

Detecting Implicit Grids in Terrain Databases using a Fast Fourier Transform

Victor Skowronski

Thomas Stanzione

TASC

55 Walkers Brook Drive

Reading, MA 01867

email: vjskowronski@tasc.com

Keywords:

Terrain Grids, Terrain Representation, Format Conversion

ABSTRACT: *The format of terrain data can greatly affect the size and efficiency of a terrain database. In particular, an elevation grid format is much more efficient than a triangulated irregular network (TIN). Thus efficiency gains are possible for terrain databases that can use either format, such as the Compact Terrain Database (CTDB), if the nature of the source data can be determined when the database is created. Some source formats are explicitly gridded. Others, such as S1000, are composed of polygons, but have an implicit grid pattern. In the case of S1000, data about the implicit grid is stored as part of the source data. A conversion program can read this data and adjust its grid parameters for the most efficient grid.*

However, the use of standard interfaces, such as the Synthetic Environment Data Representation Interchange Specification (SEDRIS), for transmitting terrain data, makes it less certain that information about explicit or implicit grids will be available to the conversion program. In cases where no information about grids in the source data is available, the conversion program must either select an arbitrary and possibly inefficient terrain representation, or calculate the optimal parameters itself.

This paper proposes a technique for finding an optimal set of grid parameters for a terrain database using the Fast Fourier Transform. Elevation readings in intervals along the X- and Y- axes of the database are counted. The Fourier Transform of the resulting arrays provides information about the spatial frequency of the elevation readings. This information can be used to calculate the optimum grid post origin and spacing for the database being generated. It has the advantage that it needs no special information from the source database.

1.

1. Introduction

The most compact method for storing elevation data in a terrain database is an array of elevation posts. However, this format requires that the source data be provided as a regular grid, and that the grid spacing and origin correspond to the grid spacing and origin of the target database. If the grids cannot be aligned in this manner, a less compact format must be used. Since the increase in memory may be considerable, being able to adjust the parameters of the target database so that its grid matches a grid in the source data may save a large amount of memory.

The parameters of the Compact Terrain Database (CTDB) can be adjusted so that its grid matches the source data. When created from Lockheed-Martin S1000

format data, the CTDB grid is adjusted to match the implicit grid of the S1000 data source. (Although a polygonal representation, the vertices of the S1000 polygons lie on a grid.) The S1000 Application Program Interface (API) also provides an easy means for obtaining this data.

With the development of the Synthetic Environment Data Representation Interchange Specification (SEDRIS), it is possible to create a CTDB from an arbitrary source of terrain data. Unfortunately, the SEDRIS API does not have a mechanism for querying the source data about whether the data is arranged in a grid. In order to find out whether there is an implicit grid, the user must check the data explicitly.

This paper presents a method for checking terrain data to determine the presence of an implicit grid. The method

uses a Fourier Transform to find implicit periodicities in the elevation data. The periodicities are then examined to determine if one represents a suitable grid post spacing. Once the grid post spacing has been determined, the phase angle of the selected period can be used to adjust the CTDB origin to align the CTDB grid with the implicit grid of the source data.

The next section reviews the Fourier Transform and its implementation, the Fast Fourier Transform. The application of the Fourier Transform to the problem of finding the implicit grid in a terrain database is then discussed and the algorithm is described.

Finally, the results of using the algorithm on two databases are presented.

2. The Fourier Transform and Its Implementation

This section discusses those properties of the Fourier Transform that allow it to be used to detect implicit grids in a set of elevation data. The transform itself is discussed first. The Fourier Transform's practical implementation, the Discrete Fourier Transform, is described next. Finally, the computer implementation of the Fourier Transform, the Fast Fourier Transform (FFT), is described.

2.1 The Fourier Transform

The Fourier Transform maps a function in the time or space domain (the signal) into a function in the frequency domain (the spectrum). This mapping is useful for detecting implicit grids because a signal with a well defined period will map to a spectrum with a series of well defined peaks. The frequency of the first peak is the fundamental frequency of the period and is inversely related to the period. Thus a period in a signal can easily be detected and its length can be calculated from the fundamental frequency.

