An Interactive Visualization Method for Integrating Digital Elevation Models and Geographic Information Systems Vector Layers

J. Stuart, J. Jaquish,
S. Bassett, F. Harris, W. Sherman

Department of Computer Science and Engineering
University of Nevada, Reno, Reno, NV, 89557

Desert Research Institute, Reno, NV, 89510

Abstract. Currently there are software packages that allow a user to analyze GIS data. The problem is that the software is limited in the interaction provided. This paper presents a method for merging digital elevation models (DEM) and GIS vector layers, and also provided is an interactive visualization tool for analyzing the output.

Keywords: DEM, GIS layers, Visualization

1 Introduction

Training is an essential part in military effectiveness [1, 2]. Generally, the more realistic training is to the soldier, the better prepared that soldier will be when an actual scenario arises [3, 4]. Creating a realistic training environment requires significant preparation in planning, coordination, time, money, etc. Resources such as equipment, fuel, vehicle parts, and maintenance time are used extensively, making it difficult to provide consistent, thorough training for every one.

The advent of computer technology has significantly impacted the way the military trains its personnel. Specifically, the field of computer graphics and visualization has become crucial for the creation of realistic training environments. With modern technology, the military can train its personnel in computer simulated situations, thus alleviating much of the need for travel and equipment transportation. Also, personnel can work their way through a simulation several times, honing their skills while using minimal resources.

Visualization techniques help to establish realism, but real-world geographic data is also necessary to achieve a fully immersive and realistic simulation. The geographic data represents a true-to-life landscape in the simulated environment. The data can be visualized via computer graphics and virtual reality techniques. The end goal is to give personnel the feeling they are at the real-world location, even though they are actually in a simulated environment.

In order to obtain accurate information about a location, specialists in geographic and geological disciplines must first visit and thoroughly analyze the location. This is viewed as a necessary expenditure of time and resources. However, situations occur where the resources and access necessary to collect ample data is simply not available. This lack of data can endanger missions, especially when specific training, such as helicopter flight, requires accurate knowledge and representation of an area.

There is some data that can be gathered and then later analyzed off-site. One such piece of data is a Digital Elevation Model (DEM) [5]. The name has pragmatic significance in that it consists of spatially explicit digital values of height data. Other important data are Geographic Information Systems (GIS) vector layers [6]. GIS vector layers are maps (i.e. represent a location) that store two-dimensional shapes (polygons) with attributes. Attributes can be assigned from a variety of sources including satellite imagery, site visits, etc. DEMs are also referred to as GIS raster layers, but from this point forward, any mention of a GIS layer refers to a GIS vector layer.

DEMs and GIS layers have been combined before, but this paper proposes a method for merging the two so as to be efficiently displayed in a three-dimensional virtual environment. Such an environment would give specialists better interaction with the data being visualized. It is also believed that an environment such as this would reduce the amount of time and other resources necessary to perform a thorough analysis of geographic sites.

The remainder of this paper is structured as follows. Section 2 presents background information on geographic applications, computer graphics, and immersive virtual environments. Section 3 presents our method for combining DEMs and GIS layers and displaying them in a manner conducive to fluid interactivity. We present the results of our methods in Section 4 and state our conclusions in Section 5. Section 6 presents several possibilities for future work on this topic.

2 Background

The fields of geography and computer graphics must be combined for this project to become a practical application. Geography is the study of earth and its features. Computer graphics refers to methods of creating visual imagery with computing techniques. It is incredibly difficult to interpret hundreds of tables of geographic data if it is not visualized (i.e. with computer graphics). Thus the combination of geographic data and computer visualization techniques is unavoidable.

There are several applications in use by geographic specialists that help them consolidate and visualize geographic data. One such program is called ArcGIS^{TM} [7]. ArcGIS is designed for the desktop environment on a typical two-dimensional monitor. It works primarily with GIS layer data files (mentioned in Section 1). It is useful for portraying and altering region data and associated attributes. ArcGIS is not meant to display three-dimensional data, that task is left to another application (typically packaged with ArcGIS) called Arc Scene^{TM}. However, users do not feel immersed into the environment because the data is visualized on a two-dimensional screen.

