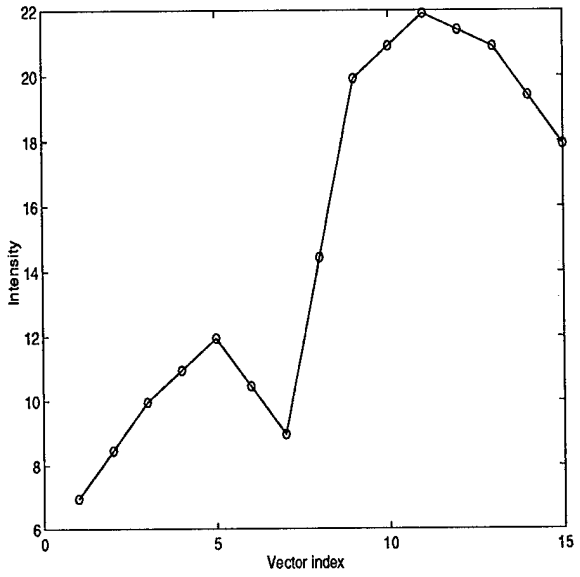


Figure 3.4 Linear Interpolation : Add one intermediate point between each two consecutive points of example vector

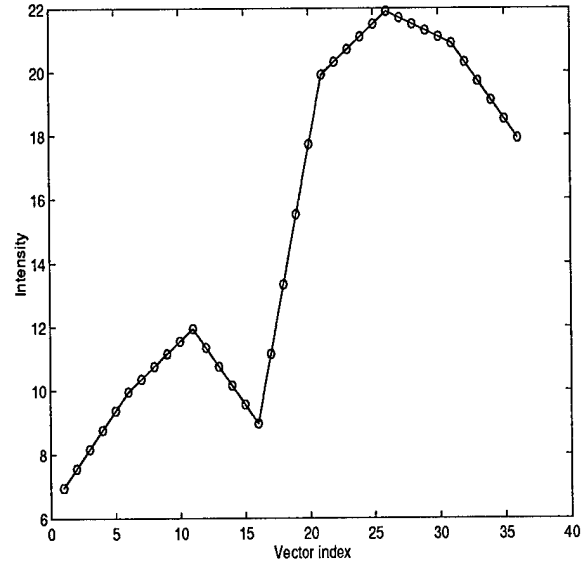


Figure 3.5 Linear Interpolation : Add four intermediate points between each two consecutive points of example vector

3.3.2 Cubic Spline Interpolation. The 'natural' cubic spline interpolation used in this thesis is a simplified case of cubic spline interpolation. The key point of natural cubic spline interpolation is to make the conditions that the second derivatives at endpoints are zero. Natural cubic spline interpolation assumes that the relationship between each two known points is a cubic polynomial function. It is different from the linear interpolation which assumes the relationship between each two known points is a monotonically increasing or decreasing function. Because each piece of the matching vector is a cubic polynomial function, the number of elements of the matching vector can not be fewer than four.

Figure 3.6 shows the matching vector interpolated with natural cubic spline interpolation and only one intermediate point added between each two consecutive points. Figure 3.7 shows four intermediate points added between each two consecutive points.

3.3.3 Fourier Interpolation. The one-dimensional Fourier interpolation used in this thesis uses the Fast Fourier Transform (FFT) method and assumes the input vector is a periodic vector sampled at equally spaced points. The periodic vector is first transformed to the Fourier domain using the FFT. The zero padding is added to resample the original vector and than it is transformed back with more points.

Figure 3.8 shows one intermediate point added and Figure 3.9 shows four intermediate points added between each two consecutive points of the example vector.

3.3.4 Image Interpolation Test. In the previous sections we presented the implementation of three interpolation methods. In this section, we present a test which uses linear, cubic spline, and Fourier interpolation methods respectively to generate an intermediate slice between the original consecutive images of number 50 and 51. This test does not apply to the matching method and assumes the matching

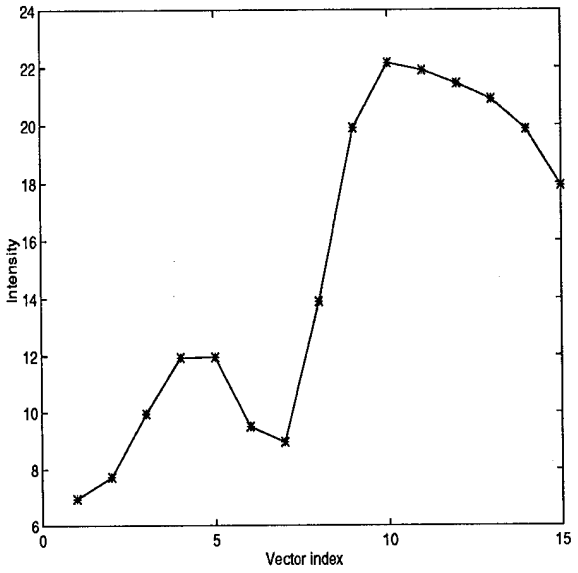


Figure 3.6 Cubic Spline Interpolation : Add one intermediate point between each two consecutive points of the example vector

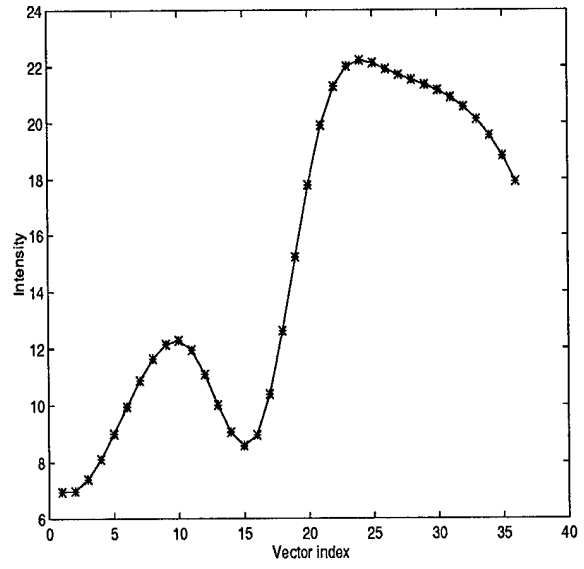


Figure 3.7 Cubic Spline Interpolation : Add four intermediate points between each two consecutive points of the example vector

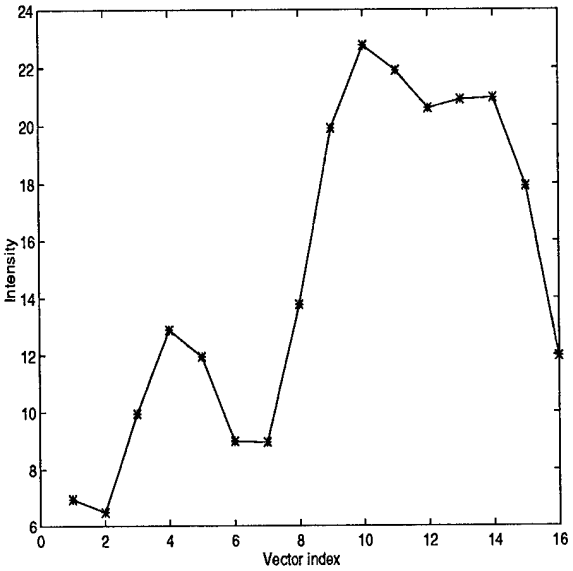


Figure 3.8 Fourier Interpolation : Add one intermediate point between each two consecutive points of the example vector

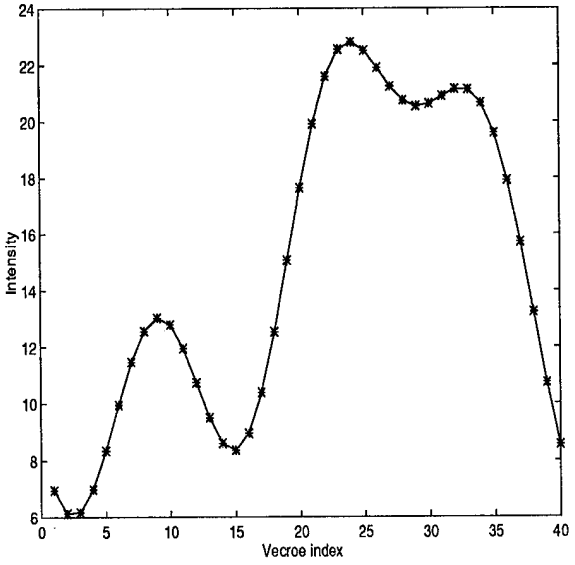


Figure 3.9 Fourier Interpolation : Add four intermediate points between each two consecutive points of the example vector

vectors are constructed by choosing the same coordinate pixel in those consecutive images.

Images number 50 and 51 are shown in Figure 3.10 and Figure 3.11 respectively. Figure 3.12 shows the intermediate slice interpolated by linear interpolation method. Figure 3.13 shows the slice interpolated by cubic spline interpolation method and Figure 3.14 shows the slice interpolated by the Fourier interpolation method.

3.4 Parameters of the Matching Process

The reliability of a matching process depends on different image properties. A matching process should have parameters that could be adjusted to the type of images matched. The matching method used in this thesis has weight factors u_1, u_2, u_3, and u_4 that can be used to adjust the intensity, gradient and geometric differences between the images. In order to determine the behavior of the matching method under variations of its parameters, we carry out three experiments which use linear interpolation and the root-mean-squared measuring the error between known and estimated slices in order to decide the proper value of the weight factors. Because each two consecutive CT images have similar intensities and gradients, there is a small disparity value and we therefore set the weight factor u_4 to 1.0 in all three experiments.

In the first experiment, we calculate root-mean-squared errors as a function of parameter u_1 and parameters u_2, u_3, and u_4 were all set to 1.0. The matching results are summarized in Table 3.1. In the second experiment, we fix u_1 using the appropriate value found in the first experiment (0.5) and calculate the root-mean-square error as a function of parameter u_2. In the last experiment, we replace u_1 and u_2 using the appropriate values found in the previous experiments (0.5 and 0.4 respectively) and calculate the root-mean-square error as a function of parameter u_3. The results of the second and the third experiments are depicted in Table 3.2 and Table 3.3.

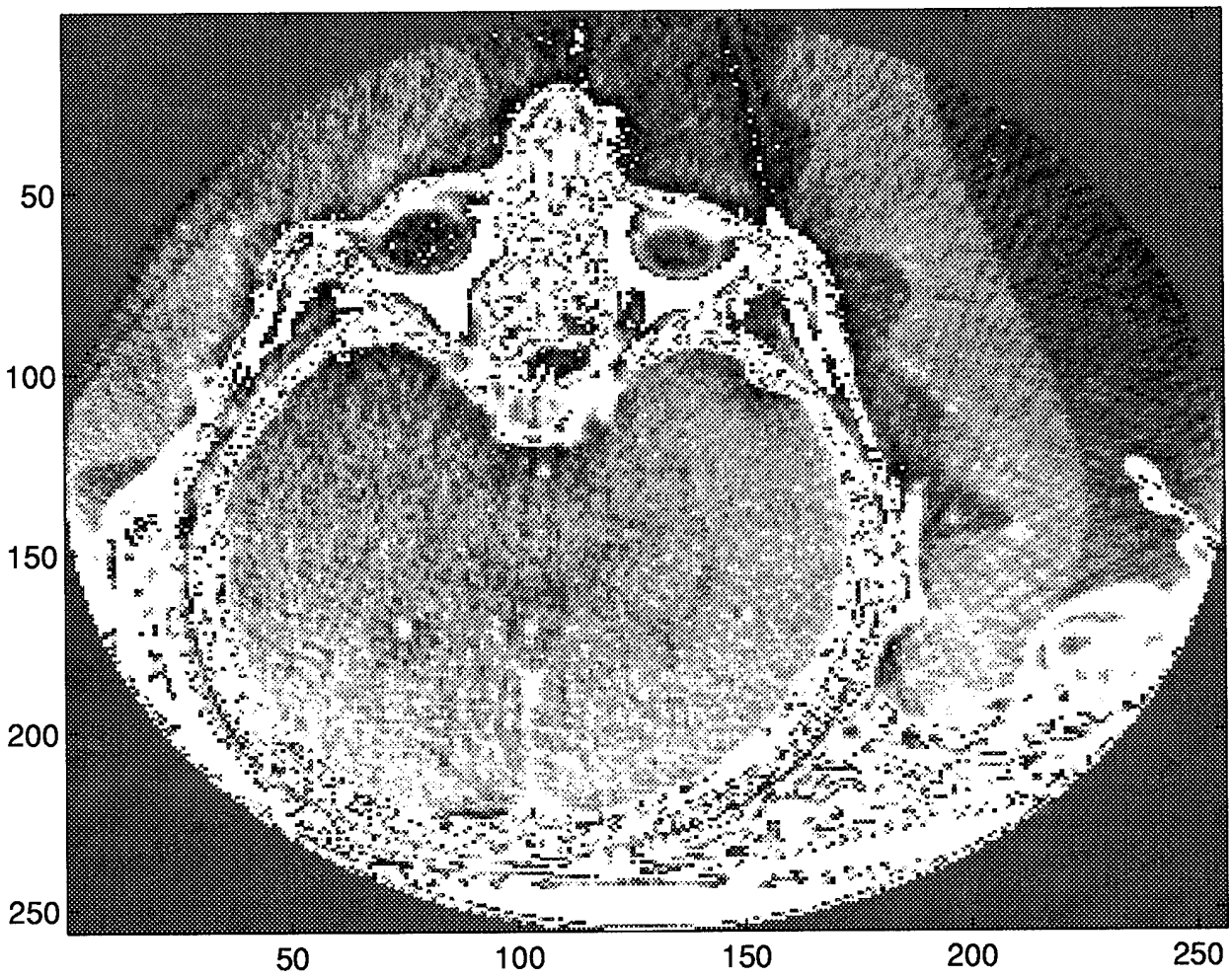


Figure 3.10 Original Image number 50

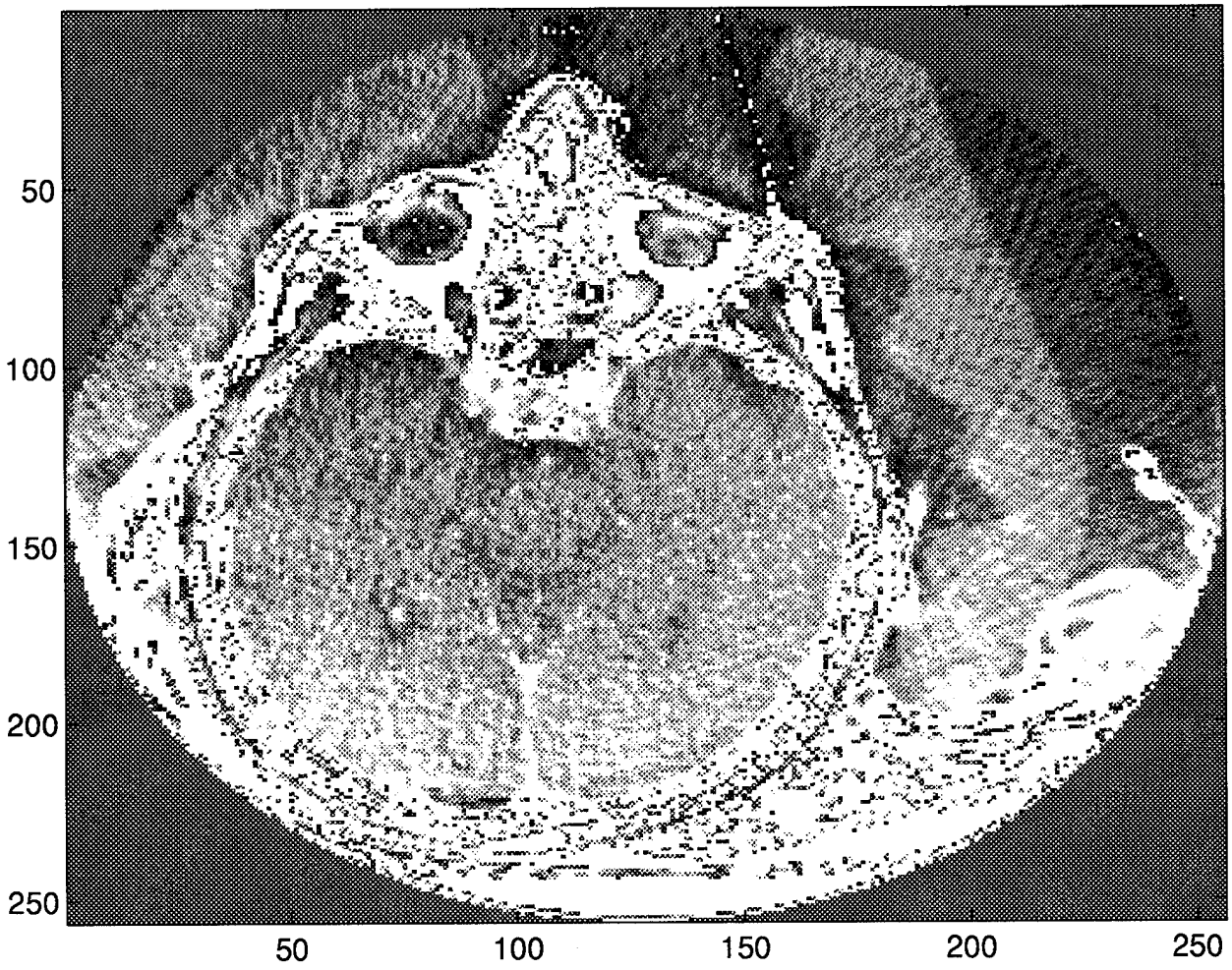


Figure 3.11 Original Image number 51

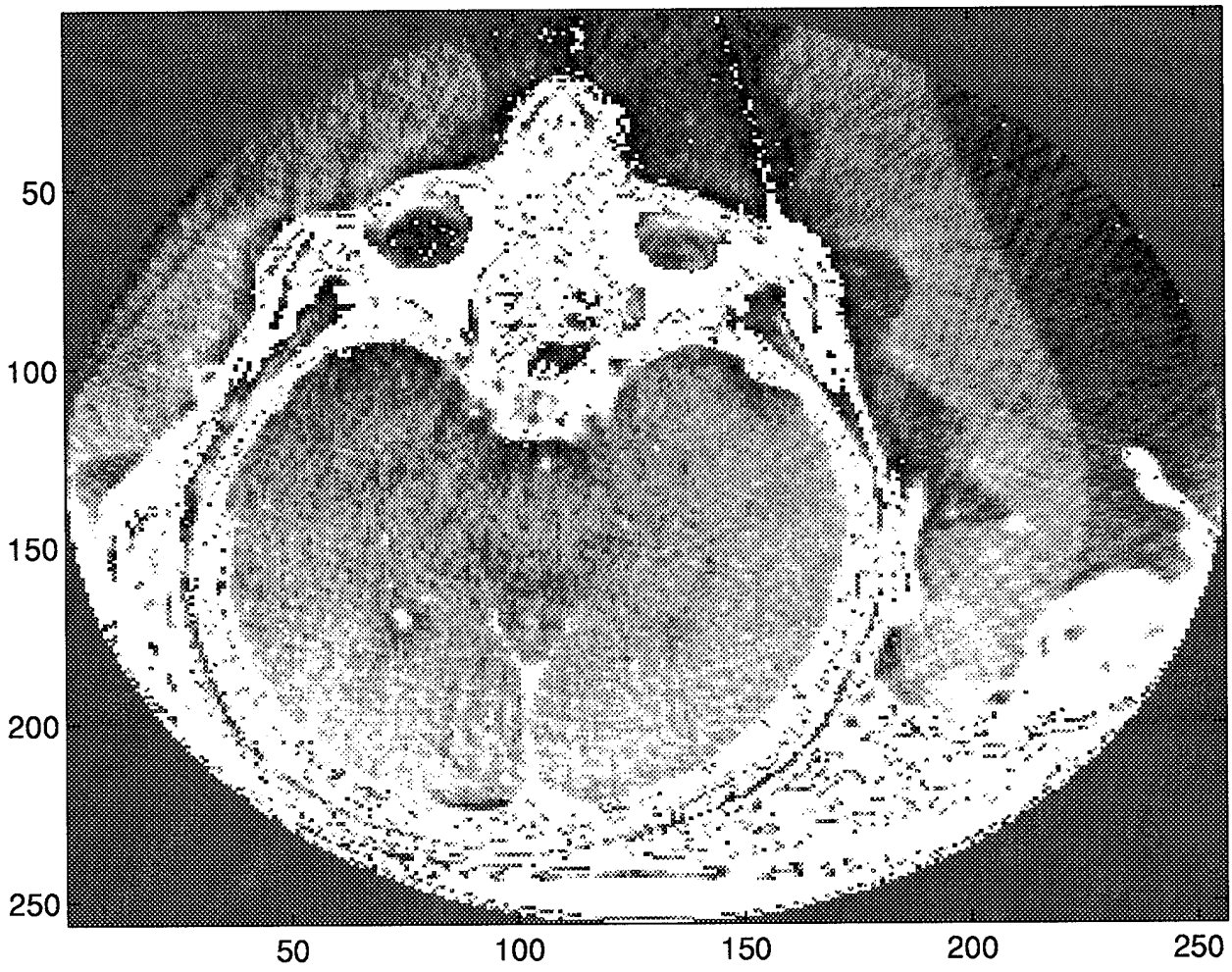


Figure 3.12 Intermediate slice : With linear interpolation and without the application of the matching method.

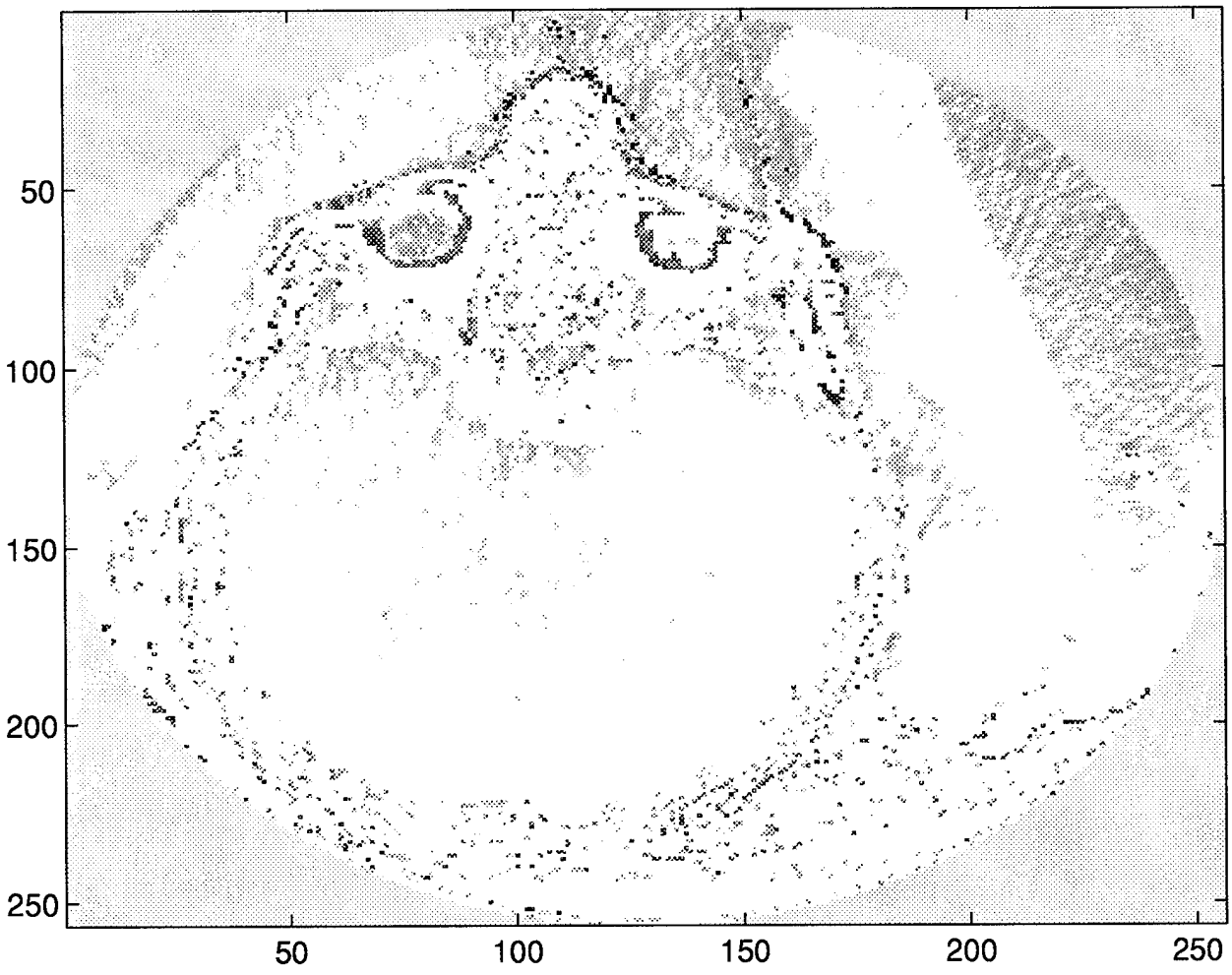


Figure 3.13 Intermediate slice : With cubic spline interpolation and without the application of the matching method.

