Interpolation of two-dimensional surfaces using the Gerchberg algorithm

Mark J. Carlotto, Victor T. Tom
The Analytic Sciences Corporation
One Jacob Way, Reading, Massachusetts 01867

Abstract

The Gerchberg algorithm has been successfully applied to signal enhancement, reconstruction and extrapolation problems where only partial information is available in the space (time) and frequency domains. In this paper, the Gerchberg algorithm is applied to the iterative interpolation of two-dimensional (2-D) surfaces from irregularly spaced data points. Specific applications presented are: the generation of hydrographic surfaces from bathymetric data obtained from the hydrographic airborne laser sounder (HALS) and the generalization of wind-flows from cloud imagery obtained from the Geostationary Operational Environmental Satellite (GOES). Experimental results obtained using a VAX 11/780 and FPS 120-B array processor system are presented.

Introduction

In signal processing one frequently encounters the problem of recovering a signal from partial information in the space (time) and frequency domains. Each partial specification is usually not sufficient to determine the entire signal; however, taken together there may be one solution or a class of solutions that are consistent with all the information.

A scheme of reconstruction which affords effective utilization of partial information in the space and frequency domains is the iterative formulation, in particular, the Gerchberg algorithm. The Gerchberg algorithm and its variants iterate back and forth between the time and frequency domains substituting known information into the estimate at each iteration step. They have been applied in problems related to bandwidth extrapolation[1], signal extrapolation[2], and reconstruction from phase or magnitude[3]. This paper addresses the related problem of interpolation, given data whose region of support can be highly irregular under the constraint that the data follows an underlying smooth (in some sense) model.

The organization of this paper is as follows: The next section describes the formulation of the Gerchberg algorithm for interpolation and discusses its stability. Issues concerning the implementation of the algorithm on an array processor are then presented in the following section. Two application areas in which this technique has been successfully applied are described in the last two sections.

Interpolation using the Gerchberg Algorithm

In this section the use of the Gerchberg algorithm to interpolate a 2-D surface to known data subject to a spatial bandwidth (smoothness) constraint is described. A block diagram of the Gerchberg algorithm is shown in Fig. 1. The objective is to interpolate an M x N point surface to K data points d_k located at (i_k, j_k) for k = 0, 1, ... (K-1). By repeatedly substituting the data back into the estimate of the surface and lowpass filtering, the Gerchberg algorithm attempts to arrive at a solution which fits the data and is consistent with the spatial frequency constraint. At each iteration, the data is substituted into the estimate s_{ij} according to

$$s_{ij} = \{1 - \delta(i-i_k, j-j_k)\} s_{ij} + \delta(i-i_k, j-j_k) d_k \tag{1}$$

for k = 0,1,...(K-1) where $\delta(i-i_k, j-j_k) = 1$ at the K data points and zero elsewhere. Letting $s = \{s_{ij}\}$ be an MN x 1 vector, Eq. 1 can be written in matrix notation as

$$s = As + Bd \tag{2}$$

where A is the MN x MN matrix $\{1 - \delta(i-i_k, j-j_k)\}$ and B is the MN x K matrix $\{\delta(i-i_k, j-j_k\}$ for k = 0,1,...(K-1). After forcing the surface to equal the known data, the spectrum is forced to be within the specified bandwidth. Letting F denote the filtering operation:

$$S_{ij} = 0 \text{ for } |i|>W_x \text{ and } |j|>W_y \tag{3}$$

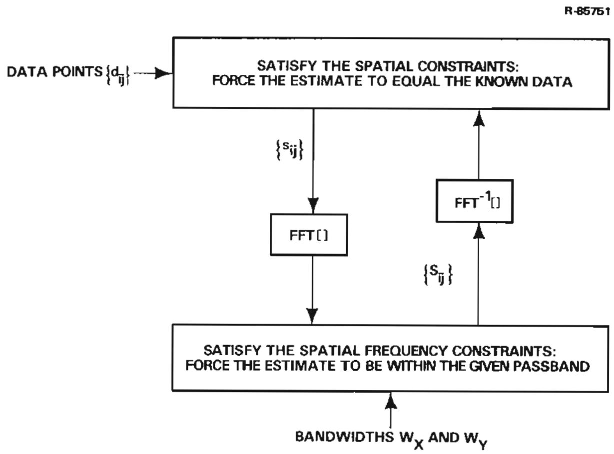


Figure 1. Block Diagram of the Gerchberg Algorithm

where S_{ij} is the discrete Fourier transform (DFT) of s_{ij}, the interpolation may be described by the 1-st order iteration:

