The Link Foundation Fellowship In Advanced Simulation and Training

A Wavelet-Based Approach to Dynamic Environments

by Glenn A. Martin

Department of Computer Science
University of Central Florida

September 1998

The Link Foundation Fellowship in Advanced Simulation and Training

A grant is awarded to foster advanced level study in simulation and training research: to enhance and expand the theoretical and practical knowledge of how to train the operators and users of complex systems and how to simulate the real-world environments in which they function; and to disseminate the results of that research through lectures, seminars and publications. This publication is the final report of the fellow's research.

LINK AWARDEES

1991-92	Joe Dumas / University of Central Florida
1992-93	Julia Carrington / University of Central Florida
1992-93	Ginger Watson-Papelis / University of Iowa
1993-94	Michael Bajura / North Carolina University
1993-94	Eric Foxlin / Massachusetts Institute of Technology
1993-94	Gary R. Geroge / Binghamton University
1993-94	Matthew P. Reed / University of Michigan
1993-94	Benjamin A. Watson / Georgia Institute of Technology
1993-94	Ronald J. Guckenberger / University of Central Florida
1993-94	Henry A. Nelson / University of Central Florida
1994-95	Kurt M. Joseph / Kansas State University
1994-95	Mark R. Mine / University of North Carolina
1994-95	James R. Otto / University of Kentucky
1994-95	Jim Chen / University of Central Florida
1995-96	Ellen Bass / Georgia Institute of Technology
1995-96	Jonathan Cohen / University of North Carolina
1995-96	Doreen Comerford / Kansas State University
1995-96	Curtis Lisle / University of Central Florida
1995-96	Hans Weber / University of North Carolina
1995-96	Donna Wilt / Florida Institute of Technology
1996-97	Robert A. Glaser / University of Massachusetts
1996-97	Leslie L. Heimenz / Ohio State University
1996-97	Mark C. Kilby / University of Central Florida
1996-97	William R. Mark / University of North Carolina
1996-97	Ellen Scher Zagier / University of North Carolina
1996-97	Douglas A. Peterson / University of South Dakota
1997-98	Robert Franceschini / University of Central Florida
1997-98	Barry Kelly / Binghamton University
1997-98	Darren Law / University of Central Florida

1997-98	Mark Livingston / University of North Carolina
1997-98	Glenn Martin / University of Central Florida
1997-98	Carol Chensey / University of Florida

The Institute for Simulation and Training and the University of Central Florida
Administer this program on behalf of the Link Foundation.
Dr. A. Louis Medin, Executive Director
Institute for Simulation and Training
3280 Progress Drive
Orlando, Florida 32826-0544

The Link Foundation Fellowship In Advanced Simulation and Training

A Wavelet-Based Approach to Dynamic Environments

by Glenn A. Martin

Department of Computer Science
University of Central Florida

September 1998

A Wavelet-Based Approach to Dynamic Environments

Glenn A. Martin
University of Central Florida

A Report to the Link Foundation Fellowship in Advanced Simulation

Abstract

This paper demonstrates a new approach for terrain representation: multiresolution surfaces based on wavelets and the process of multiresolution analysis. Because they are based on mathematical surfaces, multiresolution surfaces gain the benefits of surfaces (more level of detail control, increased number of possible resolutions, easier updates of a dynamic terrain) that lead to fewer correlation errors both in non-networked and networked simulations. Furthermore, the power of multiple resolutions (at little added cost) reduces network transmission costs and provides a framework for the use of dynamic terrain in shared virtual worlds.

1.0 Introduction

For years, polygonal and gridded data have served as the industry standards for representing terrain data. This commitment has remained true even as the demand for added realism has driven the development of dynamic environments. Recently, this situation has begun to change as work in surface-based terrain representations has provided evidence that increased realism can be achieved at little added cost. The research reported here focuses on extending surface-based representations into multiple resolutions through the use of wavelets and multiresolution analysis. The implications of this approach on networked simulations and dynamically alterable terrain are also studied.

2.0 Terrain Representation

2.1 Basic Issues

Terrain representation refers to the concept of creating a spatial database [3] to represent the ground used in a simulator. In simulation literature, terrain often includes objects on the ground such as roads, trees, treelines, tree canopies, buildings, etc. However, I use the word "terrain" to mean simply the ground itself and the word "culture" to refer to the objects on the ground (other than actual vehicles, which are known as "entities").

