Image compression based on a Multipoint Taylor series representation

GHISLAIN FRANSSENS, MARTINE DE MAZIERE, DOMINIQUE FONTEYN, DIDIER FUSSEN

BIRA - Belgian Institute for Space Aeronomy
Ringlaan 3, B-1180 Brussels, Belgium
ghislain.franssens @ oma.be

Abstract. An image compression/decompression method is proposed, based on a new interpolation formula of Hermite-Birkhoff type. It is based on a function representation that uses a truncated Multipoint Taylor (MT) series. Given a regularly tabulated image, first spatial derivatives (up to a chosen order) are computed by a FFT algorithm. Secondly a knot placing strategy selects sample point positions. The image values and spatial derivatives at these sample points are stored as the compressed data. Decompression is achieved by applying the MT interpolant to the compressed data. The quality of compression is user controlled by specifying the maximum order of spatial derivatives and the number of knots to be used. The method is found to be superior to JPEG in both reduction factor and quality, when reasonable smooth data is to be compressed with high accuracy. In this case, data reduction ratios in the range 10-100 can be obtained with a root mean square reconstruction error of the order of 1-2%. For compression of real-life color images it is inferior to JPEG and yield reduction ratios of about 4, at the treshold of notable picture degradation to the eye. A number of compresssion/decompression examples are shown.

Keywords: Image Compression, JPEG, Multipoint Taylor series, Hermite-Birkhoff interpolation.

1. Introduction

Data compression is becoming increasingly important in many domains of science and in everyday life [Sonka et al. (1994)].

Most of real-life color image compression is based on the JPEG standard [Pennebaker et al. (1993)]. Another approach to achieve large compression ratios is by fractal decomposition [Saupe et al. (1994)]. Lately, data reduction based on wavelet representations has become an active research area [Koornwinder (1993)]. In this paper we introduce still another approach to data compression for reasonable smooth varying images, as encountered in scientific applications and numerical modeling.

JPEG compression was intended to compress real-life color images that were to be judged by the human eye. JPEG compression ratios for full color images are typically in the range 10-30, at the threshold of visible degradation. For gray scale images this is less, with visible loss setting in at a ratio of about 5-10. This is due to the fact that the human eye is more sensitive to brightness variations than color variations.

The here presented technique can achieve gray scale compression ratios in the range 10-100 for smooth varying 2D images with a reconstruction error of typically 1 to 2 %. In these cases, our new method was found to be superior to JPEG in quality and compression ratios. Depending on the detail in the image, up to four times higher compression ratios have been achieved for the same accuracy than with JPEG. For real-life color pictures however, the compression rates achievable with our method are inferior to the JPEG algorithm. Our method therefore seems to be rather complementary to JPEG.

In our method, image decompression is based on a new solution for the Hermite-Birkhoff interpolation problem [Franssens (1998a,b)]. Here, the aim is to represent a function, up to a tolerated error, as a linear combination of its functional and partial derivative values (further called functional data), given at a discrete set of knot points. The knot positions and the functional data constitute the compressed data. The new interpolation method has a computational cost that is linear with the number of knots and is always numerical stable.

That this approach can result in a data reduction is motivated by the following simple observation. Suppose a 1D analytical function is equidistantly tabulated in 101 points over the interval $[0,1]$, hence the representation error is $O(h_1)$ with $h_1 = 0.01$. Now suppose this function is tabulated in 11 points, then $h_2 = 0.1$, and that its first derivatives are known in these points. Then the representation error is $O(h_2^2)$ with $h_2^2 = 0.01 = h_1$, but now only 22 values are used instead of 101, resulting in a data compression ratio of $r = 101/22 \approx 4.6$.

In the next sections we introduce our method for 2D images. Numerical examples are given for 1D and 2D data sets. However, the described algorithm is extendable to any number of dimensions.

2. The compression algorithm

We assume that the image to be compressed can be regarded as a scalar real-analytic function over a subset of the 2D plane, $D \subset R^2$, and is sampled at a dense, regular, rectangular grid of dimensions $L = L_x L_y$. For color images the algorithm is applied to the R, G and B components separately.

As compressed data we will use the functional data at a number of knot points $N < L$. Let M represent the number of values supplied at each knot, then the compressed data consists of $K = NM$ numbers. The data reduction factor is then $r = L/(NM)$.

The compression algorithm must determine the knot positions and compute the partial derivatives at the knots. In practice the algorithm consists of the following steps:

(i) Compute partial derivatives by a Fast Fourier Transform (FFT) technique.

(ii) Apply a strategy to determine knot positions.

We experimented with the following knot placing strategies:

a) regular grid,

b) adaptive grid,

c) minimum error adaptive grid iteration,

d) minimum error regular grid iteration.

