

REAL-TIME 3D ANIMATIONS IN A 3D GIS

Jinwu Ma and Salvador Bayarri
jma@esri.com, sbayarri@esri.com
ESRI, USA

Abstract

This paper clarifies the concept of real-time animation, comparing it to non-interactive animation, and describes the major categories of interest for 3D GIS applications. Examples of each category are exposed as implemented in the ArcGIS's ArcScene application. Three main animation categories are identified: camera animation, scene object animation, and environment animation. For camera animation, the focus is on the spatial relationship between the observer and the target locations, which are the main elements of the OpenGL camera model. The animation of scene objects emphasizes changes that happen at different levels of the geographic data. The environment animation involves variations in the ambient and the rendering context of the 3D view. Finally, the paper discusses the mechanisms that the ArcScene customization environment offers in practice to create real-time animations.

Keyword: Spatial Data Visualization

Introduction

Techniques for computer generation of synthetic images of 3D scenes have existed for a long time, at least since the 1960s. An important distinction must be made between two basic approaches in the creation of synthetic images. The first approach emphasizes in the quality or realism of the results, and it commonly uses time-consuming processes to achieve these results, like ray-tracing and radiosity techniques (see Foley et al., 1997). These visualization algorithms require long computing times (an image may take seconds or minutes to be generated), but they do not require specific hardware. The resulting images are recorded together in some video format to be played back later. The animation is, in this sense, final and can be modified only by using image processing or video editing techniques.

An alternative approach for generating 3D images is real-time animation, which emphasizes the speed of the process. Generating several images or frames per second, this technique allows user interaction during the animation. A flight simulator is a good example of real-time animation. Although some specific hardware is required for real-time 3D graphics, nowadays it is inexpensive and widely available in personal computers. Basic techniques are well understood (see Moller and Haines, 1999), and available software ranges from programming libraries (OpenGL and Direct3D, see <http://www.opengl.org> and <http://msdn.microsoft.com/directx/>) to development environments and out-of-the-box applications like ArcScene, the main application program for ESRI's ArcGIS 3D Analyst extension (see Booth, 2001).

Non-real-time animations have its advantages in creating higher quality 3D graphics (at the expense of long processing time) and are capable of representing spatial-temporal changes over much longer/shorter time frames, but all changes must be specified in advance. Real-time animation sequences can be recorded and compiled into videos and played back later if we want to do so, but the technique's true power lies in being a tool for two-way exploratory information analysis, whereas non real-time animations are better for one-way well-packaged analysis result presentations. In this paper we will explore the basic functionalities that real-time 3D animation opens for GIS applications.

Three major aspects can be identified in real-time animation of GIS data, depending on which elements are subject to change. These aspects are camera animation, scene object animations, and environment animation. Of course, a complete animation can combine, simultaneously or sequentially, elements of the different types.

1. Camera Animation

ArcScene's 3D scene viewer is built on top of OpenGL, a widely adopted graphics standard in the industry. OpenGL and other 3D graphics software libraries use a projection to transform the 3D coordinates of the objects into 2D coordinates in the windows (see OpenGL ARB et al., 1997). The camera model defines how this projection is done. It is common to define an observer location, from which the line of sight originates, and a target location, at which the line of sight points. Additional parameters of the camera specify the roll angle and view field angle.

Exploring a 3D scene is the primary method for analyzing the data, in which process the spatial location of the camera observer and/or target are changed frequently. These changes can be associated with the user interaction through the operation of mouse or other input devices. For instance, the default active tool in ArcScene is the Navigation tool. In the navigation mode we can make the observer rotate around the target that stays fixed. Other specific tools are used to set the target and observer points to be on the surface of an object of the scene. Panning is another commonly used interaction mode, during which both the observer and the target points are translated in the same direction (shifting). Other frequently used interaction tools include the Zooming In/Out tool, that moves the observer closer to the target or away from it, or alternatively changes the view field angle, like we do with the zoom lens of a real camera.

A very useful scene navigation tool is the Fly tool (see Figure 1) that enables the user to perform a flyby animation interactively. During the flight both the observer and the target points change dynamically along a path. The user can use the mouse to control the flight altitude, speed, flight direction, and even the inclination.



