UNIT III VISUAL COMPUTATION IN VIRTUAL REALITY

Fundamentals of Computer Graphics-Software and Hardware Technology on Stereoscopic Display-Advanced Techniques in CG: Management of Large Scale Environments & Real Time Rendering -Development Tools and Frameworks in Virtual Reality: Frameworks of Software Development Tools in VR. X3D Standard; Vega, MultiGen, Virtools etc

I. Fundamentals of Computer Graphics

To display a picture of any size on a computer screen is a difficult process. Computer graphics are used to simplify this process. Various algorithms and techniques are used to generate graphics in computers. Computer graphics is an art of drawing pictures on computer screens with the help of programming. It involves computations, creation, and manipulation of data. In other words, computer graphics is a rendering tool for the generation and manipulation of images.

Cathode Ray Tube

The primary output device in a graphical system is the video monitor. The main element of a video monitor is the Cathode Ray Tube CRT, shown in the following illustration.

The operation of CRT is very simple −

- The electron gun emits a beam of electrons cathode rays.
- The electron beam passes through focusing and deflection systems that direct it towards specified positions on the phosphor-coated screen.
- When the beam hits the screen, the phosphor emits a small spot of light at each position contacted by the electron beam.
- It redraws the picture by directing the electron beam back over the same screen points quickly.

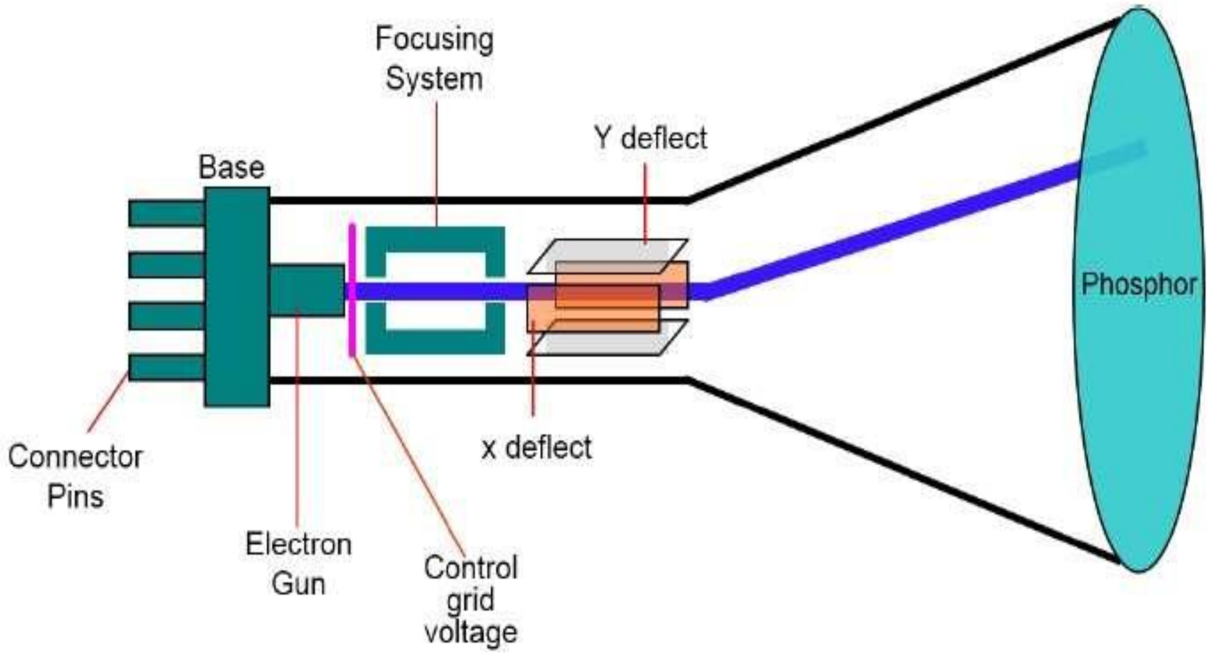


Fig. 3.1 Cathode Ray Tube

There are two ways Random scan and Raster scan by which an object can be displayed on the screen.

Raster Scan

In a raster scan system, the electron beam is swept across the screen, one row at a time from top to bottom. As the electron beam moves across each row, the beam intensity is turned on and off to create a pattern of illuminated spots. Picture definition is stored in memory area called the Refresh Buffer or Frame Buffer. This memory area holds the set of intensity values for all the screen points. Stored intensity values are then retrieved from the refresh buffer and "painted" on the screen one row scanline at a time as shown in the following illustration. Each screen point is referred to as a pixel picture element or pel. At the end of each scan line, the electron beam returns to the left side of the screen to begin displaying the next scan line.

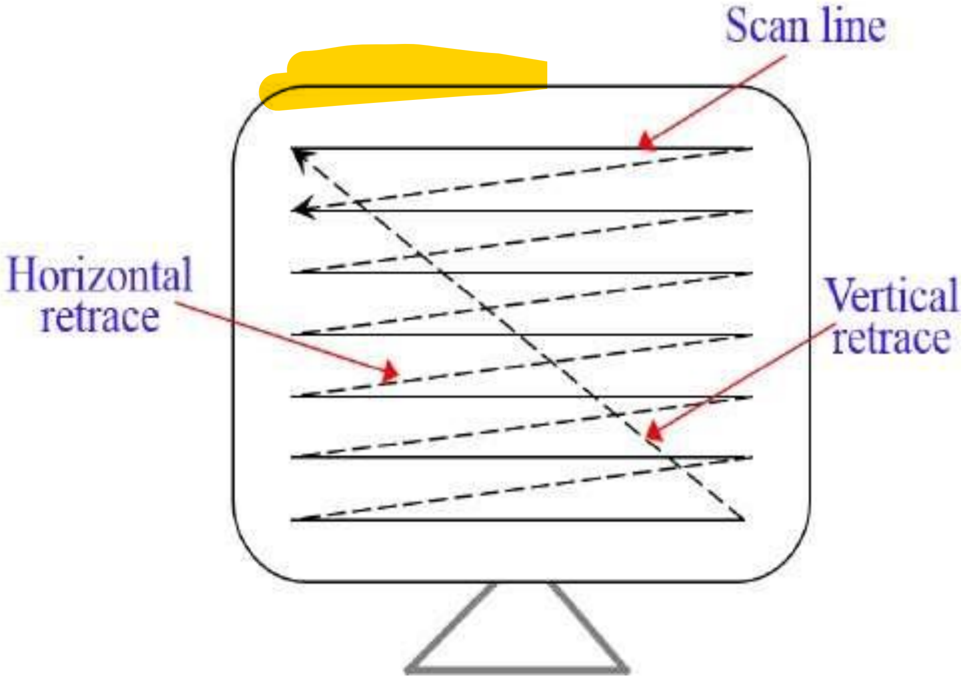


Fig. 3.2 Raster Scan

Vector Scan

In this technique, the electron beam is directed only to the part of the screen where the picture is to be drawn rather than scanning from left to right and top to bottom as in raster scan. It is also called vector display, stroke-writing display, or calligraphic display. Picture definition is stored as a set of line-drawing commands in an area of memory referred to as the refresh display file. To display a specified picture, the system cycles through the set of commands in the display file, drawing each component line in turn. After all the line-drawing commands are processed, the system cycles back to the first line command in the list. Random-scan displays are designed to draw all the component lines of a picture 30 to 60 times each second.

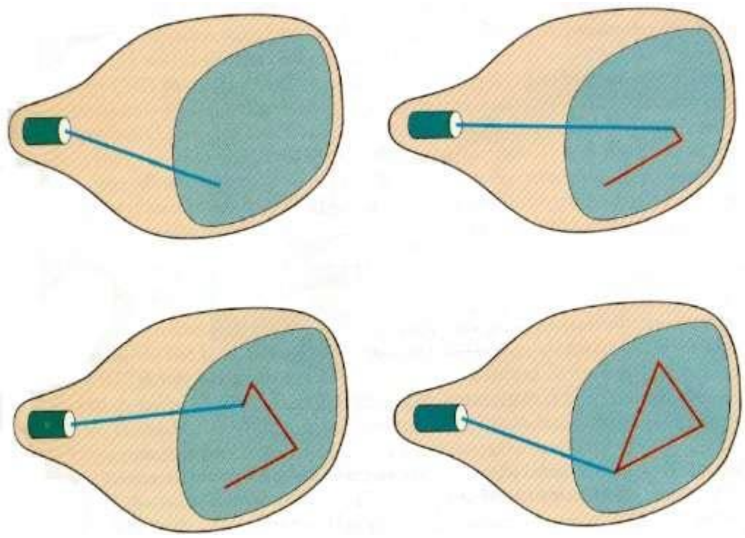


Fig. 3.3 Vector Scan

II. Software and Hardware Technology on Stereoscopic Display

STEREOSCOPY

Stereoscopy (also called stereoscopics or stereo imaging) is a technique for creating or enhancing the illusion of depth in an image by means of stereopsis for binocular vision. Any stereoscopic image is called a stereogram. Originally, stereogram referred to a pair of stereo images which could be viewed using a stereoscope.