2.1 Calculation of the derivatives

In practice partial derivatives are required at the knots for typical up to third order. Using higher orders does not contribute much to the accuracy of the representation, because of the increasing numerical errors in calculating them. Furthermore, retaining higher orders reduces the compression ratio r.

The most efficient and most accurate way for numerically computing partial derivatives of a regular sampled function is by using a FFT [Cooley et al. (1965), Brigham (1974), Bloomfield (1976)]. To compute the k,l-derivative, multiply the 2D FFT of the image with the filter

$$h^{k,l}(f_x, f_y) \stackrel{\Delta}{=} (j2\pi f_x)^k (j2\pi f_y)^l \qquad (2.1.1)$$

and inverse FFT this product. Herein are f_x, f_y the spatial frequencies in both dimensions.

2.2 Knot placing strategie

Depending on the variability of the image to be compressed, one can try various knot placing strategies.

The simplest method is using a coarse, regular, rectangular grid. This strategy often gives good results for complex images and is trivial to apply.

For images with localized features an adaptive, irregular grid might give higher compression ratios. An example of an automatic strategy for knot placing is at the minima of the function

$$t(x,y) \stackrel{\Delta}{=} |c(x,y)| \left(\left| \frac{\partial c(x,y)}{\partial x} \right| + \left| \frac{\partial c(x,y)}{\partial y} \right| \right) \qquad (2.1.1)$$

where

$$c(x,y) \stackrel{\Delta}{=} \frac{\partial^2 f(x,y)}{\partial x^2} + 2\frac{\partial^2 f(x,y)}{\partial x \partial y} + \frac{\partial^2 f(x,y)}{\partial y^2} \qquad (2.1.2)$$

is a measure of curvature. This assures that knots are placed at positions of zero and extreme curvature.

A further improvement can be obtained by using an iterative approach of knot insertion till the root mean square (rms) error between the original f and reconstructed image $\tilde{f}$,

$$\varepsilon_{RMS}(x,y) \stackrel{\Delta}{=} \sqrt{\frac{\iint_D \left(\tilde{f}(x,y) - f(x,y)\right)^2 dxdy}{\iint_D \left(f(x,y)\right)^2 dxdy}} \qquad (2.1.3)$$

drops under a given maximum. The additional knots to be inserted are most effectively positioned at the maxima of $\varepsilon_{RMS}(x,y)$.

The fourth possibility is the iterative refinement of a regular grid till the overall maximum of $\varepsilon_{RMS}(\mathbf{x})$ drops under a given maximum error.

3. The decompression algorithm

The decompression algorithm is basically an interpolation technique that uses the compressed data to reconstruct the original image. This kind of interpolation is called Hermite-Birkhoff interpolation [Stoer et al. (1980)]. It is more general than the common Lagrange interpolation, because it also uses the values

of the partial derivatives at the knot points to approximate an unknown function. The classical solution to this interpolation problem consists in the construction of the unique polynomial that has the desired properties at the knots [Ahlin (1964), Ciarlet et al. (1972), Gasca et al. (1982)]. In practice however, computing this polynomial can be very time consuming and is prone to numerical instabilities. To circumvent these problems we derived a new solution to the Hermite-Birkhoff interpolation problem, which eliminates the use of polynomials and hence all its associated drawbacks [Franssens (1998a)].

In our method the decompressed image is represented by a truncated MT series. A MT series is a new concept in function representation theory, which generalizes the well known Taylor series. Where an ordinary Taylor series represents a function in terms of the derivatives at a single point, a MT series is a representation in terms of the derivatives at $N>1$ points. The advantage of using a MT over a Taylor series representation is that we can trade higher derivatives at a single point for lower derivatives at multiple points, for the same accuracy.

The practical decompression algorithm consists of the following steps:

(i) Given the knot positions, decompose the domain by Delaunay triangulation.

(ii) Construct basis functions with compact support over a Voronoi polygon and tabulate them at the reconstruction grid points. Given the functional data at the knot positions, evaluate the truncated MT series interpolant.

(iii) An optional spectral filtering.

3.1 Domain decomposition

For practical computation, it is desirable to have a local interpolation algorithm. This means that the interpolated value at an interpolation point is only influenced by the data at the closest knot points. Then the computational cost of the algorithm becomes linear in the number of knots. The local nature of a interpolation algorithm is most conveniently achieved by introducing a decomposition of the interpolation domain D.

A very convenient decomposition method is Delaunay triangulation [Fortune (1992)]. It has the property that within the circumscribed circle of a triangle, having three knots as vertices, no other knot is found. This will assure us that the interpolation points inside a triangle are only influenced by the closest knots. Delaunay triangulation can be done for any set of irregularly distributed knots.