Figure 1. The Navigation toolbar of ArcScene showing the Fly tool.

A flight sequence can then be recorded by taking snapshot images using stored camera points, and then compiling them into a digital video file (MPEG, AVI, etc.) for playback. This is an example of video made out of a real-time sequence. If not navigating interactively, a flyby animation can also use a pre-defined flight path. A simple implementation can be achieved by moving the observer-target pair at constant distance intervals along the flight path curve¹.

During a flyby animation we can also look to the left/right or up/down of the flight path, and also fly backwards. The roll angle can be adjusted dynamically to provide a more realistic movement, increasing the inward inclination angle when the camera makes a left or right turn.

Another useful interaction mode is the Walk mode. In this case the movement of the observer is restricted to be at a certain height on top of a surface (the ground, a building floor, etc). The drawback of using this tool is that it may slow down the interaction because of the cost to calculate the observer location on the scene objects.

2. Animation of Scene Objects

Animations can also involve changes in the objects that make the 3D scene. Objects can be organized in a hierarchy of complexity, and changes can occur at any level. Here is a simple scheme of the levels of the data structure used in ArcScene (see Figure 2):

Scene Level
 Group Layer Level
 Layer Level
 Feature / Surface Unit Level
 Vertex / Node / Cell Level

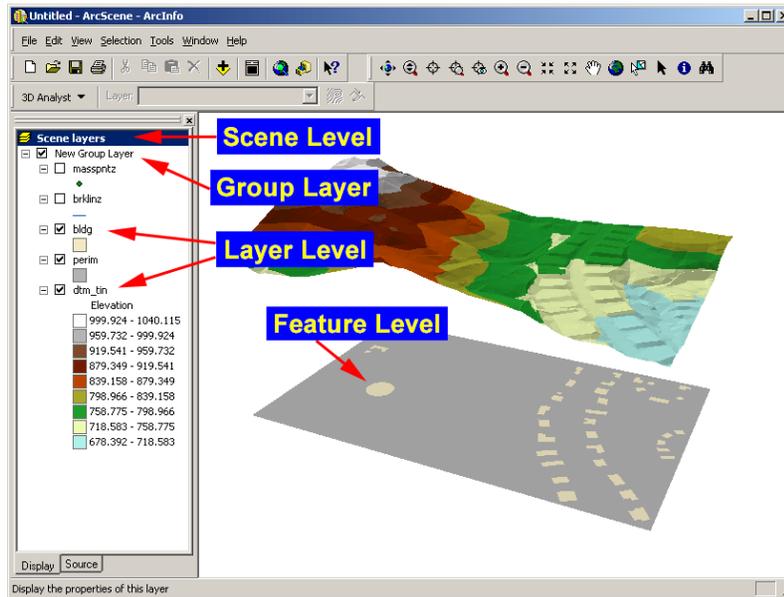


Figure 2. Data organization in ArcScene exemplified by the Table of Contents (vertex level label not shown)

The top-most structure below the scene is the group layer. It can contain one or more data layers. A layer (a feature layer, a raster layer, or a TIN layer) has its own symbology (shown in legend) and 3D properties, such as base height, offset, extrusion and so on. A feature has its own attributes and a shape (geometry) of a certain type (point, line, polygon, etc.). There are two types of surfaces: TIN surface and raster surface. TIN surfaces can have values for its surface unit (e.g. a triangle) and a geometric shape that can be decomposed in vertices or nodes. Raster surfaces can also have values and can be decomposed in raster cells. We can create animations that show changes in the geometry or symbology associated to each of these levels.

Scene Level

Some global properties may affect the way we see the scene contents, and we may want to change them during the animation. For instance, some packages provide different visualization modes (fill, wire-frame, textured). Another common property, important for GIS applications, is the vertical exaggeration or Z scale. Even if we are not interested in recording a video with changes in the Z scale, we still may want to use real-time to control it interactively until we find a suitable value. Other scene properties are related to the visualization environment, which we will describe later.