Many challenging problems arise when terrain needs to be represented accurately and efficiently. These include problems related to level of detail, resolution, correlation and the management of dynamic changes. This situation is further complicated in a networked environment by issues of distribution and more demanding correlation problems.

Level of Detail (LOD) refers to the concept of using less complex, less accurate models for representing terrain and/or objects at distances farther from the viewpoint. This reduces the cost of producing the image in real-time and allows the image generator to work at higher performance levels.

Resolution problems occur when making the tradeoff between performance and accuracy. High detail databases define the actual terrain most accurately, but at such a high cost that image generators can easily lose their real-time performance. Lower detail databases, while less costly, can adversely affect realism and, thus, compromise the quality of a simulated experience. If such a simulation is being used for training, the reliability of the exercise is open to question. When faced with this compromise, terrain modelers must balance detail and performance. Using level of detail can help, but ultimately some detail in the terrain is lost.

Because of the decisions involved in modeling terrain as well as numerical errors inherent in floating-point values, terrain databases that are supposed to represent the same terrain often end up looking quite different. A long conversion pipeline can potentially increase the number of errors. In addition, problems arising from networking heterogeneous simulators also cause errors in representation. This has motivated researchers to study correlation metrics to develop a way of quantifying differences between terrain databases [2, 11]. This field of study is very important, yet is still in its infancy and offers some very challenging problems centering on the basic question of how to compare two terrain databases that were modeled differently.

The problems of performing correlation tests become even harder when you consider dynamic terrain. Moshell *et al.* define dynamic terrain as "the capability, within a real-time graphical simulation, of rearranging the terrain surface" [8]. In other words, dynamic terrain refers to changing the terrain during a single simulation experience. In simulators with static databases, interactions with the ground do not actually change the terrain database. For example, military trainers typically represent terrain changes from munitions by overlaying a texture of a crater, achieving a desired visual effect. However, the dynamic models of the vehicles still perceive the terrain without craters. The lack of true dynamic terrain means that these simulated worlds can not be used in many situations, e.g., for training civil engineers. These limitations have led to an increasing demand to add dynamic terrain to simulators in order to give better realism and gain better training. The complexities of the above mentioned issues (LOD, resolution and correlation) will be further complicated, however.

Finally, all four previous issues (level of detail, resolution, correlation and dynamic terrain) are complicated when networking is added. The primary method for networking heterogeneous simulators in an interoperable manner is known as Distributed Interactive

Simulation (DIS). At the core of DIS is a collection of Protocol Data Units (PDUs) which define messages that can be passed between simulators. For example, the Entity State PDU is used to broadcast data about an entity's identification, type, location, orientation, linear and angular velocity, linear acceleration, etc. The velocity and acceleration terms are used in a process called "dead reckoning" which allows simulators on the network to approximate the location of the entity as it moves. With this scheme, network traffic can be reduced since the sender of the true entity data needs only to send another PDU when the approximation has lost too much accuracy.

In the early 1990s, there were a series of demonstrations at the Interservice/Industry Training Systems and Education Conferences (I/ITSEC) that tested DIS running on many different image generators from many vendors. As would be expected, many correlation problems occurred. These include floating vehicles (e.g., tanks flying above the ground because the elevations between terrain databases do not correspond), and situations where one vehicle cannot see another because a tree is blocking the view, yet the latter vehicle can see the former. Sometimes programmers try to "fix" the first problem by terrain clamping which forces the vehicles to ride on the ground. However, this almost certainly causes more instances of the second problem as well as other anomalies.

Now consider the potential problems when networking dynamic terrain. Since the updates cannot reflect how each terrain database is modeled, they must be generic enough so that all users can apply them to their databases. Furthermore, there are questions that arise when applying the update to a terrain database. How will the change affect correlation between databases? Is it possible to estimate terrain changes between updates (a terrain variant of dead reckoning) so fewer updates are needed? Both of these questions remain to be answered. Both are important, but the second question emphasizes the networking issue. If civil engineers are moving large volumes of soil, for instance while clearing roadways, they are clearly going to be making significant changes to the terrain and these will occur often. Even with a method for dead reckoning, the cost of sending these changes through a network may be unacceptable.