To make the interpolation local one needs basis functions with compact support. As support area we use the interior of the Voronoi polygon around each knot. The Voronoi polygon is defined as follows. Consider a set of N distinct knots, $S=\{\mathbf{x}_n=(x_n,y_n),\ 1\le n\le N\}\subset \mathrm{D}$. The neighbors of a knot $\mathbf{x}_n$ are the ordered set $X_n=\{\mathbf{x}_i,\ i=1,N_n\}$, such that

$$\mathrm{V}_n=\{\mathrm{T}_{i,n}=\mathrm{T}(x_n,x_i,x_{i+1}),\ i=1,N_n,(cycl)\} \qquad (3.1.1)$$

are a set of Delaunay triangles $\mathrm{T}_{i,n}$ having $\mathbf{x}_n$ as their common vertex. Then V_n is called the Voronoi polygon at the knot $\mathbf{x}_n$.

3.2 The Multipoint Taylor series representation

A MT series has the form

$$f(\mathbf{x})=\sum_{n=1}^{N}\sum_{0=k+l}^{+\infty} f_n^{k,l}U_n^{k,l}(\mathbf{x}) \qquad (3.2.1)$$

where the $f_n^{k,l}$ are

$$f_n^{k,l}=\left.\frac{\partial^{k+l}f(x,y)}{\partial x^k\partial y^l}\right|_{\mathbf{x}=\mathbf{x}_n} \qquad (3.2.2)$$

For the $f_n^{k,l}$ to be the functional data at the knots, the basis functions $U_n^{k,l}(\mathbf{x})$ must be $C^{\infty,\infty}$ and satisfy

$$\left.\frac{\partial^{i+j}U_n^{k,l}(x,y)}{\partial x^i\partial y^j}\right|_{\mathbf{x}=\mathbf{x}_m}=\delta_{mn}\delta^{ik}\delta^{jl} \qquad (3.2.3)$$

for $k+l\ge 0,\ 1\le n,m\le N$.

For reasons explained elsewhere [Franssens (1998a)], we choose higher order basis functions $U_n^{k,l}(\mathbf{x}),\ k+l>0$ as related to the zero order functions $U_n^{0,0}(\mathbf{x})$ by

$$U_n^{k,l}(\mathbf{x})=\frac{(x-x_n)^k}{k!}\frac{(y-y_n)^l}{l!}U_n^{0,0}(\mathbf{x}) \qquad (3.2.4)$$

Conditions (3.2.3) then imply that the zero order basis functions $U_n^{0,0}(\mathbf{x})$ should satisfy

$$U_n^{0,0}(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_m} = \delta_{mn},\ \ 1 \le n,m \le N$$
$$\left.\frac{\partial^{i+j} U_n^{0,0}(x,y)}{\partial x^i \partial y^j}\right|_{\mathbf{x}=\mathbf{x}_m} = 0,\quad \begin{matrix} i+j>0 \\ 1 \le n,m \le N \end{matrix} \qquad (3.2.5)$$

Using (3.2.4) the MT series can be written as

$$f(\mathbf{x}) = \sum_{n=1}^{N} \left(\sum_{0=k+l}^{+\infty} f_n^{k,l} \frac{(x-x_n)^k}{k!} \frac{(y-y_n)^l}{l!} \right) U_n^{0,0}(\mathbf{x}) = f(\mathbf{x}) \left(\sum_{n=1}^{N} U_n^{0,0}(\mathbf{x}) \right) \qquad (3.2.6)$$

Because we assumed real-analyticity of the function $f(\mathbf{x})$ in D, the N ordinary Taylor series in (3.2.6), evaluated around the N knots for any interpolation point $\mathbf{x} = (x,y) \in \mathrm{D}$, converge to the same value $f(\mathbf{x})$. A sufficient condition for convergence of a MT series is therefore that $\mathrm{D} \subset \Omega_1 \cap ... \Omega_N$, with Ω_n the region of convergence of the n-th ordinary Taylor series. This condition can be relaxed if the basis functions have local support. For basis functions having their Voronoi polygon as support, a sufficient condition for convergence is that $\mathrm{T}(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) \subset \Omega_a \cap \Omega_b \cap \Omega_c$, i.e. any Delaunay triangle should be in the intersection of the convergence regions of the three ordinary Taylor series evaluated around its vertices.

From (3.2.6) follows that, to ensure convergence to the correct function, the zero order functions $U_n^{0,0}(\mathbf{x})$ must form a partition of unity over the domain D

$$\sum_{n=1}^{N} U_n^{0,0}(\mathbf{x}) \equiv 1 \qquad (3.2.7)$$

We will call the truncated MT series

$$\tilde{f}(\mathbf{x}) = \sum_{n=1}^{N} \sum_{0=k+l}^{M_n} f_n^{k,l} U_n^{k,l}(\mathbf{x}) \qquad (3.2.8)$$

the MT interpolant for a function $f(\mathbf{x})$ with functional data $F = \left\{ f_n^{k,l}, 0 \le k+l \le M_n, 1 \le n \le N \right\} \subset \mathrm{R}$. It is readily clear that (3.2.8) is a very natural solution to the Hermite-Birkhoff interpolation problem.