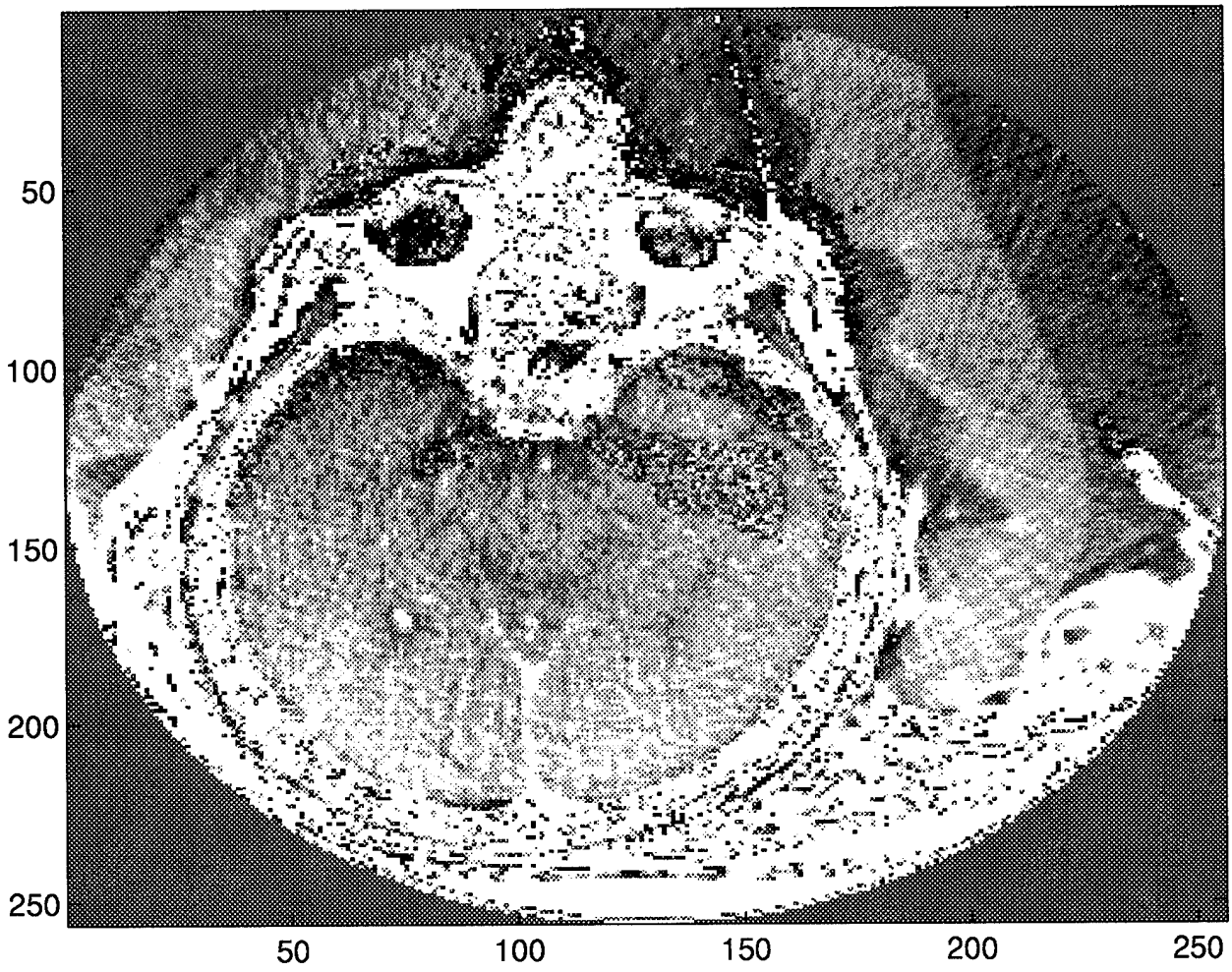


Figure 3.14 Intermediate slice : With Fourier interpolation and without the application of the matching method.

u_1	u_2	u_3	u_4	$rmse$
0.1	1.0	1.0	1.0	1.1295
0.2	1.0	1.0	1.0	1.1279
0.5	1.0	1.0	1.0	1.1257
0.8	1.0	1.0	1.0	1.1304
1.0	1.0	1.0	1.0	1.1320

Table 3.1 $u_2 = 1.0$, $u_3 = 1.0$, $u_4 = 1.0$

u_1	u_2	u_3	u_4	$rmse$
0.5	0.1	1.0	1.0	1.1268
0.5	0.2	1.0	1.0	1.1263
0.5	0.4	1.0	1.0	1.1253
0.5	0.8	1.0	1.0	1.1256
0.5	1.0	1.0	1.0	1.1257

Table 3.2 $u_1 = 0.5$, $u_3 = 1.0$, $u_4 = 1.0$

u_1	u_2	u_3	u_4	$rmse$
0.5	0.4	0.1	1.0	1.1258
0.5	0.4	0.2	1.0	1.1255
0.5	0.4	0.5	1.0	1.1243
0.5	0.4	0.8	1.0	1.1252
0.5	0.4	1.0	1.0	1.1253

Table 3.3 $u_1 = 0.5$, $u_2 = 0.4$, $u_4 = 1.0$

3.5 Conclusion

This chapter presented the 2D CT image used in this thesis, the implementation of three interpolation methods, and the experiments used to decide the appropriate parameters of the matching process. The next chapter discusses the results of interpolation images generated by the linear, cubic spline and Fourier interpolation methods. We will also use the contrast sensitivity of human perception to measure the qualities of interpolation images.

IV. INTERPOLATION RESULTS

4.1 Introduction

This chapter provides examples of image interpolation that employ three different algorithms, some of which use the matching method described in chapters two and three. We also use the human visual perception model to measure the qualities of interpolated images. The first example compares two interpolated images that both use the linear interpolation algorithm. One of them also uses the matching method. The second and the third examples carry out the same process but respectively replace the interpolation method by the cubic spline and Fourier interpolation algorithms.

The interpolated images that are processed without the matching method assume matching vectors by choosing the same coordinate pixel in each consecutive image.

4.2 Linear Interpolation Image

In this section, the results of two linear interpolation images are presented. One was carried out with the matching method and the other wasn't. The test images used in this example are slice 50 and 52. We generated the intermediate slice 51 and compared it with the original slice 51.

Figure 4.1 shows the slice 50 and Figure 4.2 shows the slice 52. Figure 4.3 shows the original slice 51. The intermediate slice 51 without the application of matching method is shown in Figure 4.4. The intermediate slice51 with the matching method is shown in Figure 4.5.

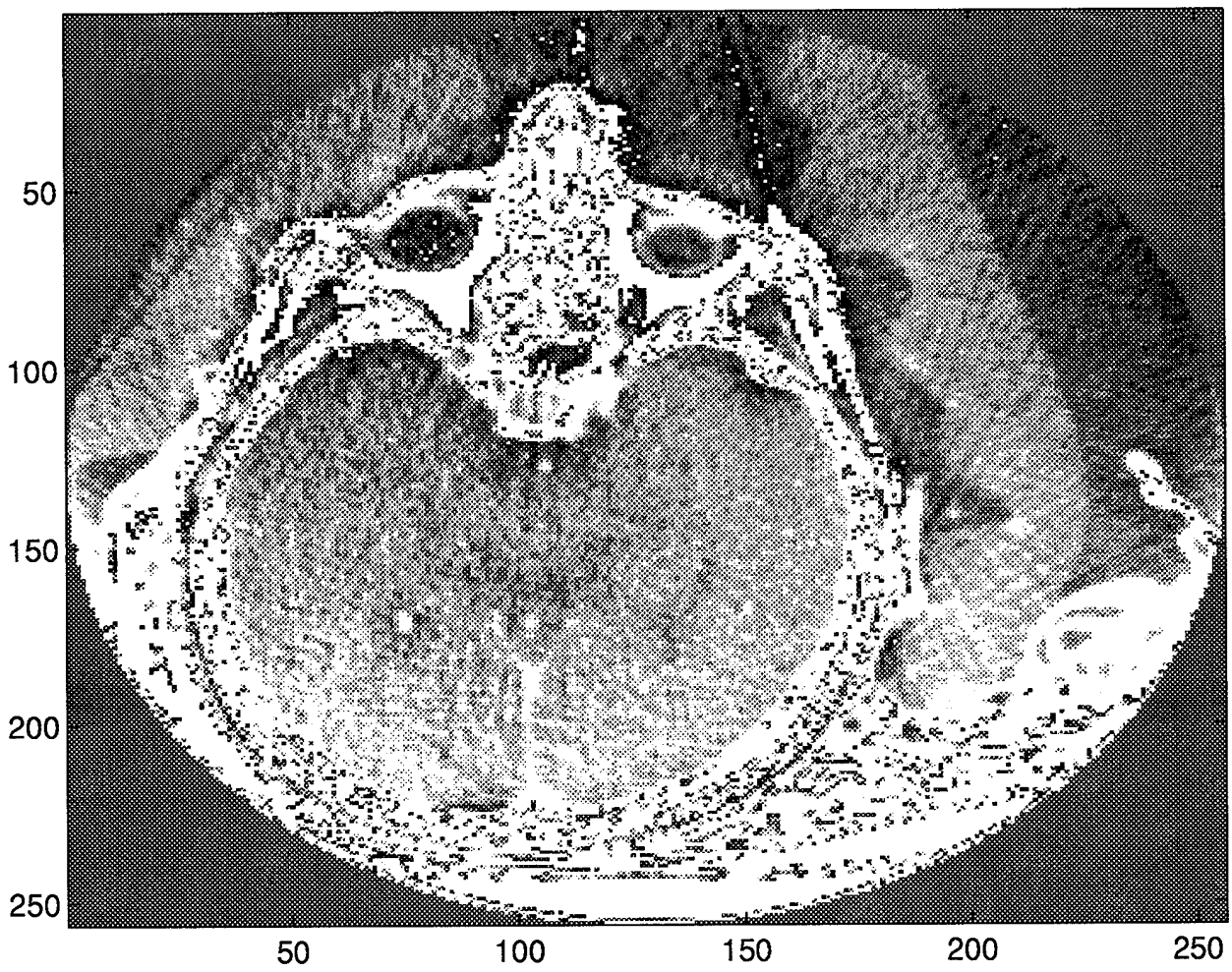


Figure 4.1 Original CT Slice 50

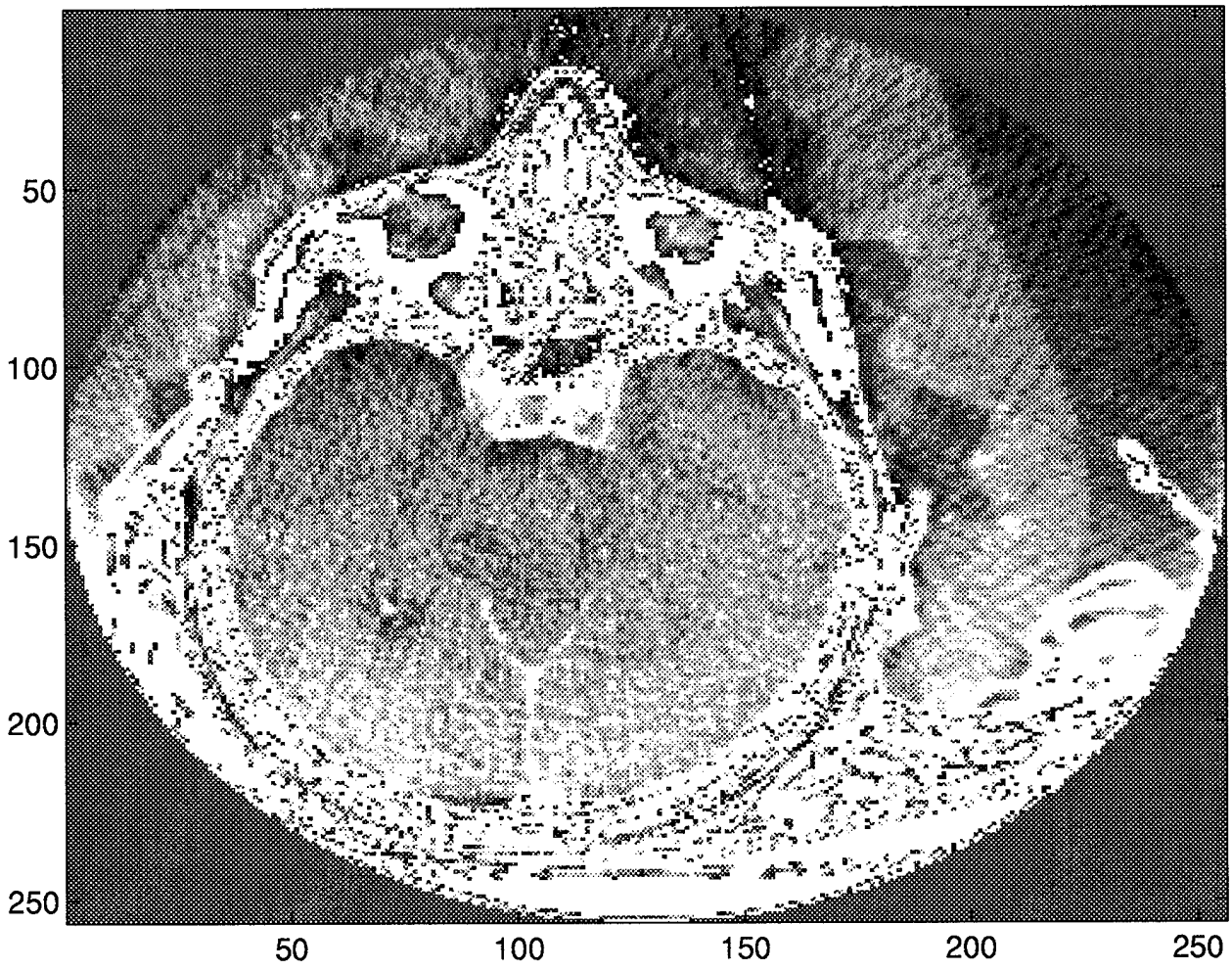


Figure 4.2 Original CT Slice 52

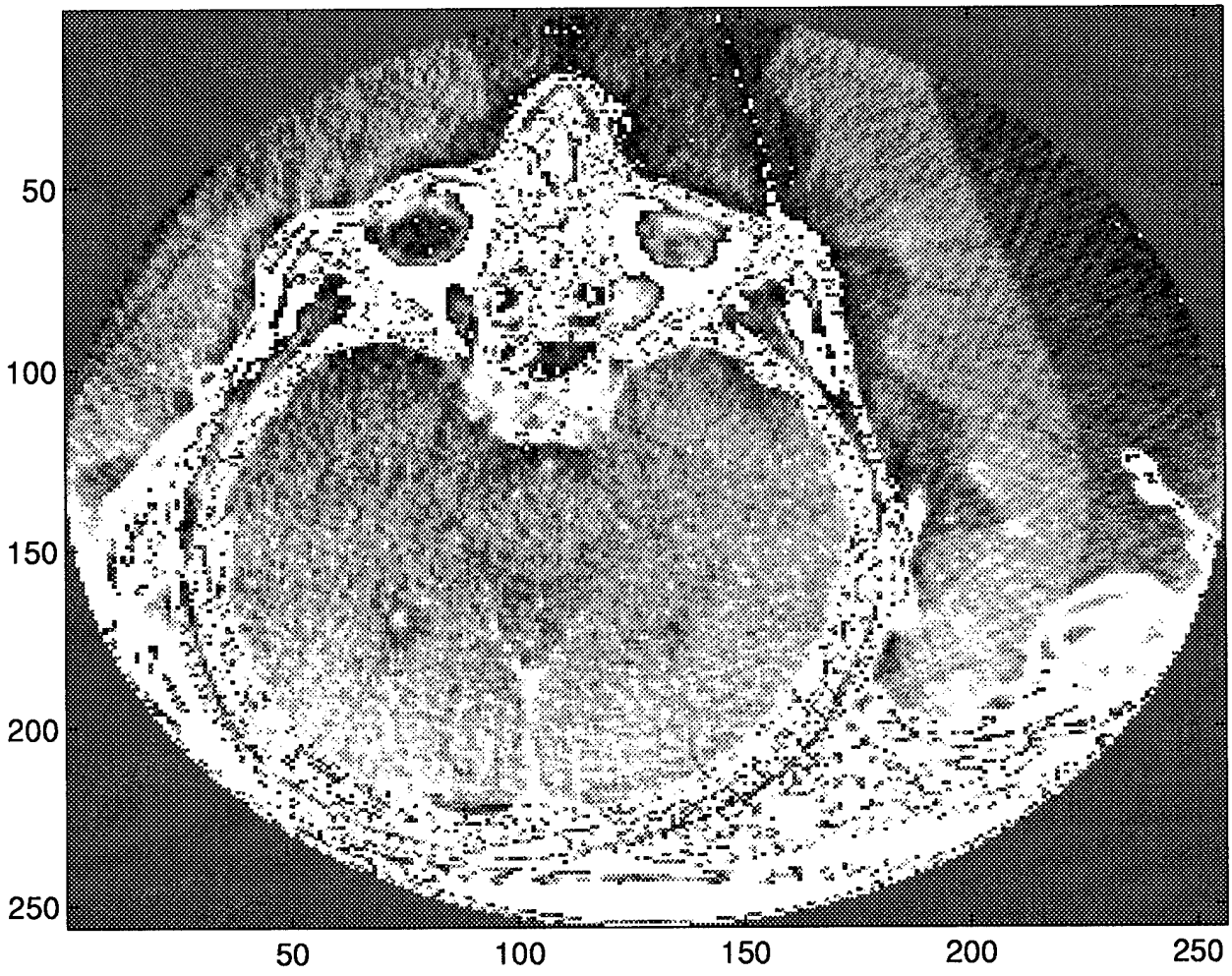


Figure 4.3 Original CT Slice 51

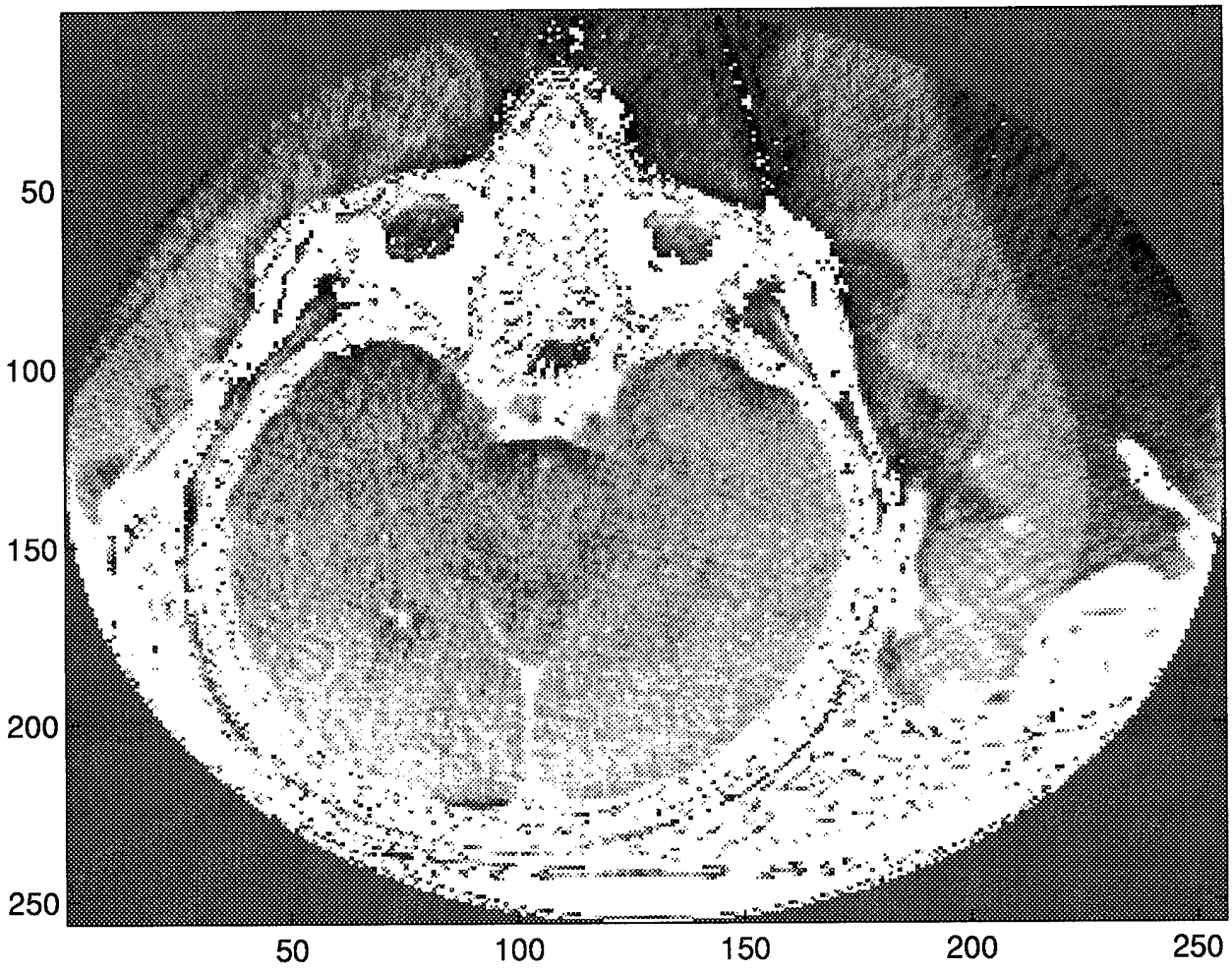


Figure 4.4 The intermediate slice 51 using linear interpolation and without applying the matching method.

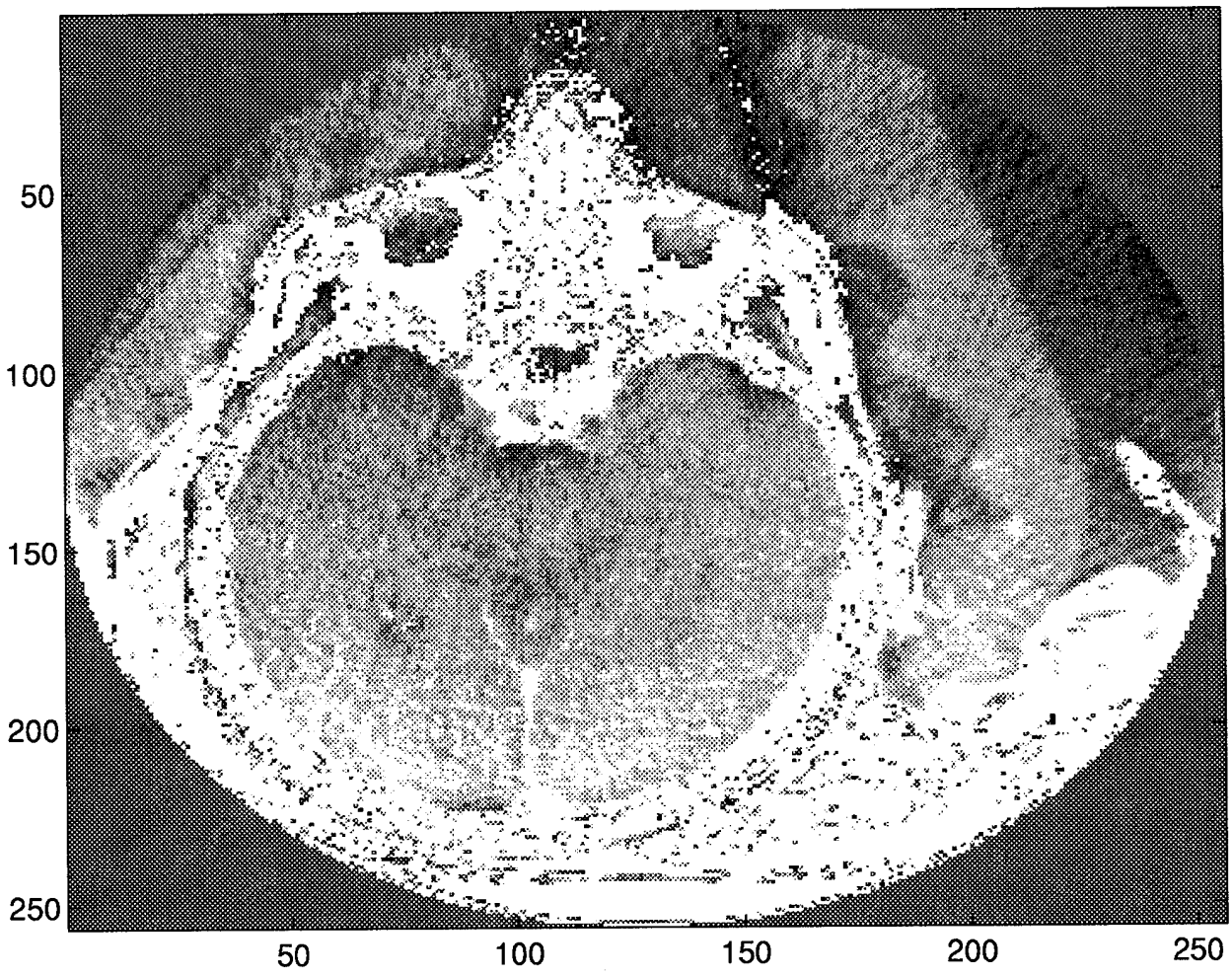


Figure 4.5 The intermediate slice 51 using linear interpolation and applying the matching method.