2.2 Methods for Representing Terrain

Representations for terrain fall into three categories: polygon-based, grid-based, and surface-based. Polygon-based and grid-based methods are classic techniques that are popular in today's image generators. Polygons and grids are simple representations but fail to provide sufficient power for new dynamic environment research. Surface-based methods, relatively new within the real-time simulation domain, try to rectify the limitations of polygons and grids. This section will describe how each representation method affects the issues mentioned in Section 2.1, and then will discuss previous work in the evolving field of surface-based representations.

2.2.1 Polygon-Based Representation

Polygons are the most popular method for representing terrain within an image generator. In fact, most image generators contain special-purpose hardware in order to process polygons quickly. However, polygons also cause many problems with terrain representation issues.

First, level of detail for polygons must be modeled in advance and is generally only used on culture objects and vehicle entities. This is achieved by modeling some number of versions of the object so that the image generator can switch between them either by snapping to the next level, or by morphing to the next level (which is a relatively new technique in LOD control). Terrain is usually not modeled through LODs because a single polygon can represent a large area of terrain, which reduces the number of polygons required for modeling the terrain in the first place. This consideration has been the primary reason that image generators have historically used static terrain databases.

Second, polygon resolution affects the quality of terrain representation. As polygons represent larger and larger areas, there is a potential of creating harsh boundaries between the polygons. For example, two polygons might create a hill that is pointed at the top rather than smoothly transitioning from one slope to the other. The rapid changes at the boundaries cause major problems in vehicle dynamics, interoperability and sometimes correlation tests.

Third, due to the number of possibilities of modeling a terrain using polygons, correlation of polygonal databases is nearly impossible. In addition, Kilby *et al.* [2] concluded "any correlation metrics developed will be difficult to apply since no standard interchange format exists between heterogeneous simulators other than the original source data (i.e. SIF/HDI)." In order to perform their correlation experiments, they were forced to sample elevation values from each of the polygonal databases under comparison.

Fourth, implementing dynamic terrain on polygons can exact a heavy price. As a polygon becomes too large to represent the changes to the terrain, it is simply broken into smaller polygons. Each smaller polygon is then modified as needed to represent the change wanted. Although this is possible, the increasing number of polygons will lower the performance of an image generator to the point where it is unusable. Li *et al.* [4] developed a relaxation algorithm for reducing the number of polygons by omitting elevation points which do not add important geometrical surface information (relatively co-planar points as defined by a programmable tolerance value). However, important features such as ridgelines and vehicle tracks are given a tolerance of 0.0 so that they are not modified. Hence, "even with the terrain relaxation approach used, exercises with many entities changing terrain will ultimately create so many polygons that CIGs are unable to render views, and non-visual entities become computationally overburdened" [4].

Finally, networking with static polygonal databases does not cause any problems in itself, but entities will suffer from any correlation errors that exist. However, networking dynamic terrain changes with a dynamic database may be difficult depending on the method. Obviously, the changes to one database may not be valid for another that employs a different polygonization. The size of an update containing individual polygons may induce a high cost as well.

2.2.2 Grid-Based Representations

Grid-based methods only store attribute data at certain points on a gridded array based upon a selected resolution (often in meters). For example, terrain might only be sampled at 10-meter spacing meaning that values are queried every 10 meters and are stored into the gridded array. Obviously, a severe amount of detail can be lost if a poor choice of resolution is made, but the amount of data increases significantly as one lowers the resolution spacing. Therefore, a trade-off must be made between accuracy and performance relative to the issues of terrain representation.

Level of detail with grids can only be achieved by resampling the space using some method of interpolation between the old grid values to get a new, more refined grid. Obviously, the accuracy of such a process is limited by the resolution of the original grid posts. Resolution, in turn, can cause a loss of data (as mentioned earlier) which causes correlation errors. Furthermore, since grid posts eventually must be converted into either polygons or surfaces (see Section 2.2.3) for drawing, they are most often used as an interchange format. However, Lisle et al. [5] did use gridded attribute data as a generic method to represent dynamic terrain updates across the network (thereby retaining compatibility with other image generators).