Conditions (3.2.5) do not uniquely determine the zero order basis functions. This is in sharp contrast with the ordinary Taylor series representation where the basis functions are the unique monomials. The non-uniqueness is no problem in the full MT series, since the exact form of the basis functions drop out if the summation is continued over all derivatives up to infinity, due to the partition of unity condition (3.2.7). But for the interpolant it is an additional freedom, which the algorithm designer can use to his advantage. For details about the construction of MT basis functions we refer to material to be published elsewhere [Franssens (1998a,b)]. We will just give here one particular form, which was found to be computational convenient.

The zero order basis function $U_n^{0,0}(\mathbf{x})$, associated to knot $\mathbf{x}_n$, was chosen to be

$$U_n^{0,0}(\mathbf{x}) = \frac{\Phi_n(\mathbf{x})}{\Phi_a(\mathbf{x}) + \Phi_b(\mathbf{x}) + \Phi_c(\mathbf{x})} \qquad (3.2.9a)$$

with

$$\Phi_n(\mathbf{x}) = \begin{cases} \Psi(w(r, r_b(\theta))), \mathbf{x} \in \mathrm{V}_n \\ 0, \qquad\qquad\quad \mathbf{x} \notin \mathrm{V}_n \end{cases} \qquad (3.2.9b)$$

$$\Psi(w) = \frac{1}{2}\left(1 + \tanh(w)\right) \qquad (3.2.9c)$$

$$w(r, r_b) = \frac{r_b^{\,2}}{4}\left(\frac{1}{r} - \frac{1}{r_b - r} \right),\ 0 \le r \le r_b \qquad (3.2.9d)$$

for the interpolation point $\mathbf{x} = \mathbf{x}_n + (r\cos\theta, r\sin\theta)$ and parametric representation of the Voronoi boundary $\partial \mathrm{V}_n = \{ \mathbf{x} : \mathbf{x} = \mathbf{x}_n + (r_b(\theta)\cos\theta, r_b(\theta)\sin\theta) \}$. Notice that due to the compact support of the $\Phi_n(\mathbf{x})$, only the three knots that form the vertices of the triangle containing the interpolation point, contribute to that point. It is therefore sufficient to sum only over these three functions in the denominator of (3.2.9a) to satisfy (3.2.7).

Any zero order basis function does not exceed in modulus one, is one at the interior knot and zero on the Voronoi boundary. In addition it has all its partial derivatives zero at the interior knot as well as on the Voronoi boundary. Values of higher order basis functions are obtained from (3.2.4). Due to the compact support of the basis functions the first sum in the interpolant (3.2.8) will also consists only of three terms.

An interesting simplification arises when the reconstruction is done from a regular grid. In this case we can take as zero order basis functions

$$U_n^{0,0}(\mathbf{x}) = \Psi\big(w((x-x_n)/\Delta x)\big)\Psi\big(w((y-y_n)/\Delta y)\big) \tag{3.2.10}$$

together with (3.2.9c-d). The grid step sizes of the decompressed image, $\Delta x', \Delta y'$, will in general be integer fractions of the grid step size of the coarse grid of the compressed data, so $\Delta x = J_x \Delta x', \Delta y = J_y \Delta y'$. This means that it is sufficient to tabulate $\Psi(w(j/J_x)), 0 \le j \le J_x$ and $\Psi(w(j/J_y)), 0 \le j \le J_y$. In this case only $J = J_x J_y$ evaluation of the transcendental zero order basis functions are necessary. On the other hand, in the case of an irregular grid of knot points, the number of zero order basis function evaluations is $J' = 3L$. A regular grid of knots therefore results in a very significant reduction in run time. In addition, a similar reduction in the computation of the $(x-x_n)^k/k!$ and $(y-y_n)^l/l!$ factors in (3.2.4) is achieved as well. In pratice one first computes a lookup table of all required basis function values, $I = MJ$ in total (not all different). The number of entries I will typically be small, say about 256. The value at an interpolation point is then found as a simple weighted average of the functional data at the four closest knots, the weights being entries from the lookup table. The total number of multiplication required for the decompression is then $I + 4ML$. Finally, notice that in this case also the triangulation of the domain is unnecessary.

To numerical evaluate the basis one should take some minor precautions to prevent overflow and underflow in (3.2.9b). It will then be clear, from (3.2.9a-d) or either from (3.2.10) and (3.2.9c-d), that the evaluation of the basis functions poses no numerical problems. Hence our interpolant (3.2.8), containing these simple trancendental functions, avoid the computational errors that are inherent in the classical interpolation approach, based on using polynomials as basis functions.