Comparing Figure 4.3 with the circled parts in Figure 4.4 and Figure 4.5, we can find that the intermediate image without applying the matching method loses more information. In order to measure the interpolated image qualities, we use the human visual perception model to calculate the visual perception differences between each intermediate image and the known image. The result is shown in Table 4.1. We also calculate the energies of the known image and the interpolated images. The result is shown in Table 4.2.

ComparisionImages	*VisualPerceptionDifference*
Original Slice 51	214.6885
Intermediate Slice 51 (without matching method)	
Original Slice 51	196.1584
Intermediate Slice 51 (with matching method)	

Table 4.1 Visual perception differences between original slice 51 and each linear interpolation image

ImageName	*Energy*
Original Slice 51	2.2485e+04
Intermediate Slice 51 (without matching method)	2.1293e+04
Intermediate Slice 51 (with matching method)	2.2422e+04

Table 4.2 Image energies of original slice 51 and linear interpolation images

4.3 Cubic Spline Interpolation

In this example the test images are slice 50, 52, 53, and 54. The generated slice is the intermediate slice 51. Figure 4.6 shows the interpolated image without

applying the matching method. Figure 4.7 shows the interpolated image carried out with the matching method. When we compare Figure 4.3 with Figure 4.6 and Figure 4.7, it is obvious that without the matching method information is lost.

Table 4.3 shows the interpolation results measured with the human visual perception model. Table 4.4 shows the energies of the original slice 51 and both interpolated images.

ComparisionImages	*VisualPerceptionDifference*
Original Slice 51	247.1396
Intermediate Slice 51 (without matching method)	
Original Slice 51	226.8381
Intermediate Slice 51 (with matching method)	

Table 4.3 Visual perception differences between original slice 51 and each cubic spline interpolation image

ImageName	*Energy*
Original Slice 51	2.2485e+04
Intermediate Slice 51 (without matching method)	2.4297e+04
Intermediate Slice 51 (with matching method)	2.2282e+04

Table 4.4 Image energies of original slice 51 and cubic spline interpolation images

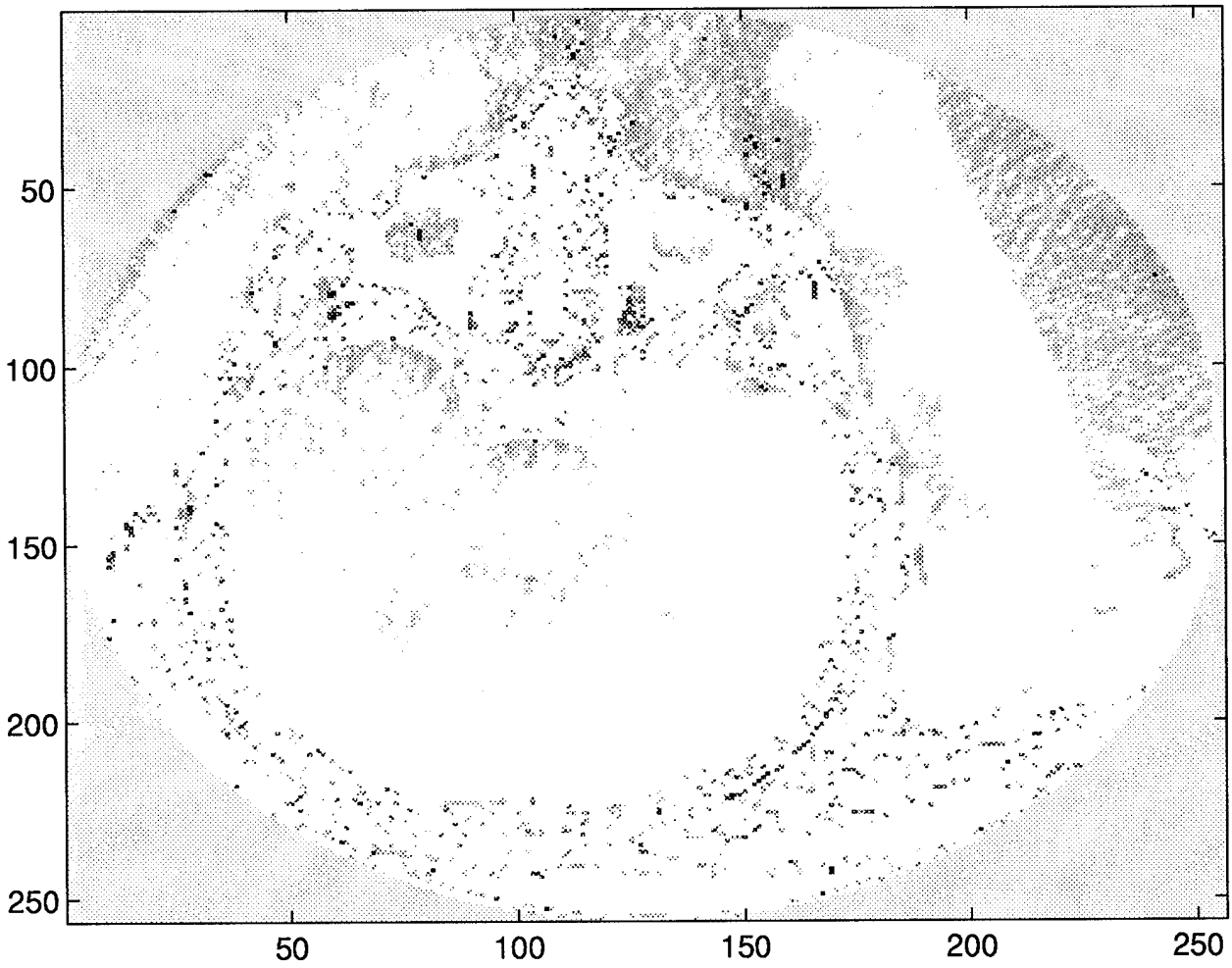


Figure 4.6 The intermediate slice 51 using cubic spline interpolation and without applying the matching method.

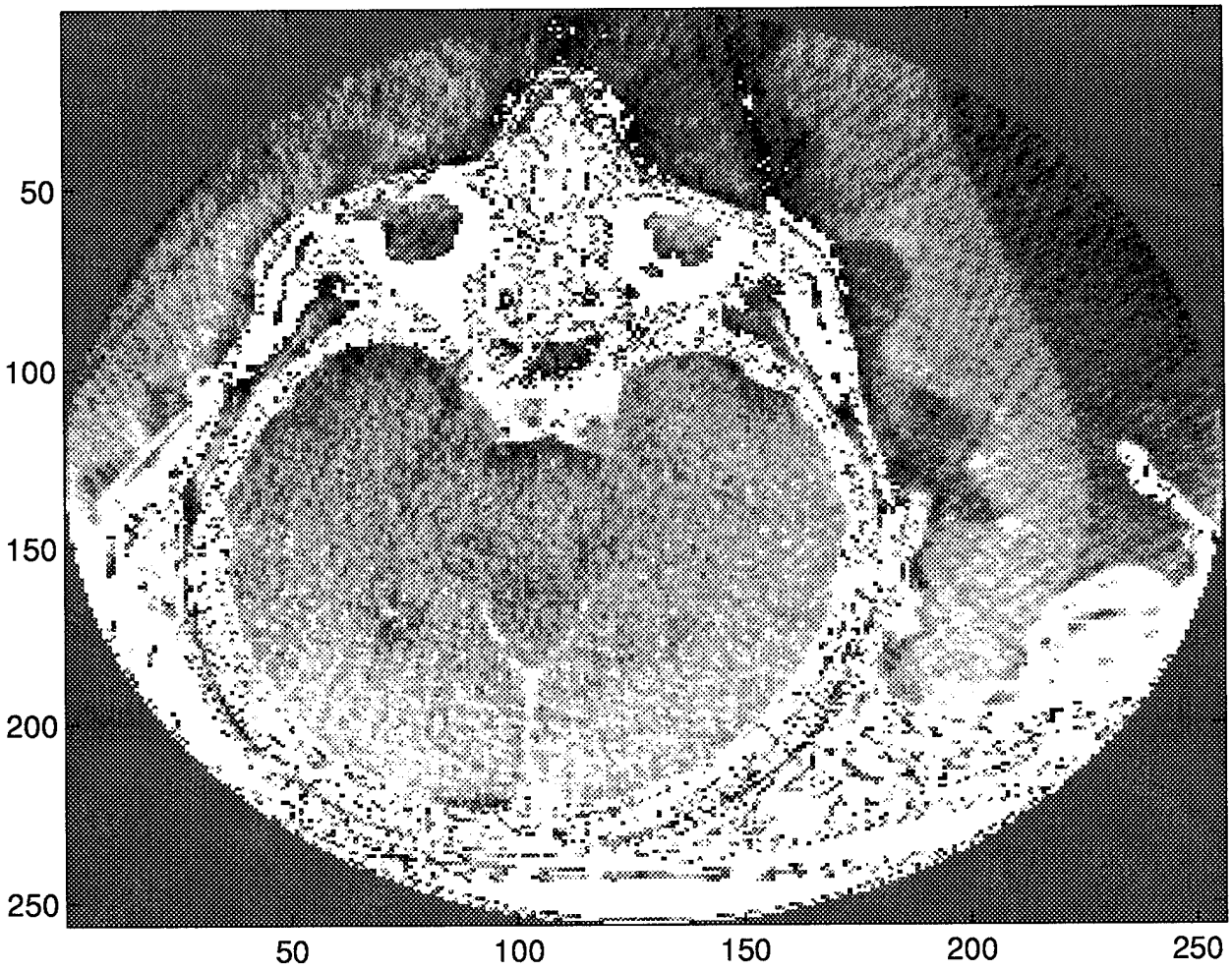


Figure 4.7 The intermediate slice 51 using cubic spline interpolation and applying the matching method.

4.4 Fourier Interpolation Image

In this example the test images are from slice 50 to slice 65. We will still assume slice 51 is unknown and generate it by Fourier interpolation. Figure 4.8 shows the interpolated image without applying the matching method. Figure 4.9 shows the interpolated image carried out with the matching method.

Comparing Figure 4.3 to Figure 4.8, we notice that, when the matching method is not applied there is noise in the image. That is, it produces unnecessary information. Comparing Figure 4.3 and Figure 4.9, we notice that applying the matching method does not produce noise and generates a better interpolated image. Table 4.5 shows the interpolation results measured by the human visual perception model. Table 4.6 shows the energies of original slice 51 and both interpolated images.

ComparisionImages	*VisualPerceptionDifference*
Original Slice 51	251.2280
Intermediate Slice 51 (without matching method)	
Original Slice 51	236.4418
Intermediate Slice 51 (with matching method)	

Table 4.5 Visual perception differences between original slice 51 and each Fourier interpolated image

ImageName	*Energy*
Original Slice 51	2.2485e+04
Intermediate Slice 51 (without matching method)	2.2033e+04
Intermediate Slice 51 (with matching method)	2.2245e+04

Table 4.6 Image energies of original slice 51 and Fourier interpolated images

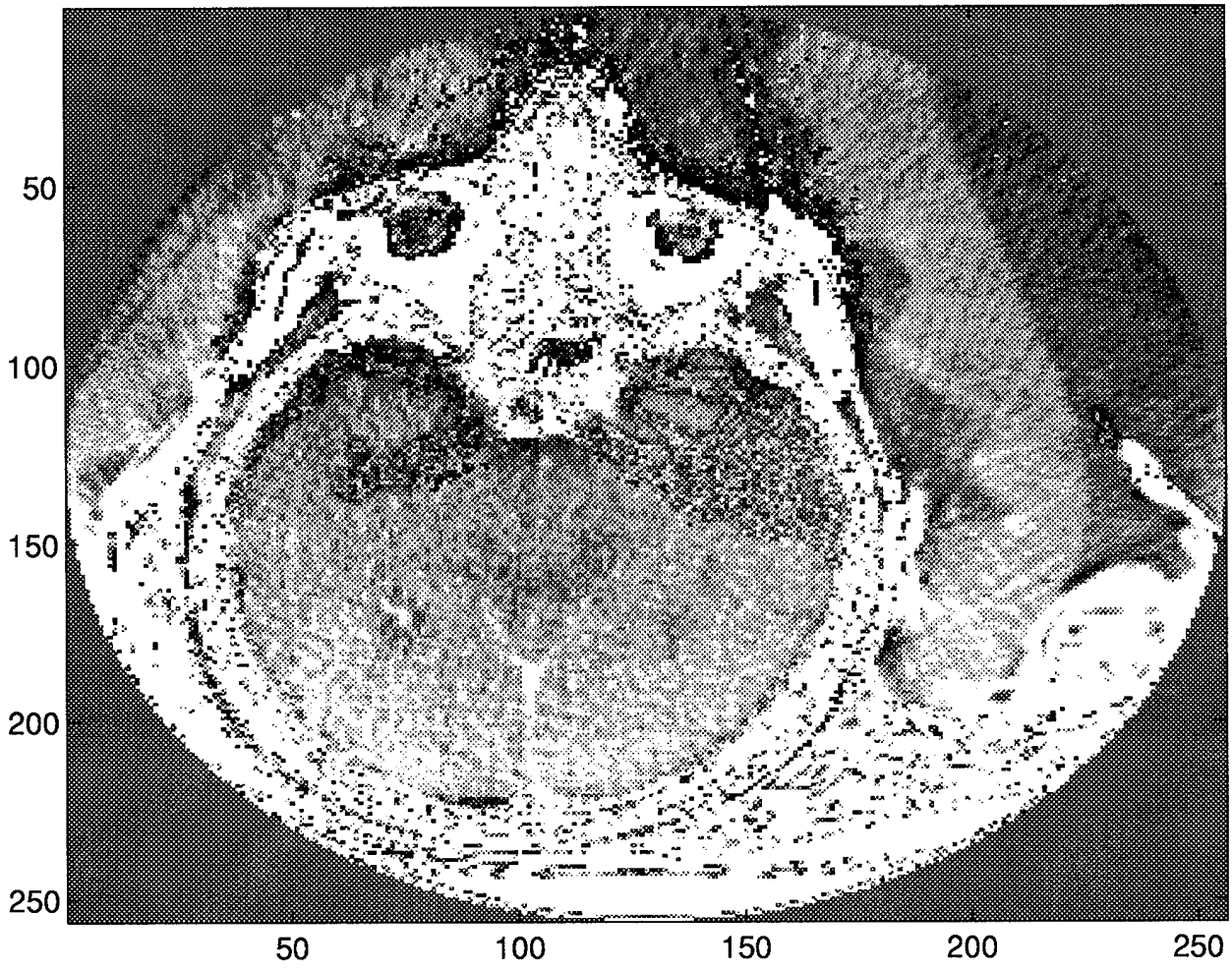


Figure 4.8 The intermediate slice 51 using Fourier interpolation and without applying the matching method.

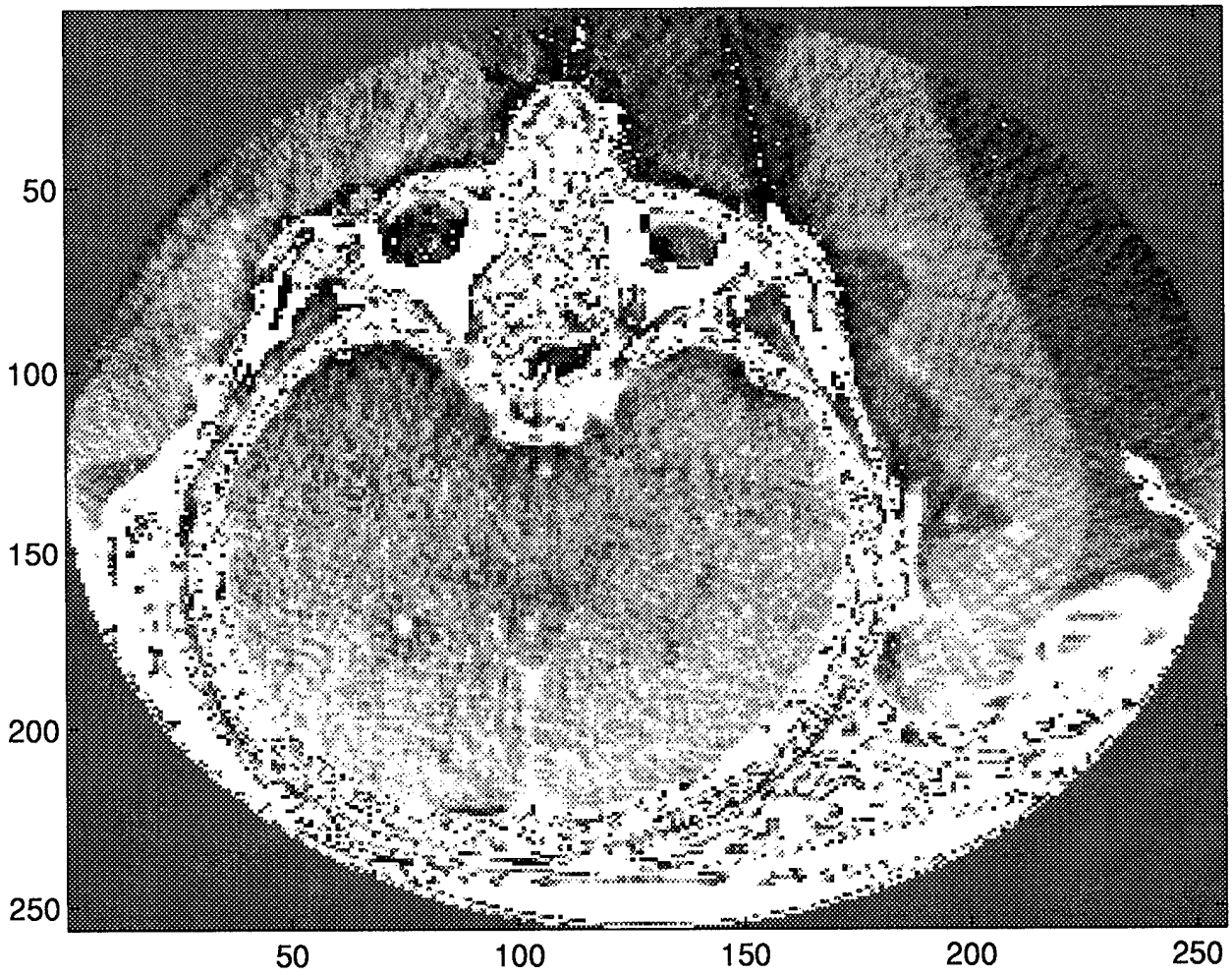


Figure 4.9 The intermediate slice 51 using Fourier interpolation and applying the matching method.

4.5 Comparison of Interpolated Images

In this section, we will compare the quality of images generated by the three different interpolation algorithms along with the matching method. Three comparison interpolated images are shown, respectively, in Figure 4.5, Figure 4.7, and Figure 4.9. Table 4.7 shows the results measured with the human visual perception model.

ComparisionImages	*VisualPerceptionDifference*
Original Slice 51	196.1584
Intermediate Slice 51 (with linear interpolation)	
Original Slice 51	226.8381
Intermediate Slice 51 (with cubic spline interpolation)	
Original Slice 51	236.4418
Intermediate Slice 51 (with Fourier interpolation)	

Table 4.7 Visual perception differences between original slice 51 and three different interpolated images

4.6 Conclusion

This chapter presented the results of interpolation images using three different interpolation algorithms. The image matching method was demonstrated to provide better interpolation results. Comparing the interpolated images, we notice that without applying the image matching method we lose information or make noise. The image qualities were measured by a human visual perception model. The results are images processed with linear interpolation and applying the images processed by matching method generate the best interpolated image.

V. CONCLUSION AND RECOMMENDATION

The conclusions and recommendations presented here are based upon the results detailed in the previous chapters.

5.1 Imaging Matching Method

Before the interpolation process is carried out, the major problem of selecting the right transformation function for mapping one image into another has to be solved. The matching method, used in this research, is a locally sensitive matching process which was developed for matching of tomographic slices. This method works on images that do not have scaling differences, have very small or no translation and rotation differences, but may have significant local geometric differences. This method has only a few parameters that need be adjusted and once these parameters are fixed, the method works without any user interaction. The method also uses intensities and gradient of points in the images to establish correspondence.

It is very difficult to design a signal image matching method that can match all types of images. The method used in this research is designed especially for the matching of tomographic images.

5.2 Interpolation Methods

Because the actual relationship between two known points is unknown, we have to assume one and test it. In this research, we use three interpolation algorithms : linear, cubic spline, and Fourier interpolation. Linear interpolation assumes that the relationship between each two known points is a monotonically increasing or decreasing function. Cubic spline interpolation assumes that the relationship between two known points is a smooth curve function. Fourier interpolation is different from the previous methods. The intensity value is transformed from the spatial domain to the frequency domain. An appropriate lowpass filter gives the interpolated image.

5.3 Qualities of Interpolated Images

When an image is processed, the root-mean-squared error is the typical method used to measure the image quality. Although this measure has a good physical and theoretical basis, it often correlates poorly with the subjectively judged distortion of the image. This is principally due to the fact that the human visual system does not process the image in a point-by-point fashion but extracts certain spatial, temporal, and chromatic features.

In this research, we use the human visual perception model developed by Wilson (27) to measure the visual perception difference and decide the qualities of interpolated images. The result of using linear interpolation and applying the image matching method generated the best interpolated image.

5.4 Recommendation for Further Research

There are several other interesting interpolation methods that can be applied. Kriging is one of the popular methods. It differs from the methods used in this research because it is a geo-statistical method. Since medical data is spatially distributed and has regions in which sample values are highly correlated (e.g., bone and tissue regions), this technique is applicable to medical data.

Further study is also recommended to construct a more detailed human visual perception model to measure the images.

Appendix A. Experimental Result Images

This Appendix provides figures for every experimental result images in finding the appropriate parameter values of the matching method cost function.