Most stereoscopic methods present two offset images separately to the left and right eye of the viewer. These two-dimensional images are then combined in the brain to give the perception of 3D depth. This technique is distinguished from 3D displays that display an image in three full dimensions, allowing the observer to increase information about the 3-dimensional objects being displayed by head and eye movements. Stereoscopy creates the illusion of three-dimensional depth from given two-dimensional images. Human vision, including the perception of depth, is a complex process, which only begins with the acquisition of visual information taken in through the eyes; much processing ensues within the brain, as it strives to make sense of the raw information. One of the functions that occur within the brain as it interprets what the eyes see is assessing the relative distances of objects from the viewer, and the depth dimension of those objects. The cues that the brain uses to gauge relative distances and depth in a perceived scene include.

- Stereopsis
- Accommodation of the eye
- Overlapping of one object by another
- Subtended visual angle of an object of known size
- Linear perspective (convergence of parallel edges)
- Vertical position (objects closer to the horizon in the scene tend to be perceived as farther away)
- Haze or contrast, saturation, and color, greater distance generally being associated with greater haze, desaturation, and a shift toward blue
- Change in size of textured pattern detail

Stereoscopy is used in photogrammetry and also for entertainment through the production of stereograms. Stereoscopy is useful in viewing images rendered from large multi-dimensional data sets such as are produced by experimental data. Modern industrial three-dimensional photography may use 3D scanners to detect and record three-dimensional information. The three-dimensional depth information can be reconstructed from two images using a computer by correlating the pixels in the left and right images. Solving the Correspondence problem in the field of Computer Vision aims to create meaningful depth information from two images.

Monocular vs. stereo cues

To distinguish between monocular and stereo cues -

1- Monocular: single eye

2- Stereo cues: two eyes

Surely, the stereo cues, using both eyes, give you more depth information; but, you can still estimate some depth information from a single photo (monocular even using your both eyes), but you also may be deceived! For instance, please focus on the figure extracted. The two yellow lines have the same length; that is exactly true for the two line segments in the right figure.

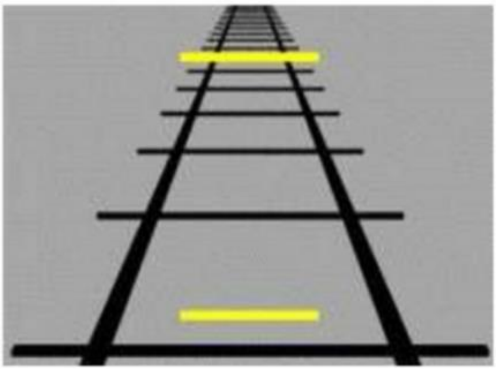


Fig. 3.4 Stereoscopy hardware equipment

` To see the power of the stereoscopy imaging, you should have specific hardware equipment that separate the two received images, one for the left eye and the second one for the right eye. Remember that, your brain must receive different images from each eye to give you the perception of 3D depth. The hardware side consists of two main items, which are the screen or the projector, and the glasses. There are many types of glasses for that purpose, depending on the type of the emitter (i.e. screen or projector).

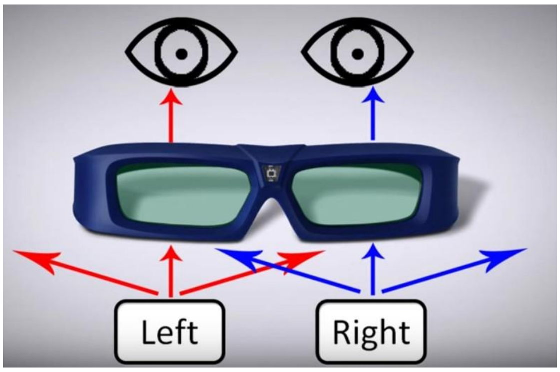


Fig. 3.5 Stereoscopy hardware equipment

STEREOSCOPIC TECHNOLOGY

Stereoscopic technology is a technology that aims to give you the illusion of depth by mimicking the real world. In other words, viewer's eyes are made to perceive two different CG, or even captured; footage likes what our eyes do to perceive real objects in our lives.

1. Parallax

In VR video on the other hand, stereoscopy is essential. As a virtual world is created, it should be made as immersive as possible and depth is an important part of that. To create depth in virtual reality, the same technique as in stereoscopic filming is used. Cameras used for 360 filming are separated into left and right cameras or sometimes algorithmically combined to create two panoramic images; one for the left eye and one for the right eye, with the same perspective difference to create a perfect parallax effect.

Parallax is the ability to see an object in two different ways based on the eye distance. That yields to our depth perception. There are three types of parallax based on the intersection point of the two eyes. See the following figures.

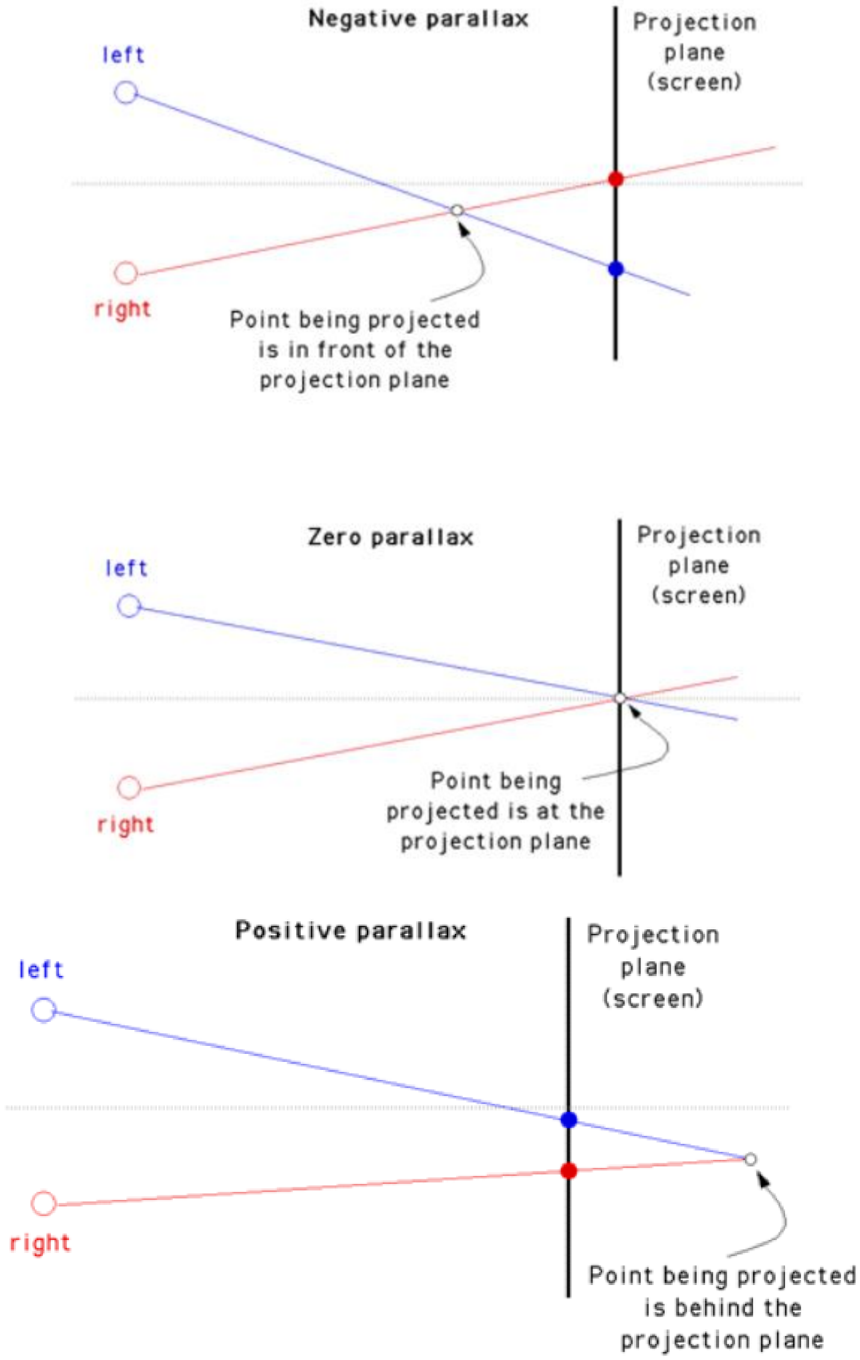


Fig. 3.6 Parallax - Types

2. Freeviewing

Freeviewing is viewing a side-by-side image pair without using a viewing device. Two methods are available to freeview:

- The parallel viewing method uses an image pair with the left-eye image on the left and the right-eye image on the right. The fused three-dimensional image appears larger and more distant than the two actual images, making it possible to convincingly simulate a life-size scene. The viewer attempts to look through the images with the eyes substantially parallel, as if looking at the actual scene. This can be difficult with normal vision because eye focus and binocular convergence are habitually coordinated. One approach to decoupling the two functions is to view the image pair extremely closes up with completely relaxed eyes, making no attempt to focus clearly but simply achieving comfortable stereoscopic fusion of the two blurry images by the "look-through" approach, and only then exerting the effort to focus them more clearly, increasing the viewing distance as necessary. Regardless of the approach used or the image medium, for comfortable viewing and stereoscopic accuracy the size and spacing of the images should be such that the corresponding points of very distant objects in the scene are separated by the same distance as the viewer's eyes, but not more; the average interocular distance is about 63 mm. Viewing much more widely separated images is possible, but because the eyes never diverge in normal use it usually requires some previous training and tends to cause eye strain.