3.3 Final spectral filtering

After interpolation, as computed by (3.2.8), an optional additional spectral filtering can be applied. The main purpose is to remove unrealistic variations in the result, having a higher spatial frequency than the Nyquist sampling frequency, determined by the placement of the knots. The result is a smoothing effect.

Good results were obtained by applying a low pass, spatial filter of Gaussian shape. Filtering is done by taking the FFT of the interpolated image, then multiplying it by the filter

$$g(f_x, f_y) = \exp\Big(-\big((f_x/f_a)^2 + (f_y/f_b)^2\big)\Big) \tag{3.3.1}$$

and inverse FFT the product. Herein are f_a and f_b filter characteristic, determining the spectral width of the filter. Typical values are 0.2 (in Nyquist frequency units).

4. Examples

Fig. 1 shows the reconstruction of a 1D function. The solid line is the reconstructed function, the dashed line is the original function and the vertical dashed lines mark the knot positions. The original function was tabulated in 721 points. The number of knots is 28 and functional data up to third order derivatives were used. The rms reconstruction error is 0.4%. The data reduction ratio is 6.4. Knots were placed at the zeroes of the second and third derivatives (i.e. the 1D version of expression (2.1.1) was used) and no iterative adaptation was applied.

Fig. 2a and b show the original and decompressed version of the simple 2D test function

$$f(x,y) = \cos(\pi x)\exp(-25y^2), \quad -1 \le x \le 1, \ -1 \le y \le 1 \tag{4.1}$$

The original function was tabulated in 360x360 points. Tables 1a,b show the rms error and reduction factor obtained with the new method and with the JPEG algorithm, respectively, for varying compression rates. The knots were placed at a regular, rectangular grid.

Derivatives	Grid size	Rms error	Ratio r
$0 \le k+l \le 3$	19x19	0.02 %	36
$0 \le k+l \le 3$	13x13	0.10 %	77
$0 \le k+l \le 3$	10x10	0.25 %	130
$0 \le k+l \le 2$	10x10	1.0 %	217
$0 \le k+l \le 1$	10x10	1.5 %	434

Table 1a. New method applied to test function.

Q factor	Rms error	Ratio r
100	0.1 %	11
75	0.35 %	40
50	0.75 %	48
25	1.0 %	53
5	3.2 %	66

Table 1b. JPEG method applied to test function.

The example shows that the new method can achieve a better accuracy and compression rates of up to four times those obtained by JPEG (cf. row of rms error 1.0 %).

Figs. 2a and b show the original and decompressed version of a 2D test data set, representing the variation of total optical IR sunlight extinction in the stratosphere due to aerosols (in arbitrary units). The reconstruction rms error in Fig. 2b is 1.2 %. The original function was tabulated in 360x360 points. Tables 2a,b show the rms error and reduction factor obtained with the new method and with the JPEG algorithm, respectively, for varying compression rates. The knots were again placed at a regular, rectangular grid and no higher than first order derivatives were used.

Grid size	Rms error	Ratio r
26x26	1.1 %	64
24x24	1.4 %	75
19x19	1.8 %	120
17x17	2.4 %	150
15x15	3.0 %	193

Table 2a. New method applied to test data set.

Q factor	Rms error	Ratio r
25	1.1 %	33
15	1.4 %	37
11	2.0 %	42
5	4.0 %	54
2	8.0 %	65

Table 2b. JPEG method applied to test data set.

Finally, Figs. 3a and b show the original and decompressed images of a RGB color photograph (scanned to 8x8x8 bits in 300 dpi, with size 512x512), using the new method. The visual quality of the decompressed image is fair, but the compression ratio is only 4. We applied our method to the R,G and B components separately. The compressed data were computed on a regular, rectangular grid of 64x64 points. Spatial derivatives up to third order were used ($0 \le k \le 3, 0 \le l \le 3$). Although the method was not designed to compress this kind of pictures, it still performs reasonably well. It is however no match for the JPEG algorithm in this case, which yields a compression ratio of 50 for this image. On the other hand, our method is still in experimental stage and some further refinements in gradient computation, the inclusion of colour transformation and the use of compressed data coding techniques (as is done in JPEG) might add an additional factor of 2 or 3 to the eventual performance.

The automatic knot placing strategy, based on expression (2.1.1), in general resulted in much lower compression ratios for equal quality. It was found that, for general 2D images, a regular grid gives the best results in quality and compression ratios. For 1D data sets on the other hand, the automatic strategy gives good results.

5. Conclusions

A data compression/decompression method was proposed based on a new algorithm for Hermite-Birkhoff interpolation.

The compression part uses a knot placing strategy and the FFT approach to compute a number of partial derivatives of the image at these knots. The compressed data consists of the knot positions and a set of functional data at the knots.