A grid of elevation values has a regular spacing. If the occurrence of an elevation value at a point is used as the signal, then the grid becomes a two-dimensional array of impulses. This array of impulses can be analyzed along each axis and transformed into spectrums along those axes. The spectrums will have peaks corresponding to the spacing of the elevation values.

The Fourier Transform also can provide information about the starting point or origin of the grid. The Fourier

Transform operates on the domain of complex numbers, so the spectrum that is calculated contains both real and imaginary components for each frequency. The relative magnitudes of the real and imaginary components are determined by the phase shift of the grid points relative to the starting point for each period. Thus the offset of the grid from the origin of the database can be computed. The offset (o) is proportional to the arctangent of the imaginary (I_m) divided by the real (R_e) component of the value of the spectrum, and to the length of the period (p).

$$o = p \tan$$

The signal to which the Fourier Transform is applied ideally extends to infinity in both directions. This signal should also be defined for all real numbers. The need for an infinite and continuous signal presents problems in implementation. These problems are addressed by the Discrete Fourier Transform, which is described in the next section.

2.2 The Discrete Fourier Transform (DFT)

The Discrete Fourier Transform differs from the classic Fourier Transform in two ways. The discrete transform samples the signal at a discrete, regular interval (l). The discrete transform is also applied to a signal over a finite interval (L).

The practical result of sampling the signal is to limit the highest frequency that can be measured. From the sampling theorem, this frequency is $1/2l$. Higher frequencies will be "aliased", and reflected onto lower frequencies. However, for sufficiently small sampling intervals, this effect can be neglected.

Sampling the signal over a finite interval limits the smallest difference in frequency that can be detected. This smallest difference is equal to the inverse of the interval being sampled or $1/L$. This result is important in interpreting the frequency scale of the Fast Fourier Transform.

Further changes to the Discrete Fourier Transform are needed for implementation on a computer. These are described in the next section on the Fast Fourier Transform.

2.3 The Fast Fourier Transform (FFT)

For the computer implementation of the Discrete Fourier Transform, the sample values of the signal are mapped to an array. The distance between successive elements of the array is the sampling interval (I).

The Fast Fourier Transform returns an array of the same size as the array of signal values, but in the frequency domain. The difference in frequency between two elements of this array corresponds to the smallest frequency difference of the discrete spectrum (I/L). Thus a peak at location n in the returned array corresponds to a frequency of n/L , and to a period or grid post spacing of L/n .

3. Finding the Implicit Grid in Elevation Data

In order to find grid spacings from elevation data, the elevation readings themselves are not important. What is important is the location of each reading.

The readings are counted along the X- and Y-axes. It is possible to construct a two-dimensional FFT, but the amount of memory required would be excessive. Instead, two separate FFT's are calculated, to find the grid post spacings along both the X- and the Y-axis. For each axis, an array is set up, corresponding to one-meter intervals. For each interval, the value of the element in the array is set to the number of elevation readings having an X (or Y) value within this interval.

Some, perhaps a majority, of the elevation readings may not be part of a grid. These readings will produce non-zero values for frequencies that are not a multiple of the grid spacing. The magnitude of these "noise" values will be smaller than the number of the readings would indicate. This is because random elevation locations will tend to cancel each other when processed by the FFT. Thus, a grid pattern will be more apparent in the frequency domain than in the spatial domain.

To find the optimum grid spacing, the algorithm looks for the peak frequency in the range of values that correspond to allowable grid spacing. For the CTDB, a grid may have a spacing between 80 and 150 meters. Peaks outside this range can be ignored. For the CTDB, the grid post spacing must also be the same along both the X- and Y-axes. The peaks along each axis must be compared to verify the grid post spacing is the same.