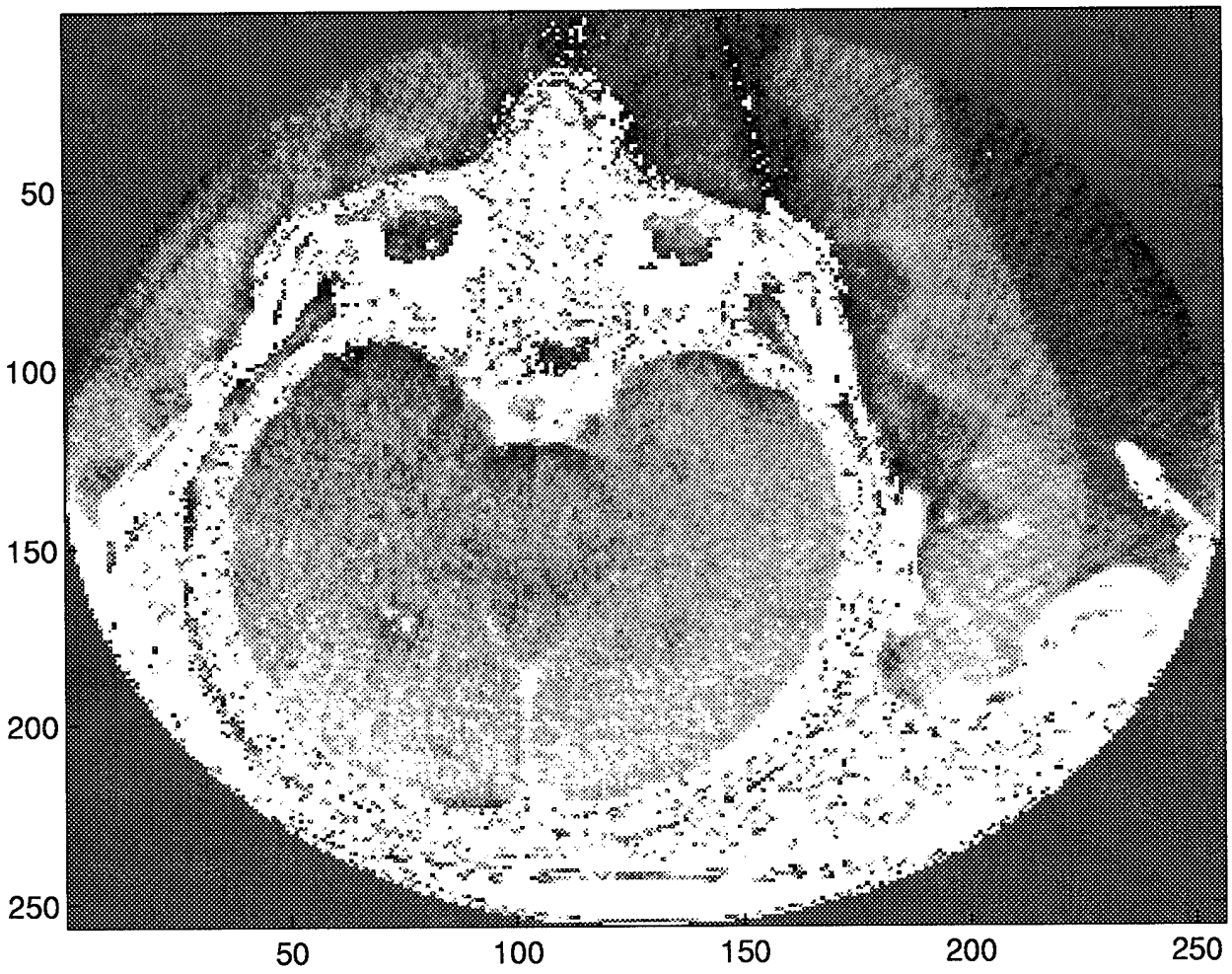


Figure A.1 Interpolated slice 51 with $u1 = 0.1$, $u2 = 1.0$, $u3 = 1.0$, $u4 = 1.0$

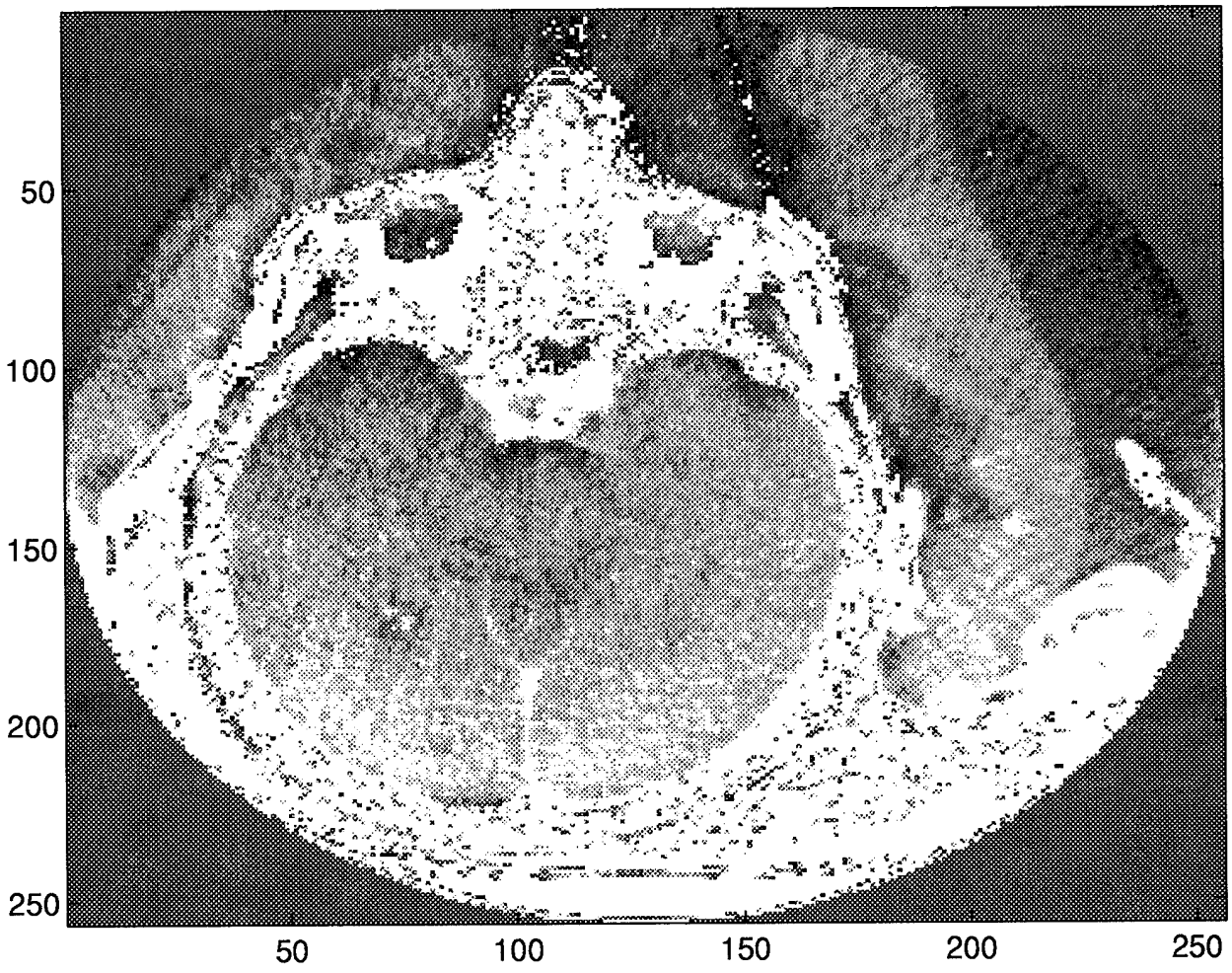


Figure A.2 Interpolated slice 51 with $u1 = 0.2$, $u2 = 1.0$, $u3 = 1.0$, $u4 = 1.0$

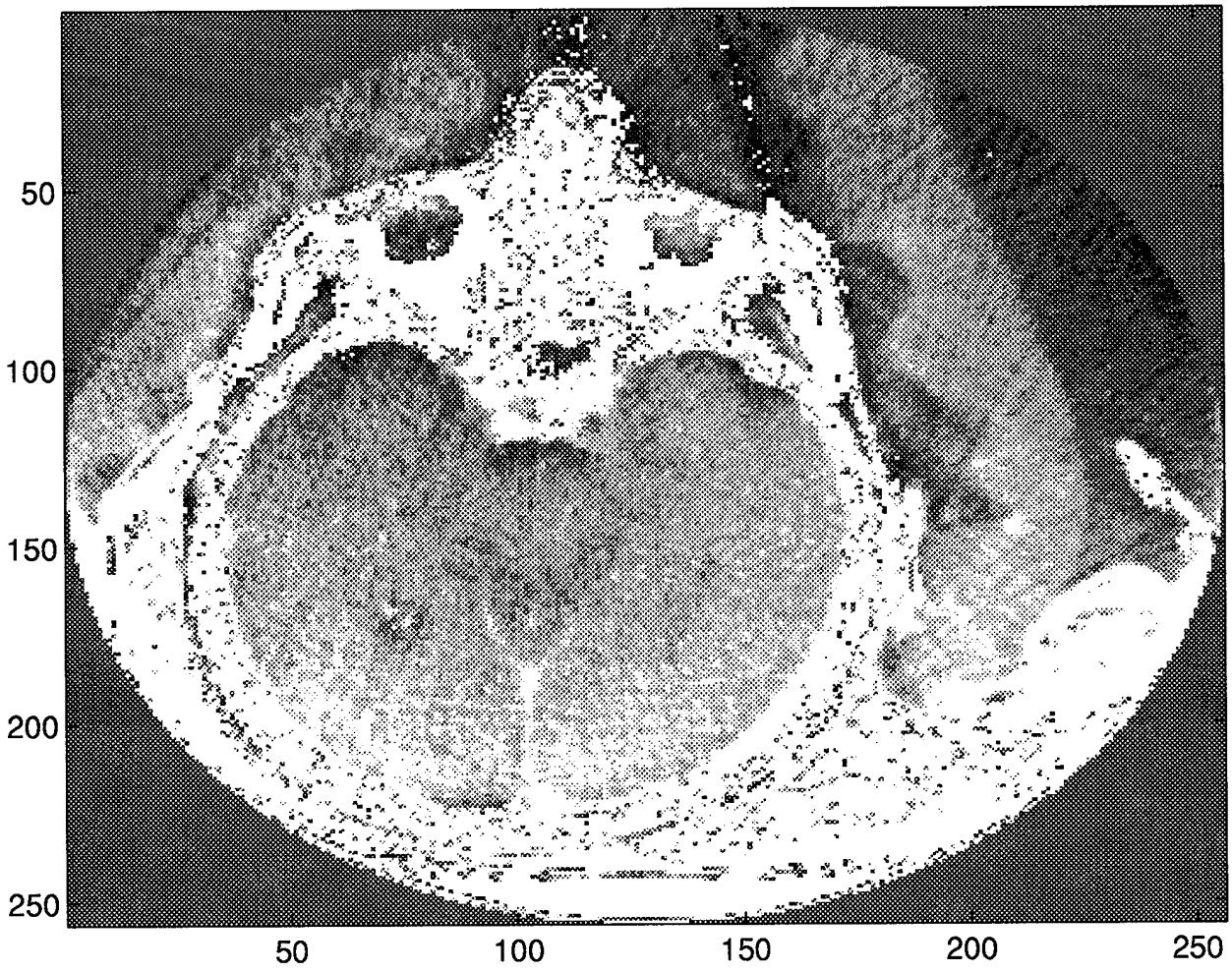


Figure A.3 Interpolated slice 51 with $u1 = 0.5$, $u2 = 1.0$, $u3 = 1.0$, $u4 = 1.0$

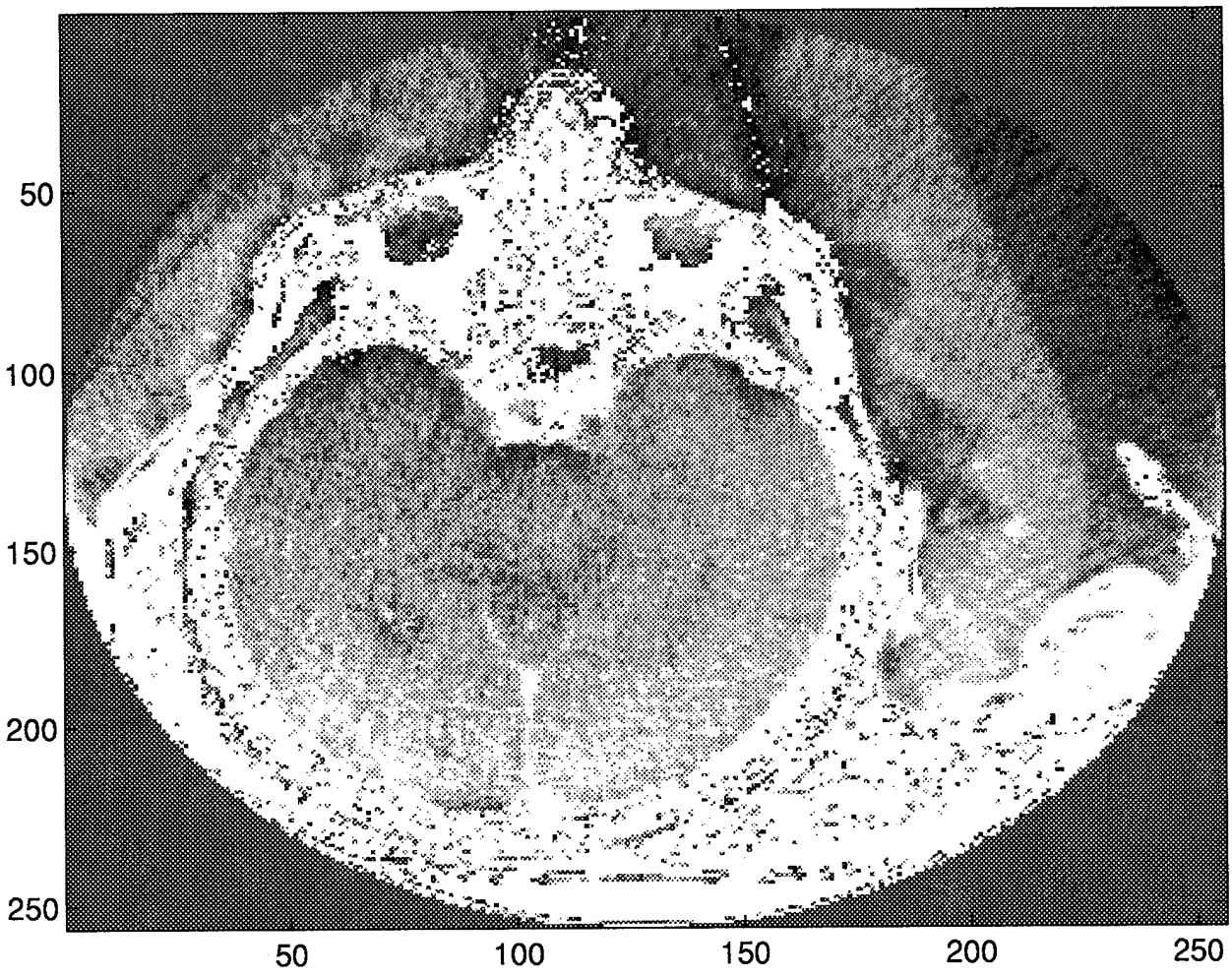


Figure A.4 Interpolated slice 51 with $u1 = 0.8$, $u2 = 1.0$, $u3 = 1.0$, $u4 = 1.0$

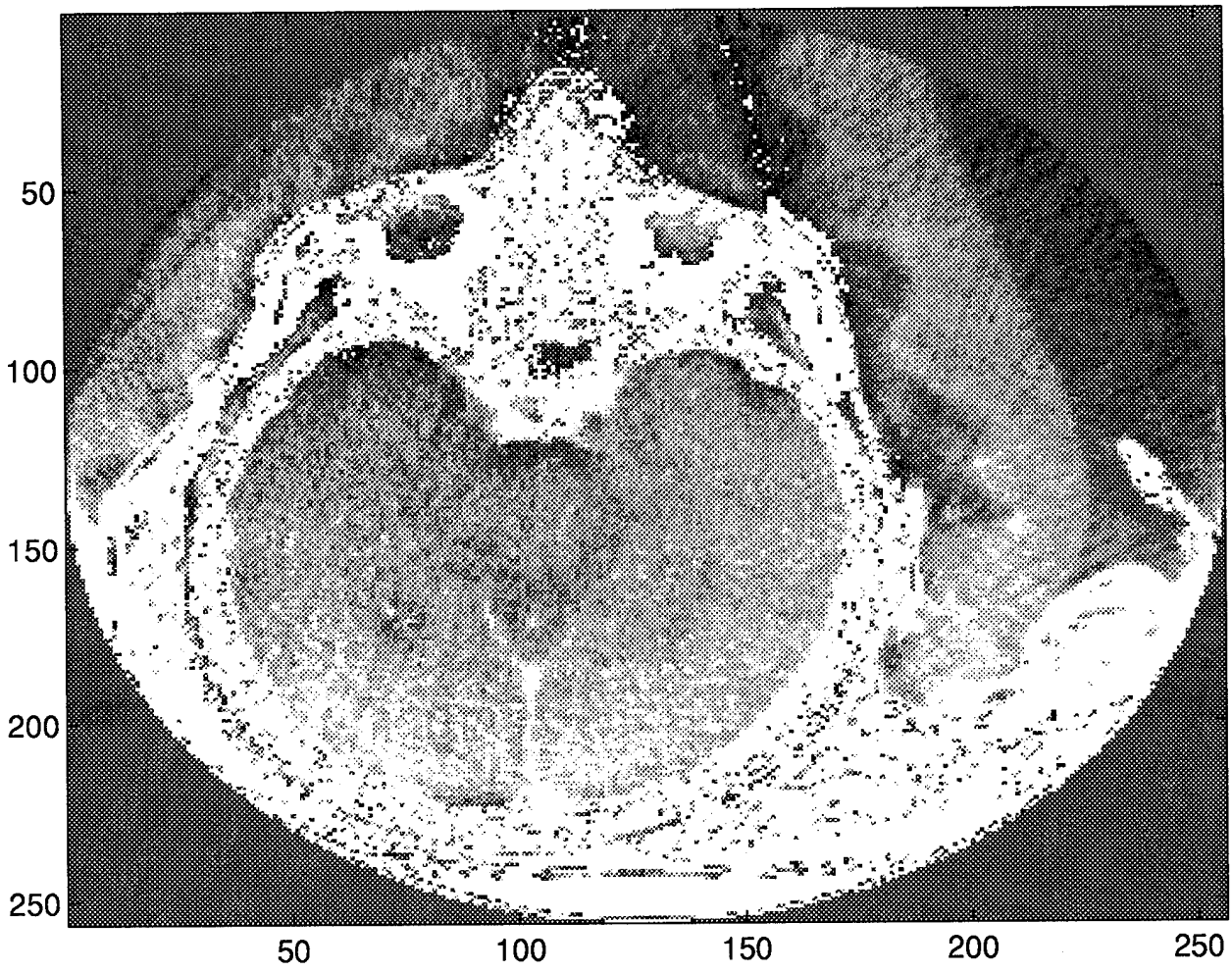


Figure A.5 Interpolated slice 51 with $u1 = 1.0$, $u2 = 1.0$, $u3 = 1.0$, $u4 = 1.0$

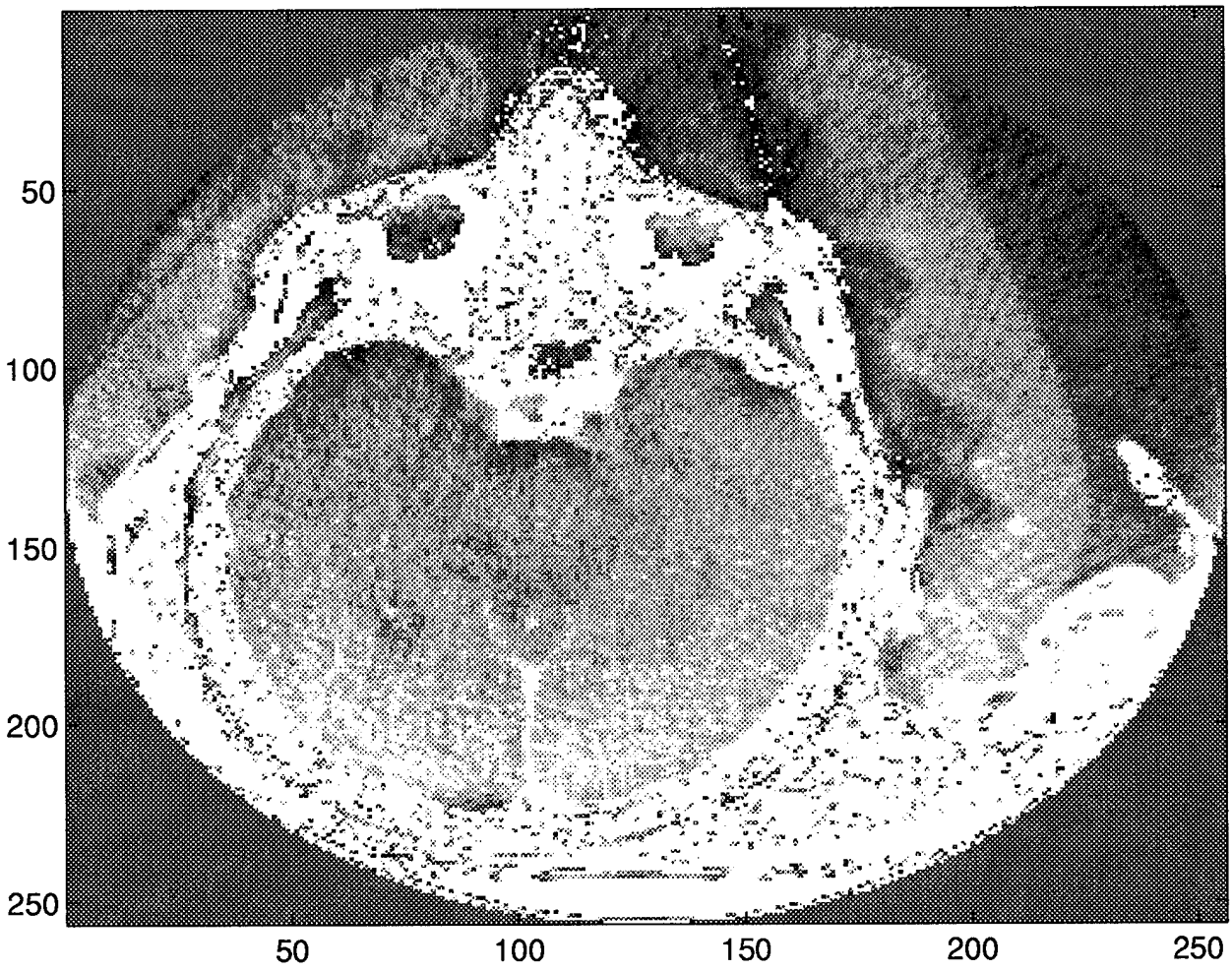


Figure A.6 Interpolated slice 51 with $u1 = 0.5$, $u2 = 0.1$, $u3 = 1.0$, $u4 = 1.0$

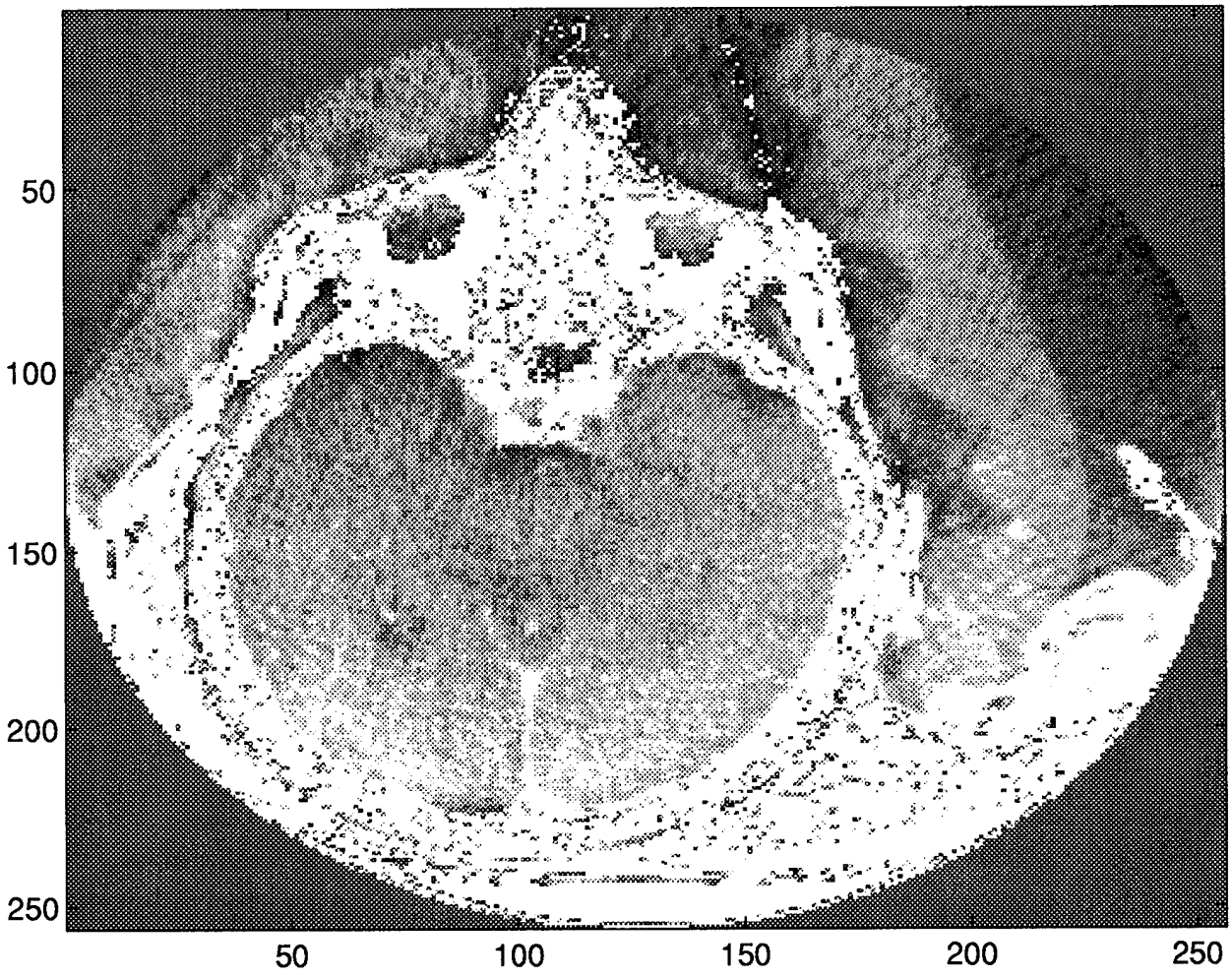


Figure A.7 Interpolated slice 51 with $u1 = 0.5$, $u2 = 0.2$, $u3 = 1.0$, $u4 = 1.0$