The decompresssoin part reconstructs the image as an interpolant that uses the compressed data for its input. This interpolant is a truncated Multipoint Taylor series. It has a computational cost that is linear with the number of knots and is numerically stable.

Several automatic knot placing strategies were considered. An iterative procedure can produce compressed data that is guaranteed to reproduce the given image to within a given error tolerance. However, for 2D images it was found that a regular grid gives the best results of quality and compression ratios. In addition, these grids significantly speed up the decompression process. The required number of floating point operations for decompression then scales as 4ML, with M the number of functional values used per knot and L the number of knots (cf. end of section 3.2).

The proposed method can achieve gray scale compression ratios in the range 10-100 for smooth varying 2D images with a reconstruction error of typically 1 to 2 % JPEG (cf. Tables 1a,b and 2a,b). In these cases, it was found to be superior to JPEG in quality and compression ratios. Depending on the detail in the image, up to four times higher compression ratios can be achieved for the same accuracy in reconstruction than with JPEG (cf. Tables 1a,b). For real-life color pictures however, the compression rates achievable with our method are inferior to the JPEG algorithm. Our method therefore seems to be complementary to JPEG and worth considering when compression of computer generated, functional data is required.

References

M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis and Machine Vision*, Chapman and Hill, London, 1994.

J. Pennebaker, J. Mitchell, *JPEG still image compression standard*, Van Nostrand Reinhold, New York, 1993.

D. Saupe, R. Hamzaoui, "A review of the fractal image compression literature", *Computer Graphics*, vol 28(4), pp. 268-279, 1994.

T. Koornwinder, Ed., *Wavelets: an elementary treatment of theory and applications*, vol. 1 in Series in Approximations and Decompositions, World Scientific, Singapore, 1993.

G. Franssens, "A new non-polynomial univariate interpolation formula of Hermite type", submitted to *Advances in Computational Mathematics*, 1998.

G. Franssens, "Multivariate Hermite interpolation based on a Multipoint Taylor series representation", *in preparation*, 1998.

J. Cooley, J. Tukey, "An algorithm for the machine calculation of complex Fourier series", *Math. Comput.*, vol 19, pp. 297-301, 1965.

E. Brigham, *The Fast Fourier Transform*, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

P. Bloomfield, *Fourier Analysis of Time Series*, Wiley, New York, 1976.

J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1980.

A. Ahlin, "A bivariate generalisation of Hermite's interpolation formula", *Math. Comput.*, vol. 18, pp. 264-273, 1964.

P. Ciarlet and P. Raviart, "General Lagrange and Hermite interpolation in R^N with applications to finite element methods", *Arch. Rational Mech. Anal.*, vol. 46, pp. 177-199, 1972.

M. Gasca and J. Maetzu, "On Lagrange and Hermite Interpolation in R^N", *Numerische Mathematik*, vol. 39, pp. 1-14, 1982.

S. Fortune, *Voronoi diagrams and Delaunay triangulations*, in Computing in Euclidean Geometry, World Scientific, Singapore, 1992.