2.2.3 Surface-Based Representations

Mathematical surfaces provide another representation for terrain. They allow for higher fidelity data that avoid the harsh transitions that the polygon-based representations create. In general, surface-based representations provide more power than polygons with little added cost. This includes cost in performance as well as cost in terrain representation issues.

2.2.3.1 Effects of Surfaces on Terrain Representation

Using surfaces, Level of Detail is performed by querying at different resolution. Because the terrain is represented using a smooth surface, few of the problems resulting from the use of polygons arise in this instance. Physically based vehicle dynamics, for example, can now query the smooth transitioning surface at any resolution needed. However, the "snapping" involved in switching between levels of detail (resolution) can be visually disconcerting.

Correlation between a surface-based database and other terrain databases is still difficult because of the differences between representations. However, the increased accuracy of a surface-based method should reduce the number of correlation errors when compared to the original source data (assuming we compute the surface accurately). In fact, the use of a surface allows enhanced accuracy without a prohibitive additional cost. As mentioned earlier, making a polygonal database more accurate entails adding polygons, which adds to the load of an image generator.

Surface-based dynamic terrain has many advantages over polygon-based methods. Terrain changes are merged in with the database by simply modifying control points used to define the surface. Because no polygons are created, there is no additional load. For a large area of update, many control points will need to be altered, however.

As with polygons, networking does not create any correlation errors but the entities will be affected by any that do exist. Dynamic terrain changes could possibly be represented by a surface's control points, but this technique could potentially require a receiver to evaluate the surface so that it could update its own terrain database. However, the use of control points, which generally number fewer than polygons or grid posts, might be an alternative for decreasing packet size.

2.2.3.2 Previous Work in Surface-Based Representation

Scarlatos [9] studied surfaces for terrain representation and how different triangulation methods could preserve important features (such as pits, valleys, peaks, ridges, and saddle points). Specifically, she studied three problems:

1. The need for spatial data representations to support multiple scales and precisions without losing critical information
2. A requirement for the database to use filtering techniques to improve performance
3. Merging techniques that allow different data representations to exist separately but work together (allowing each representation to be used to its best advantage)

Obviously, techniques that satisfy these three constraints can provide a great deal of power while lowering cost.

In their dynamic terrain work, Lisle et al. developed a software abstraction called the Dynamic Terrain Database (DTDB) [5]. It allows an arbitrary number of attributes (each represented by a mathematical surface) with a query mechanism powerful enough to provide any number of data points at any resolution. The query mechanism was intentionally decoupled from the underlying representation. How the data was stored within the database was completely hidden from the client application program. The decoupling of the user from the data representation will prove even more useful as new types of data are incorporated in the system as a whole [5].

3.0 Multiresolution Analysis

This section provides a brief introduction to wavelets and multiresolution analysis. The concept of wavelets was first introduced in approximation theory, signal processing, and physics [7]. More recently they have been applied to problems in image processing, numerical analysis, and computer graphics. Section 3.1 describes the theory of multiresolution analysis and Section 3.2 shows how filter matrices are computed.

3.1 The Theory Behind Multiresolution Analysis

Since the research reported in this paper has been based on the work of Finkelstein and Salesin [1] (which in turn is based on the work of Lounsbery *et al.* [6]), we have adopted their approach rather than the more classical one developed by Mallat [7]. To begin, consider a signal C^n that is simply a column vector of m pieces of data $[c_1,...,c_m]^T$. Now, suppose a lower-resolution version of C^n is required (call it C^{n-1}) that only has m' ($m' < m$) data items. To compute this, we simply use a form of filtering and downsampling:

$$C^{n-1} = A^n C^n \qquad \text{(EQ 1)}$$

where A^n is a filter matrix that is multiplied with C^n.

Clearly, some information is lost in the downsampling. However, if A^n is chosen in a certain way, we can capture the lost detail as another column vector (call it D^{n-1}). Another filter matrix B^n (which is related to A^n) with dimensions of $(m-m') \times m$ is used to find the lost detail:

$$D^{n-1} = B^n C^n \qquad \text{(EQ 2)}$$

The filter matrices A^n and B^n together are called analysis filters, and the process of splitting C^n into the two parts A^n and B^n is called decomposition.