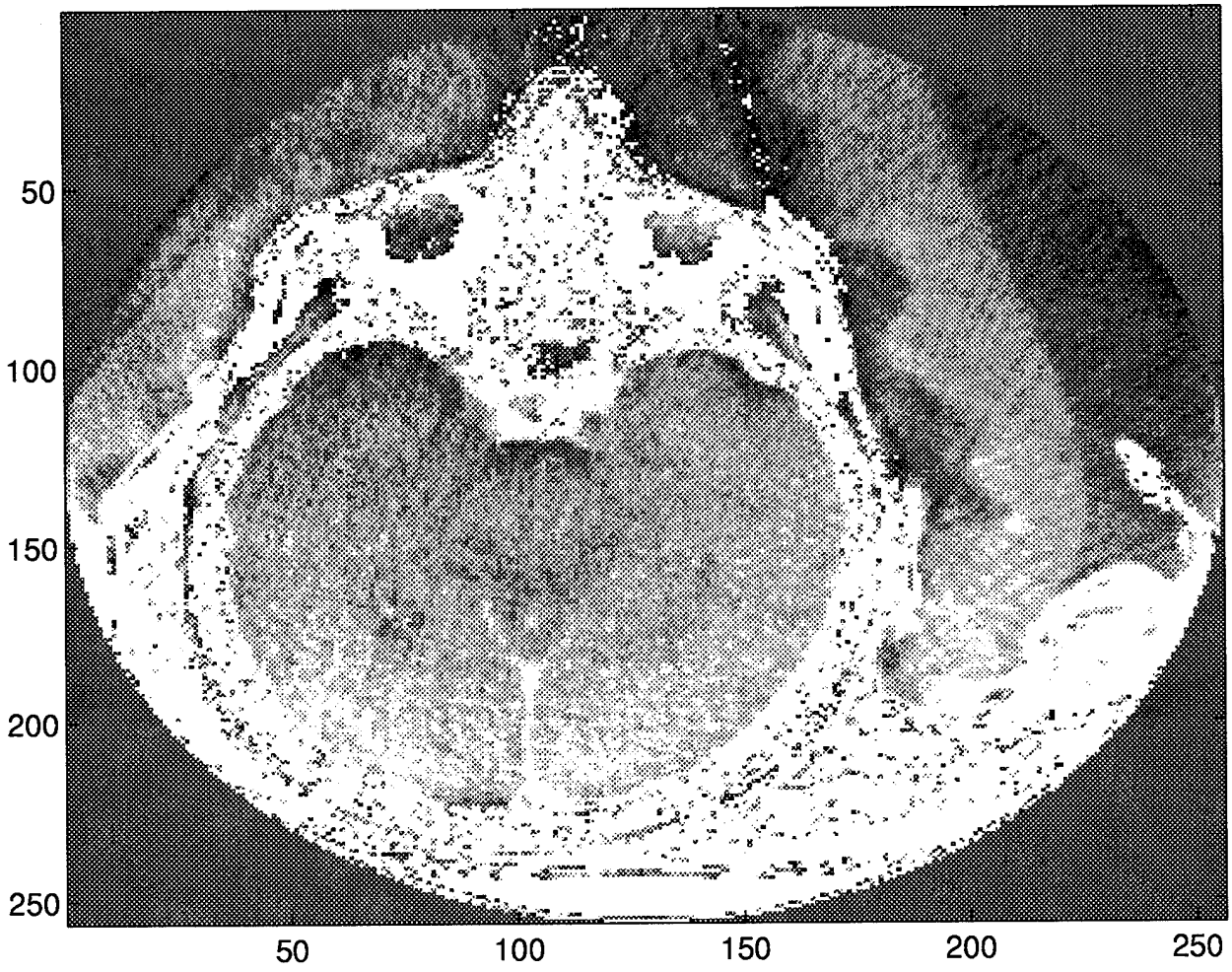


Figure A.8 Interpolated slice 51 with $u1 = 0.5$, $u2 = 0.4$, $u3 = 1.0$, $u4 = 1.0$

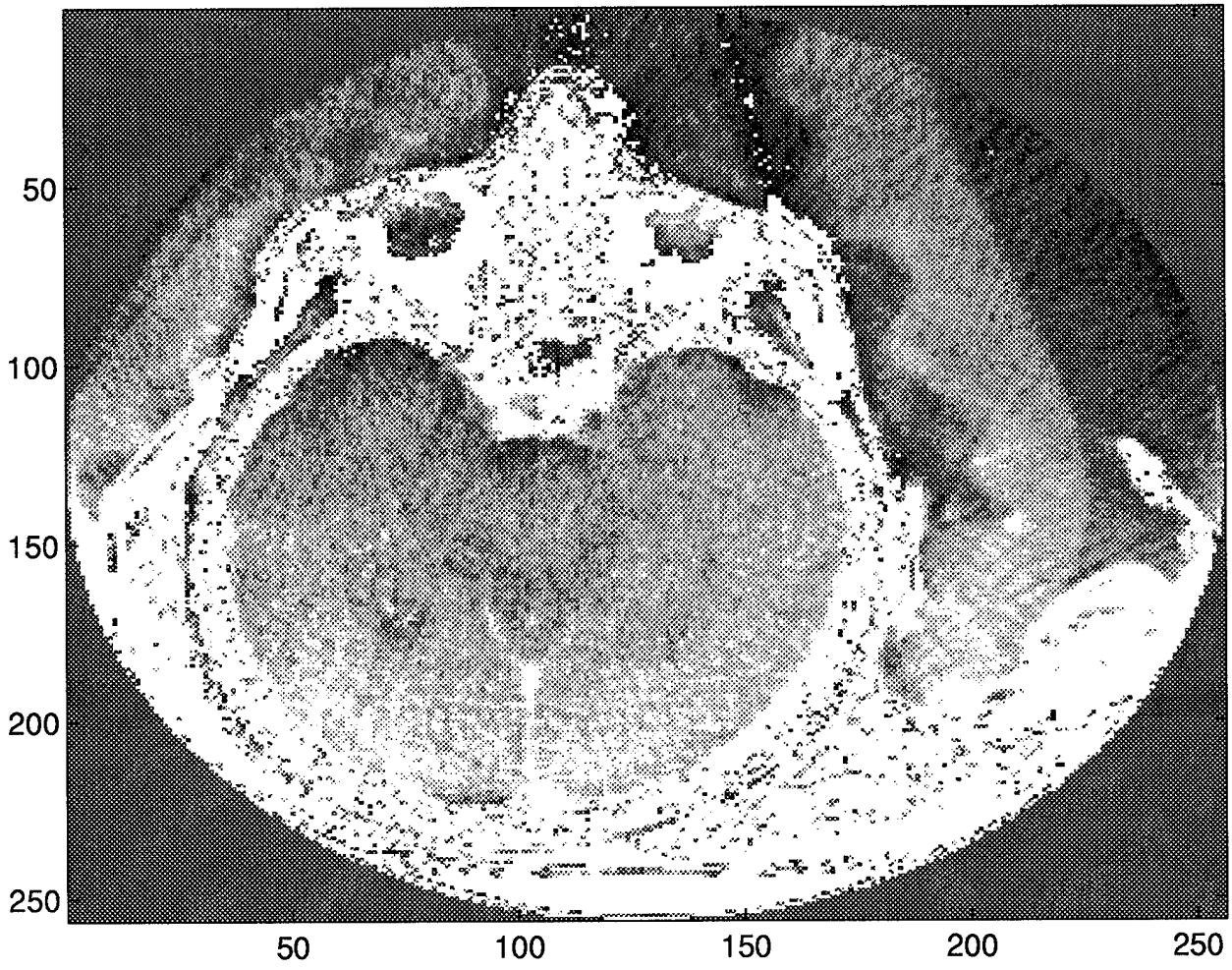


Figure A.9 Interpolated slice 51 with $u1 = 0.5$, $u2 = 0.8$, $u3 = 1.0$, $u4 = 1.0$

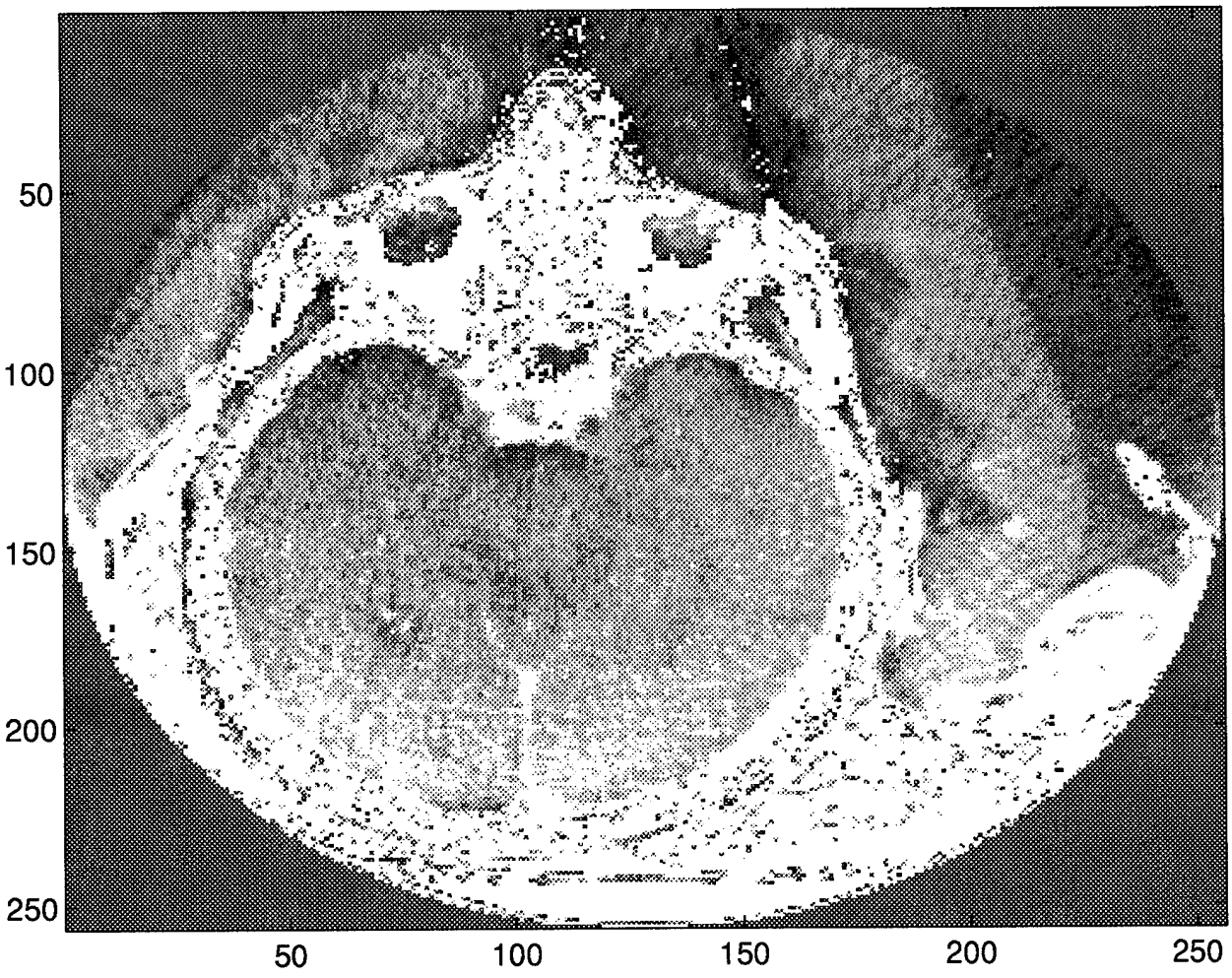


Figure A.10 Interpolated slice 51 with $u1 = 0.5$, $u2 = 1.0$, $u3 = 1.0$, $u4 = 1.0$

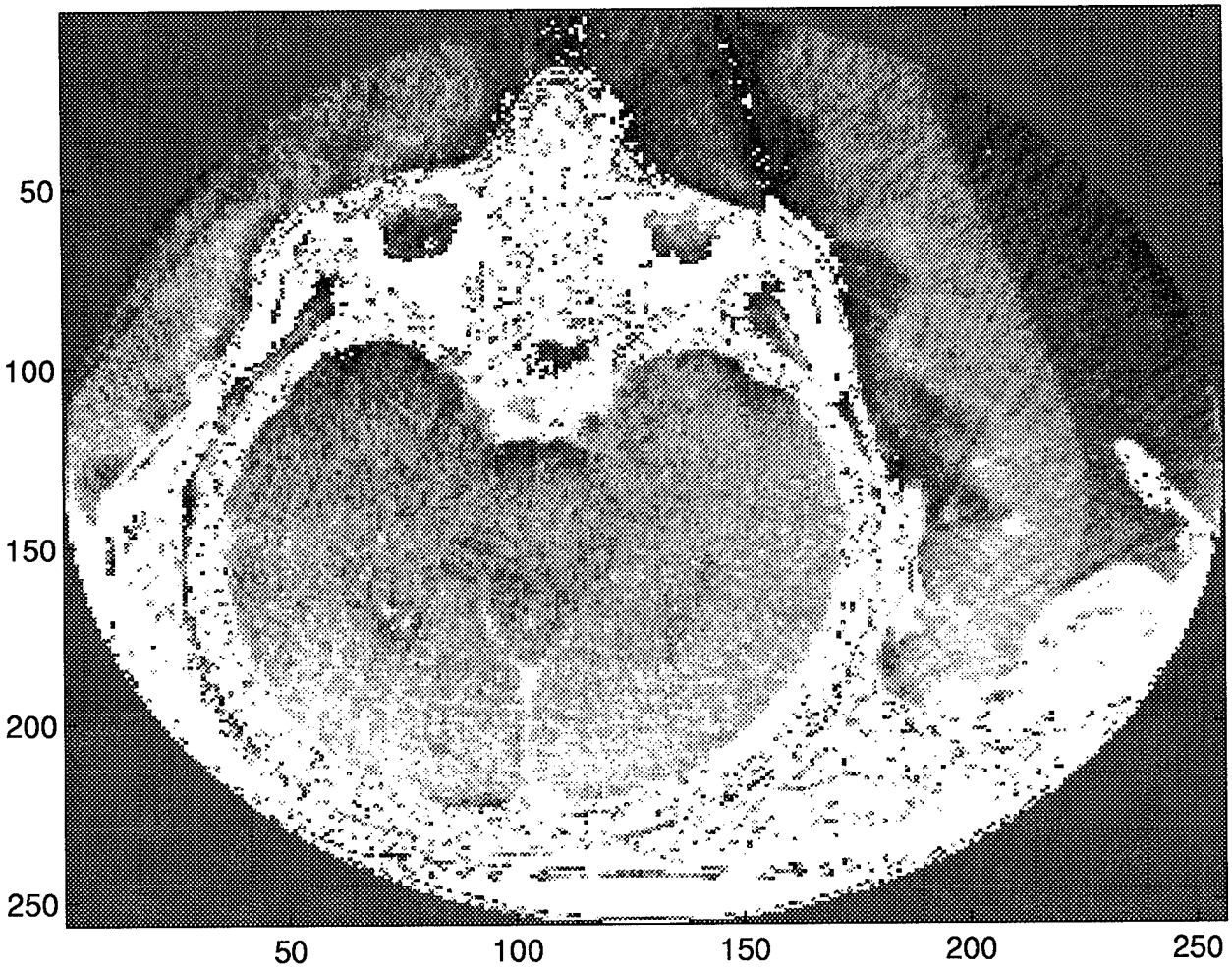


Figure A.11 Interpolated slice 51 with $u1 = 0.5$, $u2 = 0.4$, $u3 = 0.1$, $u4 = 1.0$

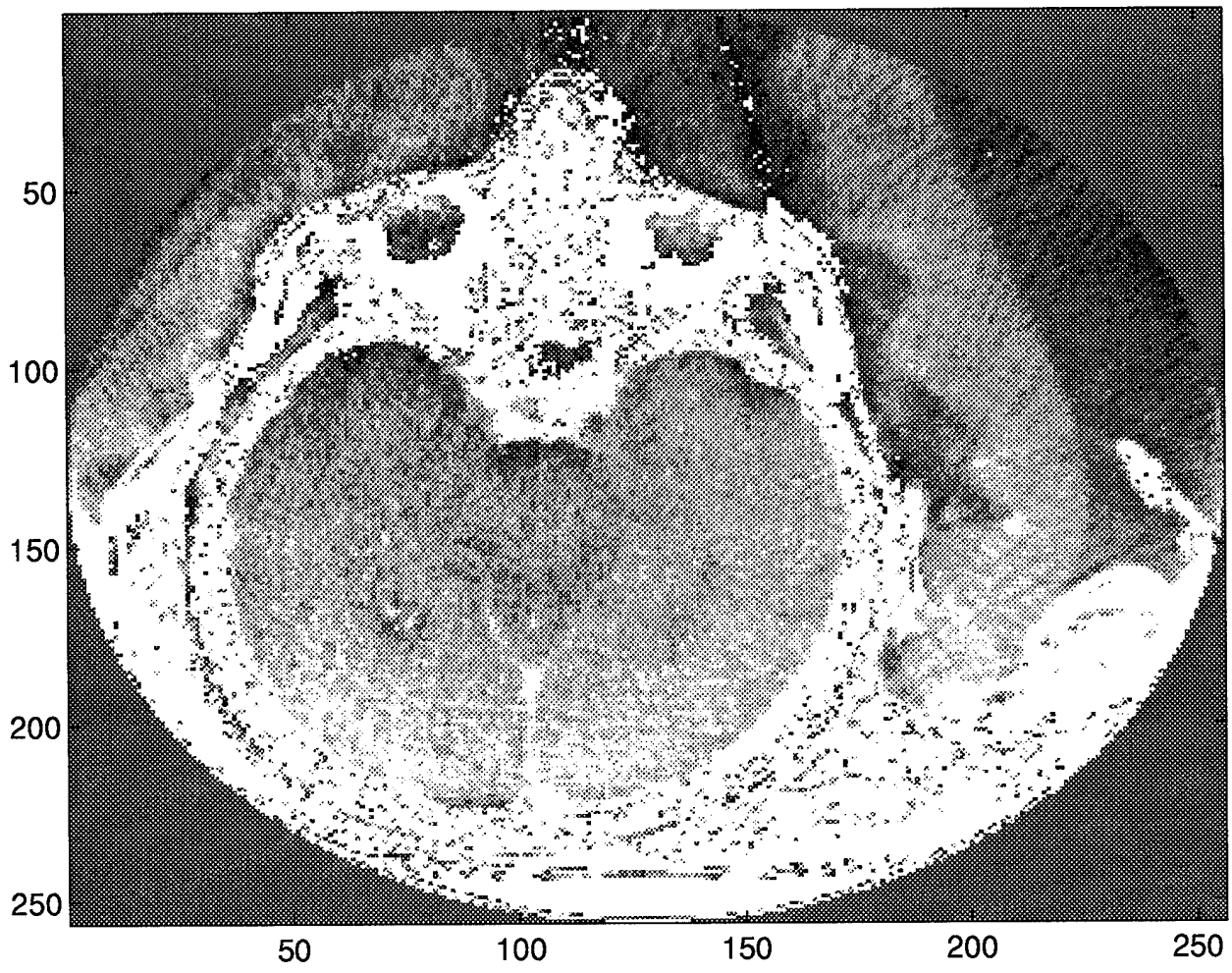


Figure A.12 Interpolated slice 51 with $u1 = 0.5$, $u2 = 0.4$, $u3 = 0.2$, $u4 = 1.0$

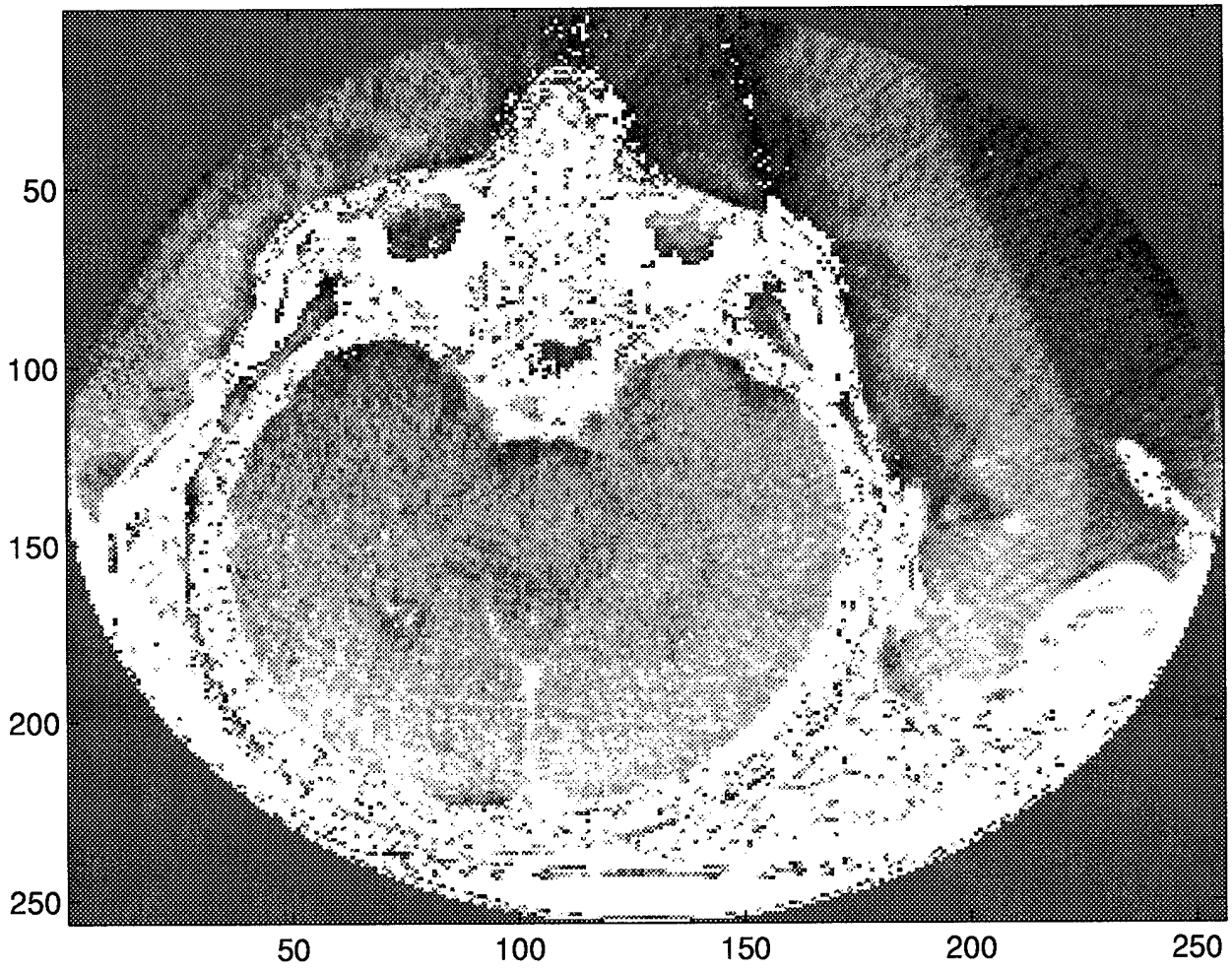


Figure A.13 Interpolated slice 51 with $u1 = 0.5$, $u2 = 0.4$, $u3 = 0.5$, $u4 = 1.0$

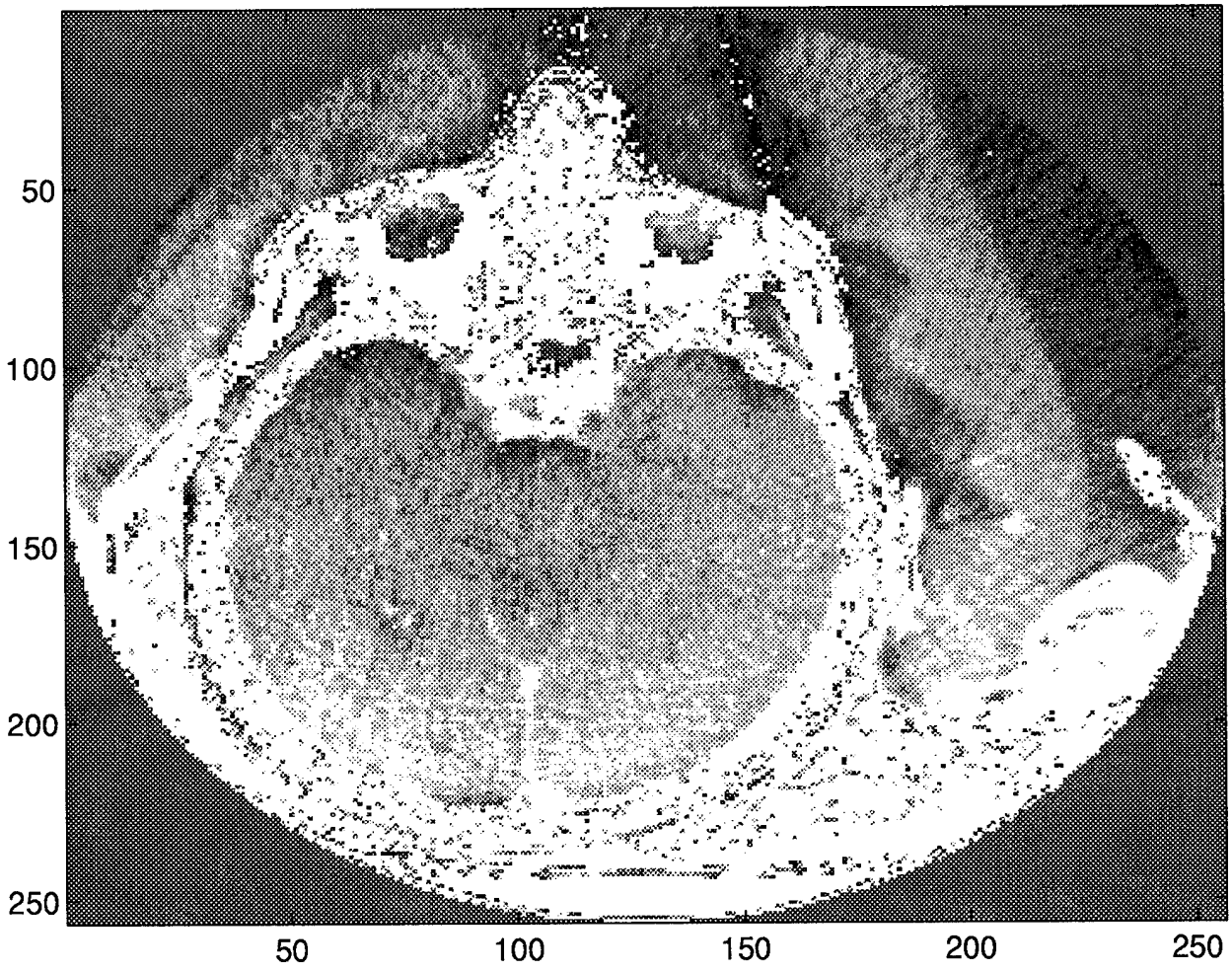


Figure A.14 Interpolated slice 51 with $u1 = 0.5$, $u2 = 0.4$, $u3 = 0.8$, $u4 = 1.0$