The feeling of immersion and realism is important because it allows a specialist to notice and analyze more features of a landscape, and to analyze those features with higher accuracy. To achieve complete immersion, we use a computer graphics technique called Virtual Reality (VR).

Effective VR is created in a virtual/immersive environment by means of several projection screens and accompanying projectors. In this environment, users are required to wear a headset or a pair of specialized glasses. Via the combination of the glasses, projection screens, and the projectors, users are given a realistic sense of depth unattainable on modern desktop systems. Several companies such as FakeSpace^{TM} and Silicon Graphics Inc.^{TM} sell full VR projection systems.

Beyond immersion in a VR environment, one must also have the means to interact with and manipulate the data being visualized. For this project, the immersive environment must allow the user to pick regions of a map and see all associated attributes. Attributes assigned to regions in GIS layers depend on the context of the research. The user must also have the ability to add, remove, and update attributes as they see fit, all while inside the VR environment.

This interaction must be done in real-time, as one of the main points of the purpose is to cut-down on a crucial resource, experts' time. For many reasons, allowing real-time interaction can be quite difficult. Geographical experts need to interact with large areas of the world at once, and the size of the area to be visualized is directly proportional to the amount of data stored and processed. The second reason is we are striving for realism and the amount of realism is also directly proportional to the amount of data stored. This all amounts to longer and longer computation time. Optimization methods are a must to visualize this data as fast and realistically as possible. We can now begin discussing our methods for quickly visualizing this data to allow for fluid interaction.

3 Proposed Methods

The primary elements of visualization dealt with by this paper are three dimensional digital elevation maps (DEMs), stored using the BIL File Format [8], and two dimensional Geographical Information Systems (GIS) layers, stored using the ESRI^{TM} Shapefile Format [9]. It is common for both DEMs and GIS layers to be complex in terms of size and structure, thus requiring special methods for visualization.

3.1 Digital Elevation Map Manipulation

Three-dimensional rendering APIs such as OpenGL [10] and Direct3D [11] expect display primitives to be represented as collections of triangles. DEMs are easily represented using triangles, so rendering a DEM can be trivial. Figure 2 shows a DEM and all of the triangles used to approximate it. However, the size of modern DEMs are approaching the point that real-time rendering with modern graphics hardware is not possible. Also, this paper uses DEMs to assign an extra dimension to a GIS layer, not just for visualization.

In order to efficiently deal with DEMs, this paper applies a lossy compression technique to DEMs called Triangle Decimation [12, 13] and Mesh Simplification [14–16]. Triangle Decimation works by examining coincident triangles, and removing unnecessary triangles. Figure 1 shows a set of triangles that can be reduced and the possible effects of triangle decimation. Triangle decimation is called lossy because once it is performed, it is not always possible to regain the full representation of the DEM. However, the effects of the decimation aim to have little-to-no visual side effects. Figure 2 shows a DEM that is a prime candidate for triangle decimation since it has many coincident triangles that are nearly coplanar.

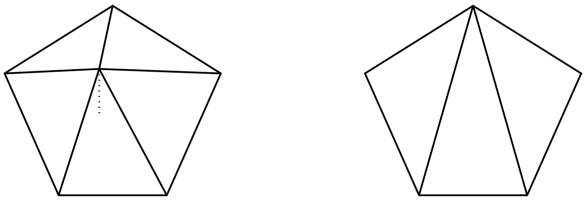

Fig. 1. Candidate triangles and the result of their decimation. The small amount of height lost in the left area is barely noticeable in the right area.