$$s(t+1) = AFs(t) + Bd \tag{4}$$

where s(t) is the t-th estimate for $t = 0,1,\ldots,T$ and the initial estimate $s_{ij}(0) = 0$. For Eq. 4 to converge, the moduli of the eigenvalues of the matrix AF must be between zero and one. If Eq. 4 is represented by the mapping

$$s(t+1) = \Gamma s(t) \tag{5}$$

from the t-th to the (t+1)-st estimate where $\| s(t) \| < \infty$ and Γ represents the combined operations of spatial replacement and filtering, a sufficient condition for Eq. 5 (hence Eq. 4) to be stable is that Γ be a non-expansive mapping[4]; i.e.,

$$\| \Gamma s(t) - \Gamma u(t) \| \leq \| s(t) - u(t) \| \tag{6}$$

for any s(t) and u(t). The Gerchberg and related algorithms have been shown to be stable for the application of recovering a signal from partial information in the time and frequency domains[5]. Substituting Eq. 4 into the left-hand side of Eq. 6 yields

$$\| \Gamma s(t) - \Gamma u(t) \| = \| AFs(t) - AFu(t) \| . \tag{7}$$

The replacement operation A forces the differences between the filtered estimates to be zero at the K data points and leaves the filtered estimate unchanged at the other points. Therefore

$$\| \Gamma s(t) - \Gamma u(t) \| \leq \| Fs(t) - Fu(t) \| \tag{8}$$

where Eq. 8 holds with equality only when the filtered estimates Fs(t) and Fu(t) are identical at the K data points. The filtering operation F forces the spectrum to be equal to zero beyond the specified W_x and W_y bandwidths so that

$$\| Fs(t) - Fu(t) \| \leq \| s(t) - u(t) \| \tag{9}$$

where Eq. 9 holds with equality only when the S(t) and U(t) spectra are identical outside the passband. Combining Eq. 7 through Eq. 9 yields Eq. 6 which holds with equality when the estimates equal the data at the K data points and the support of the spectra are contained in the specified passband. Thus Γ is non-expansive and the interpolation algorithm is stable.

For the solution s to be unique, the number of discrete Fourier components passed by the filter (single-sided) must be less than or equal to the number of data points processed (i.e., $W_xW_y \leq K$). In addition, for the solution to converge to the true solution, the number of data points must be equal to or greater than the number of non-zero discrete Fourier coefficients in the DFT of the true surface. The rectangular filter function (Eq. 3) coarsely delimits the frequency response of the interpolator. Since the support of the true DFT and the filter will generally be different, the estimated solution will be an approximation to the true signal.

Implementation Considerations

This section describes the implementation of an interpolator based on the Gerchberg algorithm on a VAX 11/780 and FPS 120-B array processor (AP) system. A comparison of the Gerchberg algorithm with other interpolation techniques (polynominal fitting, least-squares approximation, and local area interpolation, e.g., cubic spline) was performed[6]. The Gerchberg algorithm was found to be computationally more efficient than polynominal techniques in terms of the number of operations performed. The regularity of the computations and the ability to easily change the order of the interpolation was significantly better than local area techniques. Finally, the memory requirements of the Gerchberg algorithm were lower than least-square techniques (which require the generation and storage of the orthonormal basis functions).