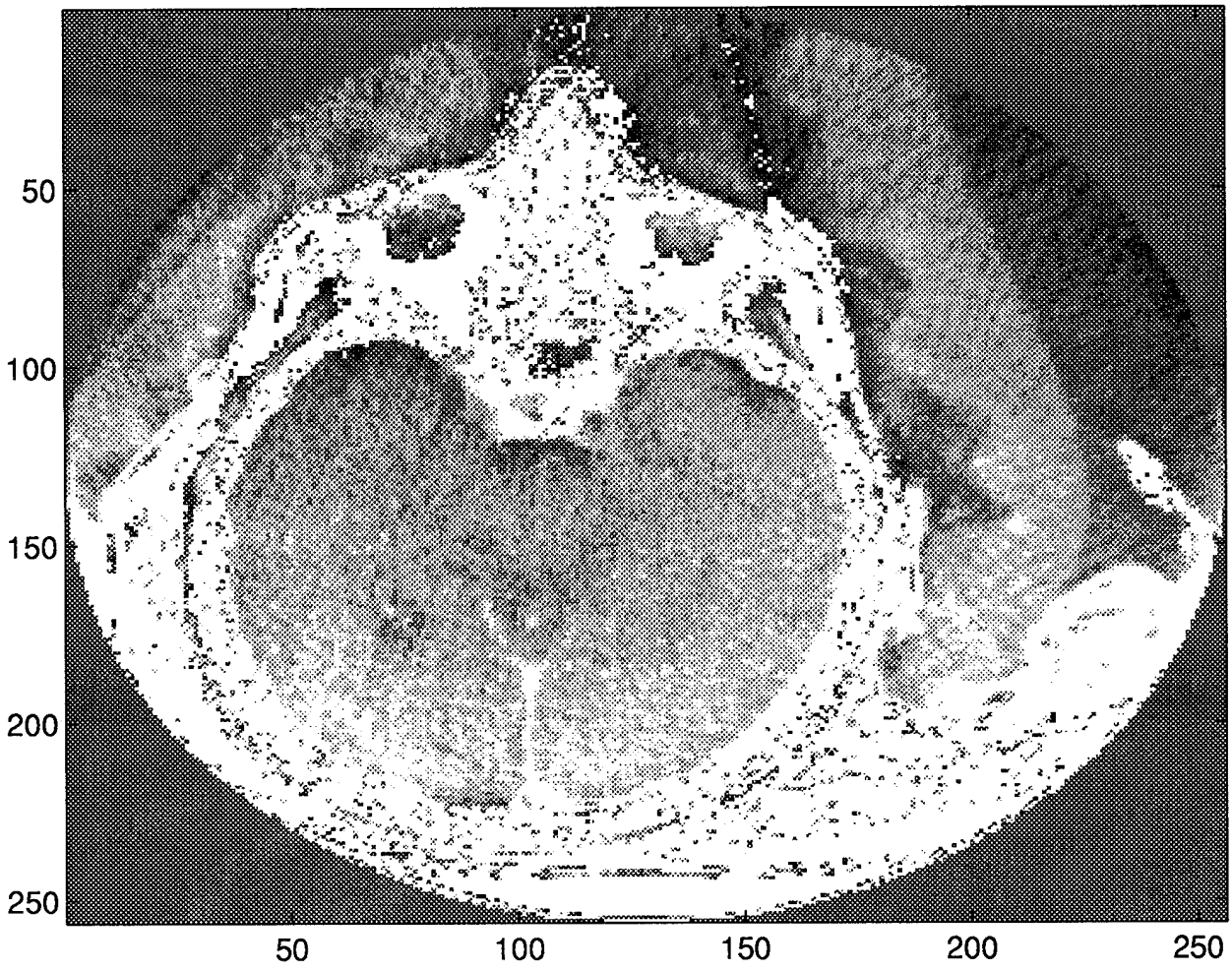


Figure A.15 Interpolated slice 51 with $u1 = 0.5$, $u2 = 0.4$, $u3 = 1.0$, $u4 = 1.0$

Appendix B. Test Images

This Appendix provides figures used in three interpolation processes.

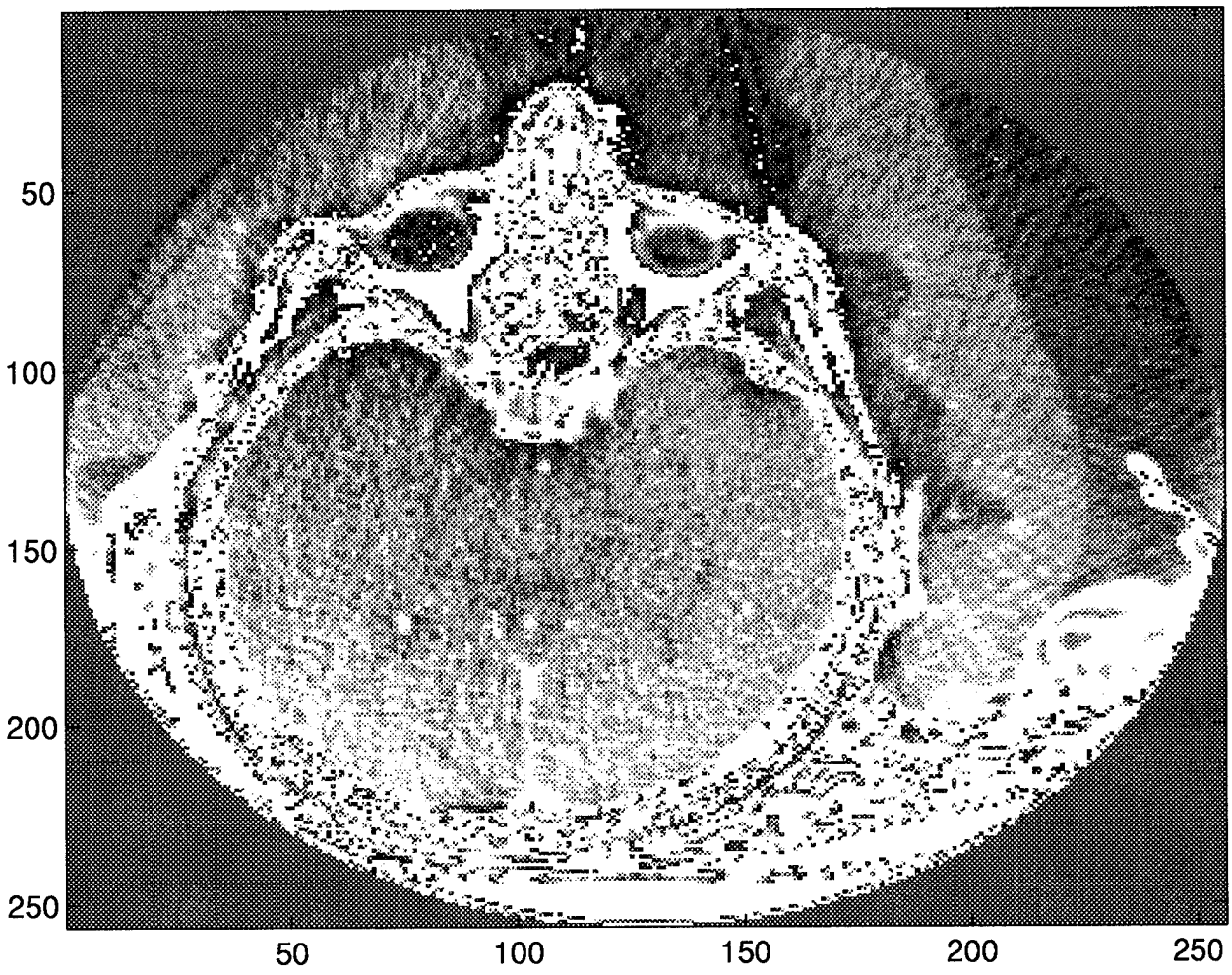


Figure B.1 Original CT Slice 50.

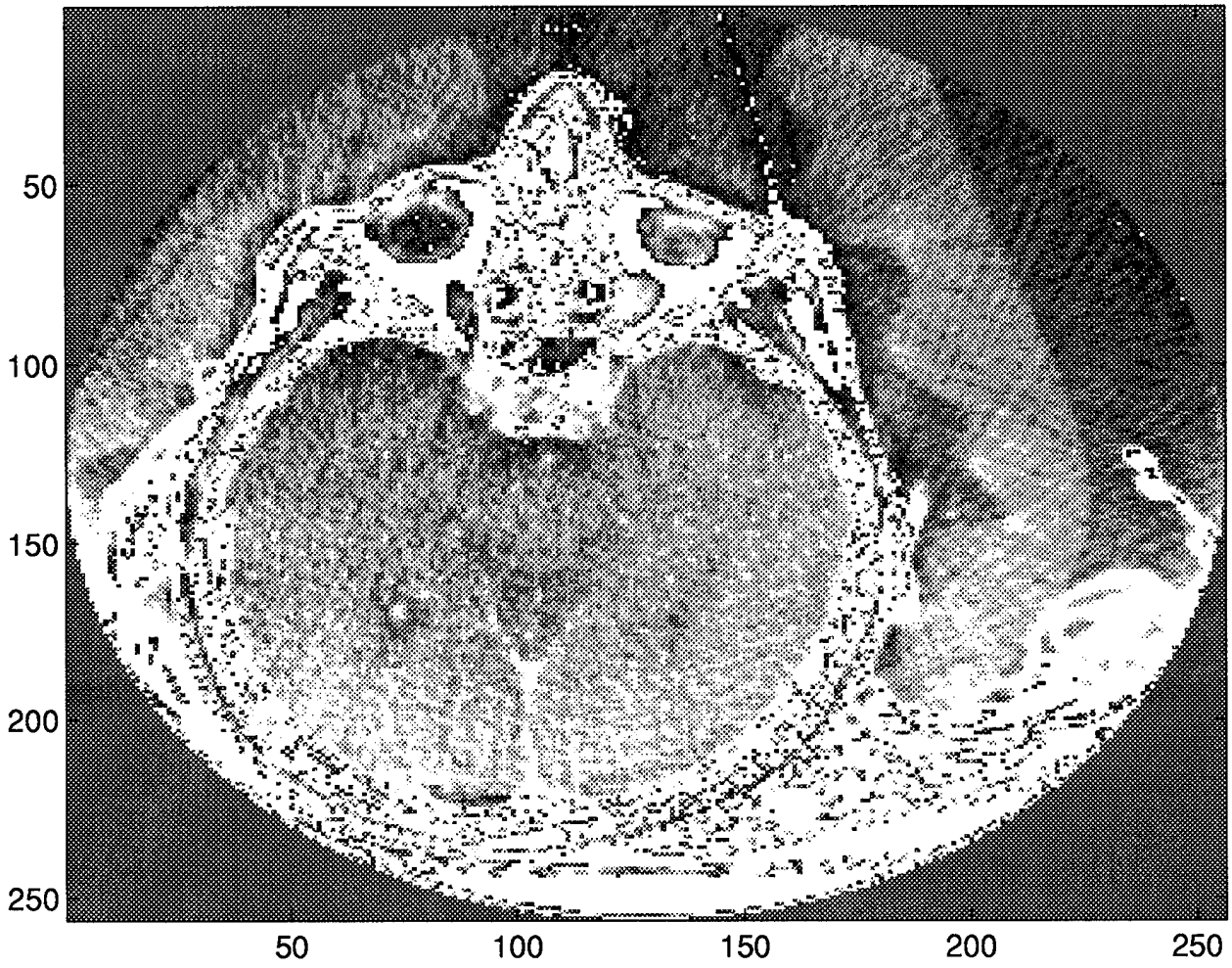


Figure B.2 Original CT Slice 51.

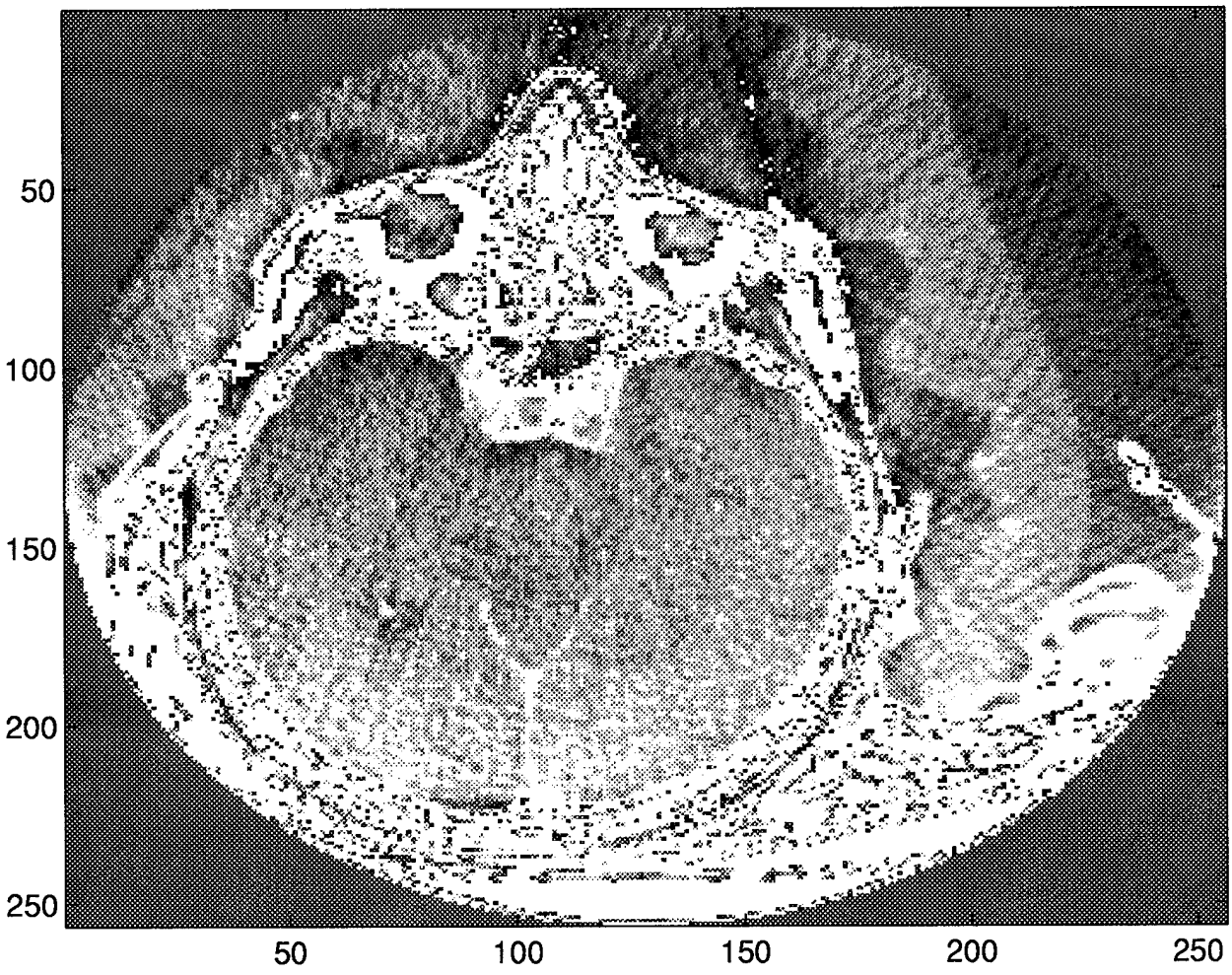


Figure B.3 Original CT Slice 52.

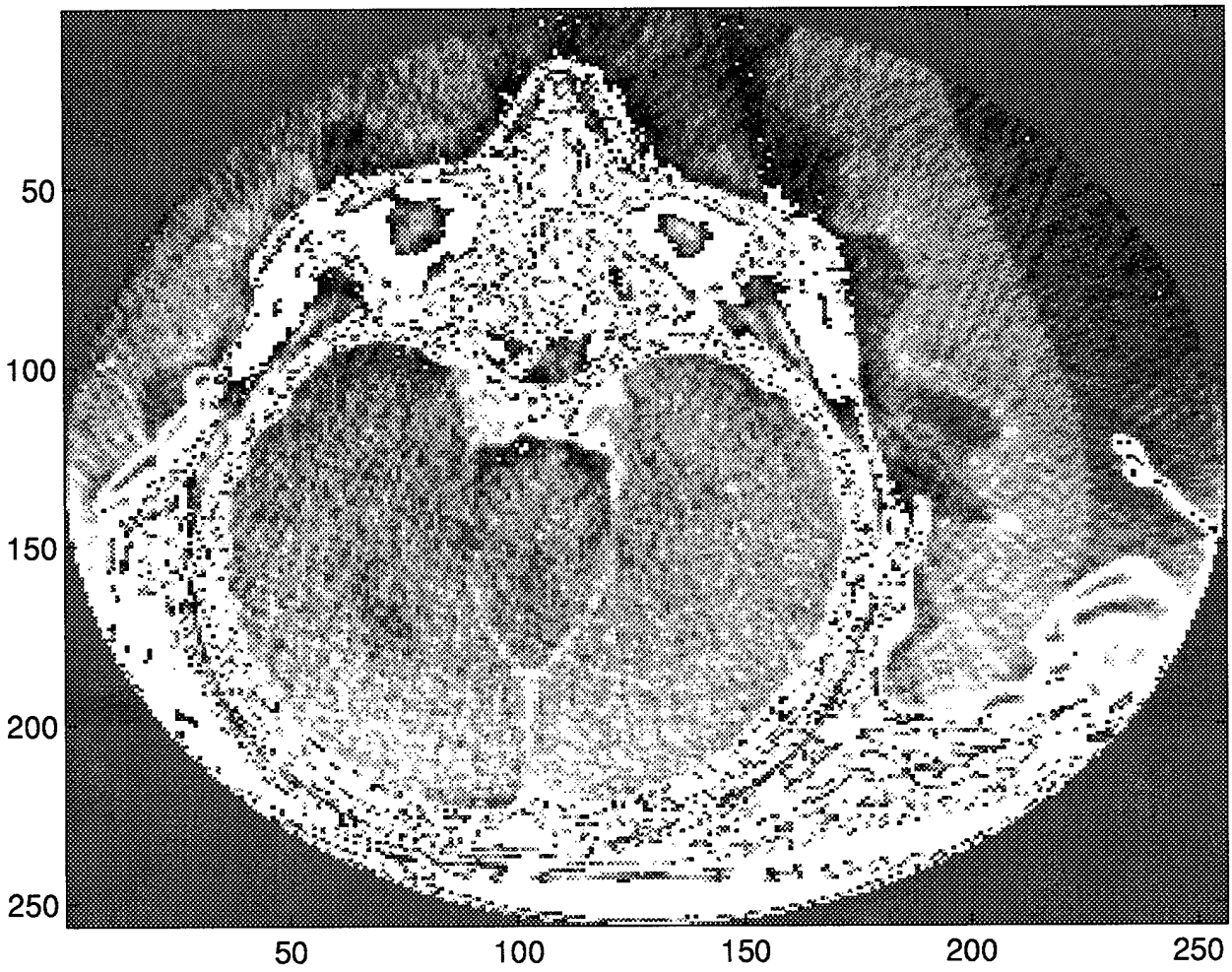


Figure B.4 Original CT Slice 53.

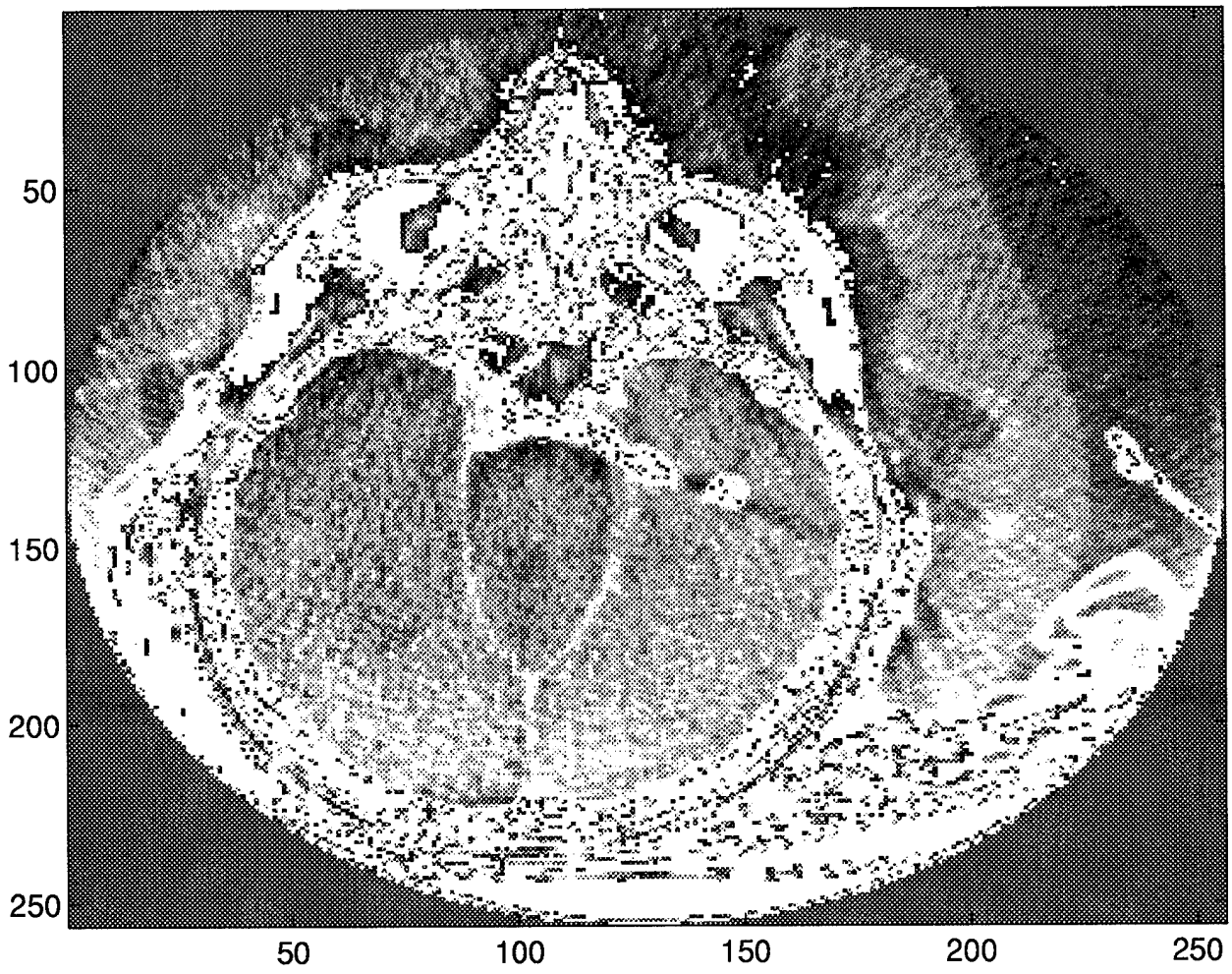


Figure B.5 Original CT Slice 54.

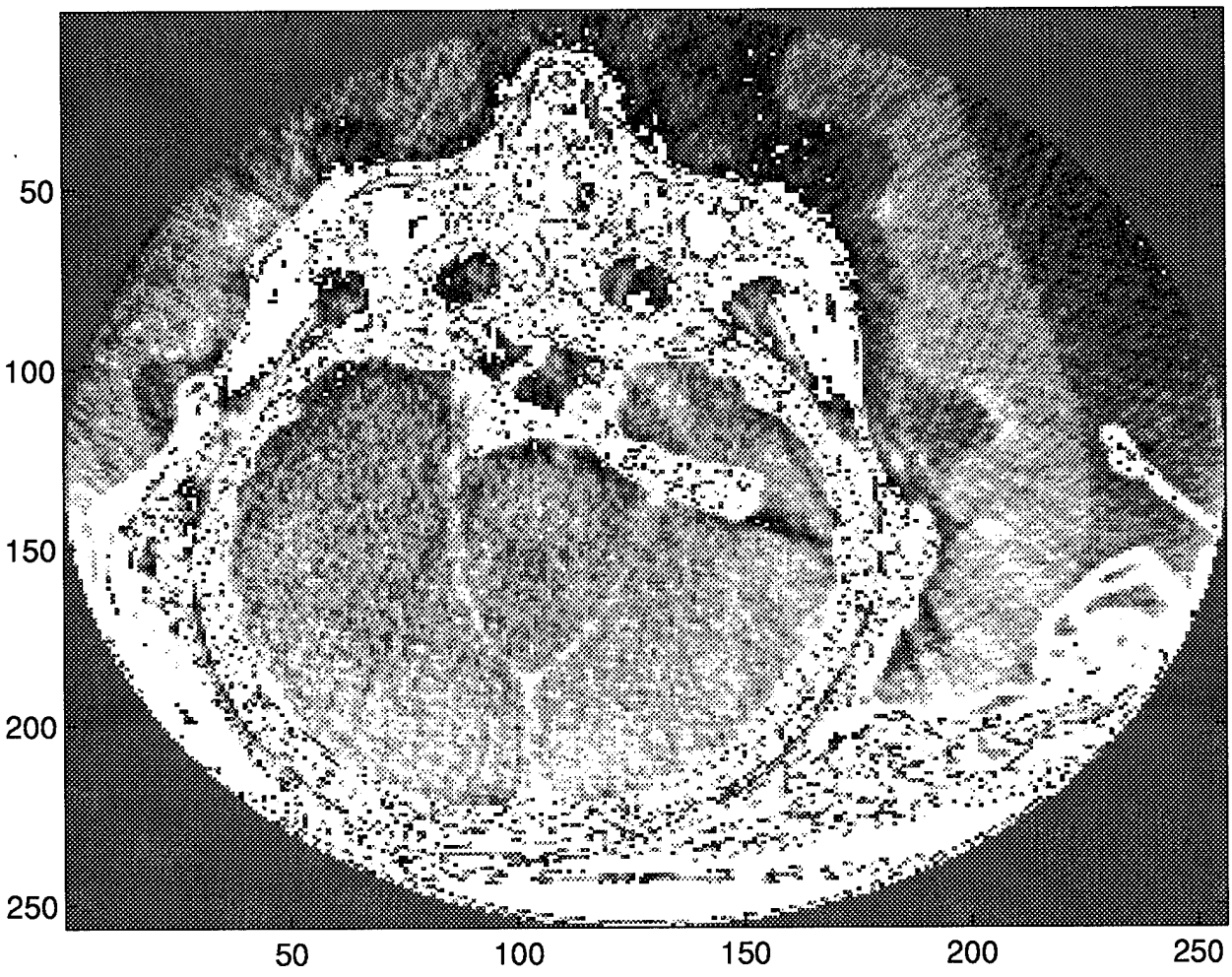


Figure B.6 Original CT Slice 55.