Layer Group Level

A very powerful animation technique consists in predefining a group of layers that represent snapshots of the evolution of an original dataset. For example, we may have one layer in the group for each week of the year, representing the distribution of ocean temperatures for that week. In order to make apparent the effect of time we can create a simple animation by looping through the layers in the group, making only one of them visible each time.

This technique is very flexible. We can introduce any type of change in each layer of the group, and they can even come from different datasets. The main drawback is that group layers normally contain a large number of individual layers and can consume a considerable amount of computing resources. It is usually better to keep a single layer and try to perform the changes in it, without the need of duplicating it for each change.

Layer Level

As we mentioned, a geo-data layer can have some properties of its own. The most evident is its spatial position in the scene. When several layers coincide in the same location we may want to separate them artificially for easier data analysis. In that case we can introduce an elevation offset and maybe a sideways translation in each layer. This type of layout positioning and animation is very useful for presentations.

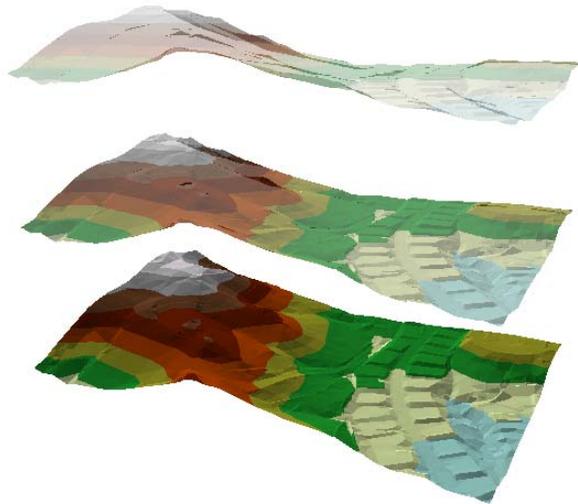


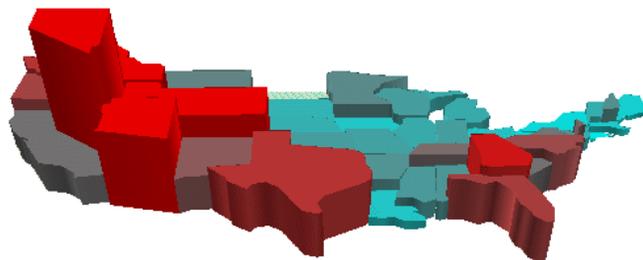
Figure 3. TIN layers with different vertical offsets and transparency values

Another way to present combined information coming from several layers in the same location is transparency. The Transparency tool in the 3D Effects toolbar of ArcScene lets you change value interactively to create different effects, enhancing the informational content of the scene. The symbology applied to the layer can also be changed, and this constitutes a way to explore patterns and extract important information. However, real-time performance in legend changes can only be achieved if the 3D GIS application provides the proper architecture.

Feature / Surface Level

Features represent well-defined objects, like buildings, points of interest etc. whose location we may want to change interactively while we create the dataset or analyze different options (e.g. we may be planning where new buildings should be built). Simple transformations, composed of scaling, rotation and translation operations are suitable for this purpose, and are present in many packages.

A different type of feature transformation happens when we want to use the geometry as a vehicle to present attribute information. A clear example is the use of the extrusion elevation of a feature to symbolize the value of an attribute. In Figure 4 we see an animation where the evolution of the US state population is animated by changing the extrusion value of the polygon feature that represents each state.



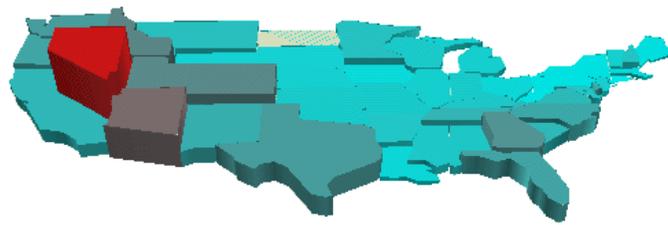


Figure 4. Animated simulation of US state population increases, exemplified by polygon height (and color).