The regularity of the alternating spatial replacement and filtering operations (Eqs. 2 and 3) allowed an efficient implementation of the Gerchberg algorithm on the AP. A program running in the VAX reads the data files from disk, sections the data set into sub-blocks for processing, and coordinates the operation of the AP. The host supplies a sub-block of data, the number of iterations to be performed, and the filter bandwidths. The AP returns the maximum difference between the interpolated surface and the data:

$$\varepsilon = \text{maximum } \{s_{i_k j_k} - d_k\} \tag{10}$$

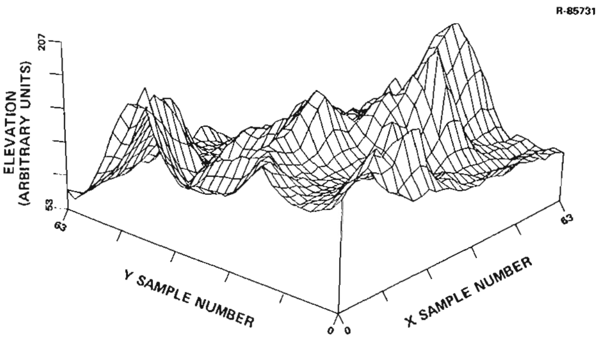
(This is computed prior to the last replacement operation since each replacement operation forces this difference equal to zero). When the differences in Eq. 10 is less than a predetermined threshold ε_T, the interpolated surface is read out of the AP.

To demonstrate the behaviour of the Gerchberg algorithm and to verify the operation of the interpolator, a 64 x 64 point synthetically generated terrain elevation surface was sampled and an interpolation was performed. (Given the one-quarter megabyte of data memory in the AP, up to 64 x 64 point interpolations can be performed at a time). A 2-D perspective plot of the original terrain surface is depicted in Fig. 2a. Half-power bandwidths (single-sided) in the x- and y-directions of $W_x = 8$ and $W_y = 5$ frequency bins were computed for this surface. The surface was sampled and a sparse array of approximately 60 data points was generated. The results after 1, 10, and 100 iterations are shown in Figs. 2b-2d. Peak differences between the estimated surface and the data (Eq. 10) were 117.7, 83.34, and 17.61 respectively. The corresponding root-mean-square (rms) differences were 4.37, 2.31, and 0.22.

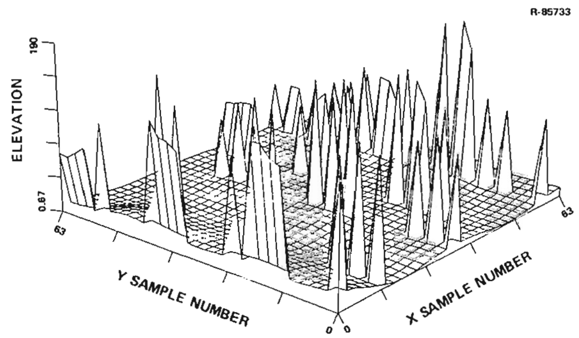
In practice, since the original surface is unknown, the bandwidths must be determined indirectly. A technique successfully used to interpolate the HALS and GOES data initializes the filter bandwidths to a low value (typically $W_x = W_y = 1$). After performing a specified number of iterations, if the difference ε is greater than the threshold ε_T, the bandwidths are increased by some predetermined increment. In addition, these differences are an indication of the degree to which the algorithm has converged and can be used to terminate the iteration automatically without supervision.

As stated earlier, the M and N array dimensions are determined by the amount of data memory available in the AP. In order not to be limited by the AP memory, a scheme for partitioning large data sets into sub-blocks for processing was developed. Local area techniques have been found to be superior to global interpolation techniques, particularly when the order of the interpolation is large[7]. Since the bandwidths of sub-blocks are adjusted independently, a method of block processing the data which maintains continuity across the data set is required.