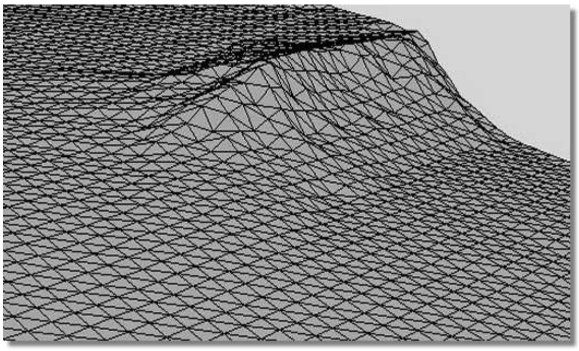

Fig. 2. Approximated Digital Elevation Map

3.2 GIS Layer Manipulation

As mentioned previously, rendering APIs expect primitives to be broken into triangles. The GIS layers found in Shapefiles are represented using two-dimensional polygons. These polygons are clean, meaning they contain no self-intersecting lines. Examples for both clean and non-clean polygons are shown in Figure 3. Since there is no restriction on the structure of the polygons beyond that of "cleanness," one cannot assume anything of the convexity of the polygons in GIS layers. This paper uses several techniques to triangulate the polygons in the GIS layer; ear-clipping [17], monotone decomposition [18], and trapezoidilization [17]. The output of all these techniques is a set of two-dimensional, non-intersecting triangles.

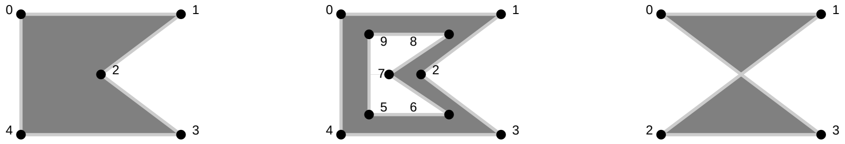

Fig. 3. Two clean (left, middle) polygons and a non-clean (right) polygon.

3.3 Combining DEMs and GIS Layers

In order to combine the DEM and the GIS layer, a six step approach is proposed:

1. Read and store the raw DEM and GIS layer
2. Convert the GIS layer into sets of two-dimensional triangles
3. Convert the DEM into triangles
4. Perform triangle decimation on the DEM
5. Add a third dimension to the GIS layer
6. Add the DEM points to the GIS layer

Step 1 of the approach is trivial. Steps 2, 3, and 4 can be done using the algorithms discussed in 3.1 and 3.2. Step 5 can be accomplished using the $O(t_d * v_g)$ algorithm presented Figure 4, where t_d is the number of triangles in the DEM and v_g is the number of vertices in the GIS layer. Step 6 can be computed using the $O(v_d * t_g)$ algorithm shown in Figure 5, where v_d is the number of vertices in the DEM and t_g is the number of triangles in the GIS layer.

The algorithm relies on the ability to split a triangle into one, two, or three triangles new triangles. The algorithm, which we call **SPLIT**, takes as parameters a triangle and a point. **SPLIT** works by examining the location of the point; if the point lies on a vertex of the triangle, **SPLIT** returns the original triangle. If the point lies on the perimeter of the triangles, but not on a vertex of the triangle, **SPLIT** returns two new triangles with the point being a vertex on both triangles. Finally, if the point resides in the interior of the triangle, **SPLIT** creates three new triangles using the vertices of the triangle and the new point. Examples for the three cases of **SPLIT** can be seen in Figure 6. In the figure, the dark dot represents the second parameter to **SPLIT**.

```
$T_0 \leftarrow$ Triangles in DEM
$V_0 \leftarrow$ Vertices in GIS layer
for all $t_0 \in T_0$
  for all $v_0 \in V_0$
    if $v_0$ has only two dimensions
      if $v_0$ is contained by $t_0$
        $v_1 \leftarrow v_0$ projected onto $t_0$
        $V_0 \leftarrow V_0 - \{v_0\}$
        $V_0 \leftarrow V_0 \cup \{v_1\}$
```

Fig. 4. Algorithm for assigning height values to a GIS layer

```
$V_0 \leftarrow$ Vertices in DEM
$T_0 \leftarrow$ Triangles in GIS layer
for all $v_0 \in V_0$
  for all $t_0 \in T_0$
    if $v_0$ is contained by $t_0$
      $T_1 \leftarrow$ SPLIT $(t_0, v_0)$
      $V_0 \leftarrow V_0 - \{v_0\}$
      $T_0 \leftarrow T_0 - \{t_0\}$
      $T_0 \leftarrow T_0 \cup T_1$
```

Fig. 5. Algorithm for incorporating DEM points with a GIS layer

4 Results

The overall algorithm we presented takes two pieces of data for its input. The first is a DEM, with an example given in Figure 2. The second is a GIS layer, presented in Figure 7. The algorithm combines its input and produces a set of three-dimensional triangles. The algorithm was implemented as a preprocessing stage for our visualization tool.