The number of features within a feature layer can also be changed; new features can be added, and existing features can be dropped. Typical examples of this kind of animation are the evolution of human settlements or vegetation species in a geographical area. Again, the possibility of performing this animation in real-time will depend on the ability of the GIS application to support fast updates in the layers.

Symbols and legends play an important role in conventional cartography. Good 2D maps use appropriate cartographic symbols. But the conventional 2D map symbology is static; the set of symbols does not change in timeⁱⁱ. The use of real-time graphics in a 3D geo-spatial animation allows feature symbols to be changed on the fly, creating a varied and vivid visual effect with rich informational content. The changes in symbology will usually be linked to feature attributes that represent values of a magnitude in time.

We mentioned transparency at the layer level, but it can also be used at feature level to give relative importance to the features, focusing on the relevant ones, or creating fading effects. Color is of course the primary source of symbol information, and can also be animated. More sophisticated symbols include texture symbols, that can be used to show imagery or raster data on features or surfaces (see Figure 5).

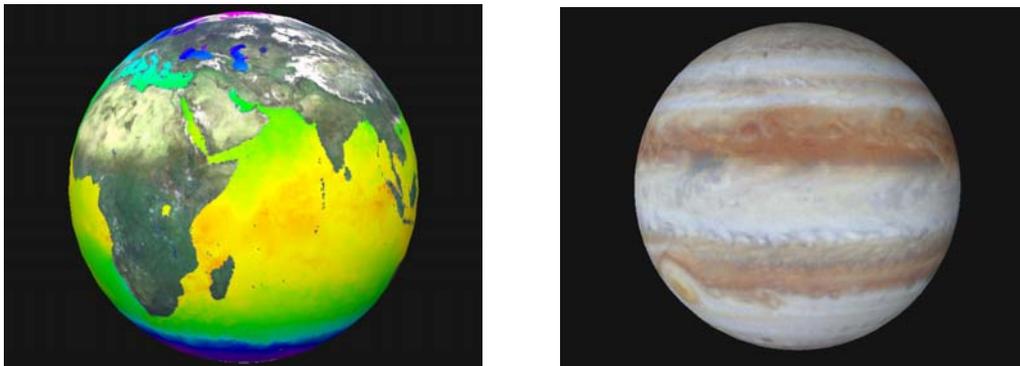


Figure 5. Images mapped as textures on spheres representing planets (the Earth and Jupiter).
In the animation version different images are shown in sequence.

Vertex Level

Features and Triangulated Irregular Network (TIN) surfaces are made of vertices, whose number and relative positions can change in time. They are usually modified during the editing or modeling stage of a project, but they can also be changed on-the-fly to handle dynamically the level of detail (i.e. number of vertices used in the features or TIN surfaces) of complex scenes. A specific type of vertex-based animation is sometimes used to output the results of simulation calculations. A common example is the simulation of erosion processes that change the elevation of the surface vertices (and possibly also the number of vertices, if the surface is a TIN).

3. Environment Animation

In this context, by environment we mean the visual elements and properties that affect how the geo-spatial data of the scene are displayed. These elements are used to enhance the appearance of the scene or to provide additional information.

Background and Backdrop

We can define two different types of background. The first one is two-dimensional, a bitmap drawn behind the scene objects, filling the entire scene viewer area. It can be used to display corporate logos or photographic images, but it is usually static and not integrated with the 3D scene.