We can also take this process one step further. If A^n and B^n are chosen in a certain manner, then the original C^n can be recomputed through another pair of matrices P^n and Q^n:

$$C^n = P^n C^{n-1} + Q^n D^{n-1} \qquad \text{(EQ 3)}$$

P^n and Q^n are referred to as synthesis filters and the process of recomputing C^n is known as reconstruction.

The process of decomposition can be applied repetitively to produce a hierarchy of lower-resolution signal and detail coefficients. This process is known as a filter bank and the signal composed of C^0 and all the detail coefficients is known as the wavelet transform of the original signal.

3.2 The Construction of Filter Matrices

Depending on the signal under consideration, the decomposition and synthesis filters are chosen appropriately. There are currently two methods for computing these filters. The first was developed by Mallat [7] and creates so-called "first generation" wavelets. These wavelets are formed by translations and dilations of a "mother" wavelet which makes them valuable to analyze data that occur on lines (1-D) or planes (2-D). Furthermore, the Fourier Transform is used to construct these wavelets (since translation and dilation become simple algebraic operations in the Fourier domain).

The second is called the Lifting Scheme and was developed for computer graphics by Schroder and Sweldens [10]. The idea was to find wavelets to use in other situations (such as triangulations and surfaces) where translations and dilations cannot be used. Lifting takes a simple wavelet transform and slowly improves properties such as smoothness and vanishing moments. As a comparison, these "second generation" wavelets do not have coefficients that are the same throughout but rather change locally to reflect the surface.

4.0 Multiresolution Curves

Finkelstein and Salesin [1] developed some very impressive applications of multiresolution analysis to a common case of cubic B-spline curves defined on a knot sequence that is uniformly spaced everywhere except at its endpoints (where its knots have a multiplicity of 4). These B-splines are known as endpoint-interpolating cubic B-splines. Storing the curve's control points into the signal and applying the method discussed earlier performs the multiresolution analysis. Applications include smoothing, editing and scan conversion of the curves.

4.1 Sample Applications

Editing curves can be achieved with remarkable versatility. This includes both editing the sweep (the general shape) of a curve as well as editing the character (the detail) of a curve. As the level decreases, more and more of the curve is affected until finally at lowest level the entire curve is affected.

Editing the character of a curve is nearly trivial. Given the control points of a curve, simply run the decomposition process to a certain level, and replace the detail coefficients with a new set. The final step is to simply perform the reconstruction process.

One final application of multiresolution curves is in reducing scan conversion costs. If we can smooth a curve so that it has fewer control points while keeping within a particular error tolerance, costs relating to scan conversion will be reduced. Finkelstein and Salesin applied this curve compression to the domain of printers using the Postscript language. However, we can also apply these ideas to reducing network costs of dynamic terrain updates as well.

4.2 Applications to Terrain Representation

In the domain of terrain representation, we can apply multiresolution curves as a method for storing culture data such as linear features. This gives us the power of continuous level of detail rather than the few levels available in many applications today. A road far off in the distance could be drawn with much fewer points (i.e. at a lower level), but can increase its number of points as the viewer approached it until the original curve for the road is reached.

Editing at different levels with different number of control points could aid the update process used in dynamic terrain. We can actually change larger and larger pieces of the overall terrain database as we use lower levels in the decomposition process while keeping the same number of update points. This applies both to the sweep of the curve as well as its character.

As alluded to earlier, the ideas behind curve compression (and possibly curve smoothing) can be used to reduce networking costs. If we can approximate a curve with fewer control points, then the packet we send out will be smaller. Furthermore, if more detail is needed at some point later in the simulation exercise, we can send the detail coefficients with the assurance that the total cost will be no more than if the highest precision data were sent all at the beginning. This also allows receivers to perform the decomposition and reconstruction methods as needed to alter the resolution of the representation. As the desire to network simulators increase, networking issues are being raised more frequently so the flexibility of this system will be very useful.

5.0 Applying Multiresolution Surfaces to Terrain Modeling

This section describes and analyzes, both from a correlation and a complexity viewpoint, the extensions that have been made to multiresolution curves.

5.1 Multiresolution Surfaces

The first step in extending the multiresolution analysis process to surfaces is to recognize that C^n now becomes a more complex matrix since we must represent two dimensions of the surface. However, we cannot simply add the second dimension and continue to use the same decomposition and reconstruction methods. We must consider the surface's control points as a two-dimensional image and adjust the multiresolution analysis process.