With our data in hand, we constructed an interactive visualization tool. This tool provides better interaction with GIS data than current packages because of

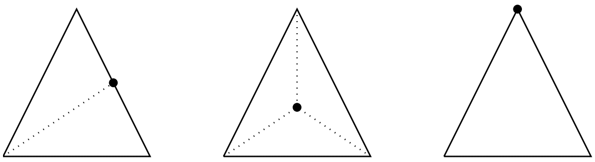

Fig. 6. Cases for **SPLIT**: Splitting a triangle into (from left to right), 2 triangles, 3 triangles, and 1 triangle.

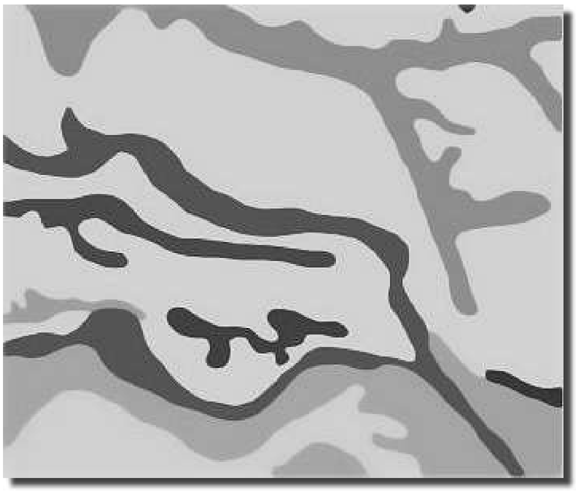

Fig. 7. Two-dimensional GIS Layer

the ability to accentuate specific features of the data, and because unlike other packages you can interact with and manipulate the data. Using our tool, one can accentuate the data by stretching, in various directions, the data set. This, combined with the ability to become immersed in the data in three dimensions and the capability to “fly through” the data and view the data from multiple angles, provides the user a greater ability to discover features that would have otherwise been missed. A screenshot of the visualization is shown in Figure 8.

5 Conclusions

The visualization methods presented in this paper will serve as the basis for larger and more complex visualizations. The methods presented have proven to be robust, capable of handling a wide array of data. The improved interaction

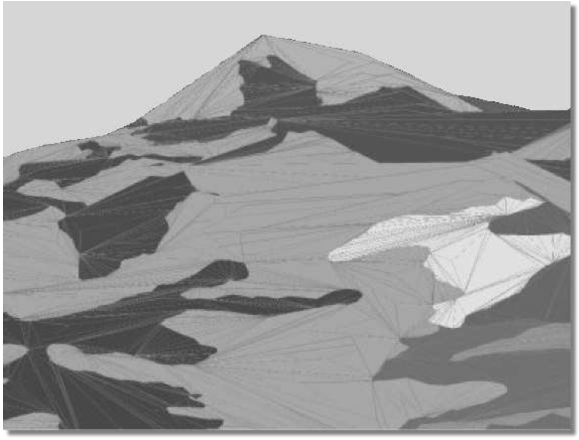

Fig. 8. Three-dimensional visualization of combined DEM and GIS layer. Outlining the triangles helps accentuate the height.

and immersion in the VR environment provides a more efficient and beneficial experience for GIS experts than some of the common packages available today.

The method devised by this paper is by no means the end to a line of work. In fact, the work presented by this paper will serve as the foundation for a series of visualization tools. Because the VR environment is geared towards better interaction, experts can work more efficiently. The added sense of depth provides the user with more detail than is typical with two-dimensional displays, allowing a user more accurately and efficiently analyze GIS data.