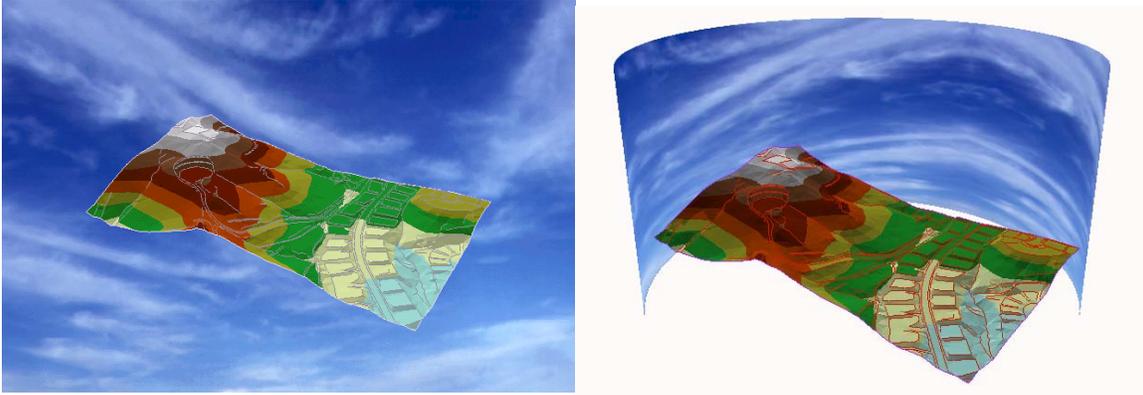


Figure 6. Sky backdrops: on the left the sky is treated as a wall paper to the whole scene; on the right the sky is mapped to a cylinder geometry, which is a scene layer by itself.

The second type of background mimics the appearance of distant objects, such as surrounding mountains or the sky. It is integrated in the 3D scene, so when we navigate and look around, the background changes accordingly (see Figure 6). It is usually implemented by adding a backdrop geometry (cylinder or spherical shape) that surrounds the central scene objects. A texture adds detail and visual realism to the backdrop.

Illumination

In order to enhance a volume effect in the scene objects, 3D scenes use light sources to modulate the colors of the features and surfaces. For each light source there are a number of parameters that we can change interactively. For instance, we have the direction of the light, usually defined by altitude and azimuth. If we use the light source to simulate the Sun, then the parameters reflect the Sun's altitude and azimuth (see Figure 7). We can combine the change of these parameters with the color of the sunlight to represent different times of the day or seasonal changes.

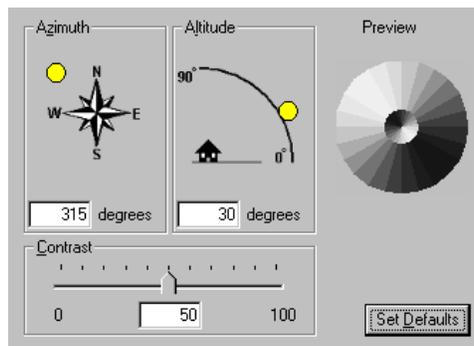


Figure 7. Illumination properties shown on ArcScene's Scene Properties dialog.

Atmospheric Effects

One way to obtain more realism in a scene is to use atmospheric effects such as fog or haze. Using fog in a scene strengthens the sense of distance and volume, and can be used in visibility analysis. Most graphics systems include a distance-based fog that basically blends the fog color with the color of the objects, depending on its distance to the viewer (see Figure 8). It is also possible to create localized fog by using transparent geometries. All these effects can also be modulated in time.

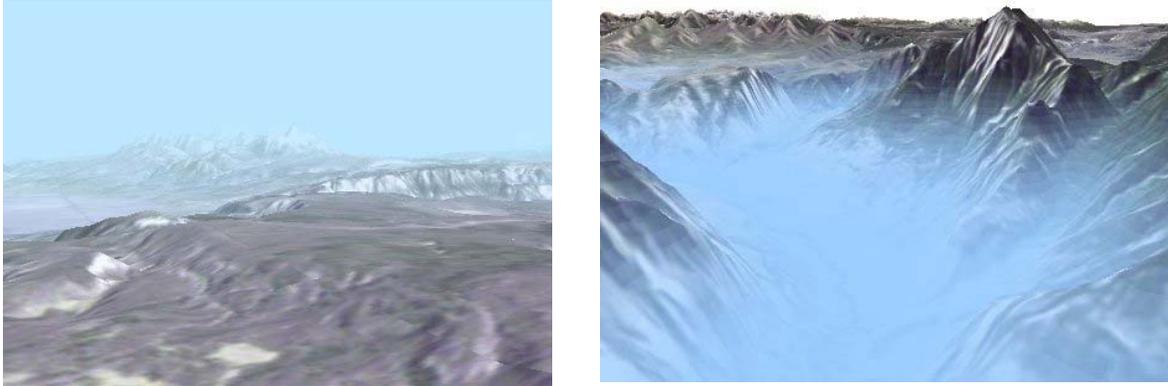


Figure 8. Using fog effect in a scene: on the left the fog is distance-based; on the right the fog is elevation-based.