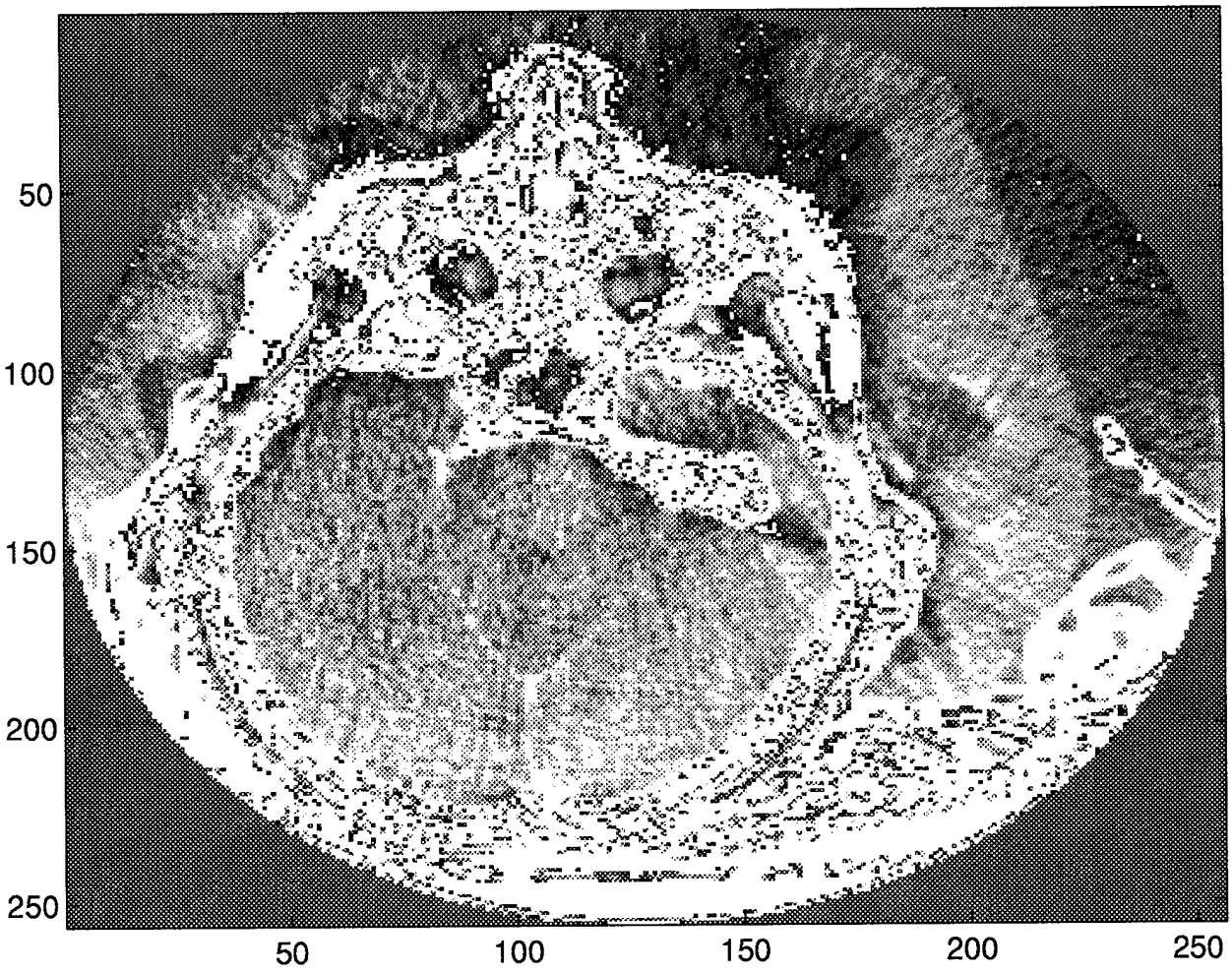


Figure B.7 Original CT Slice 56.

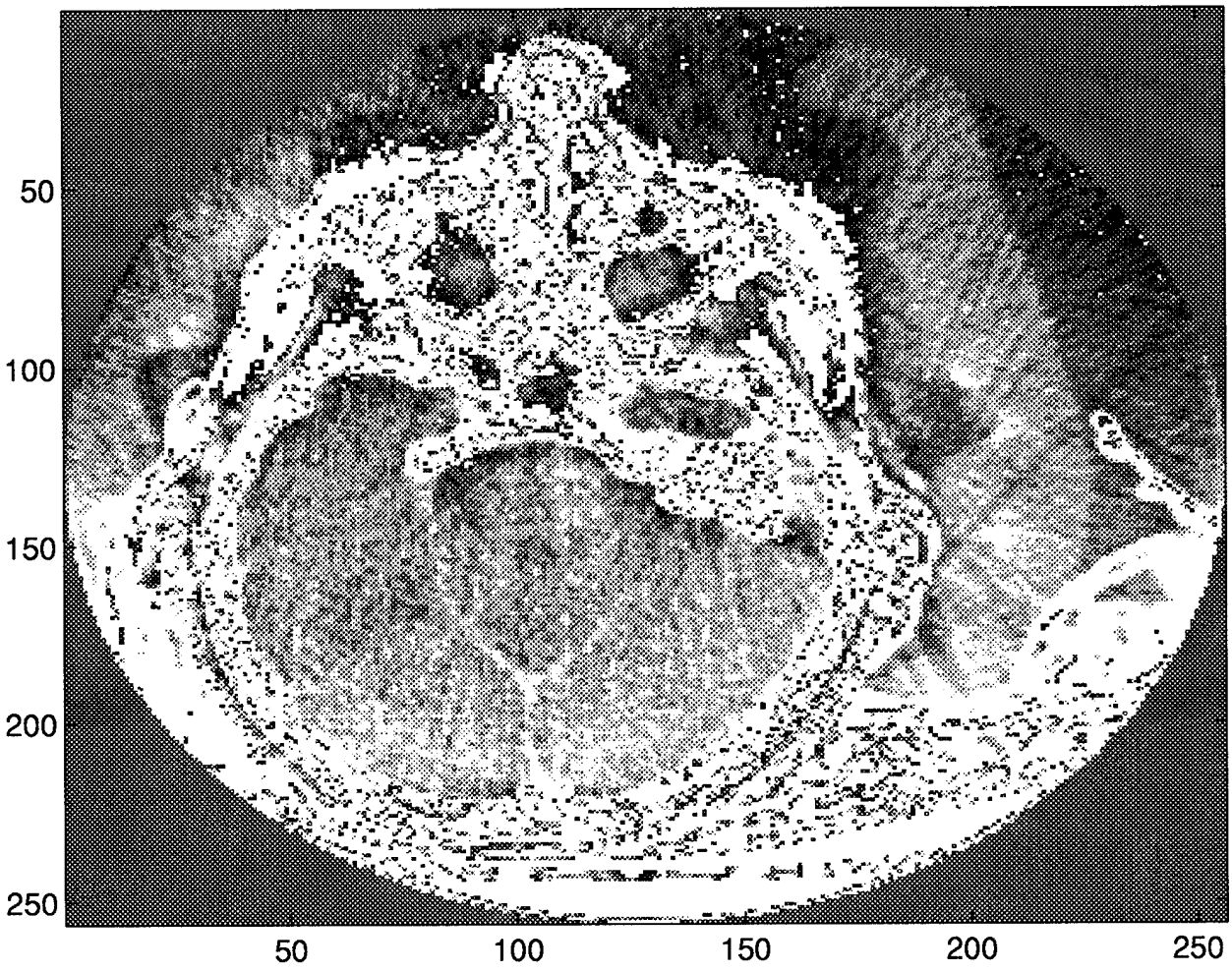


Figure B.8 Original CT Slice 57.

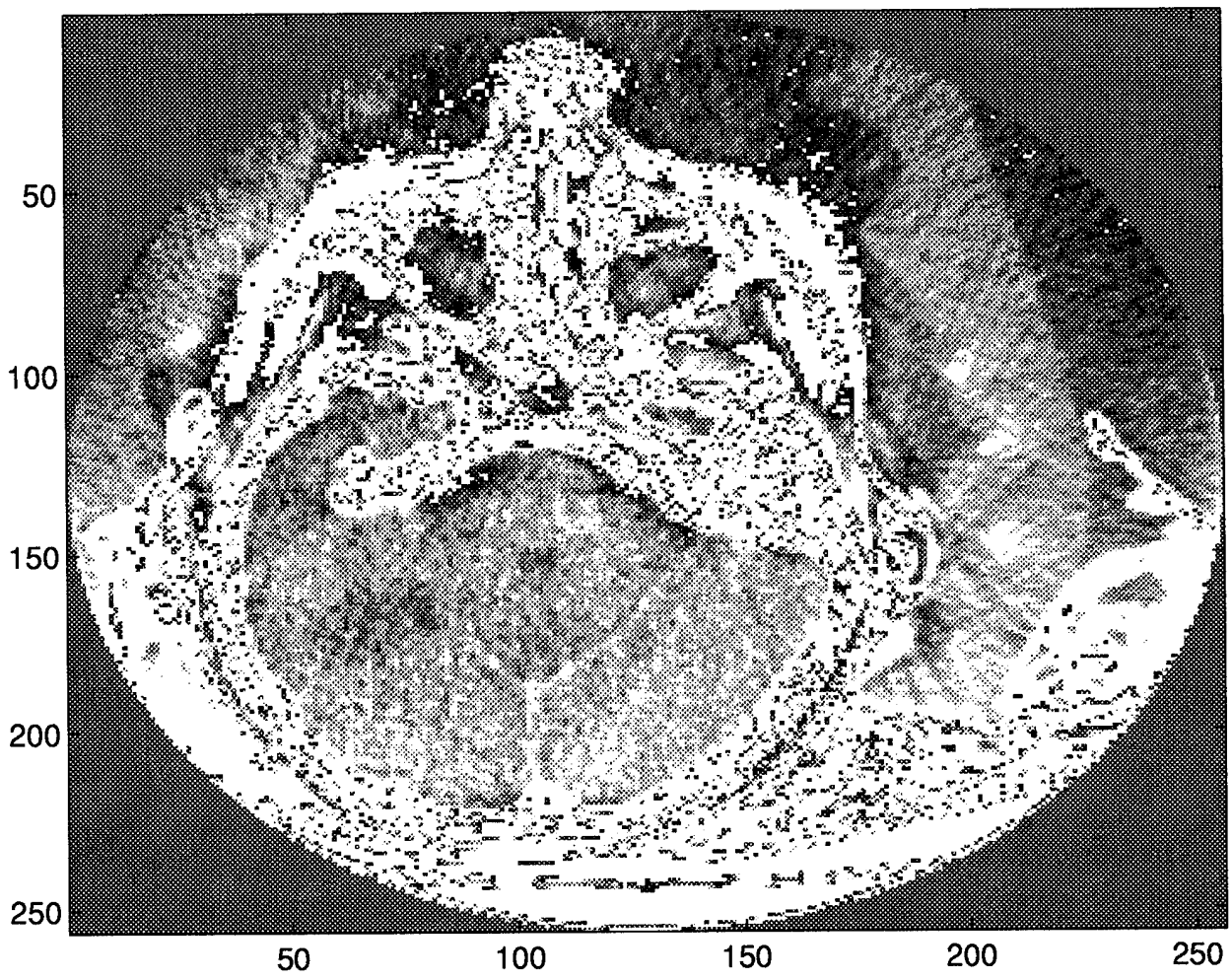


Figure B.9 Original CT Slice 58.

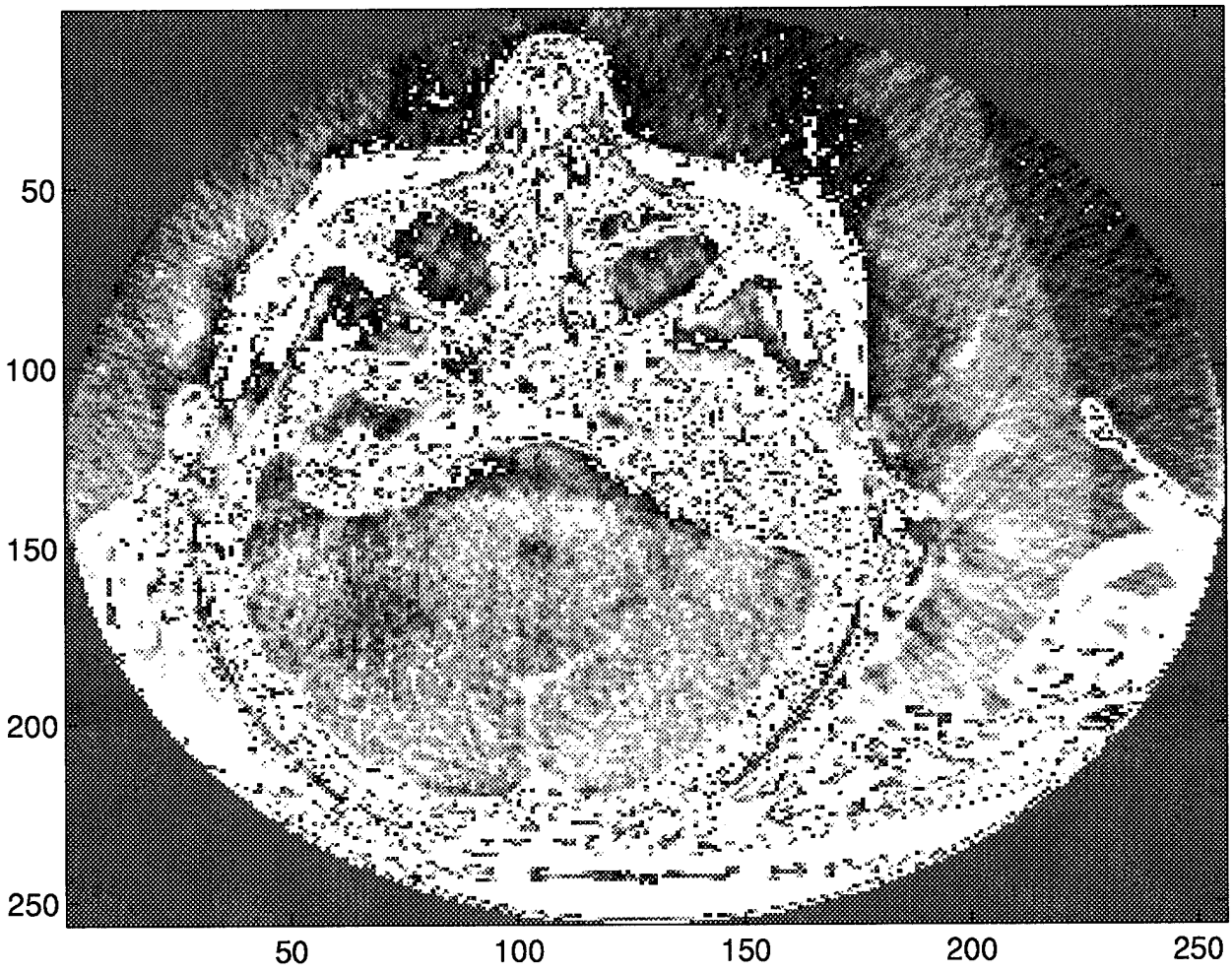


Figure B.10 Original CT Slice 59.

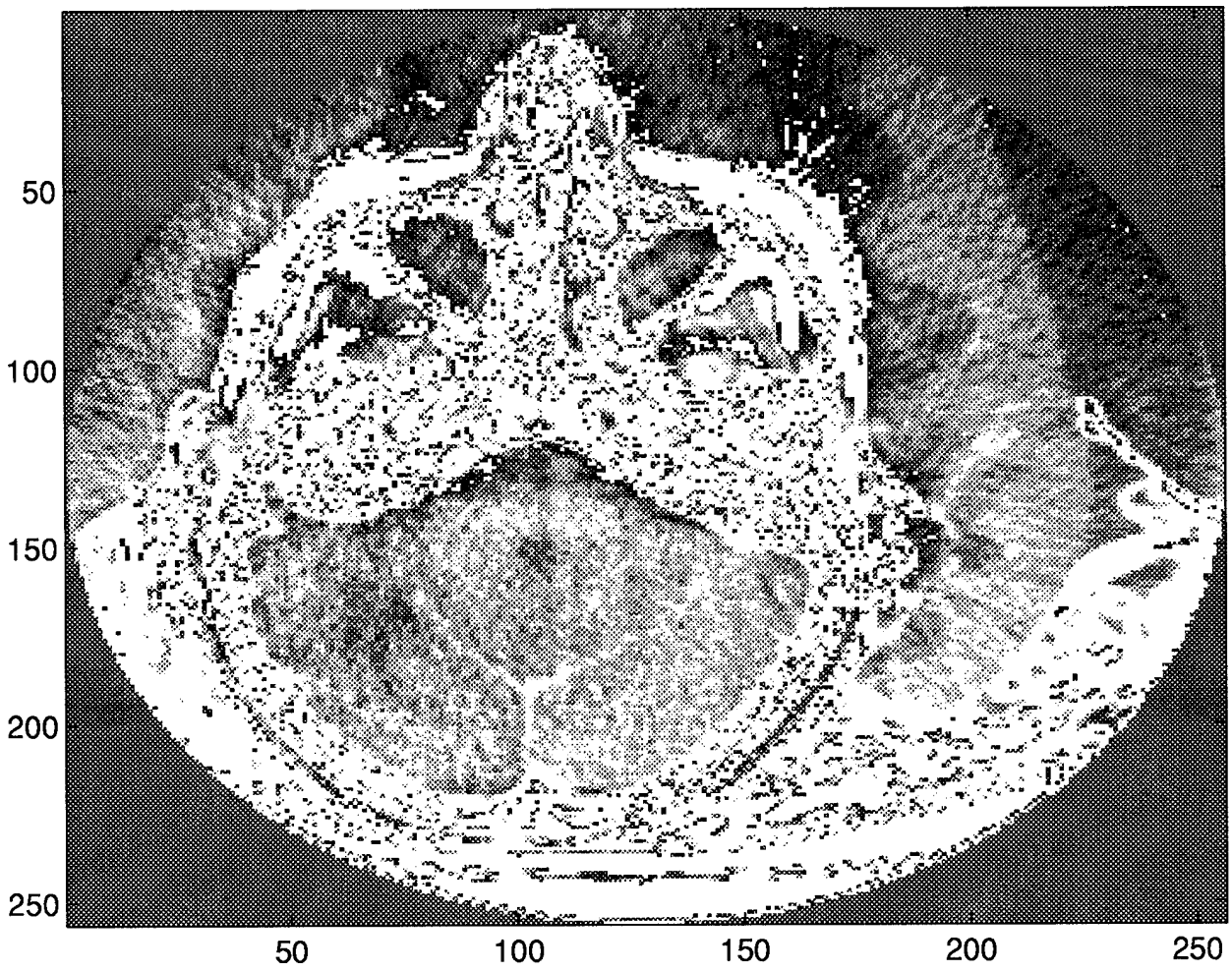


Figure B.11 Original CT Slice 60.

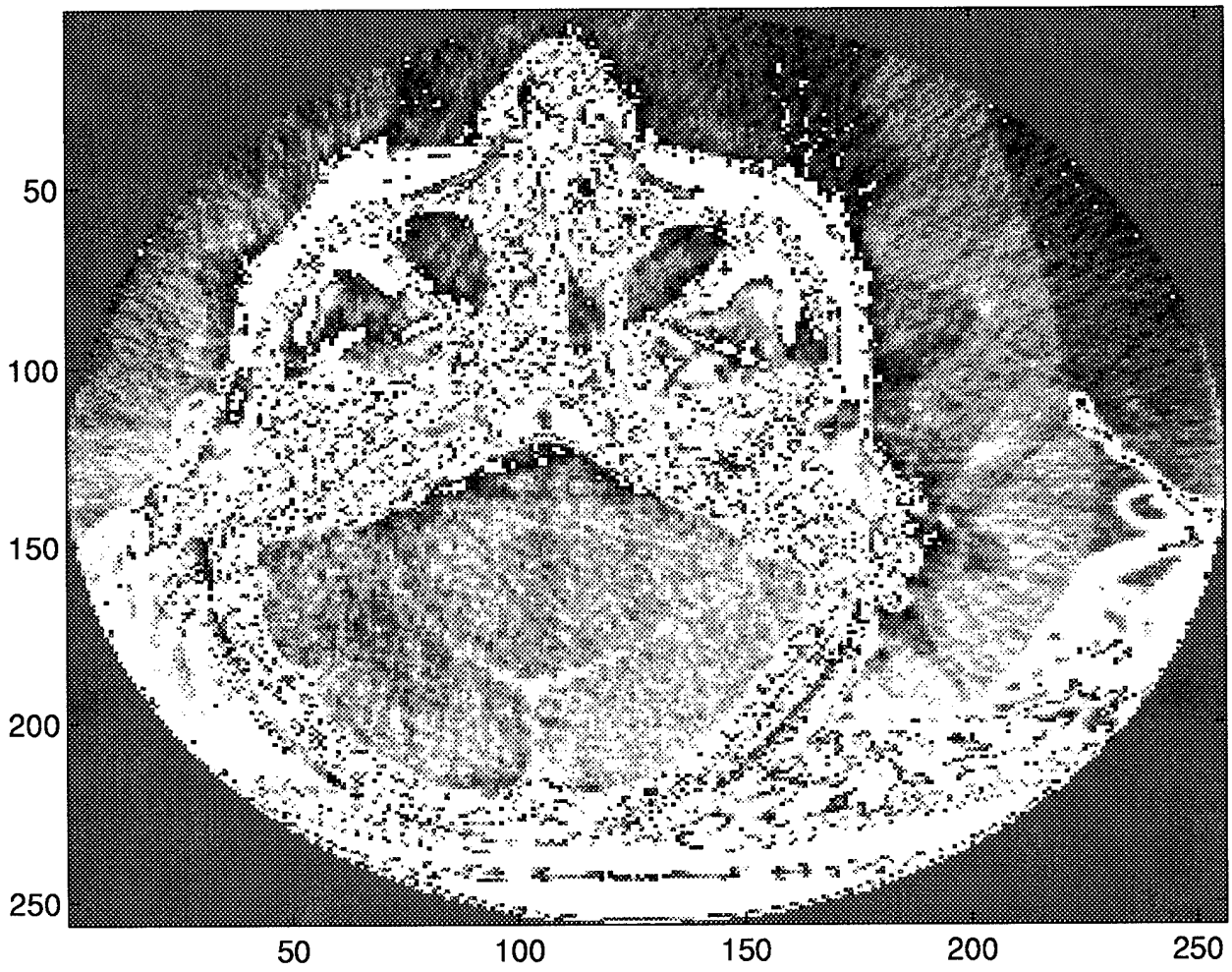


Figure B.12 Original CT Slice 61.