Data set sizes used ranged from tens of polygons, less than a thousand triangles, and a 100 × 100 DEM, all the way to hundreds of polygons, tens of thousands of triangles, and a DEM that was bigger than 1300 × 1100. While the running time and response time of our methods are affected by the size of the input, they still cope excellently with large input sets and allow for real-time interaction.

Packages like ArcGIS and Arc Scene provide users a rich set of features for handling GIS data. However, these packages aim to provide as many services as relevant (and rightfully so), instead of focusing on one or two very specific areas. Since our method runs as a standalone application, we have the ability to fine-tune the visualization and the level of interaction.

6 Future Work

The possibilities for future work are numerous. As stated before, the methods we proposed will serve as the foundation for more advanced visualizations. Another immediate possibility is the refinement of our code.

One larger goal we have in mind is to visualize dust kick-up in real-time. A significant analysis of terrain is key to modeling dust, and we believe that the methods implemented for this paper will serve as an excellent aid to this project.

The source code written for this paper is robust, capable of handling a wide variety of data sets. However, this robustness comes at a sacrifice to speed. An immediate avenue of future work is to optimize the source code, and to integrate faster algorithms into our visualization methods.

References

1. Smith, R.: Essential Techinques for Military Modeling and Simulation. Volume 1 of Simulation Conference Proceedings. (1998) 805–812
2. Walton, G.H., Patton, R.M., Parsons, D.J.: Usage Testing of Military Simulation Systems. Winter Simulation Conference (2001) 771–779
3. Service, D.M.F.P.: Former POW learns value of military training. TRADOC News Service (2003)
4. Leu, M.C., Hilgers, M.G., Agarwal, S., Hall, R.H., Lambert, T., Albright, R., Nebel, K.: Training in Virtual Environments for First Responders. ASEE Midwest Section Meeting (2003)
5. geo Community: USGS SDTS format Digital Elevation Model data (DEM). `http://www.dofactory.com/Patterns/Patterns.aspx` (Accessed June 13. 2005)
6. Meuser, M.R.: What is GIS and How Does it Work? `http://www.dofactory.com/Patterns/Patterns.aspx` (Accessed June 13. 2005)
7. ESRI: ArcGIS. `http://www.esri.com/software/arcgis/` (Accessed June 13. 2005)
8. Softwright: TopoScript .BIL File Format. `http://www.softwright.com/faq/support/toposcript_bil_file_format.html` (Accessed April 2. 2005)
9. ESRI: ESRI Shapefile Technical Description. `http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf` (Accessed March 17. 2005)
10. Board, O.A.R.: OpenGL - The Industry Standard for High Performance Graphics. `http://www.opengl.org/` (Accessed June 13. 2005)
11. Corporation, M.: Microsoft DirectX: Home Page. `http://www.microsoft.com/windows/directx/default.aspx` (Accessed June 13. 2005)
12. Schroeder, W.J., Zarge, J.A., Lorensen, W.E.: Decimation of Triangle Meshes. ACM SIGGRAPH (1992) 65–70
13. Schroeder, W.J.: A Topology Modifying Progressive Decimation Algorithm. IEEE Visualization (1997) 205–212
14. Kobbelt, L., Campagna, S., Seidel, H.: A General Framework for Mesh Decimation. Graphics Interface Conference (1998)
15. Schaufler, G., Sturzlinger, W.: Generating Multiple Levels of Detail from Polygonal Geometry Models. Virtual Environments 9́5 (1995) 33–41
16. Algorri, M.E., Frances-Schmitt: Mesh Simplification. Volume 15 of Computer Graphics Forum. (1996) 77
17. O'Rourke, J.: Computational Geometry in C. 2 edn. Cambridge University Press (1998)
18. de Berg, M., van Kreveld, M., Overmars, M., Shcwarzkopf, O.: Computational Geometry - Algorithms and Applications. 2 edn. Springer (1998)