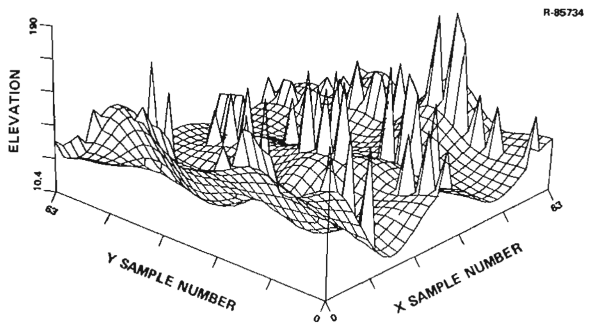
If x is a P x Q data set (i.e., a sparse array of data points) and s is the M x N array ($M < P$ and $N < Q$) which represents the domain of the interpolator, the blocking scheme is as follows: Read the first M x N sub-block from x, interpolate a surface s to fit the data in the sub-block, and write the result back into x. If x is processed in a raster fashion from left to right and top to bottom, the second sub-block in x initially contains the right-most M-element column of the previous surface s plus (N-1) columns of data. The second sub-block is read, processed, and written back into x. After covering the top M rows (of Q elements each) in x, the interpolator moves down (M-1) rows and continues to read, process, and write x. This continues until all P rows of x have been processed. Continuity is thus maintained across the data set by using the edge row and/or column from adjacent sub-block(s) previously processed as boundary values for succeeding sub-blocks. An added advantage of this technique is that it is performed in-place thus reducing storage requirements. This method is applied specifically to interpolate the bathymetric data in the next section.



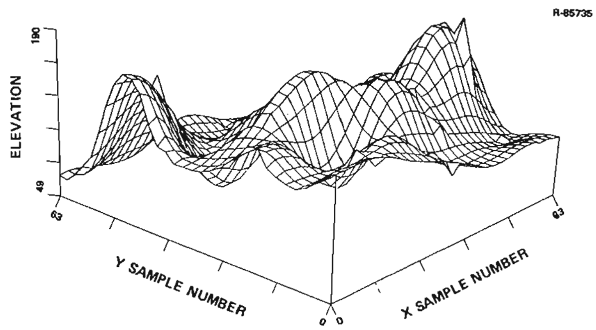
a) 64x64 Point Synthetic Terrain Surface



b) Surface Estimate after 1 Iteration



c) Surface Estimate after 10 Iterations



d) Surface Estimate after 100 Iterations

Figure 2. Experimental Verification of Gerchberg Algorithm

Interpolation of Bathymetry Data

This section describes the interpolation of 2-D surfaces from data obtained from the hydrographic airborne laser sounder (HALS). HALS is a laser bathymetry device for measuring the depth of coastal waters (up to 50 meters in depth depending on water clarity). The data is collected in a spiral sampling pattern (Fig. 3) traced out by a pulsed blue-green laser beam scanned in an elliptical pattern and translated by the motion of the aircraft. The 256 data points shown in Fig. 3 represents only part of the 4400 depth measurements made in the 1000 x 100 meter survey area processed. The measured depth is proportional to the time-delay between the air-water and bottom return pulses.

The survey area was gridded into a 512 x 64 block of data (spaced 2 meters between samples). The array was partitioned into eight 64 x 64 sub-blocks and was processed as described earlier. In each sub-block, 20 iterations were performed at a given bandwidth. If the peak difference between the data and the interpolated surface was greater than a threshold (ε_T = 3 or 10% of maximum depth), the bandwidth was doubled. The W_x and W_y bandwidths were initialized to one frequency bin (0.0039 cycles/meter) at the start of processing each sub-block. The resultant surface is shown in Fig. 4. Since the hydrography was fairly smooth in sub-blocks 0-3 and 6-7, the corresponding bandwidths were small (Table 1). In the shoal area (sub-blocks 4-5), higher bandwidths were required by the interpolator in order to generate a surface of sufficient order to fit the measurements. The 512 x 64 sample survey area was processed in approximately 5 minutes under normal system loading.