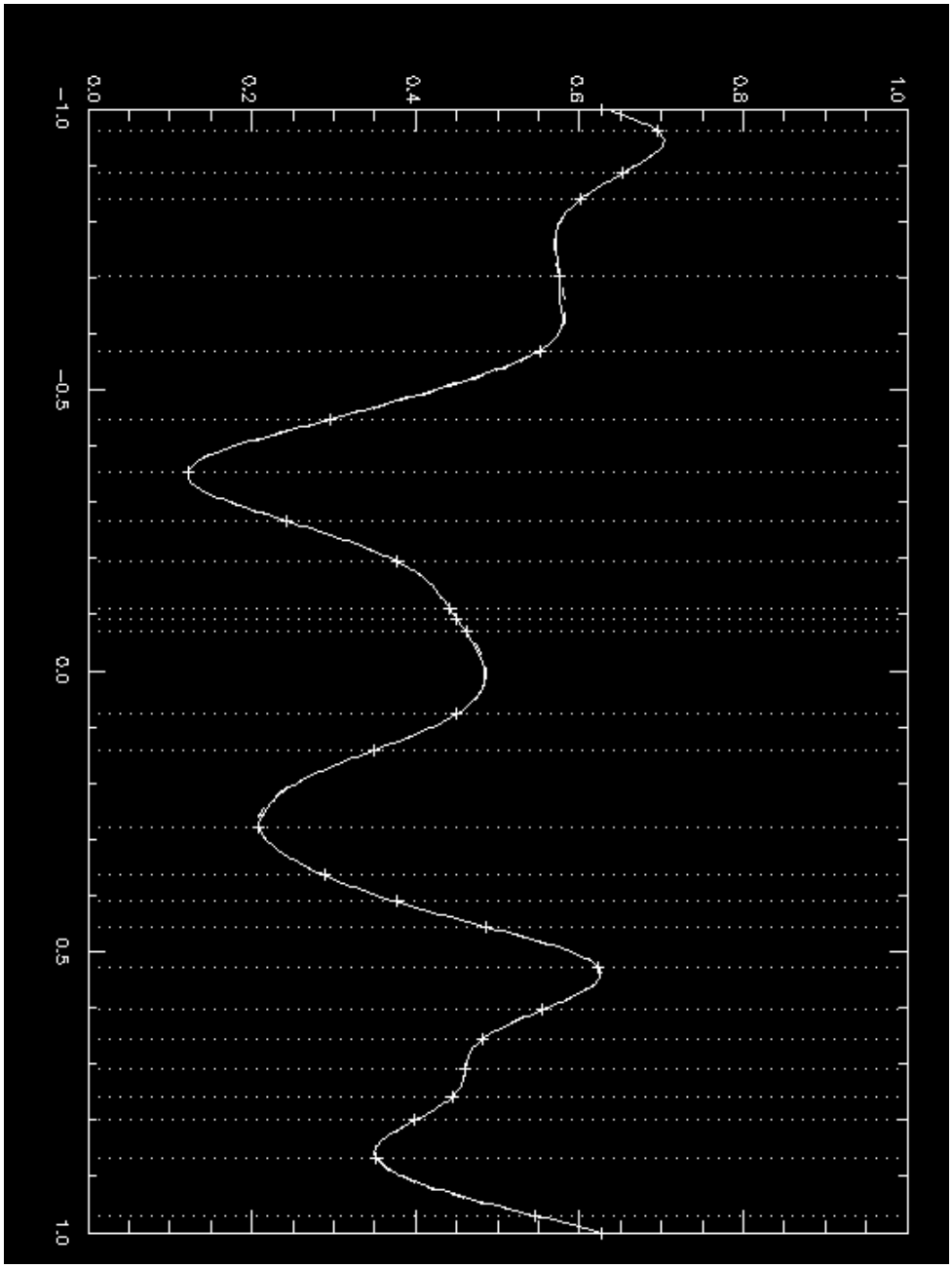

Fig. 1 Reconstruction of a one dimensional function.