- The cross-eyed viewing method swaps the left and right eye images so that they will be correctly seen cross-eyed, the left eye viewing the image on the right and vice versa. The fused three-dimensional image appears to be smaller and closer than the actual images, so that large objects and scenes appear miniaturized. This method is usually easier for freeviewing novices. As an aid to fusion, a fingertip can be placed just below the division between the two images, then slowly brought straight toward the viewer's eyes, keeping the eyes directed at the fingertip; at a certain distance, a fused three-dimensional image should seem to be hovering just above the finger. Alternatively, a piece of paper with a small opening cut into it can be used in a similar manner; when correctly positioned between the image pair and the viewer's eyes, it will seem to frame a small three-dimensional image.

3. Shutter system

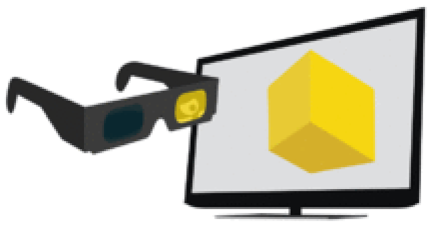


Fig. 3.7 Functional principle of active shutter 3D systems

A shutter system works by openly presenting the image intended for the left eye while blocking the right eye's view, then presenting the right-eye image while blocking the left eye, and repeating this so rapidly that the interruptions do not interfere with the perceived fusion of the two images into a single 3D image. It generally uses liquid crystal shutter glasses. Each eye's glass contains a liquid crystal layer which has the property of becoming dark when voltage is applied, being otherwise transparent. The glasses are controlled by a timing signal that allows the glasses to alternately darken over one eye, and then the other, in synchronization with the refresh rate of the screen. The main drawback of active shutters is that most 3D videos and movies were shot with simultaneous left and right views, so that it introduces a "time parallax" for anything side-moving: for instance, someone walking at 3.4 mph will be seen 20% too close or 25% too remote in the most current case of a 2x60 Hz projection.

4. Polarization systems

Fig. 3.8 Functional principle of polarized 3D systems

To present stereoscopic pictures, two images are projected superimposed onto the same screen through polarizing filters or presented on a display with polarized filters. For projection, a silver screen is used so that polarization is preserved. On most passive displays every other row of pixels is polarized for one eye or the other. This method is also known as being interlaced. The viewer wears low-cost eyeglasses which also contain a pair of opposite polarizing filters. As each filter only passes light which is similarly polarized and blocks the opposite polarized light, each eye only sees one of the images, and the effect is achieved.

5. Interference filter systems

This technique uses specific wavelengths of red, green, and blue for the right eye, and different wavelengths of red, green, and blue for the left eye. Eyeglasses which filter out the very specific wavelengths allow the wearer to see a full color 3D image. It is also known as spectral comb filtering or wavelength multiplex visualization or super-anaglyph. Dolby 3D uses this principle. The Omega 3D/Panavision 3D system has also used an improved version of this technology. In June 2012 the Omega 3D/Panavision 3D system was discontinued by DPVO Theatrical, who marketed it on behalf of Panavision, citing ″challenging global economic and 3D market conditions″.

6. Color anaglyph systems

Fig. 3.9 Anaglyph 3D glasses

Anaglyph 3D is the name given to the stereoscopic 3D effect achieved by means of encoding each eye's image using filters of different (usually chromatically opposite) colors, typically red and cyan. Red-cyan filters can be used because our vision processing systems use red and cyan comparisons, as well as blue and yellow, to determine the color and contours of objects. Anaglyph 3D images contain two differently filtered colored images, one for each eye. When viewed through the "color-coded" "anaglyph glasses", each of the two images reaches one eye, revealing an integrated stereoscopic image. The visual cortex of the brain fuses this into perception of a three dimensional scene or composition.

7. Holography

Laser holography, in its original "pure" form of the photographic transmission hologram, is the only technology yet created which can reproduce an object or scene with such complete realism that the reproduction is visually indistinguishable from the original, given the original lighting conditions. It creates a light field identical to that which emanated from the original scene, with parallax about all axes and a very wide viewing angle. The eye differentially focuses objects at different distances and subject detail is preserved down to the microscopic level. The effect is exactly like looking through a window. Unfortunately, this "pure" form requires the subject to be laser-lit and completely motionless—to within a minor fraction of the wavelength of lightduring the photographic exposure, and laser light must be used to properly view the results. Most people have never seen a laser-lit transmission hologram. The types of holograms commonly encountered have seriously compromised image quality so that ordinary white light can be used for viewing, and non-holographic intermediate imaging processes are almost always resorted to, as an alternative to using powerful and hazardous pulsed lasers, when living subjects are photographed.

Although the original photographic processes have proven impractical for general use, the combination of computer-generated holograms (CGH) and optoelectronic holographic displays, both under development for many years, has the potential to transform the half-century-old pipe dream of holographic 3D television into a reality; so far, however, the large amount of calculation required to generate just one detailed hologram, and the huge bandwidth required to transmit a stream of them, have confined this technology to the research laboratory.

In 2013, a Silicon Valley company, LEIA Inc, started manufacturing holographic displays well suited for mobile devices (watches, smartphones or tablets) using a multi-directional backlight and allowing a wide full-parallax angle view to see 3D content without the need of glasses.

8. Volumetric displays

Volumetric displays use some physical mechanism to display points of light within a volume. Such displays use voxels instead of pixels. Volumetric displays include multiplanar displays, which have multiple display planes stacked up, and rotating panel displays, where a rotating panel sweeps out a volume. Other technologies have been developed to project light dots in the air above a device. An infrared laser is focused on the destination in space, generating a small bubble of plasma which emits visible light.

9. Integral imaging

Integral imaging is a technique for producing 3D displays which are both autostereoscopic and multiscopic, meaning that the 3D image is viewed without the use of special glasses and different aspects are seen when it is viewed from positions that differ either horizontally or vertically. This is achieved by using an array of microlenses (akin to a lenticular lens, but an X–Y or "fly's eye" array in which each lenslet typically forms its own image of the scene without assistance from a larger objective lens) or pinholes to capture and display the scene as a 4D light field, producing stereoscopic images that exhibit realistic alterations of parallax and perspective when the viewer moves left, right, up, down, closer, or farther away.

10. Wiggle stereoscopy

Wiggle stereoscopy is an image display technique achieved by quickly alternating display of left and right sides of a stereogram. Found in animated GIF format on the web, online examples are visible in the New-York Public Library stereogram collection. The technique is also known as "Piku-Piku".

SOFTWARE FOR STEREOSCOPY

Bino

Bino plays stereoscopic videos, also known as 3D videos.

KMovisto (Version 0.6.1)

KMovisto is a molecule viewer for using in quantum chemistry. You are able to import GAUSSIAN 94 and GAUSSIAN 98 files (obtained from UNIX or MS Windows systems) or XYZ files and view your results in several view modes or edit the molecule geometry. Especially the 3D view modes (anaglyph or stereo pair) make it possible to enjoy stereoscopic impressions of the molecular structure - so this is what KMovisto makes a real 3D molecule viewer.

Mplayer

It is a command line video player but is possibly the most popular on Linux because of its capacity to play almost any anything, especially if you count GUIs that use it as a base such as gnome-mplayer and smplayer.

Plascoin

Plascolin is a Linux X11 tool to create and to view anaglyph stereo images or to display the left and right image on separate output devices (e.g. projectors).

SIV

SIV (Stereoscopic Image Viewer) is capable of displaying JPS stereo images and MPO stereo images in different stereo modes. It was tried in fullscreen/windowed mode with anaglyphic and quad buffered stereo mode.Main features in the 1.0 version are quad buffered stereo and vr920 headtracking.

Split MPO

This script takes a folder of .MPO files, extracts left and right images, and assembles them into pairs suitable for cross-eye, side-by-side and over-under (View Magic) use. The script seems to run fine on Linux and Mac. The shell is specified as "bash", but most should work as well. The script output files are easy to size in Open Office Draw. This script and Open Office Draw are a simple solution for anyone with a Mac or Linux to enjoy stereo photos from this fine Fuji camera.

Fuji W3 3D QuickLook Plugin (Version 1.0.0)

This QuickLook plugin enables a Fuji W3 3D MPO format image to be viewed by default in the finder and other applications using QuickLook on Mac OSX 10.5 or later. QuickLook isn't available for systems below 10.5

StereoPress (Version 1.4.0-E)

StereoPress helps you to make a stereo photo from your stereo pair. It is an application for Power Macintosh. Very easily, you can get an black & white anaglyph stereo image, color anaglyph stereo image or interleave stereo format.

3D Slide Projector (Version 1.05)

This software for creating your 3D slide show runs on Windows computers. It can make 3D images for anaglyphs or for interleave images or for dual projectors from your Left & Right stereo images taken by your digital cameras or scanner, and it can sync wav sounds. All order of your slides show and sounds are indicated by 'order.txt' file at the same folder of this software. If you have two PC-projectors and dual VGA video card, you will be able to have 3D projection by using dual screen mode of this program.

Anaglyph Maker (Version 1.08)

- free program to make black & white as well as color anaglyphs and interlaced images for LC-shutter glasses. Requires Windows™ 95, 98, 2000, Me, NT or XP.