Clip Planes and Analysis Tools

Analysis tools offering real-time results allow the user to query and explore interactively the datasets, quickly finding the relevant information. An example of this is the use of clipping planes, whose effect is to cut off parts of the scene, helping us to see the profile of a surface or the internal distribution of the data. OpenGL includes user-defined clipping planes that can be modified interactively.

The extraction of contours and steepest (flow) paths are traditional techniques used in surface analysis. Many applications offer tools to calculate these lines for a set of given locations or from a location selected by the user (usually by pointing at it with the mouse). If the GIS application can perform this calculation fast enough, it is possible to create the contours and steepest paths in an interactive way, updating them in real-time when the user changes the starting location by moving the mouse.

Scene Elements

To finish with the enumeration of animation resources, we must realize that we do not depend solely on the scene datasets themselves to represent information. We can create auxiliary objects and animate them to enrich the user experience. Traditional mapping elements like north arrows, axes, spatial buffers and labels can be easily adapted to the 3D environment (see Figure 9).

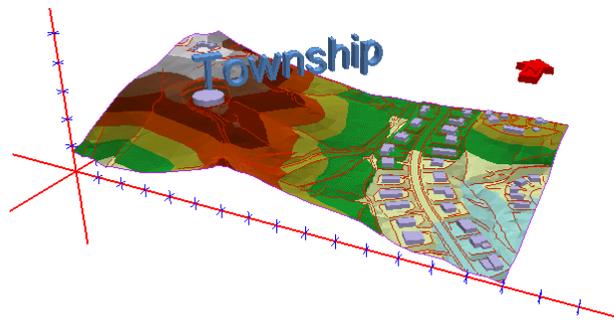


Figure 9. Using 3D text labels, axes, and north arrows in a scene.

Conclusion

We have seen the wide spectrum of resources that interactive animation offers to explore, analyze and present 3D GIS information. The software used in the examples of this paper is ArcScene, the main desktop application supplied with 3D Analyst, which is an extension of ArcGIS. Many of the animation effects can be obtained out-of-the-box, others can be achieved with minimal customization (adding sample tools to the user interface). More sophisticated or specific custom animation tools can be developed using Visual Basic for Applications (VBA), which is bundled with ArcScene. In addition, Microsoft Visual Basic (VB) and C++ can be used to create ActiveX/COM Commands (such as buttons and tools etc.) that can be then integrated into ArcScene. Moreover, external programming can be used to extend the core ArcScene, creating custom applications based on ArcGIS and 3D Analyst COM objects.

One thing worth mentioning here is that ArcScene 3D display capabilities can be combined with direct calls to the OpenGL API through VB/VBA customizationⁱⁱⁱ, a mechanism that we used to create the layered fog as one example. This extended capability assures that any real-time graphics functionality that is available in OpenGL can be integrated with ArcScene, and this opens a new world of possibilities.

Reference:

Booth B., "Using ArcGIS 3D Analyst", ESRI Press, 2001.

Foley J. D., van Dam A., Feiner S. K., and Hughes J. F., "Computer Graphics: Principles and Practice", 2nd Edition in C, Addison-Wesley Publishing Company, 1997.

Moller T. and Haines E., "[Real-Time Rendering](#)", A K Peters, Ltd., 1999.

OpenGL Architecture Review Board, Woo M., Neider J., and Davis T., "OpenGL Programming Guide", 2nd Edition, Addison-Wesley, 1997.

ⁱ The Fly-by Animation Builder sample extension (available from the ESRI's ArcScripts web site) used this method. The sample extension was originally developed for ArcView 3.x 3D Analyst, the precursor of ArcInfo 8.1 3D Analyst.

ⁱⁱ Conventional 2D maps referred to here means paper maps. It is possible to perform animations with digital 2D maps.

ⁱⁱⁱ A reference to a Visual Basic OpenGL type library needs to be added before OpenGL calls can be made.