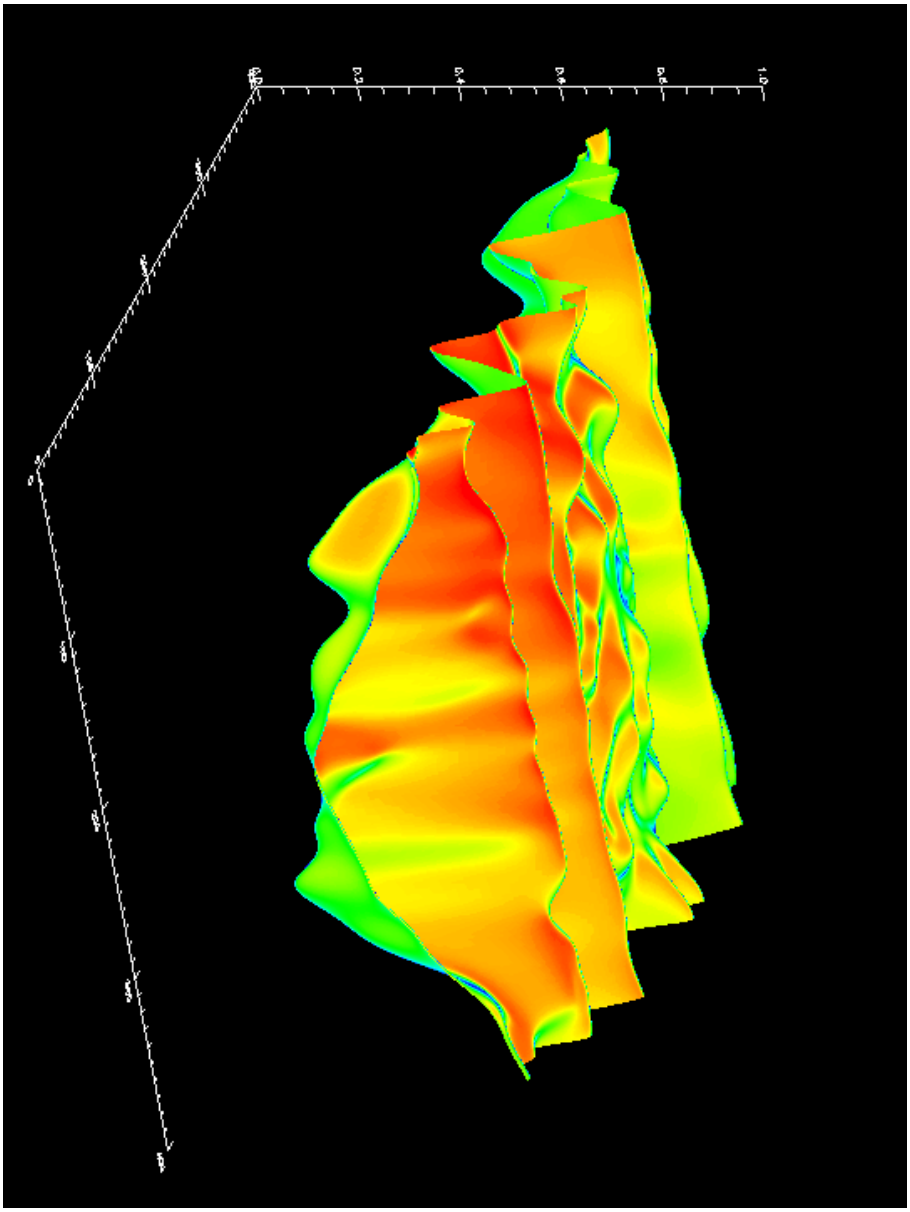

Fig. 2a Original test data set.

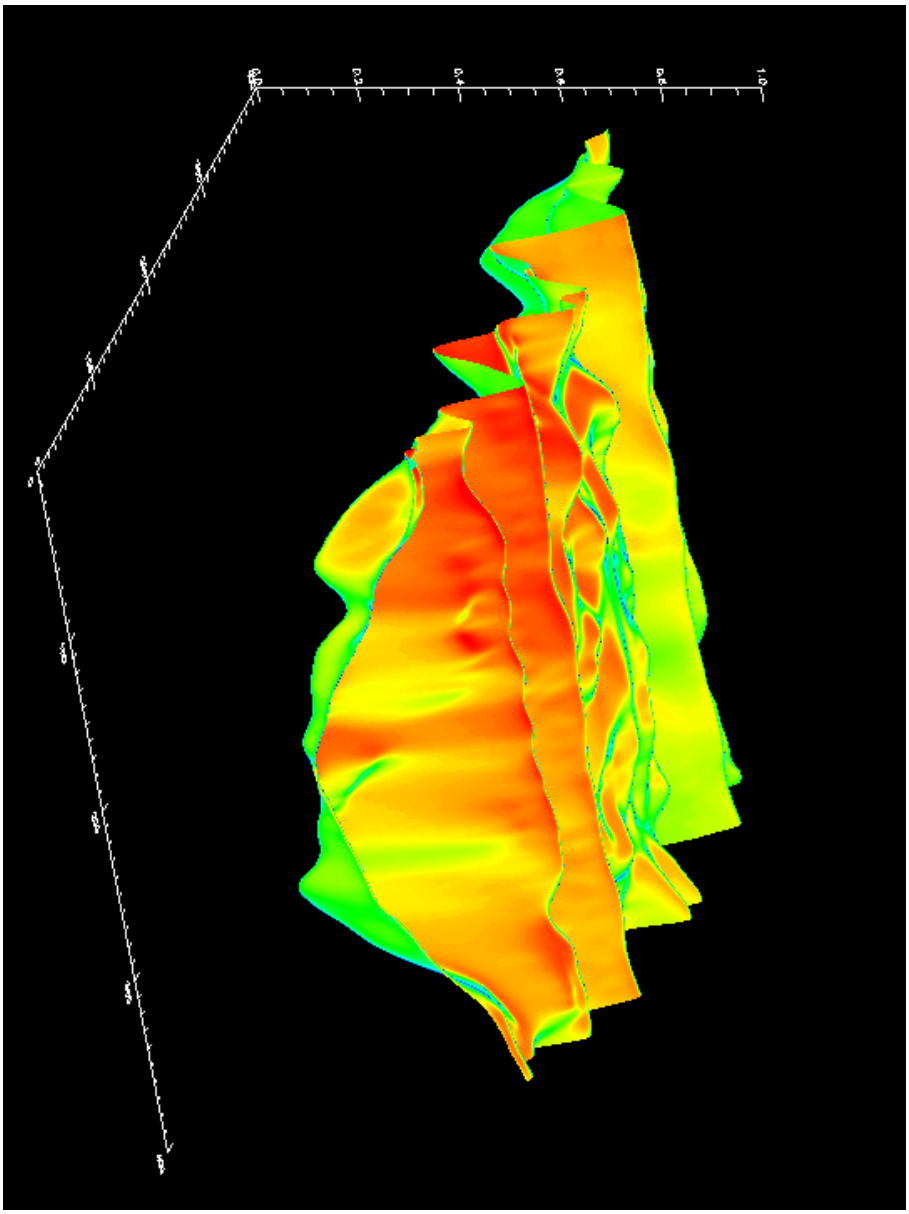

Fig. 2b Reconstructed test data set.

Fig. 3a Original image.

Fig. 3b Decompressed image.