Fully aligning the CTDB grid with the implied grid in the source data also requires that the offset of the source grid with the CTDB origin be calculated. Once the grid spacing has been determined, this offset can be calculated for each axis using the offset equation given previously. Once the offsets have been calculated, the origin of the CTDB can be adjusted so that the CTDB grid is aligned with the grid in the source data.

4. Results

The algorithm was run on two databases, both using the S1000 format. The Bellevue database is a small (10,000 x 10,000 meters) database. It mainly contains polygons whose vertices lie on a regular grid suitable for conversion to an elevation grid in the CTDB. The Moba database is larger (24,000 x 24,000 meters) and contains polygons that are less regular. The X-axis spectrums for each of these databases are shown in Figures 1 and 2. The horizontal axis in each figure shows the spatial frequency (elevation readings per meter). Only a subset of spatial frequencies is plotted since the higher frequencies would provide too dense a grid. The vertical axis plots the magnitude of the frequency. The implementation used returns the total rather than the average of the signal. (The value at 0 is the total count of elevation readings rather than the average per sample period.)

The first figure shows the Bellevue database, which is gridded. The peak occurs at a grid spacing of 10,000/80 or 125 meters. This is the actual grid spacing of the Bellevue database. The magnitude of the peak is about 75% of the total number of elevation readings, as indicated by the value for zero frequency. The remaining readings are composed of two different classes. The first class is composed of elevation readings that are not part of the grid. Bellevue has a small tinned section in its northwest quadrant. The second class contains locations that were returned by the SEDRIS search but are not elevation readings, such as road or canopy polygons. The search could be modified to exclude these locations, but only at the cost of making the search algorithm more complex. Excluding these locations is not really necessary since their random locations prevent them from creating a false peak.

The second figure shows the Moba database. This database is primarily tinned. As a result, the peaks are much smaller in relation to the value at zero frequency than was the case with Bellevue. There are also three

separate sets of grid spacings. The most prominent occurs at a spacing of $24,000/192 = 125$ meters. Unfortunately, so few of the elevation readings are part of this grid (less than 30%) that it is unlikely that any patch in the database would only have points from this grid. The other two sets of grid points have periods of

$24,000/48 = 500$ meters and $24,000/800 = 30$ meters. The first of these appears to have occurred as a result of clipping polygons to the patch boundary of 500 meters. The second is the result of a section of polygons that was generated from a 30-meter grid.