Stereo Movie Builder (Version 0.3)

- software for building (stereoscopic) videos from a set of still pictures with various effects as zoom, pan and transitions. StereoMovieBuilder can generate standard AVI files, WMV files or Quicktime movie files. Input images can be in the JPEG or PNG format and videos. Input images can be monoscopic or stereoscopic (side by side images). SMB uses scripts written in a simple format for adding special effects like Ken Burns and transitions. StereoMovieBuilder can resize the pictures, transpose a stereo picture and generate various stereo format (anaglyph, half-frame, interlaced, ...) StereoMovieBuilder will run on any PC with Microsoft Windows 98/Me, 2000, XP, Vista or Windows 7.

III. Advanced Techniques in CG

Advanced Computer Graphics

Advanced computer graphics is a field that encompasses a vast range of topics and a large number of subfields such as game engine development, real-time rendering, global illumination methods and non-photorealistic rendering. Indeed, this field includes a large body of concepts and algorithms not generally covered in introductory graphics texts that deal primarily with basic transformations, projections, lighting, three-dimensional modelling techniques, texturing and rasterization algorithms.

Real Time Rendering

Real-time rendering is concerned with rapidly making images on the computer. It is the most highly interactive area of computer graphics. An image appears on the screen, the viewer acts or reacts, and this feedback affects what is generated next. This cycle of reaction and rendering happens at a rapid enough rate that the viewer does not see individual images, but rather becomes immersed in a dynamic process. The rate at which images are displayed is measured in frames per second (FPS) or Hertz (Hz). At one frame per second, there is little sense of interactivity; the user is painfully aware of the arrival of each new image. At around 6 FPS, a sense of interactivity started to grow. Video games aim for 30, 60, 72, or higher.

Real-time rendering is the process in which animations and images are quickly rendered. The process is so quick that the images seem to appear in real time. If you play or watch video games, you're experiencing real-time rendering. This technology takes many images and calculates them to match the frame rate of the human eye. The images appear on the screen as though you're experiencing it in real life. Video game designers have been using this technology for decades while architecture and construction designers are just hopping on the wagon. Most renderings are 3D representations of an image done on your computer. Real-time rendering is similar to the cinematography and photography process as it also uses light to create images. The rendering process can take anywhere from a few seconds to days to create a frame. The faster one is real-time rendering and the slower one is pre or offline rendering. These are the two main types of 3D rendering you will find. There are three main ingredients in real-time rendering. They are the application, the geometry, and the rasterizing stages. Together, they form life-like 3D representations of images. This allows designers and clients to see what the end-result of a project will look like in architecture and construction. The primary goal of real-time rendering is for the rendering to appear as real as possible. Images must appear at 24 frames per second to seem realistic to the human eye.

Benefits of Real-Time Rendering

Real-time rendering allows you to dig deep into your creative side. You don't need to worry about losing money and valuable time when you try crazy new ideas. With that flexibility, architects and designers can create and test ideas to see how customers will react. If the tests go well, architects and designers can move forward with building their ideas. This keeps building designs, both inside and out, evolving for the better. Clients and customers can view and edit building layouts before the building process starts. During the construction phase, 3D blueprint renderings replace the traditional 2D ones. Both construction workers and clients can find and solve problems efficiently.

In real estate, video tours allow you to view the interior of homes and apartments before moving in. This is a perfect option for people who can't visit the property in person. It's also great for marketing buildings that are still in the building process. It's easier to imagine an interior renovation with a 3D representation. Interior designers use real-time rendering to create fresh interior layouts for clients. Clients and designers can experiment with different flooring, cabinetry, and wall colors. All decisions can be finalized before the renovation process starts.

In computer graphics, real-time rendering is the immediate visual representation of a virtual scene. Changes to such scenes update within a short period of time, in tens of milliseconds, which are too minor for humans to interpret as a delay.

Real-Time and Offline Rendering

Basically, in real-time rendering, the computer is producing all the images from 3D geometry, textures, etc. on the fly and displaying it to the user as fast as possible (hopefully above 30 frames a second). The user can interact with the 3D scene using a variety of input devices such as mouse/keyboard, gamepad, tablet, etc. You'll find real-time graphics in everything from the iPhone to your computer and video game consoles. Your CG application viewport is using real-time rendering.

Offline rendering refers to anything where the frames are rendered to an image format, and the images are displayed later either as a still, or a sequence of images (e.g. 24

frames make up 1 second of pre-rendered video). Good examples of offline renderers are Mental Ray, VRay, RenderMan. Many of these software renderers make use of what's known as a 'raytracing' algorithm.

Purpose of a rendering pipeline

Essentially, the whole purpose of 3D computer graphics is to take a description of a world comprising of things like 3D geometry, lights, textures, materials, etc. and draw a 2D image. The final result is always a single 2D image. If it's a video game, it renders 2D images at high frame rates. At 60fps, it's rendering one image every 1/60th of a second. That means that your computer is doing all these mathematical calculations once every 60th of a second. The reason for pipelining is - once one process is done, it can pass the results off to the next stage of the pipeline and take on a new process. In the best-case scenario, this streamlines everything as each stage of the pipeline can work simultaneously. The pipeline can slow down if a stage of a pipeline takes longer to process. This means that the entire pipeline really can only run at the speed of the slowest process.

The pipeline

Stages of the pipeline:

- Application – anything that the programmer wants. Interactions, loading of models, etc. Feed scene information into Geometry stage.
- Geometry – Take scene information and transform it into 2D coordinates
    - Modeling transformations (modeling to world)
    - View transformations (world to view)
    - Vertex shading
    - Projection (view to screen)
    - Clipping (visibility culling)
- Rasterization – Draw pixels to a frame buffer
- Display

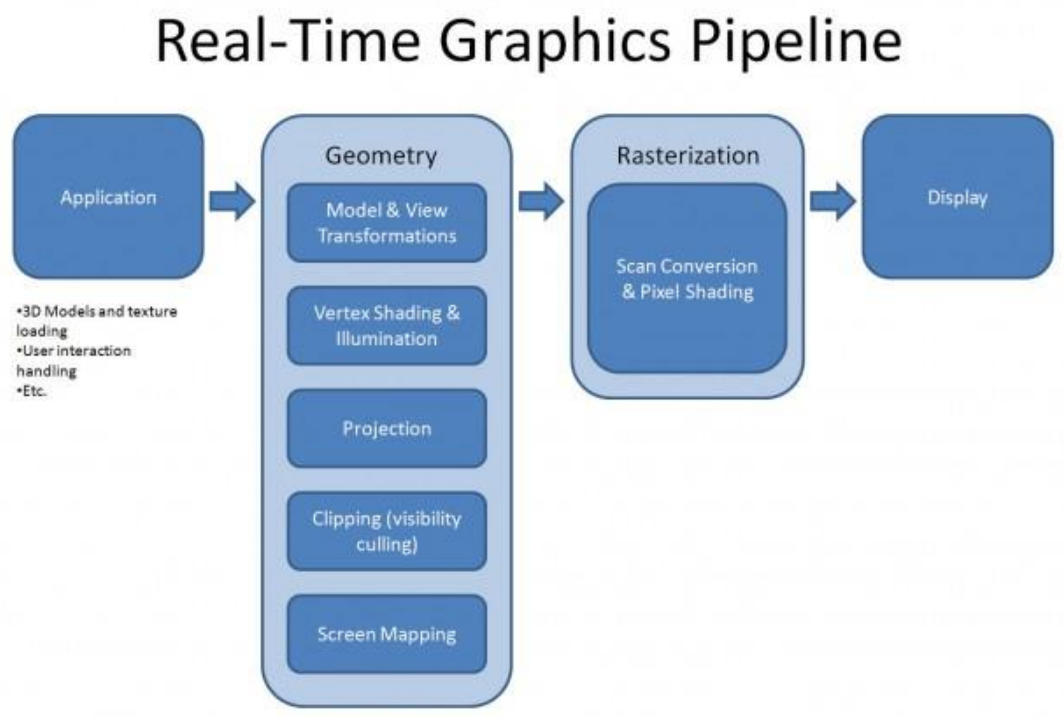


Fig. 3.10 Real-Time Graphics Pipeline

Application Stage

The Application stage of the pipeline is pretty open and left to programmers and designers to figure out what they want the application or game to do. The end result of the application stage are rendering primitives such as vertices/points, triangles, etc. that are fed to the next (Geometry) stage. At the end of the day, all that beautiful artwork that you've created is just a collection of data to be processed and displayed.

Some things that are useful for the application stage would be user input and how this affects objects in a scene. For example, in your 3D application, if you were to click on an object and move it, it needs to figure out all the variables: what object you want to move, by how many units, what direction. It's the same in a video game. If you have a character and move the view, the application stage is what takes user input such your mouse positions and figures out what angle the camera should rotate.

Another thing about the application stage is loading of assets. For example, all the textures and all the geometry in your scene needs to be read from disk and loaded into memory, or if it's already in memory, be manipulated if needed.

The Geometry Stage

The basic idea of the geometry stage is: Note that at this point in the pipeline, it hasn't actually drawn anything yet. It's just doing "transformations" of 3D primitives and turning them into 2D coordinates that can be fed into the next stage (Rasterization) to do the actual drawing.