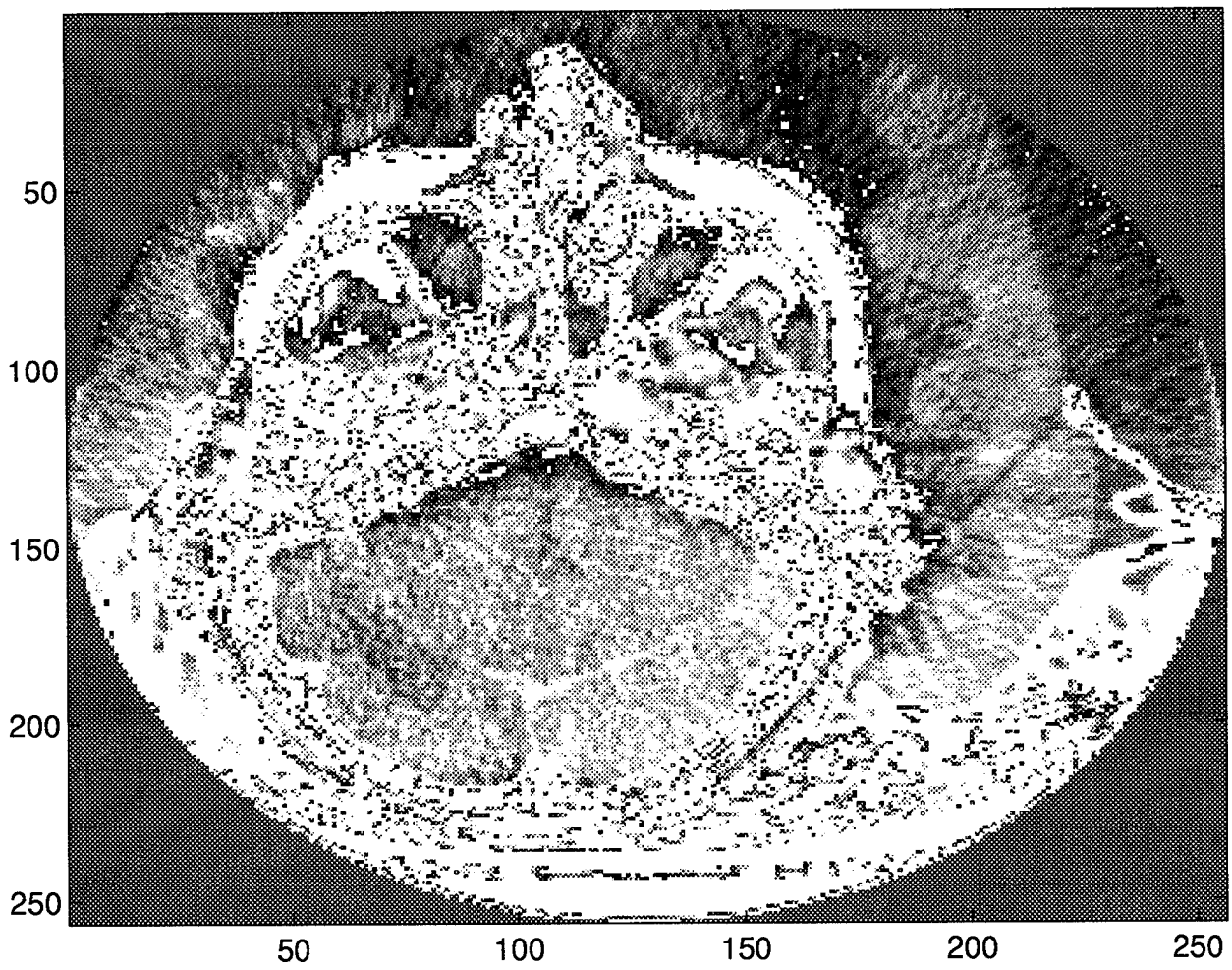


Figure B.13 Original CT Slice 62.

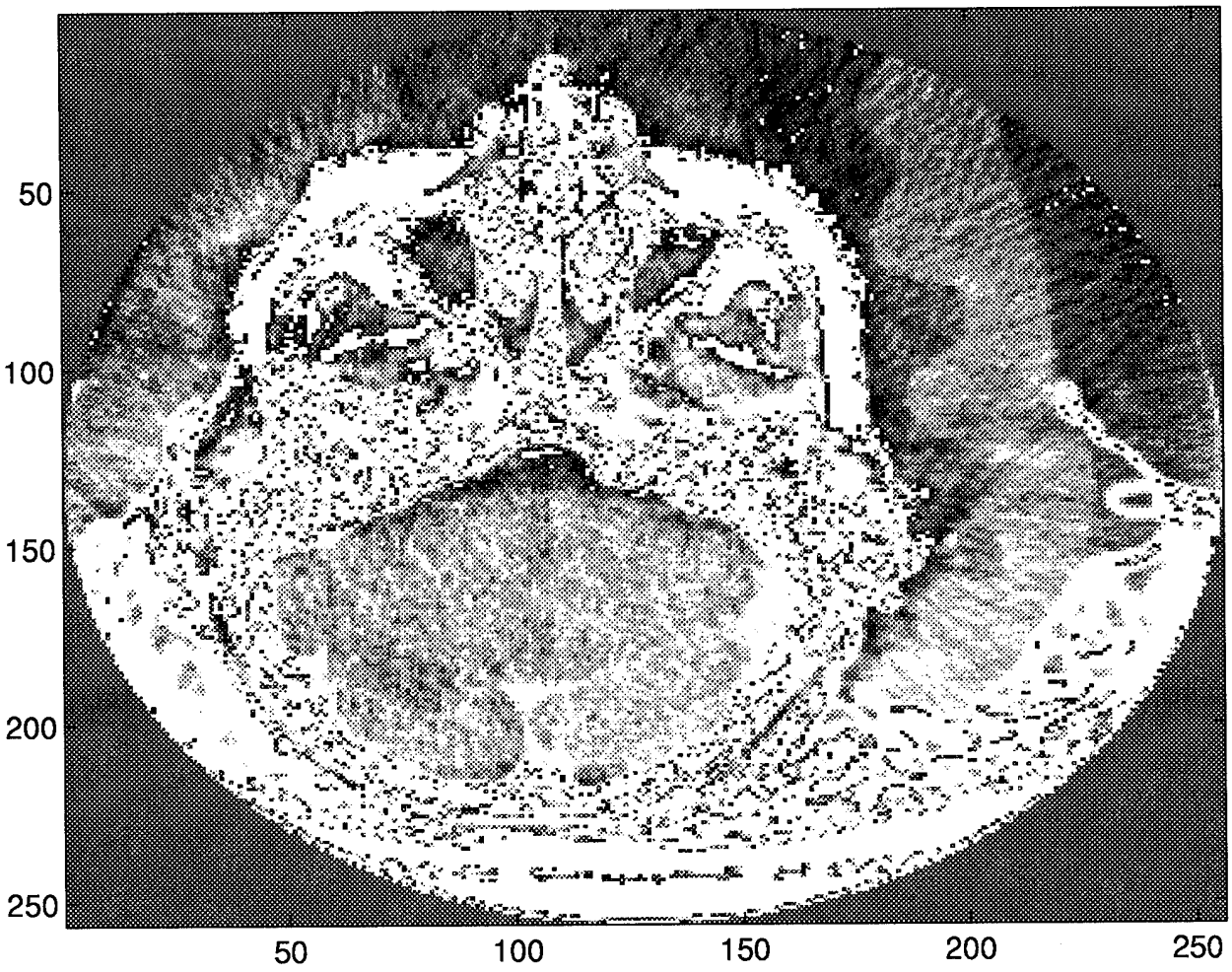


Figure B.14 Original CT Slice 63.

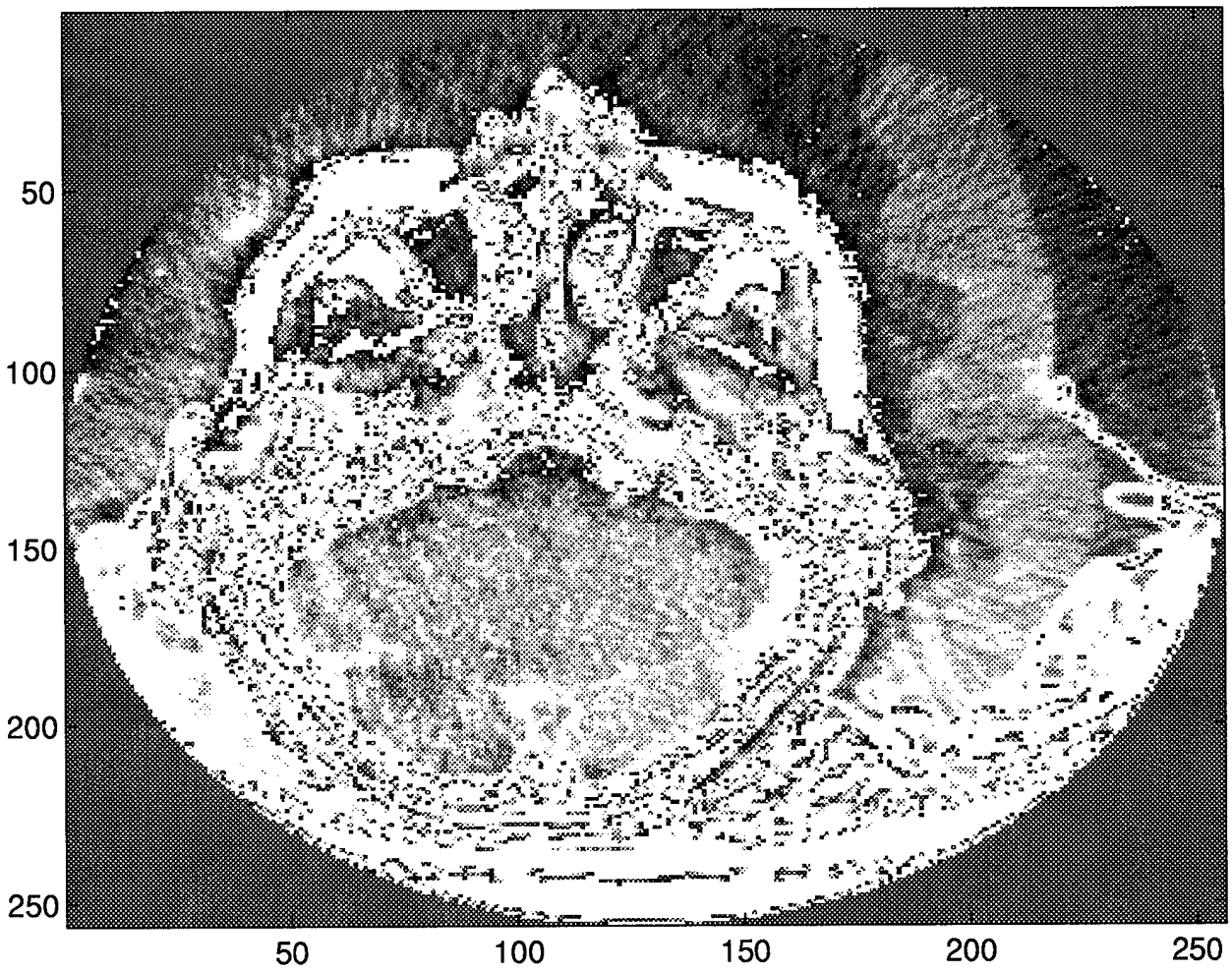


Figure B.15 Original CT Slice 64.

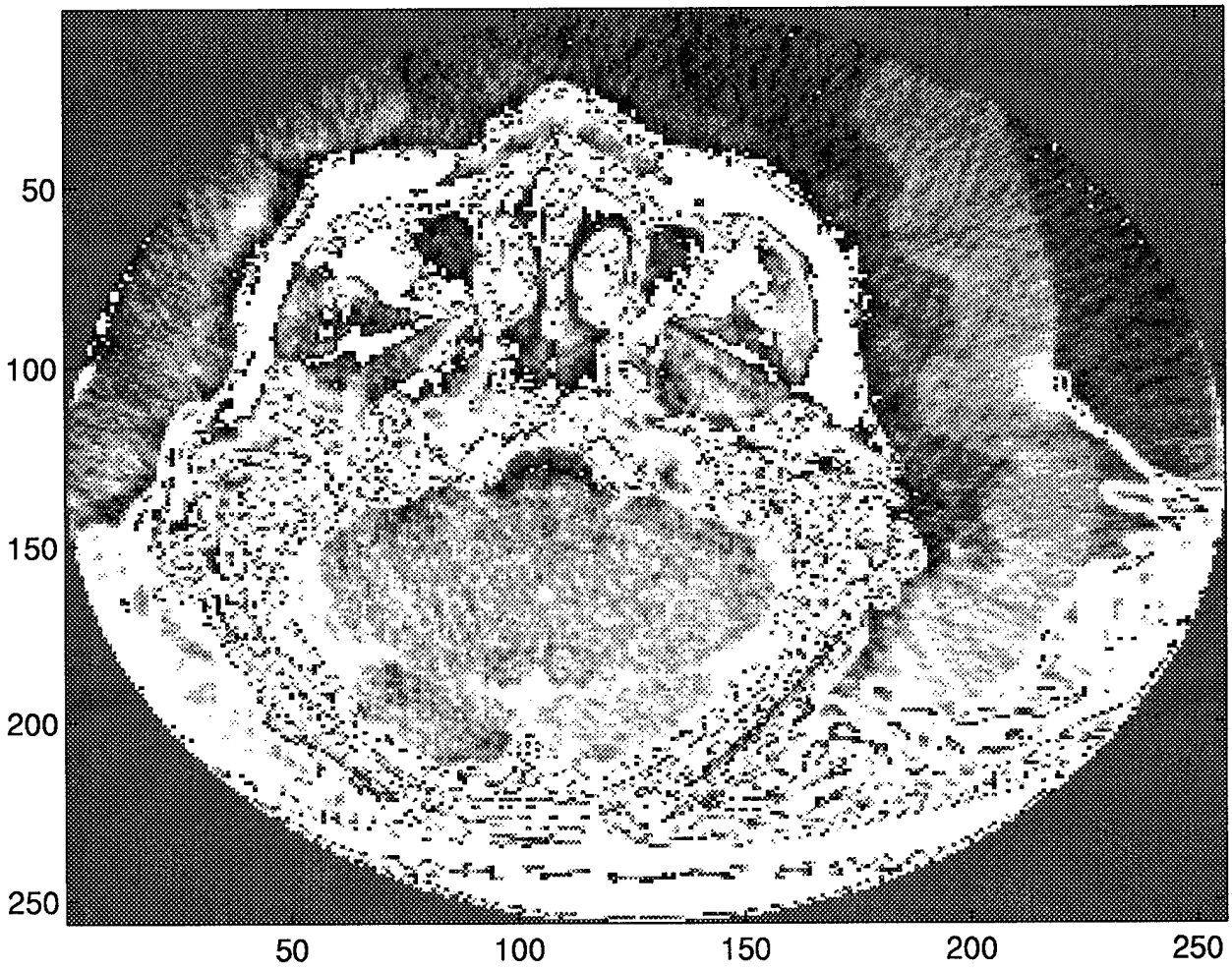


Figure B.16 Original CT Slice 65.

Appendix C. Cubic spline interpolation function

This Appendix provides the cubic spline interpolation function used in this research.

```
function output = spline(x,y,xx)
%SPLINE Cubic spline data interpolation.
%       Given data vectors X and Y, and a new abscissa vector XI, the
%       function YI = SPLINE(X,Y,XI) uses cubic spline interpolation
%       to find a vector YI corresponding to XI.
%
%       Here's an example that generates a coarse sine curve, then
%       interpolates over a finer abscissa:
%
%           x = 0:10;  y = sin(x);
%           xi = 0:.25:10;
%           yi = spline(x,y,xi);
%           plot(x,y,'o',xi,yi)
%
%       PP = spline(x,y) returns the pp-form of the cubic spline interpolant
%       instead, for later use with  ppval, etc.
%
%       See also INTERP1, INTERP2, PPVAL, MKPP, UNMKPP, the Spline Toolbox.

%       Carl de Boor 7-2-86
%       Revised 11-24-87 JNL, 6-16-92 CBM.
%       Copyright (c) 1984-94 by The MathWorks, Inc.

% Generate the cubic spline interpolant in pp form, depending on
```

```
% the number of data points.  (using the not-a-knot end condition).

n=length(x);[xi,ind]=sort(x);xi=xi(:);
output=[];
if n<2,
   fprintf('There should be at least two data points!\n')
elseif all(diff(xi))==0,
   fprintf('The data abscissae should be distinct!\n')
elseif n~=length(y),
   fprintf('Abscissa and ordinate vector should be of the same length!\n')
else
   yi=y(ind);yi=yi(:);
   if (n==2), % the interpolant is a straight line
      pp=mkpp(xi',[diff(yi)./diff(xi) yi(1)]);
   elseif (n==3), % the interpolant is a parabola
      yi(2:3)=diff(yi)./diff(xi);
      yi(3)=(yi(3)-yi(2))/(xi(3)-xi(1));
      yi(2)=yi(2)-yi(3)*(xi(2)-xi(1));
      pp = mkpp([xi(1),xi(3)],yi([3 2 1])');
   else % set up the sparse, tridiagonal, linear system for the slopes at  xi .
      dx=diff(xi);divdif=diff(yi)./dx;xi31=xi(3)-xi(1);xin=xi(n)-xi(n-2);
      c = spdiags([ [dx(2:n-1);xin;0] ...
           [dx(2);2*[dx(2:n-1)+dx(1:n-2)];dx(n-2)] ...
           [0;xi31;dx(1:n-2)] ],[-1 0 1],n,n);
      b=zeros(n,1);
      b(1)=((dx(1)+2*xi31)*dx(2)*divdif(1)+dx(1)^2*divdif(2))/xi31;
      b(2:n-1)=3*(dx(2:n-1).*divdif(1:n-2)+dx(1:n-2).*divdif(2:n-1));
      b(n)=...
```

```
      (dx(n-1)^2*divdif(n-2)+(2*xin+dx(n-1))*dx(n-2)*divdif(n-1))/xin;

        % sparse linear equation solution for the slopes
      mmdflag = spparms('autommd');
      spparms('autommd',0);
      s=c\b;
      spparms('autommd',mmdflag);
        % convert to pp form
      c4=(s(1:n-1)+s(2:n)-2*divdif(1:n-1))./dx;
      c3=(divdif(1:n-1)-s(1:n-1))./dx - c4;
      pp=mkpp(xi',[c4./dx c3 s(1:n-1) yi(1:n-1)]);
   end
   if nargin==2,
      output=pp;
   else
      output=ppval(pp,xx);
   end
end
```

Appendix D. Transformation of CT data set

This Appendix provides the function to transform the CT data set from binary file to ASCII file.

The original CT data set used in this research is binary file. Before we carry out any further processes, we have to convert it to ASCII file. In order to display the image and carry out the image interpolation process by using MATLAB software package, we transformed and stored the data set as 113 matrices and each matrix size is 256 x 256. We also normalized the image intensities to vary between 0-255.

```
function ctdata = getctdata(slice)

endslice = slice;
numlines = 256;
linewidth = 256;
startslice = slice;

fid = fopen('/home/pakh1/mchen/Tools/CThead','r')       %open file
fseek(fid,2*linewidth*256*(startslice - 1),0)        %set pointer to beginning
            %of desired slice

 for j = startslice:endslice,

   ctdata = zeros(256,256);

        for i = 1:numlines,
          temp = fread(fid, linewidth, 'ushort');    %read the first line
          ctdata(i,:) = temp';                       %append to band matrix
        end                                          % numlines for loop
```

```
 end                                                    %end endslice for loop


fclose(fid);

return
```

Appendix E. Visual perception difference function

This Appendix provides the function to calculate the visual perception difference.

```
function perceptualdiff = dist(imagea, imageb)

zeroprotection = .00000000001;

X = [0 140 180 220 250 249 248 247 246 220 195 185 170 160 150 145 140 130 120 110 1

C = zeros(21,21);

for m = 1:21
         for n = 1:21

         C(m,n) = sqrt((X(m)^2) + (X(n)^2));

         end %end for m

       end %end for n



fftimagea = abs(fft2(imagea));
acimagea = fftimagea(1:21,1:21);
ennormalacimagea = acimagea./(sqrt(sum(sum(acimagea.^2))) + zeroprotection);
```

```
fftimageb = abs(fft2(imageb));
acimageb = fftimageb(1:21,1:21);
ennormalacimageb = acimageb./(sqrt(sum(sum(acimageb.^2))) + zeroprotection);

diff = abs(ennormalacimagea - ennormalacimageb);

perceptualdiff = sum(sum(C.*diff(1:21,1:21)));

return
```

Bibliography

1. A. Goshtasby, S. Gage and J. Bartholic. "A two-stage cross-correlation approach to template matching," *IEEE Trans. Pattern Anal. Machine Intell., PAMI-6*:pp. 374 - 378 (1984).
2. Ardeshir Goshtasby, David A. Turner and Launers V. Ackerman. "Matching of Tomographic Slices for Interpolation," *IEEE Transactions on Medical Imaging, 11*(4) (December 1992).
3. Campbell, F. W. and J. G. Robson. "Application of Fourier Analysis to the Visibility of Gratings," *Journal of Physiology, 197* (1968).
4. Christopher Watkins, Alberto Sadum and Stephen Marenka. *Modern Image Processing : Wraping, Morphing, and Classical Techniques*. Boston: Academic Press, 1993.
5. Cornsweet, T. N. *Visual Perception*. New York, NY: Academic Press, 1970.
6. Cowger, Ronald I., "A Measurement of the Tnisotropic Modulation Transfer Function of the Extrafovial Human Visual System."
7. Crochiere and Rabiner. *Multirate Digital Signal Processing*. Englewood Cliffs, N.J.: Prentice-Hall, 1983.
8. Davson, H. *The Eye, Volume 2A: Visual Function in Man*. New York, NY: Academic Press, 1976.
9. G. Q. Maguire, Jr., M. E. Noz E. M. Lee and J. H. Schimpf. *Correlation methods for tomographic images using two- and three-dimensional techniques*. New York, NY: Martinus Nijhoff Publishers, 1986.
10. Gerlot, P. and Y. Bizais. *Image Registration : A review and a strategy for medical application in Information Processing Medical Imaging*. New York, NY: Plenum, 1988.
11. Greenspan, Donald. *Introduction to Numerical Analysis and Applications*. Chicago: Markham Publishing Company, 1990.
12. Guidry, Roland D., "A High Resolution Mesurement of Anisotropic Modulation Transfer Function of the Human Visual System."
13. Hildebrand, F. B. *Introduction to Numerical Analysis* (Second Edition). New York, NY: McGraw-Hill Book Company, 1974.
14. Kaj L. Nielsen, Ph.D. *Methods in Numerical Analysis*. New York, NY: The Macmillan Company, 1956.
15. Libbey, Robert L. *Signal and Image Processing Sourcebook*. New York, NY: Van Nostrand Reinhold, 1994.
16. Oppenheim, Alan V. and Ronald W. Schafer. *Discrete-Time Signal Processing*. Englewood Cliffs, N.J.: Prentice Hall, 1989.

17. Pavlidis, Theodosios. *Algorithms for Graphics and Image Processing.* Rockville, MD: Computer Science Press, 1982.

18. Peli and Eli. "Contrast in Complex Images," *Optical Society of America, 7(10)*

19. Powell, M. J. D. *Approximation Theory and Methods.* London, New York, NY: Cambridge University Press, 1983.

20. Richard A. Robb, Ph.D. *Three-Dimensional Biomedical Imaging, 1.* Boca Raton, Florida: CRC Press, Inc, 1991.

21. Richard H. Bartels, John C. Beatty and Brain A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling.* Los Altos, Calif: M. Kaufmann Publishers, 1986.

22. Richard L. Burden, Douglas J. Faires and Albert C. Reynolds. *Numerical Analysis* (Second Edition). Boston, Massachusetts: Prindle, Weber and Schmidt Publishers, 1981.

23. Sanders, Mark S. and Ernest J. McCormick. *HumanFactors in Engineering and Design* (7th Edition). New York, NY: McGraw-Hill, Inc., 1993.

24. Schumaker, Larry L. *Spline Functions : Basic Theory.* New York, NY: John Wiley and Sons, Inc, 1981.

25. Stark, Peter A. *Introduction to Numerical Methods.* New York,NY: The Macmillan Company, 1970.

26. Udupa, Jayaram K. and editors Gabor T. Herman. *3D Imaging in Medicine.* Boston: CRC Press, 1991.

27. Wilson, Cpt. Terry A., "Perceptual Based Image Fusion with Applications to Hyperspectral Image Data."

Vita

Major Ming-Chung Chen was born in Tainan, Taiwan (R.O.C.) on 23 March 1961. He was accepted into the Chung-Chen Institude of Technology in 1980 and graduated with a bachelors degree in surveying engineering in 1984. After completing his education, he was assigned to the Photographic Squadron in the Air Force. From 1991 to 1992, Major Chen was assigned to be a staff member in the Headquarters of the Air Force. Before he came to AFIT, he was the captain of the drafting department in the Photographic Squadron.

Permanent address: 2235k Chapel Drive
Fairborn, Ohio 45324

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1995	Master Thesis

4. TITLE AND SUBTITLE
BIOMEDICAL DATA INTERPOLATION
FOR 3-D VISUALIZATION

5. FUNDING NUMBERS

6. AUTHOR(S)
Ming-Chung Chen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AFIT/ENG
WRIGHT PATTERSON AFB, OHIO 45433

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GCS/ENG/95J-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Medical imaging devices that produce three-dimensional data usually produce the data in the form of image slices. In such images, the resolution in z direction is lower than in x and y directions. Before extracting and displaying objects in such images, an interpolated 3-D gray-scale volume image can be generated via image interpolation techniques to fill in the missing information.

The subject of this thesis is the applying three different interpolation techniques to generate intermediate slices and comparing their qualities. The three interpolation techniques are linear interpolation, cubic spline interpolation, and Fourier interpolation. We also apply the CT image matching method which can determine the correspondence between points in two images. Finally, we use the human visual perception model to measure the qualities of interpolated images.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Interpolation, Imaging Matching Method, Human Visual Perception.	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL