A Global 3D Interpolator for Large Voxel Datasets*

H. H. Atkinson I. Gargantini R. E. Webber
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7

Abstract

A new computational approach to interpolating large datasets is presented. Its basic features are: (i) formulation and implementation in three dimensions; (ii) capability to work in a binary as well as in a gray-level environment; and (iii) incorporation of smoothing to handle the discontinuities present at an object's boundary. The new interpolator is first introduced in one dimension and then extended to two and three dimensions. The amount of computation involved in each case is given. Applications to the magnification of a binary object (from a $2^7 \times 2^7 \times 2^7$ raster grid to a $2^9 \times 2^9 \times 2^9$ raster grid) and of a gray-level object (from a $2^8 \times 2^8 \times 36$ raster grid to a $2^9 \times 2^9 \times 2^9$ raster grid) are also given.

1 Introduction

Current scanning and sensory technologies offer extensive capabilities to gather large quantities of sample values (some representative of the surface or interior of an object — others of a physical phenomenon in progress). In general, these values are taken at regular intervals in each of the corresponding dimensions, although the sampling rate used for one parameter need not necessarily be the same as for some other. Interpolation between contiguous values in each of the space coordinates is then used to reconstruct the shape or volume of the sensed object. In this paper, we propose a novel computational technique which interpolates an arbitrary number of new values between sampled points.

The proposed algorithm — formulated in 3D and tailored to volume reconstruction — stems from well-known properties of sampling theory as well as from a technique published in 1978 by Hatori and Taki [5] in connection with Pulse Code Modulation for color television transmission.

The basic idea behind our interpolator is to assume a continuous approximation for the given set of sampled points in terms of its Fourier representation, to introduce harmonics of zero-amplitude in the frequency domain, and to evaluate the inverse Fourier Transform of the resulting function in the frequency domain in order to yield new, equally spaced values between original samples.

The artifacts arising from the periodicity imposed on the original data is counteracted by suitably repeating acquired values around the delimiting boundaries of the sampled object, while to eliminate the ripples caused by the presence of high-frequency components, σ-weighting [9] is introduced, with the choice of σ being under user's control.

While our global technique has been designed to handle large sets of data, such as those, for instance, acquired as computer tomographs, the testing has been carried out **also** on well-known geometric shapes, where the application of the above techniques and the effect of distortions can be clearly identified and evaluated. Comparison with the commonly used trilinear interpolation have also been carried out, and results given.

The proposed 3D algorithm — implemented on an SGI 4D/GTX in the language C — finds practical applications in the accurate magnification of acquired data and in the creation, out of sparse data, of new solid models for natural objects [6].

2 Fourier Representation and Transforms

Given N samples $x(0), x(1), ..., x(N-1)$ of a continuous function $x(t)$, its discrete Fourier transform is

$$X(f) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \exp^{-\frac{2\pi i}{N} nf} \quad . \qquad (1)$$

If sampling occurs at a rate equal to or above Nyquist's frequency, then

$$x(t) = \sum_{f=0}^{N-1} X(f) \exp^{\frac{2\pi i}{N} ft} \quad . \qquad (2)$$

*The authors gratefully acknowledge the financial support of the Canadian Government through NSERC grant STR004106.

Another representation for $x(t)$ ([8], page 153) is given by

$$x(t) = \sum_{n=-\infty}^{+\infty} x(n)\text{sinc}(t-n) = \sum_{n=-\infty}^{+\infty} x(t)\delta(t-n)\text{sinc}(t-n) \quad (3)$$

$$\text{where } x(N-n) = x(n),$$

$$\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t} \ ,$$

and $\delta()$ is the impulse or Dirac function defined by $\int_{t_0^-}^{t_0^+} \delta(t-t_0)dt = 1$.

3 Our Interpolation Technique: 1D

In this section we explain our basic approach which consists of assuming a model function, sampling it at regular intervals, and introducing new points.

Model

The model is represented by a function $x(t)$, continuous over $[-\infty, +\infty]$ and sampled at a rate equal to or above Nyquist's; its band-limited Fourier Transform is denoted by $X(f)$.

Input Data

The given data is represented by N samples of $x(t)$, namely $x(n), n = 0, 1, ..., N-1$ with (initially) N even.

Goal

To determine additional samples between any two given ones, so that $x(t)$ can be approximated by hN values as follows:

$$x(n/h) \qquad h = 2, 3, ..; n = 0, 1, .., N-1.$$

Method

Determine the discrete Fourier Transform of $\{x(n)\}$, namely $\{X(f)\}, f = 0, 1, ..N-1$, and enlarge the frequency interval $[0, N-1]$ to $[0, hN-1]$ while defining $\hat{X}(f)$ as follows:

$$\begin{aligned} \hat{X}(f) = &\quad X(f), \quad f = 0, 1, .., \tfrac{N}{2}; \\ \hat{X}(f) = &\quad 0, \quad f = \tfrac{N}{2}, \tfrac{N}{2}+1, .., hN-\tfrac{N}{2}-1; \\ \hat{X}(f) = &\quad X(f-(h-1)N), \\ &\quad f = hN-\tfrac{N}{2}, hN-\tfrac{N}{2}+1, .., hN-1. \end{aligned}$$

The inverse Fourier Transform of $\hat{X}(f)$ yield $x(n/h)$.

Example

Model function $x(t)$, sample points, spectrum $|X(f)|$, enlarged spectrum and the results of interpolation are shown in Figure 1.

Note that if Δx is the initial spacing, the new sampling interval is $\Delta x/h$. With this in mind, we can rewrite $x(n/h)$ as $\hat{x}(n), n = 0, 1, .., hN-1$.

4 Related Work

Interpolation is one of the oldest field of numerical mathematics. Two basic approaches are of importance: one based on techniques formulated in the sample space and the other dealing, at least conceptually, with resampling in the frequency domain and remapping these new values in the sample space.

Linear interpolation between any two sample points $x(i)$ and $x(i+1)$ simply means to approximate all values between $x(i)$ and $x(i+1)$ with the straight-line passing through these two points. Similarly, polynomial interpolation of N points means to approximate all values between $x(0)$ and $x(N-1)$ with the polynomial of degree $(N-1)$ passing through the given samples. If the model function $x(t)$ extends to infinity while remaining bounded in value, polynomial approximations are inadequate and a combination of polynomial, rational, exponential or trigonometric functions must be devised to fit the behaviour of each particular $x(t)$.

Fourier theory offers an alternative to deal with this problem. In the frequency domain linear interpolation is carried out by windowing $X(f)$ with a triangular function (or, equivalently by convolving $x(t)$ with the $sinc^2$ function). Different windows generate different forms of interpolation — the most well-known being the rectangular wave, which corresponds to convolve $x(t)$ with the *sinc* function. This form is the best possible in the sense that, in the limit, it coincides with model function $x(t)$ over $[-\infty, +\infty]$.

Conceptually speaking, most methods formulated in the frequency domain require: (i) resampling the Fourier Transform and (ii) inverting the latter to obtain new samples in the original space — an equivalent approach is to convolve the given samples with the Fourier Transform of the adopted window. References can be found in [2, 10, 12, 13].

A slightly different approach is presented by Hatori and Taki [5] who were confronted with avoiding distortions in the design of low-pass filters for Digital-to-Analog conversion. Since they worked in an environment which imposed sampling frequency only equal to Nyquist's, these authors focused on spacing spectrum replicas to create a guard-band wide enough to protect against aliasing during the transmission of signals. To achieve their goal, they inserted one new value between any two given samples using the *sinc* function.

Our approach is similar to these authors' in the sense that we **also** enlarge the frequency domain of the discrete spectral components. However, our interpolator differs in several ways: (i) it operates directly in the spectral domain; (ii) it inserts an arbitrary number of

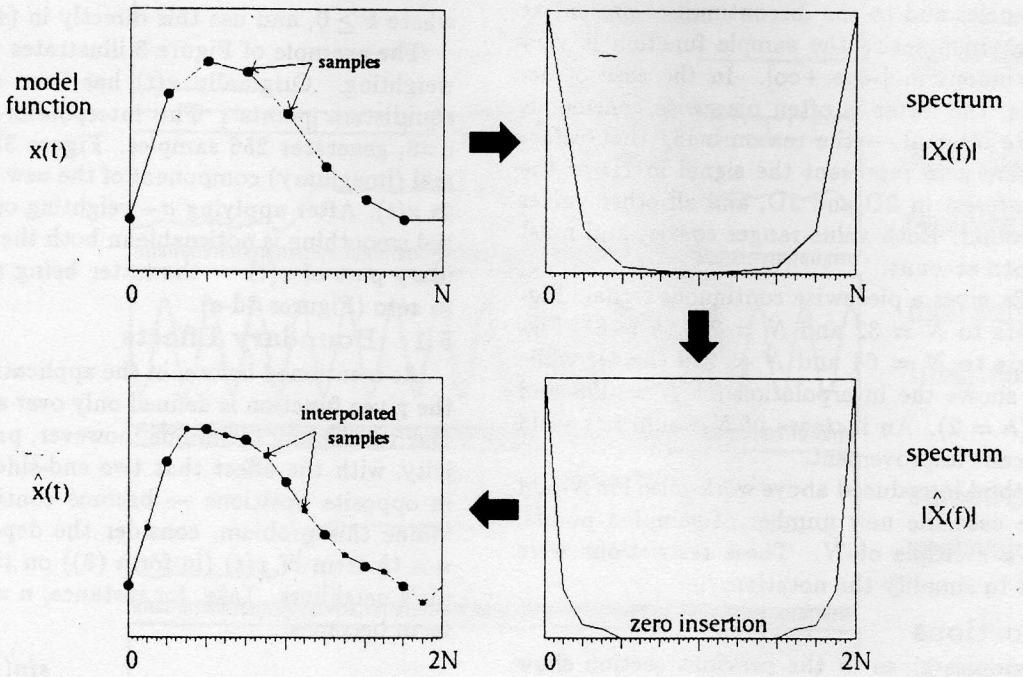

Fig.1: Interpolation Procedure

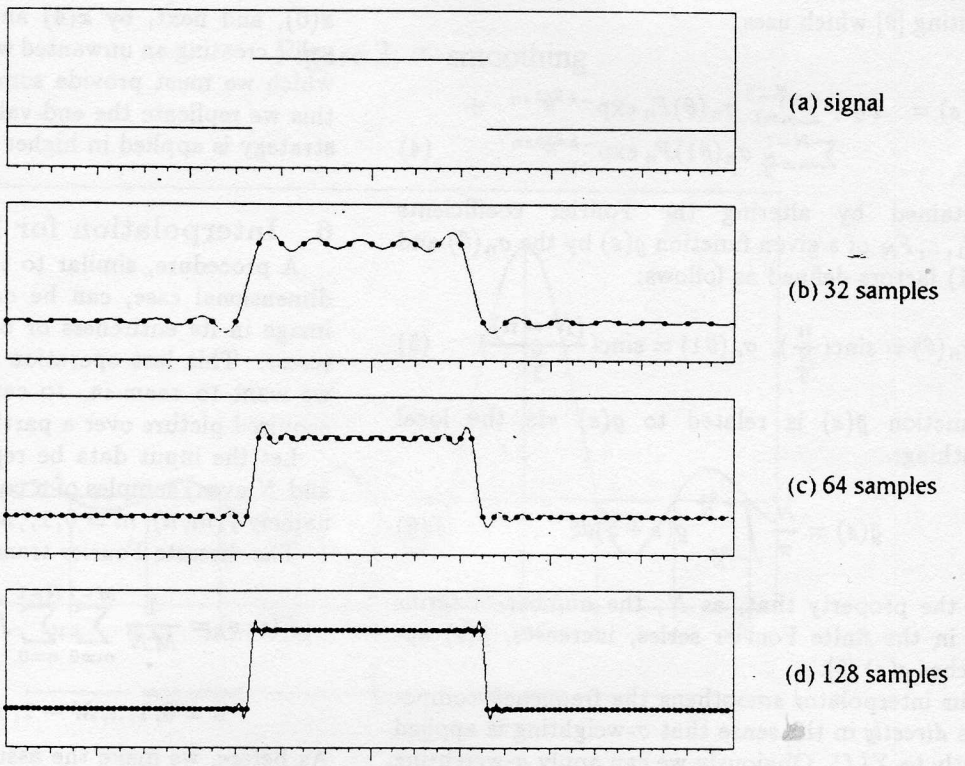

Figure 2: Interpolation of a piecewise continuous signal

new samples; and (iii) it includes effective techniques to counteract the distortions due to the existence of high frequencies and to the discontinuities present at boundaries. In practise the sample function is very rarely continuous in $[-\infty, +\infty]$. In the case of acquired data, the latter is often piecewise continuous over a finite interval — the reason being that *values* different from zero represent the signal in 1D or the object of interest in 2D and 3D, and all other *values* the background. Both value-ranges coexist and must be taken into account.

Figure 2a gives a piecewise continuous signal. Figure 2b refers to $N = 32$ and $\hat{N} = 256$ ($h = 8$), Figure 2c refers to $N = 64$ and $\hat{N} = 256$ (h=4), while Figure 2d shows the interpolation for $N = 128$ and $\hat{N} = 256$ ($h = 2$). An increase of $\hat{N}$ would not yield any significant improvement.

The method introduced above works also for N odd and in the case the new number of sampled points, $\hat{N}$, is not a multiple of N. These restrictions were introduced to simplify the notation.

5 Distortions

The examples given in the previous section show that, while $x(n)$ interpolates quite well in mid-range, at the end-intervals the presence of high frequencies introduces some distortions — an effect known as Gibb's phenomenon. To counteract it, we adopt Lanczos σ-weighting [9] which uses

$$\bar{g}(s) = F_0 + \sum_{n=1}^{\frac{N}{2}-1} \sigma_n(\theta) F_n \exp^{-h\frac{2\pi i}{N}sn} + \sum_{n=\frac{N}{2}}^{N-1} \sigma_n(\theta 1) F_n \exp^{-h\frac{2\pi i}{N}sn} \quad (4)$$

obtained by altering the Fourier coefficients $F_0, F_1, .., F_N$ of a given function $g(s)$ by the $\sigma_n(\theta)$ and $\sigma_n(\theta 1)$ factors defined as follows:

$$\sigma_n(\theta) = \mathrm{sinc}(\frac{n}{\frac{N}{2}}),\ \sigma_n(\theta 1) = \mathrm{sinc}(\frac{(N-n)}{\frac{N}{2}}) \quad (5)$$

Function $\bar{g}(s)$ is related to $g(s)$ via the local smoothing:

$$\bar{g}(s) = \frac{N}{\pi} \int_{-\frac{2\pi}{N}}^{+\frac{2\pi}{N}} g(s+\xi) d\xi \quad , \quad (6)$$

with the property that, as N, the number of terms used in the finite Fourier series, increases, $\bar{g}(s)$ approaches $g(s)$ [9].

Our interpolator smoothens the frequency components *directly* in the sense that σ-weighting is applied directly to $X(f)$. Obviously we can apply σ-weighting all over again, or, equivalently, define

$$\sigma^k(\theta) = \mathrm{sinc}^k(\theta) \quad (7)$$

where $k \geq 0$, and use this directly in (4).

The example of Figure 3 illustrates the effect of σ-weighting. Originally $x(t)$ has been sampled at 32 equidistant points. The interpolator, applied with h=8, generates 256 samples. Figure 3b(3c) gives the real (imaginary) component of the new approximation to $x(t)$. After applying $\sigma-$weighting once, a substantial smoothing is noticeable in both the real and imaginary part of $x(t)$ — the latter being almost reduced to zero (Figures 3d-e).

5.1 Boundary Effects

As mentioned before, in the applications of interest, the given function is defined only over a finite interval. Our resampling technique, however, produces periodicity, with the effect that two end-sides — originally in opposite positions — become contiguous. To examine this problem, consider the dependence of the $n-th$ term of $x(t)$ (in form (3)) on the $n-th$ sample's neighbors. Take, for instance, $n = 2$ so that this term becomes

$$x_2(t) = x(2)\ \mathrm{sinc}(t-2) = x(2)\frac{sin(\pi(t-2))}{\pi(t-2)} \quad (8)$$

(plotted in Figure 4). A new sample between $x(1)$ and $x(2)$ is influenced by $x(1)$ and $x(2)$, by $x(3)$ and $x(0)$, and next, by $x(4)$ and $x(N-1)$ — this last value creating an unwanted wrap-around effect against which we must provide some safeguard. To achieve this we replicate the end-values two times. A similar strategy is applied in higher dimensions.

6 Interpolation for 2D Arrays

A procedure, similar to that outlined for the one-dimensional case, can be devised to resample a 2D image in its entireness or over a small region of interest. This last operation is quite important when we want to *zoom-in* to capture more details of an acquired picture over a particular area.

Let the input data be represented by $M \times N$ (M and N even) samples of a continuous function $f(x,y)$, namely $f(m,n)$, $m = 0, 1, , M-1$, and $n = 0, 1, ..., N-1$. The discrete Fourier transform is

$$F(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \exp^{-2\pi i(\frac{mu}{M}+\frac{nv}{N})} \quad (9)$$

$$u = 0, 1, .., M-1; v = 0, 1, .., N-1.$$

As before, we make the assumption that model function $f(x,y)$ is band-limited, so that we can choose an