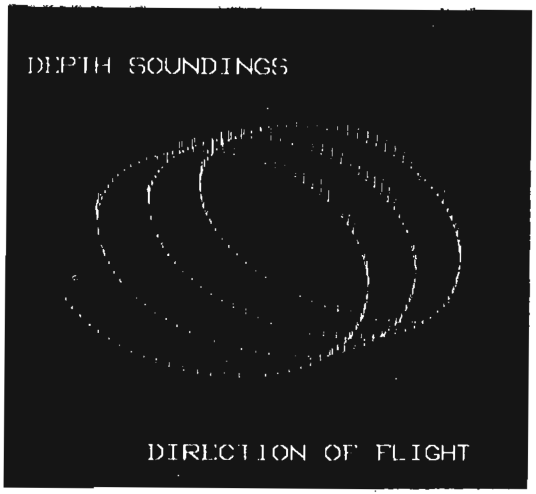


Figure 3. 256 Sample Section of HALS Bathymetry Data

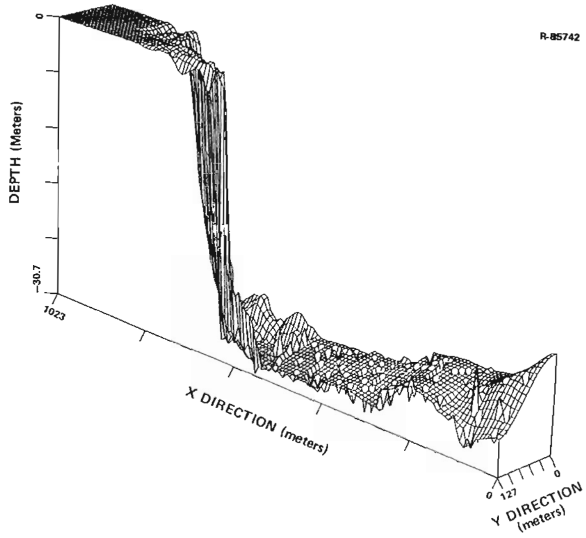


Figure 4. 512x64 Point Surface Estimate (1000x100 meters)

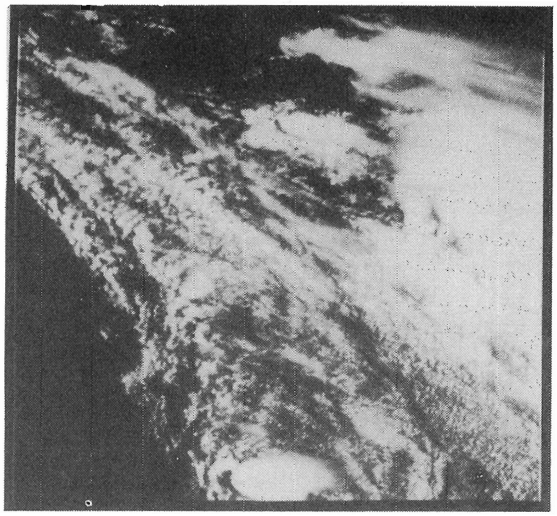
Table 1. Results of HALS Interpolation

Sub-Block	Peak Difference	Bandwidth (x 0.0039 cycles/meter)
0	2.26	2
1	2.58	1
2	1.28	1
3	1.38	1
4	2.40	32
5	2.83	8
6	2.99	4
7	0.13	1

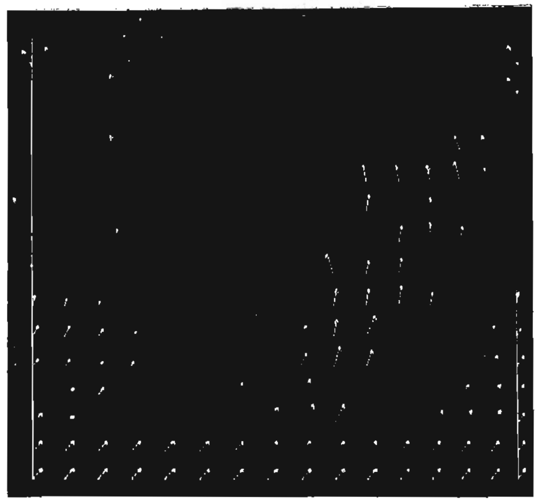
Generalization of Atmospheric Wind-Flows