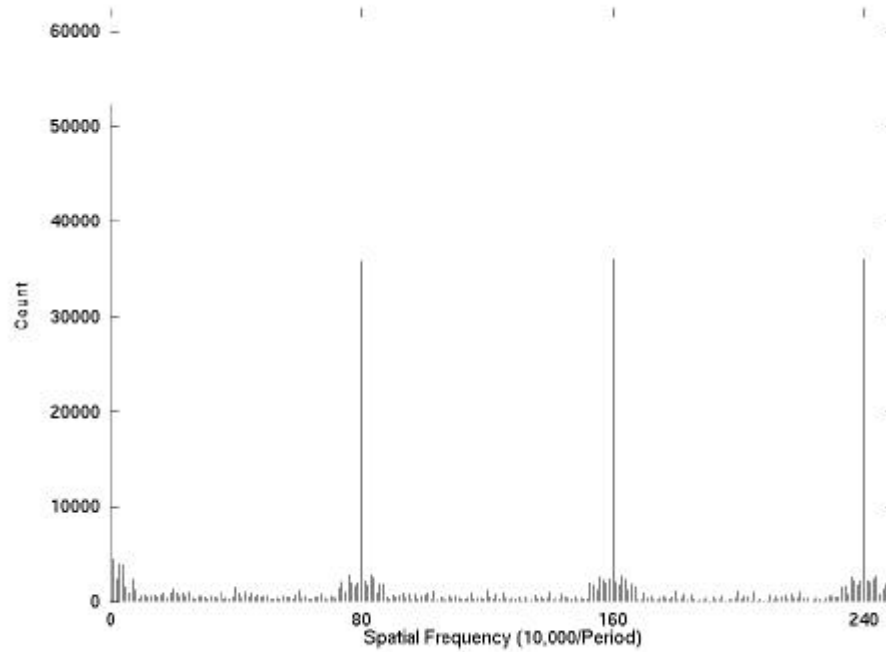


Figure 1: X-Axis FFT of Bellevue Database

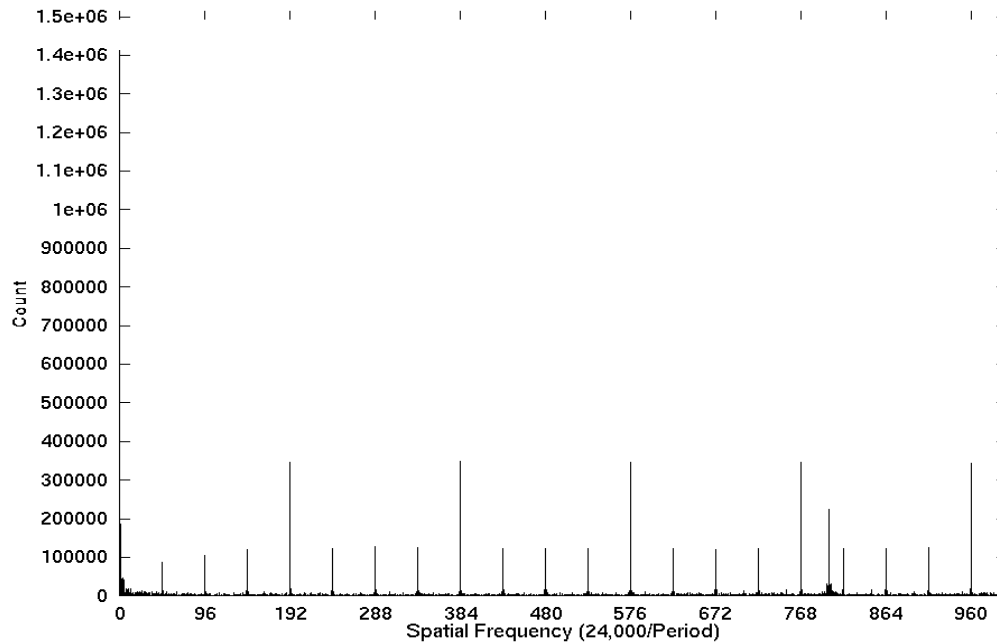


Figure 2: X-Axis FFT of Moba Database

1.

5. Conclusion

This paper has demonstrated the feasibility of detecting an implicit grid in a set of elevation readings. The ability to detect implicit grids allows the creation of optimized CTDB databases from source data, such as SEDRIS, that do not store grid parameters explicitly. This ability to derive parameters from an analysis of the source database should become more important in the future as terrain data is received and processed from more varied sources.

6. Acknowledgment

The Fast Fourier Transform software used in this work is the FFTW package, version 1.2, written by Matteo Frigo and Steven Johnson. For further information about this software contact the authors at fftw@theory.lcs.mit.edu.

7. References

- [1] E. O. Brigham, "The Fast Fourier Transform", Prentice-Hall, 1974

Author Biographies

DR. VICTOR SKOWRONSKI is a Senior Member of the Technical Staff at TASC, where he investigates terrain representation for CGF systems. Prior to joining TASC in 1996, he did research in solid modeling at Rensselaer Polytechnic Institute, where he also earned a PhD in Computer Engineering. Victor earned a M.E. and a B.E. in Electrical Engineering from Stevens Institute of Technology. He is a licensed Professional Engineer in New York and Massachusetts.

THOMAS STANZIONE is the manager of the Computer Generated Forces Section at TASC. He is the Program Manager for the DARPA Framework of Reusable Objects in the Synthetic environment (FROST) project. His interests include data representation for terrain reasoning and terrain database generation for simulation applications. Mr. Stanzone

has a Masters of Science Degree in Photographic Science from the Rochester Institute of Technology.