We begin by letting C^n be a matrix of surface control points. Furthermore, we need to alter the methods used in decomposition and reconstruction so that we do not undersample the signal. Otherwise, we will lose information and will not compute accurate signals. Basically, this means that every combination of the filters must be used in the two dimensions of the signal.

We now redefine the decomposition process as follows:

$$C^{n-1} = A^n C^n (A^n)^T \qquad \text{(EQ 4)}$$

$$D_1^{n-1} = A^n C^n (B^n)^T \tag{EQ 5}$$

$$D_2^{n-1} = B^n C^n (A^n)^T \tag{EQ 6}$$

$$D_3^{n-1} = B^n C^n (B^n)^T \tag{EQ 7}$$

The reconstruction equation becomes:

$$C = P^n C^{n-1} (P^n)^T + P^n D_1^{n-1} (Q^n)^T + Q^n D_2^{n-1} (P^n)^T + Q^n D_3^{n-1} (Q^n)^T \tag{EQ 8}$$

We now have the equations necessary to perform multiresolution analysis on the control points of surfaces. In this case, the matrices P^n, Q^n, A^n and B^n are the same matrices used by Finkelstein and Salesin because endpoint-interpolating cubic B-splines can model terrain quite effectively.

5.2 Correlation Analysis

In order to compare the accuracy of the multiresolution surfaces, a simple correlation experiment was developed. Basically, surfaces based on varying numbers of control points were defined and then queried using the same number of data points (thereby creating several data point to control point ratios). This was performed for both Lisle's Dynamic Terrain Database as well as the multiresolution surfaces using several tests of two different types of terrain (one with smooth terrain, and one with a more rough terrain). An average-of-differences error metric was then computed over the data points in each representation.

Figure 1 shows a graph of the error metric for the DTDB and multiresolution surfaces using a smooth terrain. As shown in the graph, the multiresolution surface seems to

Figure 1:
Correlation Metric of DTDB vs. MRS
for Smooth Terrain

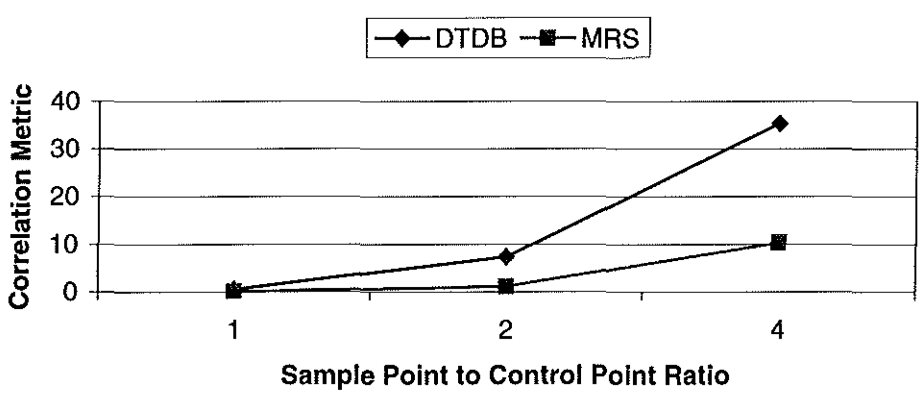
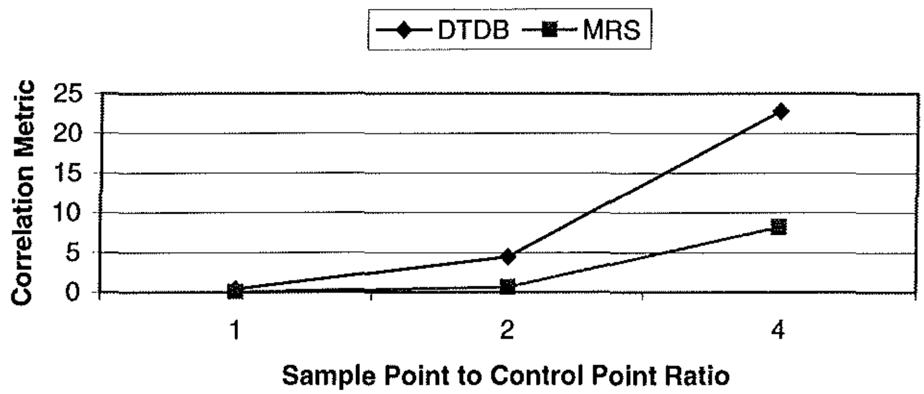


Figure 2:
Correlation Metric of DTDB vs. MRS
for Rough Terrain



perform well relative to the DTDB. Because multiresolution surfaces can favor highly changing area, a reduced control point surface can better represent the overall area.