This section describes the interpolation of wind vector fields from data obtained from two consecutive frames of GOES cloud imagery (Fig. 5a). An initial wind vector field on a square grid was first computed by calculating normalized correlation coefficients ρ for 32 x 32 sub-blocks and their corresponding search windows. The offsets of the correlation peaks indicate the cloud displacement in x and y, and the maximum correlation values are indicative of the goodness of fit between frames. These initial vectors were then edited to include only those vectors that were due to clouds with 2 kilometer base heights. An additional screening was then performed to retain only those vectors which correspond to high values of ρ, thereby mitigating the effects of changing cloud morphology on this procedure. The remaining vectors (Fig. 5b) are then used as the basis for interpolation.

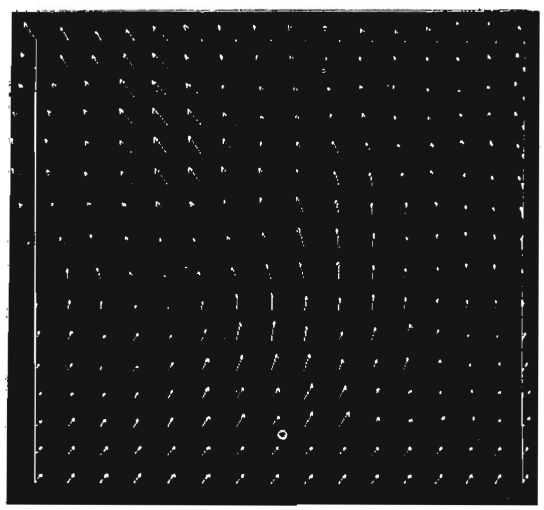
For the present example, the Gerchberg iteration was applied to the x- and y- components of the wind field separately. For each component, 20 iterations were performed at a given bandwidth, similar to the bathymetric processing. The interpolated wind field is shown in Fig. 5c. This particular example did not utilize the block processing capability, although with much larger data sets, it would have been necessary.



a) GOES Cloud Image



b) Edited Correlation Vectors



c) Interpolated Wind Vectors

Figure 5. GOES Wind Field Generation

Summary

The Gerchberg algorithm has been shown to be an effective interpolation tool for generating smoothly varying surfaces from incomplete data. A description of the algorithm was presented as well as a discussion of its stability behavior. Several key issues regarding its implementation were presented including

- Array processor considerations
- Adaptive bandwidth processing
- Block processing with boundary constraints.

Finally the effectiveness of the procedure was demonstrated by interpolating examples of bathymetric and wind field data.

References

1. Gerchberg, R.W., "Super-Resolution Through Error Energy Reduction," Optica Acta, Vol. 21, No. 8, pp 709-720, 1974.
2. Papoulis, A., "A New Algorithm in Spectral Analysis and Bandlimited Signal Extrapolation," IEEE Trans. Circuits Syst., Vol. CAS-22, pp. 735-742, September 1975.
3. Hayes, M.H., "Signal Reconstruction from Phase or Magnitude, ScD. Dissertation, Dept. Elec. Eng. Computer Sci., Massachusetts Inst. Technology, Cambridge, MA, 1981.
4. Schafer, R.W., Mersereau, R.M., Richards, M.A., "Constrained Iterative Restoration Algorithms," Proceedings of the IEEE, Vol. 69, No. 4, 1981.
5. Tom, V.T., Quatieri, T.F., Hayes, M.H., McClellan, J.H., "Convergence of Iterative Nonexpansive Signal Reconstruction Algorithms," IEEE Trans. Acoust., Speech Signal Processing, Vol. ASSP-29, pp. 1052-1058, 1981.
6. Carlotto, M.J., An Iterative Technique for HALS Post-Processing, TIM-3244-4, The Analytic Sciences Corporation, Aug. 1982.
7. Conte, S.D., deBoor, C., Elementary Numerical Analysis: An Algorithmic Approach, McGraw Hill Book Company, p. 231, 1972.