Model and View Transformations

The idea behind Model and View Transformations is to place your objects in a scene, then view it from whatever angle and viewpoint you'd like. Modeling and Viewing Transformations confuse everyone new to the subject matter. Let's use the analogy of a taking a photograph.

In the real world, when you want to take a photograph of an object, you take the object, place the object in your scene, then place a camera in your scene and take the photo.

In graphics, you place the objects in your scene (model transform), but the camera is always at a fixed position, so you move around the entire world (view transform) so that it aligns to the camera at the distance and angles that you want, then the picture can be taken.

Table 3.1 Taking a photo in the real world vs the CG world

Step	Real world	CG world	CG term
1	Place objects (e.g. teapots, characters, etc. at various positions)	Place objects (e.g. teapots, characters, etc. at various positions).	Model transform
2	Place and aim camera. The world is fixed. You move the camera into position.	Transform the entire world to orient it to the camera. The camera is fixed. The world transforms around it.	View transform
3	Take the photo	Calculate lighting, projection, rasterize (render) the scene.	Lighting, Projection, Rasterization

Vertex Shading

At this point of the pipeline, 3D scene is described as geometric primitives, and the information needed to orient and move the scene around in the form of a Model-View Matrix is available. Now manipulation of the vertices should be performed based on programs called vertex shaders. So basically, a "shader" (a text file with a program in it) will be available and it runs on each vertex in the scene.

Nothing is drawn yet. Our light sources and objects are all in 3D space. Because of this, how lighting is affected can be calculated at any given vertex. These days, this was done using Vertex Shaders. Output of these calculations can be taken and passed them on to a Pixel Shader (at a later stage of the pipeline), which will make further calculations to draw each pixel.

Because Vertex Shaders run on GPU, they are generally quite fast (accessing memory directly on the video card) and can benefit from parallelization. So, programmers now use vertex shaders to do other types of vertex manipulation such as skinning and animation. (As a side note, the terminology causes a bit of confusion as 'vertex shader' implies that it deals with shading/illumination, while a shader can be used for many other things too).

Projection

Projection- is "what does the camera see?"

In computer graphics, it is done by what's called a "view frustum". A "frustum" is a pyramid with the top chopped off. With a view frustum, anything inside the volume of the frustum is drawn. Everything outside is excluded or "clipped". In the camera settings in a 3D application, you typically have settings for "Near" and "Far" clipping planes. These are the near and far planes of the view frustum. The left, right, top, bottom planes can all be defined by mathematical means by setting a Field of View angle.

Clipping

There's no point in doing calculations for objects that aren't seen, so there is a need to get rid of anything that is not in the view volume. This is harder than it sounds, because some objects/polygons may intersect one of the view planes. Basically, this can be handled by creating vertices for the polygons at the intersection points, and getting rid of the rest of the polygons that won't be seen.

Screen Mapping

All of the operations that we've done above are still in 3D (X, Y, Z) coordinates. These coordinates should be mapped to the viewport dimensions. For example, you might be running a game at 1920×1080, so all the geometric coordinates should be converted into pixel coordinates.

Rasterization

Next is displaying a 2D image. That's where "rasterization" comes in. A raster display, is basically a grid of pixels. Each pixel has color values assigned. A "frame buffer" stores the data for each pixel. 3D data is mapped to screen space coordinates, and some brute force calculations will be done to figure out what color each pixel should be:

- At each pixel, figure out what is visible.
- Determine what color the pixel should be.

Many of the ideas behind scan conversion were developed in the early days of graphics, and they haven't fundamentally changed since then. The algorithms are fast (enough) and many hardware manufacturers have embedded the algorithms into physical silicon to accelerate them.

In graphics, the term "Scan conversion" and "rasterization" are fairly interchangeably as they kind of mean the same thing. The line drawing and polygon filling algorithms use the idea of "scan lines" where each row of pixels are processed at a time.

Some of the things you do at Scan Conversion:

- Line drawing
- Polygon filling
- Depth test
- Texturing

Line Drawing

Line drawing is one of the most fundamental operations as it lets us draw things like wireframes on the screen. For example, based on 2 end points of a line, Bresenham Line Algorithm can be used.

Polygon Filling

Polygon filling is another interesting challenge. A popular algorithm, scan-line method is used to figure out which pixels to light up that is 'inside' the shape of the polygon.

Depth Testing

Typically in a scene, you've got objects in front of each other. So "depth test" should be done where whatever is closest to the viewer is only drawn. This is basically what depth-testing is. You may have already heard of something called a "Z-buffer". There's a similarity here. With all the vertices and polygons in 3D, and a Z (depth) value for them, the closest object can be determined to draw at each pixel.

Texturing

Rasterization is also where Textures are applied. At this stage, the pixels are glued from a texture onto the object. From a UV map, pixels will be found out on the texture to get the color information. Basically what it's doing is mapping from the device coordinate system (x, y screen space in pixels) to the modeling coordinate system (u, v), to the texture image (t, s). Artists create UV maps to facilitate this projection, so that during rasterization, it can look-up what pixels to read color information from.

Pixel Shading

During rasterization, Pixel Shading can also be done. Like vertex shaders, a pixel shader is simply a program in the form of a text file, which is compiled and run on the GPU at the same time as the application. Specifically, the pixel shader will run the program on each pixel that is being rasterized by the GPU. So, Vertex shaders operate on vertices in 3D space. Pixel shaders operate on pixels in 2D space.

Pixel shaders are important because there are many things that you want to do to affect each individual pixel as opposed to a vertex. For example, traditional real-time lighting and shading calculations were done per vertex using the Gouraud Shading Algorithm. This was deemed 'good enough' and at least it was fast enough for older generation hardware to run. But it really didn't look realistic at all, so you really wanted to do a lighting calculation at a specific pixel. With the advent of pixel shaders, per-pixel shading algorithms such as the Blinn-Phong shading model can be run.

Another thing to point out about pixel shaders is that they can get information fed to them from vertex shaders. For example, in the lighting calculations, certain information is needed from the vertex lighting calculations done in the vertex shader. This combination of vertex and pixel shaders enables a wide variety of effects that can be achieved. This has been a very high level overview of the real-time rendering pipeline.

IV. Management of Large Scale Environments & Real Time Rendering

Realistic Real-Time Rendering for Large-Scale Forest Scenes

Rendering a realistic large forest scene in real-time plays an important role in virtual reality applications, such as, virtual forestry, virtual tour and computer games. Since a forest consists of an extensive number of highly complex geometric models, real-time forest rendering is still a challenge. Several techniques are there to render highly realistic forests with realtime shadows. Since a forest with thousands of plants contains a vast amount of geometry, an efficient level of details (LOD) algorithm can be used to generate multiresolution (MR) models according to forest features. Leaf modeling method is used to have leaf models match leaf textures well. Parallel-split shadow mapping (PSSM) generation scheme can be used in rendering system. The data of tree models are organized into vertex buffer objects to enhance the rendering performance. A tree clipping operation is designed both in the view frustum and in the light frustum to avoid rendering models outside the current frustum and to remove the popping up and off effects. The combination of these techniques allows to realistically render a large forest with a large number of highly detailed plant models in real-time.

TREE MODELING AND MODEL PROCESSING

Modeling of tree foliage and processing of tree model are two key aspects of this work. Special techniques are used to construct LOD tree foliage models.

Tree modeling and simplification

A new technique with texture mapping of LOD tree foliage models is presented in this subsection.

Branch model processing

Tree skeleton models could be obtained from plant modeling software where tree skeleton models can also be obtained by two input images: sketches of main branches and crown silhouette on one input image, and sketches of two boundary branches and crown silhouette on anther input image. A series of static multiresolution models are constructed from the tree skeleton, which are used in real-time scene rendering talked next section. This method is used to construct branches only because it can generate continual LOD models. Continual LOD models are useful not only for efficient memory cost but also for model switching while wandering in a forest.

Leaf model processing

A leaf can be approximated as a mesh by two rows of quadrilaterals. The major axis of the rectangular mesh coincides with the main vein of the leaf. It is observed that the main vein usually determines the curl degree of the leaf, so a quadratic function is used to fit the main vein and the leaf geometry can then be obtained. Accordingly, the leaf LOD models can be easily constructed. However, as the main vein curves more and more in space, the two boundary edges of the mesh along the direction of the major axis should curl accordingly in order for better approximation of the mesh to the leaf shape, as shown in Figure 3.10(b).

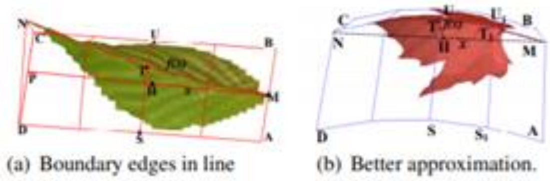


Fig. 3.10. Leaf modeling

A main vein, call as an arc, can be seen as a part of a parabola. The arc length, that is the leaf length L, and the arc height D can be measured from a real leaf. Leaf curl degree can be defined by the dihedral angle which can be represented by 6 AMB. The coordinates of A,B,M,N and the leaf unit normal vector $\vec{n}$ are given before a leaf is drawn in a designated position. A parabola function can be used to model the main vein:

$$f(x) = 4D(L-x)/L^2,\ 0 \le x \le L \qquad (1)$$

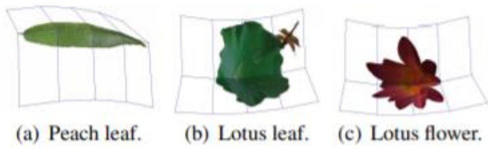


Fig. 3.11. A leaf model with different texture images.