Figure 2 shows a graph of the error metric using a rough terrain. In this case, the multiresolution surface also performs better than the DTDB. Again, this is because the multiresolution surface computes a least-square-error surface that best fits the original surface, but the DTDB blindly resamples the surface at the precise spacing needed for the quantity of points requested.

5.3 Progressive Transmission

Progressive transmission is a term applied to a technique of sending a rough copy of an object followed by more detailed versions. For example, JPEG images transmit a very blurry (but low data) version first and then fill in details to make the final image. We can use the same technique for altering multiresolution surfaces. If a change is made, an update in the form of the lowest resolution level of C^n can be sent followed by the various resolution levels of D^n. This allows a quick although low fidelity update of the terrain database.

Another idea is possible. Rather than sending all the resolution levels of D^n, network bandwidth could be saved if only the levels of D^n that made the surface look "good enough" were sent. In this technique, less data are actually sent than if the original (highest resolution) C^n was sent. Multiresolution surfaces and progressive transmission has allowed us to represent the terrain database accurately as well as save bandwidth.

6.0 Conclusions

This paper demonstrates a new approach for terrain representation and transmission: multiresolution surfaces based on wavelets and the process of multiresolution analysis. Because they are still based on mathematical surfaces, multiresolution surfaces gain the benefits of surfaces (more level of detail control, increased number of possible resolutions, easier updates of a dynamic terrain) that lead to fewer correlation errors both in non-networked database comparisons as well as during a networked simulation exercise. Furthermore, the power of multiple resolutions (at little added cost) provides a very important framework for more easily performing dynamic terrain changes to the database and result in lower network transmission costs.

7.0 References

[1] Finkelstein, Adam and David H. Salesin. "Multiresolution Curves." Proceedings of SIGGRAPH 1994.

[2] Kilby, Mark, Larry Crochet, Chen Jinxiong, Glenn Martin, and Tinahui Fu. "Correlation Studies: Final Report." IST Technical Report IST-TR-93-17, June 15, 1993.

[3] Laurini, Robert and Derek Thompson. *Fundamentals of Spatial Information Systems*. Academic Press, Inc. San Diego, CA, 1992.

[4] Li, Xin, Dale D. Miller, Mark Illing, Mark Kenworthy and Mark Heinen. "Dynamic Terrain Database Design for Real Time Image Generation." Proceedings of the 16th I/ITSEC, 1994.

[5] Lisle, Curtis, Marty Altman, Mark Kilby and Michelle Sartor. "Architectures for Dynamic Terrain and Dynamic Environments in Distributed Interactive Simulation." 10th DIS Workshop, Orlando, Florida, March 1994.

[6] Lounsbery, M., T. DeRose, and J. Warren. "Multiresolution Surfaces of Arbitrary Topological Type." Technical Report 93-10-05B, University of Washington, Department of Computer Science and Engineering, January 1994.

[7] Mallat, S. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation." IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 11, number 7, pages 674-693, July 1989.

[8] Moshell, J. Michael, Brian Blau, Xin Li, and Curtis Lisle. "Dynamic Terrain." Simulation, January 1994.

[9] Scarlatos, Lori. "Spatial Data Representations for Rapid Visualization and Analysis." Ph.D. Dissertation, Department of Computer Science, State University of New York at Stony Brook, August 1993.

[10] Schroder, P. and T. Sweldens. "Spherical Wavelets: Efficiently Representing Functions on the Sphere." Proceedings of SIGGRAPH 1995.

[11] Schiavone, Guy, Russell Nelson, and Brian Goldiez. "Statistical Certification of Terrain Databases." Proceedings of the 16th I/ITSEC, 1994.