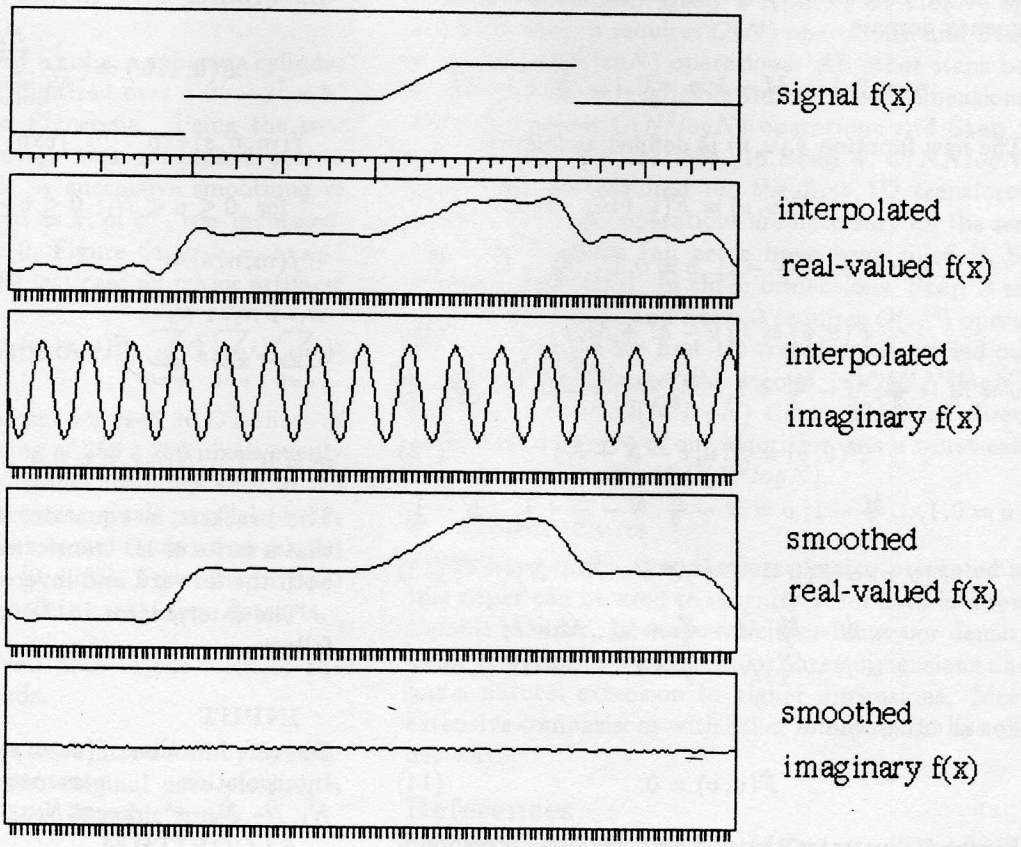

Figure 3: σ-smoothing

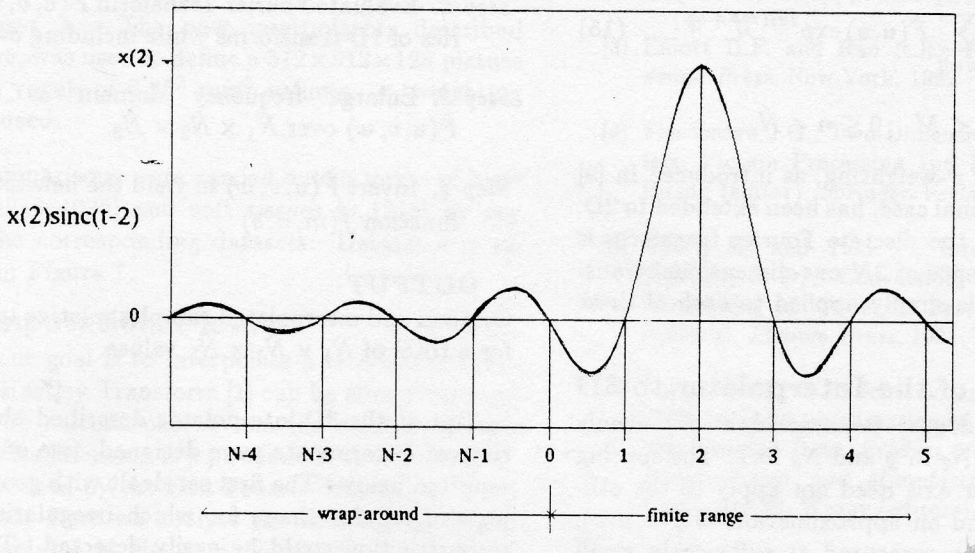

Figure 4: $x_2(t)$

$\hat{M} > M$ and an $\hat{N} > N$ to create an enlarged frequency domain

$$m = 0, 1, .., \hat{M} - 1; n = 0, 1, .., \hat{N} - 1.$$

The new function $\hat{F}(u, v)$ is defined as follows:

$$\hat{F}(u, v) = F(u, v) \quad (10)$$

$$u = 0, 1, .., \tfrac{M}{2} - 1; \quad v = 0, 1, .., \tfrac{N}{2} - 1.$$

$$\hat{F}(u, v) = F(u - \hat{M} + M, v) \quad (11)$$

$$u = \hat{M} - \tfrac{M}{2}, \hat{M} - \tfrac{M}{2} + 1, .., \hat{M} - 1; \; v = 0, 1, .., \tfrac{N}{2} - 1.$$

$$\hat{F}(u, v) = F(u, v - \hat{N} + N) \quad (12)$$

$$u = 0, 1, .., \tfrac{M}{2} - 1; \; v = \hat{N} - \tfrac{N}{2}, \hat{N} - \tfrac{N}{2} + 1, .., \hat{N} - 1.$$

$$\hat{F}(u, v) = F(u - \hat{M} + M, v - \hat{N} + N) \quad (13)$$

$$u = \hat{M} - \tfrac{M}{2}, \hat{M} - \tfrac{M}{2} + 1, .., \hat{M} - 1;$$
$$v = \hat{N} - \tfrac{N}{2}, \hat{N} - \tfrac{N}{2} + 1, .., \hat{N} - 1.$$

For all other values

$$\hat{F}(u, v) = 0. \quad (14)$$

Figure 5 illustrates the non-zero regions corresponding to (10)-(13) as well as the central region for which $\hat{F}(u, v) = 0$. The interpolating function can be reconstructed by applying the inverse Fourier transform, i.e.

$$\hat{f}(m, n) = \sum_{u=0}^{\hat{M}-1} \sum_{v=0}^{\hat{N}-1} \hat{F}(u, v) \exp^{2\pi\imath(\frac{mu}{\hat{M}} + \frac{nv}{\hat{N}})} \quad (15)$$

$$0 \leq m < \hat{M} \;\; ; 0 \leq n < \hat{N} \;\; .$$

The technique of σ-weighting, as introduced in [9] for the one-dimensional case, has been extended to 2D. Since, in this case, the discrete Fourier transform is evaluated as a sequence of $2N$ one-dimensional transforms, formula (4) is simply applied to each of these in turn.

7 Extensions of the Interpolator to 3D

Assume that $f(x, y, z)$ is sampled at N_1 evenly spaced points in x, N_2 in y and N_3 in z. The spacing along any particular axis need not apply to the others. We want to find an approximation to $f(x, y, z)$. If $f(x, y, z)$ has been measured at sufficiently small intervals in each direction and is band-limited in the frequency domain, then the Fourier pair is:

$$F(u, v, w) = (\frac{1}{N_1} \sum_{m=0}^{N_1} (\frac{1}{N_2} \sum_{n=0}^{N_2} (\frac{1}{N_3} \sum_{s=0}^{N_3}$$
$$f(m, n, s) \exp^{-\imath \frac{2\pi sw}{N_3}}) \exp^{-\imath \frac{2\pi nv}{N_2}}) \exp^{-\imath \frac{2\pi mu}{N_1}}) \quad (16)$$

$$\text{for } \; 0 \leq u < N_1; \; 0 \leq v < N_2; \; 0 \leq w < N_3 \;\; ;$$

$$f(m, n, s) =$$

$$(\sum_{u=0}^{N_1-1} (\sum_{v=0}^{N_2-1} (\sum_{w=0}^{N_3-1} F(u, v, w) \exp^{\imath \frac{2\pi sw}{N_3}}) \exp^{\imath \frac{2\pi nv}{N_2}}) \exp^{\imath \frac{2\pi mu}{N_1}}) \quad (17)$$

$$m = 0, 1, .., N_1 - 1, n = 0, 1, .., N_2 - 1, s = 0, 1, .., N_3 - 1.$$

The brackets in equations (16) and (17) emphasize that a series of 1D transforms can be used to calculate both the forward and inverse transforms.