The leaf geometry model has an advantage over other models in that different texture images (in alpha format) can be changed conveniently to a same leaf model with non-degenerative visual effect, as displayed in Figure 3.11.

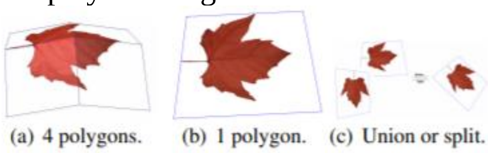


Fig. 3.12.Leaf models change base on quadric function

Not only can the leaf models easily match a given leaf texture, but also they can balance the visual effect and rendering speed when a large forest are to be displayed. If a tree is closed to the view point, the highly refined leaf model can be adopted; if it is far away from the view point, the model of low resolution is employed. Figure 3.12 shows a series of leaf models with different complex degree. Transition between a polygons model as in Figure 3.12 (a) and the model of one quadrilateral as in Figure 3.12(b) can be finished by using the function 1. If the distance between the tree and the view point decreases, the value of D decreases and the number of polygons decreases too. Transition between the Figure 3.12(b) and Figure 3.12 (c) is performed using the a method that can simplify plant organ following the structure of leaf phyllotaxy and flower anthotaxy, it is adopted to manage foliage in rendering for high simplification. In addition, the number and distribution of phyllotaxy in each branch are considered.

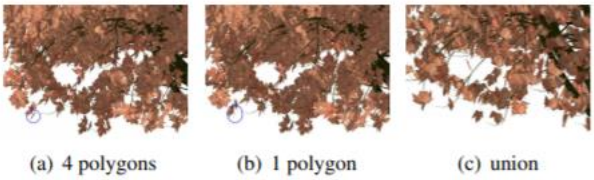


Fig. 3.13 Rendering results when a leaf represented with different polygons.

Phyllotaxy is the basic information of the leaf and flower distribution in the tree structure, so phyllotaxy geometry is constructed with experience or by measuring a real phyllotaxy of a tree. Some important parameters, such as the number of leaves in each node, the angle between two leave, the angle between axis and leaf, should be taken into account. The visual realism of a tree becomes less pleasing when the number of applied polygons for a leaf decreases. Figure 3.13 (a) shows a part of a tree whose leaves are represented with 4 polygons, and it looks realistic. In Figure 3.13 (b), each leaf is represented with one rectangle and the visual realism decreases. And in Figure 3.13 (c), several leaves is represented with one quad by union strategy and its realism is the weakest too.

Construction of Plant LOD models

A large forest often consists of trees from thousands to millions. If each tree is modeled with full information, memory will be exhausted quickly. In order to save memory while keeping the realistic, LOD models are often used in practice. However, too many LOD models also exhaust the memory for a large forest. Therefore, 4 or 5 models of different resolution is used as the LOD series to represent a tree instance. Because occlusion is common in forest rendering, the trees in the distance can be simplified greatly. With the simplification methods, five LOD models are selected for a tree instance according to rendering effect at different ranges of distance. The closer the tree is to the view point, the finer its selected model is. Figure 3.14 illustrates a LOD series. The number of polygons in each model is shown in the subtitle of each sub-figure.

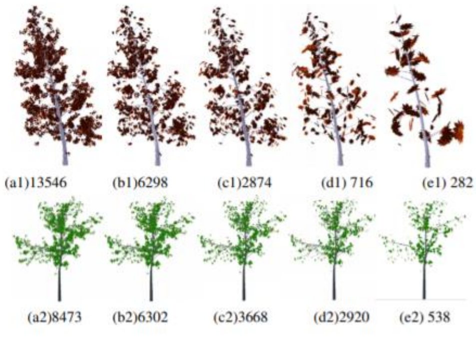


Fig. 3.14 LOD series of tree models

REAL-TIME FOREST RENDERING

PSSM is employed to produce real-time and anti-aliasing shadows. Figure 3.15 shows the rendering results of different single trees used in our system.

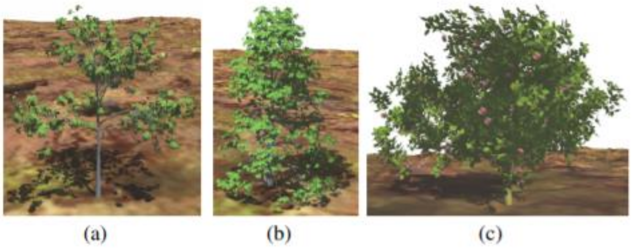
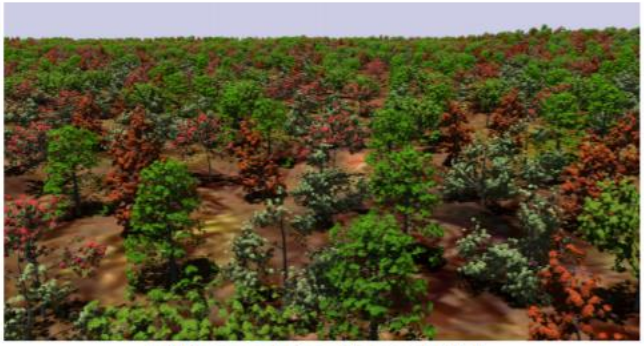


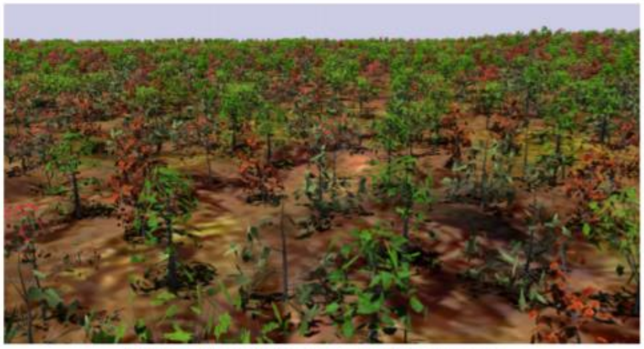
Fig. 3.15 Some single trees rendered with real-time and antialiasing shadows. (a) A simple tree. (b) A black popular with complex branching structure. (c) An apple tree with fruit and dense foliage.

Forest Scene Layout

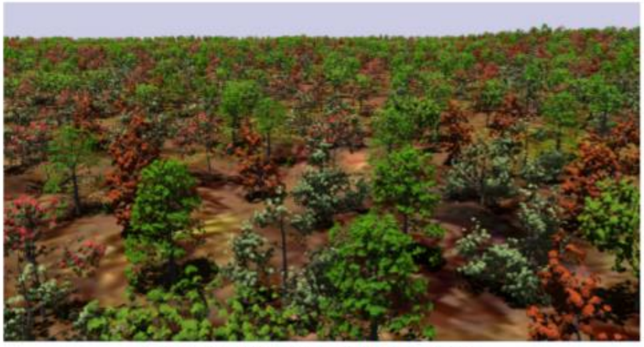
With some LOD tree models created with the method presented above, a large forest scene can be constructed as follows. A digital terrain model (DTM) covers an area of 262144 square meters. In the digital terrain model, there are 5 instances which produce 7446 tree positions with uniform distribution. Each instance owns 4 LOD models (from highest resolution to lowest one denoted as t1,t2,t3,t4)and one of them will be used according to the distance dct between the view point and the tree position. Figure 3.16(c) shows the experiment result with real-time shadow. In the scene, there are 7446 trees and 1671 in view port which displays 2162588 polygons, and the time costs about 0.083s per frame. Without LOD strategy, the forest costs more than 0.33s per frame, if only the detail models are used at all positions (Figure 3.16 (a)). If all the trees are represented with the simplest model(Figure 3.16(b)), it costs about 0.0625s per frame, but the realism is poor.



(a) Forest rendering with most detailed.



(b) Forest rendering with simplified models.



(c) Forest rendering with LOD models.

Fig. 3.16 Forest rendering with real time shadows.

Forest Rendering

Several techniques are employed to improve the rendering performance, such as clipping tree operation, vertex buffer objects. Since it is unnecessary to render trees outside the current camera frustum, clipping operation for each tree is used to cull those outside the frustum. Eight corners of each model's bounding box are projected from the object coordinates to the viewport coordinates. If neither of the projected points is in the window's viewport space, the tree will not be rendered. This approach is effective to cull trees far away from the camera. However, it will cause popping up and popping out artifact such that the trees near the viewer are unseen. To fix these popping effects, two additional points are checked while culling. One is the center of the front face of the bounding box and the other is the back face's center. If any projection of the four points (including the bounding box's left bottom and the right top points) is in the window's viewport, the tree will be rendered. Another restriction is the distance. A clipping distance threshold μ is set which is not too large. Trees with distances to the viewer less than μ will be rendered no matter whether they are in the camera frustum or not. Although a few trees outside the camera frustum could be rendered, it brings little overhead to the overall performance. The clipping operation is employed in both view frustum before rendering and the subdivided light frustums when generating the shadow maps. The clipping operation is based on the bounding boxes of the trees. Taking full advantage of the GPU capacity, the tree models' vertices, normals and texture coordinates are organized into vertex buffer objects.

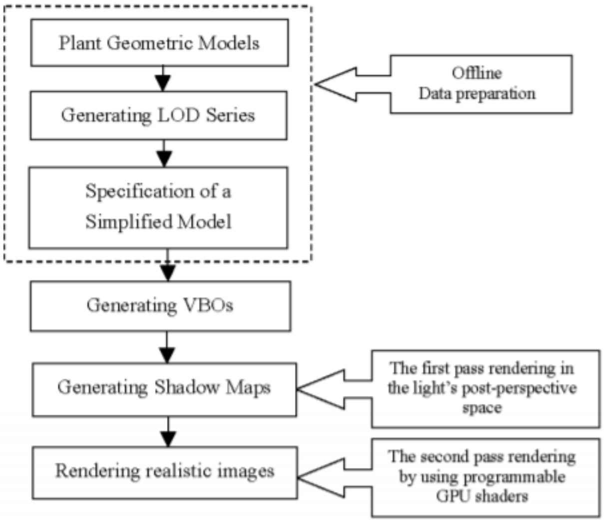


Fig. 3.17 Data processing flow chart

Vertex buffer object is an OpenGL extension. It provides an interface that allows the array data to be stored in high performance graphics memory, thereby promoting an efficient data transfer and avoiding repetitive calls of graphics functions. This technique dramatically enhances the performance. LOD are used for tree rendering based on distances to the viewer. Four levels of detail can be made for each tree species to reduce the rendering burden. In the implementation, the finest LOD is made up of 6000 - 8000 triangles and the coarsest LOD consists of 800 - 1100 triangles. The detailed data processing flow of the rendering system is shown in Figure 3.17.

The techniques presented to render large forest scenes consists of tens of thousands of highly detailed trees at interactive frame rates, even with a realistic real-time shadowing effect. Close-up viewing for trees, walk-through and flyover a forest are available. It can be applied easily to the video games and interactive visualization.

V. Development Tools and Frameworks in Virtual Reality

Frameworks of Software Development Tools in VR

Unity

- Unity is famous for game development, however, it helps to build VR solutions for many other sectors too.
- E.g., you can create VR solutions for automotive, transportation, manufacturing, media & entertainment, engineering, construction, etc. with Unity.
- Unity is a cross-platform game engine initially released by Unity Technologies, in 2005.
- The focus of Unity lies in the development of both 2D and 3D games and interactive content.
- Unity now supports over 20 different target platforms for deploying, while its most popular platforms are the PC, Android and iOS systems.
- Unity features a complete toolkit for designing and building games, including interfaces for graphics, audio, and level-building tools, requiring minimal use of external programs to work on projects.

Amazon Sumerian

- Brings a new dimension to your web and mobile applications with Amazon Sumerian.
- 3D immersive experiences are breathing new life into user experiences on the web, increasing customer engagement with brands and improving productivity in the workplace.
- Amazon Sumerian makes it easy to create engaging 3D front-end experiences and is integrated with AWS services to provide easy access to machine learning, chatbots, code execution and more.
- Amazon Web Services offers a broad set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security and enterprise applications.
- These services help organizations move faster, lower IT costs, and scale.
- As a web-based platform, our immersive experiences are accessible via a simple browser URL and are able to run on popular hardware for AR/VR.

Google VR for everyone

- Google Cardboard is a virtual reality (VR) platform developed by Google.
- Named for its fold-out cardboard viewer into which a Smartphone is inserted, the platform was intended as a low-cost system to encourage interest and development in VR applications.
- Users can either build their own viewer from simple, low-cost components using specifications published by Google, or purchase a pre-manufactured one.
- To use the platform, users run Cardboard-compatible mobile apps on their phone, place it into the back of the viewer, and view content through the lenses.
- The platform was created by David Coz and Damien Henry, French Google engineers at the Google Cultural Institute in Paris
- It was introduced at the Google I/O 2014 developers conference, where a Cardboard viewer was given away to all attendees.
- The Cardboard software development kit (SDK) was released for the Android and iOS operating systems;
- the SDK's VR View allows developers to embed VR content on the web as well as in their apps.
- Through March 2017, over 160 million Cardboard-enabled app downloads were made.
- By November 2019, over 15 million viewer units had shipped.
- After the success of Cardboard, Google developed an enhanced VR platform, Daydream, which was launched in 2016.
- Following declining interest in Cardboard, Google announced in November 2019 that it would open-source the platform's SDK.
- In March 2021, the Google Store stopped selling Cardboard viewers.

CRYENGINE

- CryEngine (officially stylized as CRYENGINE) is a game engine designed by the German game developer Crytek.
- It has been used in all of their titles with the initial version being used in Far Cry, and continues to be updated to support new consoles and hardware for their games.
- It has also been used for many third-party games under Crytek's licensing scheme, including Sniper: Ghost Warrior 2 and SNOW.
- Warhorse Studios uses a modified version of the engine for their medieval RPG Kingdom Come: Deliverance.
- Ubisoft maintains an in-house, heavily modified version of CryEngine from the original Far Cry called the Dunia Engine, which is used in their later iterations of the Far Cry series.
- According to various anonymous reports in April 2015, CryEngine was licensed to Amazon for $50–70 million.

- Consequently, in February 2016, Amazon released its own reworked and extended version of CryEngine under the name of Amazon Lumberyard.
- Well-known to 3D game developers, CRYENGINE is a robust choice for a VR software development tool.
- You can build virtual reality apps with it that will work with popular VR platforms like Oculus Rift, PlayStation 4, Xbox One, etc.
- CRYENGINE
    - Can incorporate excellent visuals in your app.
    - Creating a VR app or VR game is easy with CRYENGINE since it offers sandbox and other relevant tools.
    - Can easily create characters.
    - There are built-in audio solutions.
    - Can build real-time visualization and interaction with CRYENGINE, which provides an immersive experience to your stakeholders.

Features

- Simultaneous WYSIWYG on all platforms in sandbox editor
- "Hot-update" for all platforms in sandbox editor
- Material editor
- Road and river tools
- Vehicle creator
- Fully flexible time of day system
- Streaming
- Performance Analysis Tools
- Facial animation editor
- Multi-core support
- Sandbox development layers
- Offline rendering
- Resource compiler
- Natural lighting and dynamic soft shadows

Unreal Engine 4 (UE4)

- Unreal Engine is a game engine developed by Epic Games, first showcased in the 1998 first-person shooter game Unreal.
- Initially developed for PC first-person shooters, it has since been used in a variety of genres of three-dimensional (3D) games and has seen adoption by other industries, most notably the film and television industry.
- Written in C++, the Unreal Engine features a high degree of portability, supporting a wide range of desktop, mobile, console and virtual reality platforms.
- The latest generation is Unreal Engine 4, which was launched in 2014 under a subscription model.
- Unreal Engine (UE4) is a complete suite of creation tools for game development, architectural and automotive visualization, linear film and television content creation, broadcast and live event production, training and simulation, and other real-time applications.

- Unreal Engine 4 (UE4) offers a powerful set of VR development tools.
- With UE4, you can build VR apps that will work on a variety of VR platforms, e.g., Oculus, Sony, Samsung Gear VR, Android, iOS, Google VR, etc.
- The UE4 platform has many features –
    - It offers access to its C++ source code and Python scripts, therefore, any VR developer in your team can study the engine in detail and learn how to use it.
    - UE4 has a multiplayer framework, real-time rendering of visuals, and a flexible editor.
    - With the Blueprint visual scripting tool offered by UE4, you can create prototypes quickly.

It's easy to add animation, sequence, audio, simulation, effects, etc.

Features

- From design visualizations and cinematic experiences to high-quality games across PC, console, mobile, VR, and AR, Unreal Engine gives everything you need to start, ship, grow, and stand out from the crowd.
- Pipeline Integration
- World Building
- Animation
- Rendering, Lighting and Materials
- Simulation and Effects
- Game play and Interactive Authoring
- Integrated Media Support
- Virtual Production
- Developer Tools
- Platform Support

3DS Max

- 3ds Max is a computer graphics program for creating 3D models, animations, and digital images.
- 3ds Max is often used for character modeling and animation as well as for rendering photorealistic images of buildings and other objects.
- When it comes to modeling 3ds Max is unmatched in speed and simplicity.
- formerly 3D Studio and 3D Studio Max, is a professional 3D computer graphics program for making 3D animations, models, games and images.
- It has modeling capabilities and a flexible plugin architecture and must be used on the Microsoft Windows platform.
- It is frequently used by video game developers, many TV commercial studios, and architectural visualization studios.
- It is also used for movie effects and movie pre-visualization.
- Known for its modeling and animation tools,
- Latest version of 3ds Max also features shaders (such as ambient occlusion and subsurface scattering), dynamic simulation, particle systems, radiosity, normal map creation and rendering, global illumination, a customizable user interface, new icons, and its own scripting language.