The interpolator in 3D can then be described as follows.

INPUT

Discrete function: $f(m, n, s)$ over $N_1 \times N_2 \times N_3$
Interpolation parameters (user-controlled): $\hat{N}_1, \hat{N}_2, \hat{N}_3$; σ^k; *repeat-No.*

ALGORITHM

Step 1. Replicate layers of boundary values (a number of times equal to *repeat-No*) to avoid the effect of artificial periodicity

Step 2. Evaluate Fourier Transform $F(u, v, w)$ as a series of 1D-transforms while including σ-weighting

Step 3. Enlarge frequency domain as to define $\hat{F}(u, v, w)$ over $\hat{N}_1 \times \hat{N}_2 \times \hat{N}_3$

Step 4. Invert $\hat{F}(u, v, w)$ to yield the new interpolated function $\hat{f}(m, n, s)$

OUTPUT

Original and interpolated samples relative to $f(x, y, z)$ for a total of $\hat{N}_1 \times \hat{N}_2 \times \hat{N}_3$ values.

To test the 3D-interpolator described above, a variety of experiments were designed, two of which are reported below. The first set dealt with geometric objects of regular shape for which irregularities in the reconstruction could be easily detected. The second set dealt with Computer Tomography data of a cat's head segmented to yield hard and soft tissues.

7.1 Experiment Number 1: a Binary Image

Here four solids, namely a cube, a sphere, a cylinder and an octahedron were digitized over a $2^7 \times 2^7 \times 2^7$ raster grid, as shown in Figure 6a. Using the new interpolator, the 3D image was magnified over a $2^9 \times 2^9 \times 2^9$ raster grid. A successive smoothing of the objects can be noticed as k, of σ^k, was increased. Figure 6b refers to $k = 0$, Figure 6c to $k = 1$, and Figure 6d to $k = 2$. In the last case all major artifacts had been eliminated.

7.2 Experiment Number 2: a Gray-Level Image

For the second experiment we used 36 CT-slices of a cat's head, each consisting of 256×256 pixels evenly spaced at 0.5 mm. The distance from one slice to the next was 1 mm thus generating small parallelepipeds of 0.25 mm^3 volume. Each sample point was related to a density in the range $[0, 4095]$. From the original data, interpolation was used to create four datasets.

- Dataset 1 - The given data in the form of the original parallelepipeds.
- Dataset 2 - New slices were introduced by linearly interpolating every two original ones yielding a picture with $256 \times 256 \times 70$ cubic voxels.
- Dataset 3 - Tri-linear interpolation was used to create a picture with $512 \times 512 \times 128$ voxels, each being a cube of volume 0.25^3 mm^3. Tri-linear interpolation determined new samples values obtained by averaging the original density data.
- Dataset 4 - The new interpolator, described above, was used to define a $512 \times 512 \times 128$ picture with voxels of 0.25^3 mm^3 volume. σ_1-weighting was used.

Visual comparisons were carried out in terms of hard tissues $[1216, 4095]$ and soft tissues $[0, 1215]$ by ray tracing the corresponding datasets. Dataset 4 is visualized in Figure 7.

8 Computational Issues

Since our goal is to interpolate a real-valued function, the Hartley Transform [1] can be effectively used to speed up calculations. When the number of samples in each dimension is a power of 2, the same shortcuts introduced by the Fast Fourier Transform [3] can be adopted. Here we analyse the time complexity of our interpolator in the case $N_1 = N_2 = N_3 = N$, $\hat{N}_1 = \hat{N}_2 = \hat{N}_3 = \hat{N}$ and both N and $\hat{N}$ are a power of 2.

In one dimension, **Step 2** requires $O(NlogN)$ operations, **Step 3** requires $O(N)$ operations, and **Step 4** requires $O(\hat{N}log\hat{N})$ operations. All other steps being carried out in negligible time. In two dimensions, **Step 2** requires $O(N^2logN)$ operations and **Step 3** requires $O(N^2)$ operations. In **Step 4**, $O(N\hat{N}log\hat{N})$ operations are required for the first 1D transform, while $O(\hat{N}^2log\hat{N})$ operations are necessary for the second 1D transform (all zeros have been filled in by the first transform). In three dimensions, **Step 2** requires $O(N^3logN)$ and **Step 3** requires $O(N^3)$ operations. In **Step 4** the first 1D transform is carried out in $O(N^2\hat{N}log\hat{N})$ time, the second in $O(N\hat{N}^2log\hat{N})$, and the third in $O(\hat{N}^3log\hat{N})$ time. Thus the three-dimensional version of our algorithm has a worst-case run-time complexity of $O(\hat{N}^3log\hat{N})$.

9 Conclusions

We have shown that the interpolator presented in this paper can be used to magnify voxel data in a reasonable manner. It works on either binary or density data. It performs well in two or three dimensions and has a natural extension to higher dimensions. More extensive comparisons with other interpolators are under way.

References

[1] Bracewell R.N., The Hartley Transform, *The Oxford University Press*, New York, 1986.

[2] Dunne S., Napel S., and Rutt B., Fast Reprojection of Volume Data, *IEEE Proc. First Conference on Visualization in Biomedical Computing*, Atlanta, Ga., May 22-25, 1990, pp. 11-18.

[3] Elliott D.F. and Rao R.R., Fast Transforms, *Academic Press*, New York, 1982.

[4] Fiasconaro J.G., Two-Dimensional Nonrecursive Filters, Picture Processing and Digital Filtering, T.S. Huang (editor), *Springer-Verlag*, 1975, pp. 69-129.

[5] Hatori M. and Taki Y., Interpolation to Reduce Difficulty in D/A Conversion, in *Real-Time/Parallel Computing*, M. Onoe, K. Preston and A. Rosenfeld (editors), *Plenum Press*, 1981, pp. 227-240.

[6] Gargantini I., G.F. Schrack, and A. Kwok, Software Tools for the Amalgamation of Acquired and Graphically Created Data, *Product Modeling for Computer-Aided Design and Manufacturing*, J. Turner, J. Pegna, and M. Wozny (editors), *North-Holland*, 1991, 53-71.

[7] Hoffmeister J.W., Rinehart G.C., and Vannier N.W., Three-dimensional Surface Reconstructions using a

General Purpose Image Processing System, *Computerized Medical Imaging and Graphics*, Vol. 14, No. 1, pp. 35-42, 1990.

[8] Hsu H.P., Fourier Analysis, *Simon and Schuster*, 1967.

[9] Lanczos C., Discourse on Fourier Series, *Oliver & Boyd*, 1966.

[10] Parker J., A Comparison of Interpolating Methods for Image Resampling, *IEEE Trans. Med. Imaging*, Vol.8, No.4, 1983, pp.31-39.

[11] Rosenfeld A. and Kak A.C., Digital Picture Processing, Volumes 1 & 2, *Academic Press*, New York, 1982.

[12] Smith A. R., Digital Filtering Tutorial, Tech. Rep. No. 27, Lucasfilm Ld., March 1983 (reprinted in: Course Notes, *State-of-the Art in Volume Visualization*, No. 28, SIGGRAPH 1991 Tutorials).

[13] Udupa J.K. and Herman G.T., 3D Imaging in Medicine, *CRC Press*, 1991.

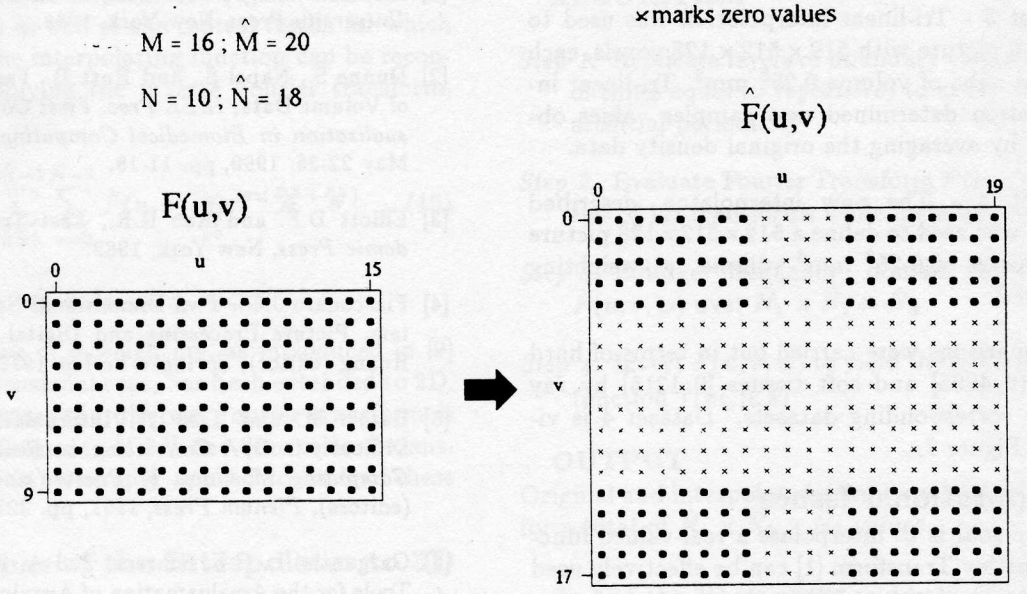

Figure 5: Zero insertion

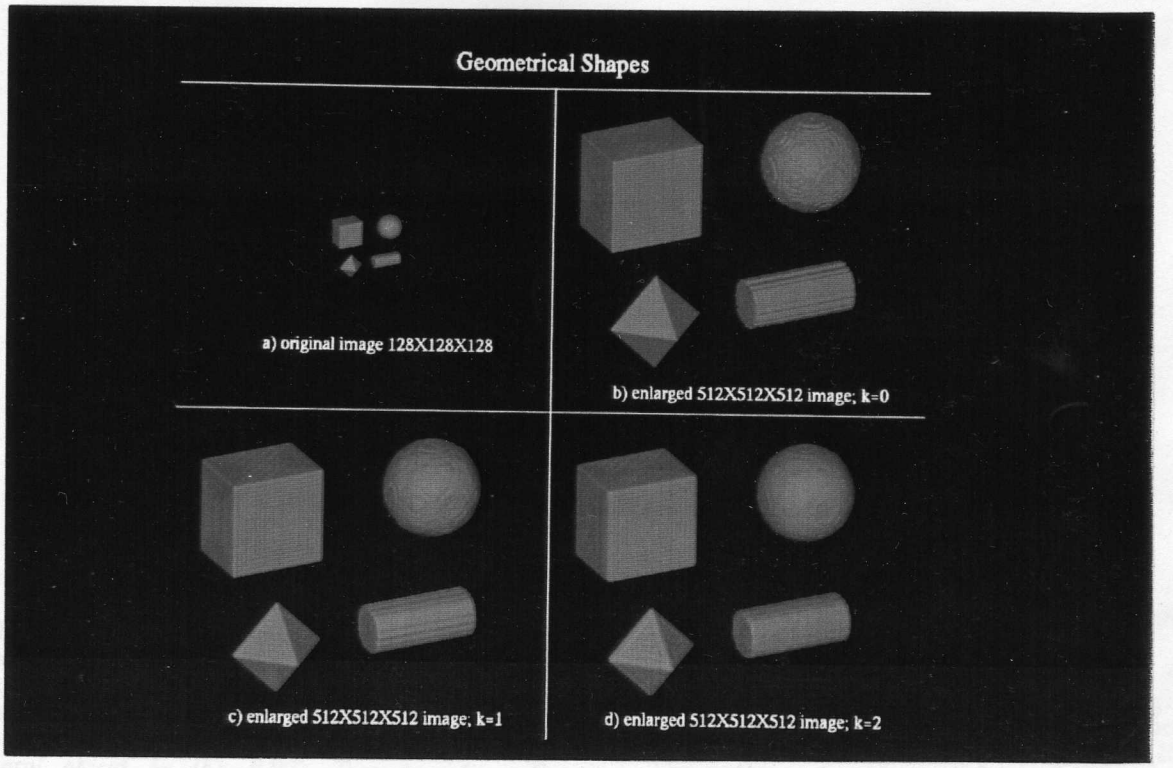

Figure 6: Geometrical Objects

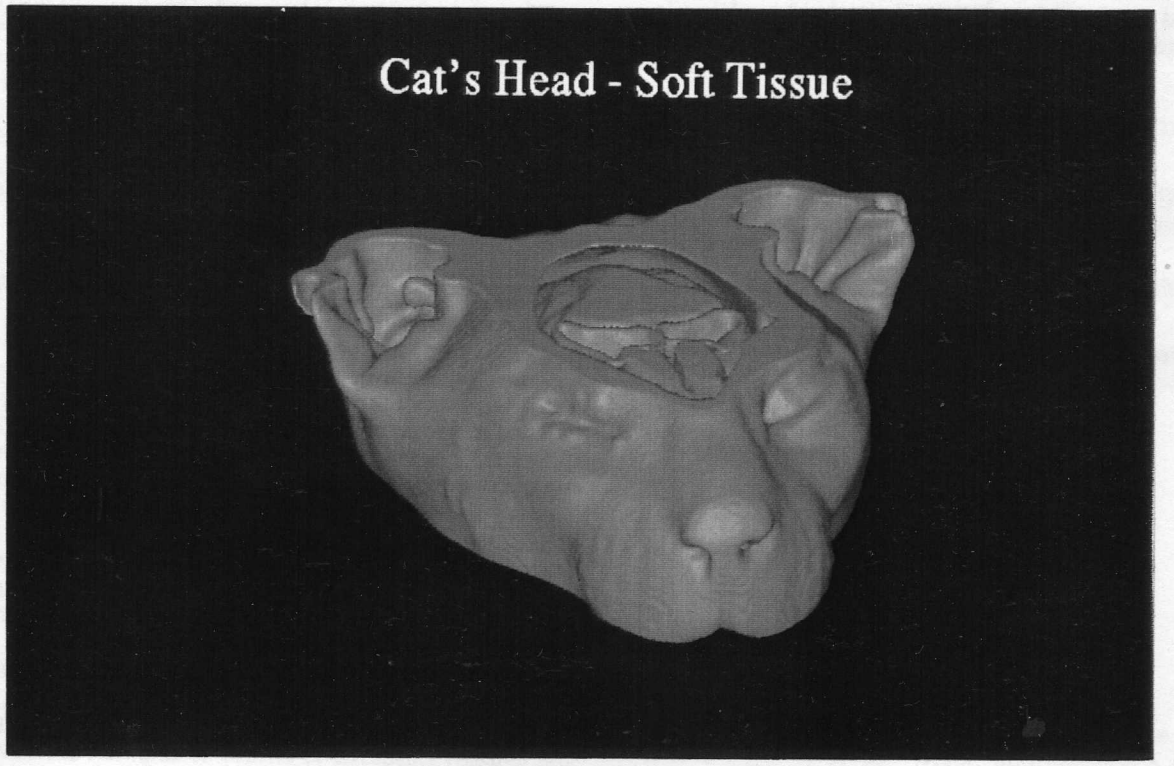

Figure 7: Visualization of Data Set 4