Maya

- Maya is an application used to generate 3D assets for use in film, television, game development and architecture.
- The software was initially released for the IRIX operating system.
- However, this support was discontinued in August 2006 after the release of version 6.5.
- Maya is a 3D computer graphics application that runs on Windows, macOS and Linux, originally developed by Alias Systems Corporation (formerly Alias|Wavefront) and currently owned and developed by Autodesk.
- It is used to create assets for interactive 3D applications (including video games), animated films, TV series, and visual effects.
- Users define a virtual workspace (scene) to implement and edit media of a particular project.
- Scenes can be saved in a variety of formats, the default being .mb (Maya D).
- Maya exposes a node graph architecture.
- Scene elements are node-based,
- each node having its own attributes and customization.
- As a result, the visual representation of a scene is based entirely on a network of interconnecting nodes, depending on each other's information.
- The widespread use of Maya in the film industry is usually associated with its development on the film Dinosaur, released by Disney in 2000.
- In 2003, when the company received an Academy Award for technical achievement,
    - it was noted to be used in films such as The Lord of the Rings: The Two Towers, Spider-Man (2002), Ice Age, and Star Wars: Episode II – Attack of the Clones.
- By 2015, VentureBeat Magazine stated that all ten films in consideration for the Best Visual Effects Academy Award had used Autodesk Maya and that it had been "used on every winning film since 1997."

SketchUp

- SketchUp is a 3D modeling computer program for drawing applications such as architectural, interior design, landscape architecture, civil and mechanical engineering, film and video game design.
- It is available as a web-based application, SketchUp Free, and a paid version with additional functionality, SketchUp Pro.
- SketchUp is owned by Trimble Inc., a mapping surveying and navigation equipment company.
- The program includes drawing layout functionality, surface rendering in different "styles", enables placement of its models within Google Earth.
- 3D Warehouse is an open library in which SketchUp users may upload and download 3D models to share.

Three. Js

- Three.js is a cross-browser JavaScript library and application programming interface (API) used to create and display animated 3D computer graphics in a web browser using WebGL.
- Three.js allows the creation of graphical processing unit (GPU)-accelerated 3D animations using the JavaScript language as part of a website
- This is possible due to the advent of WebGL.
- WebGL (Web Graphics Library) is a JavaScript API for rendering interactive 3D computer graphics and 2D graphics within any compatible web browser without the use of plug-ins.

A-Frame

- A-Frame is an open-source web framework for building virtual reality (VR) experiences.
- It is maintained by developers from Supermedium (Diego Marcos, Kevin Ngo) and Google (Don McCurdy).
- A-Frame is an entity component system framework for Three.js where developers can create 3D and WebVR scenes using HTML.
- Originally developed within the Mozilla VR team during mid-to-late 2015.
- Created in order to allow web developers and designers to author 3D and VR experiences with HTML without having to know WebGL.
- A-Frame's first public release was on December 16, 2015.
- On December 16, 2019 A-Frame version 1.0.0 was released.
- All online IDEs support A-Frame as a result of being based on HTML.

React VR

- React VR is a JavaScript framework developed by Oculus, a division of Facebook with the aim of creating web based virtual reality apps.
- a framework for the creation of VR applications that run in your web browser.
- It pairs modern APIs like WebGL and WebVR with the declarative power of React, producing experiences that can be consumed through a variety of devices.
- The declarative model that is used in React can be adopted in the React VR framework to create content for 360-degree experiences.

- Developers can access the virtual reality devices that are on the web using the WebVR API.
- Without using any plug-ins, developers can render 3D graphics in any compatible browser using the WebGL API (Web Graphics Library API).
- Since React VR mimics React JaveScript Framework for the most part, developers who have previous experience of building React apps will have no trouble creating virtual reality experiences using Facebook's React VR.
- React VR - suffers from significant limitations, such as performance issues and support for more immersive content

VI. X3D Standard

- Extensible 3D (X3D) Graphics is the royalty-free open standard for publishing, viewing, printing and archiving interactive 3D models on the Web.
- X3D standards are developed and maintained by the Web3D Consortium.
- X3D is an ISO-ratified, file format and run-time architecture to represent and communicate 3D scenes and objects.
- X3D fully represents 3-dimensional data.
- X3D has evolved from its beginnings as the Virtual Reality Modeling Language (VRML).
  - VRML is used to illustrate 3-D objects, buildings, landscapes or other items requiring 3-D structure and is very similar to Hypertext Markup Language (HTML).
  - VRML also uses textual representation to define 3-D illusion presentation methods. VRML is also known as Virtual Reality Markup Language.
- X3D provides a system for the storage, retrieval and playback of real time 3D scenes in multiple applications, all within an open architecture to support a wide array of domains and user scenarios.

X3D Strengths

- X3D is a hub for publishing 3D data.
- X3D acts as a central hub that can route 3D model information between diverse 3D applications.
- A higher-level language to compose several 3D assets into a meaningful 3D Web applications with interactivity.
- Geometric data and metadata is written and read with open, non-proprietary tools.
- When data is presented in an X3D file it can be visualized with X3D players available over all platforms integrated with WebGL, glTF, HTML5 and the DOM.
  - glTF(GL Transmission Format) is a royalty-free specification for the efficient transmission and loading of 3D scenes and models by engines and applications.
- There are several workflows and tools to import and export data between X3D and other open and proprietary formats.

X3D Features

- XML Integrated: Cross-platform, usable with Web Services, Distributed Networks, inter-application model transfer
- Componentized: allows lightweight core 3D run-time delivery engine

- Extensible: allows components to be added to extend functionality for vertical market applications and services
- Profiled: standardized sets of extensions to meet specific application needs
- Evolutionary: easy to update and preserve VRML97 content as X3D
- Broadcast/Embedded Application Ready: from mobile phones to supercomputers
- Real-Time: graphics are high quality, real-time, interactive, and include audio and video as well as 3D data.
- Well-Specified: makes it easier to build conformant, consistent and bug-free implementations for various encodings

X3D Supports

- 3D graphics and programmable shaders - Polygonal geometry, parametric geometry, hierarchical transformations, lighting, materials, multi-pass/multi-stage texture mapping, pixel and vertex shaders, hardware acceleration
- 2D graphics - Spatialized text; 2D vector graphics; 2D/3D compositing
- CAD data - Translation of CAD data to an open format for publishing and interactive media
- Animation - Timers and interpolators to drive continous animations; humanoid animation and morphing
- Spatialized audio and video - Audio-visual sources mapped onto geometry in the scene
- User interaction - Mouse-based picking and dragging; keyboard input
- Navigation - Cameras; user movement within the 3D scene; collision, proximity and visibility detection
- User-defined objects - Ability to extend built-in browser functionality by creating user-defined data types
- Scripting - Ability to dynamically change the scene via programming and scripting languages
- Networking - Ability to compose a single X3D scene out of assets located on a network; hyperlinking of objects to other scenes or assets located on the World Wide Web
- Physical simulation and real-time communication - Humanoid animation; geospatial datasets; integration with Distributed Interactive Simulation (DIS) protocols
- Security: compatibly supports XML Security through use of XML Encryption and Digital Signature (authentication)
- Portability: in addition to XML, functionally identical encodings (ClassicVRML, Compressed Binary, JSON) and programming languages (JavaScript, Java, C++) are available for X3D scene interchange.
- Extensible: scene authors can create full-fledged language functionality using Scripts, Inlines, Prototypes, and Components/Profiles

VII. Vega

- Vega is a visualization development toolkit for real-time simulation it has improved the functionality of Performer, which is a rendering toolkit based on OpenGL.
- Although Vega is an expensive platform-dependent tool, it performs better than Java3D in terms of the frame rate for continuous scenes.

- Even though Java3D is free and platform-independent, it does not perform as well as Vega.
    - Enhance. Visualize. Immerse.
- Vega Prime is a comprehensive visualization toolkit that not only lets you create and deploy game-quality visuals and electro-optical sensor views for simulations,
- but allows to scale and extend the application to achieve high-density scenes across wide geographic areas in real-time.
- Providing an extremely flexible 3D visualization environment, Vega Prime's modular environment lets developers add or modify features, and seamlessly connect, interoperate and synchronize across systems.
- Reach unprecedented levels of realism using dynamic shadows, high-resolution detail, sophisticated atmospheric models, 3D clouds, natural vegetation, and realistic night scenes.
- Vega Prime is ideally suited for the efficient rendering of very large, high-resolution areas – from out-of-the-window content to highly realistic sensor views when combined with Ondulus-family sensors.
- Vega Prime supports VR devices.
- Built with the OpenVR SDK, Vega Prime supports devices such as
  - Oculus Rift and
  - HTC Vive (virtual reality headset).
    - no longer available

Benefits

- Add, Modify, and Extend Features: Flexible architecture lets - stay current with the market’s new demands and innovations.
- Maintain and Reuse Content across Systems: Platform independence lets -develop on one platform and deploy on another.
- Designed for Training and Simulation: From marine and coastal to land and air, supports true-to-life visuals with country-sized databases.
- Fast, Real-Time Performance: Smart resource management lets you avoid bottlenecks and diagnose problems to deliver 60Hz deterministic performance.
- Presagis M&S Suite: Integration within the Presagis M&S Suite means uninterrupted workflow and collaboration in the creation of databases; from terrain and models, to simulation and visualization.
- Vega is a visualization grammar, a declarative language for creating, saving, and sharing interactive visualization designs.
- Vega Visualization provides the building blocks to quickly create custom, server-side visualization rendering for large datasets using the power of SQL.
- With Vega, - can describe the visual appearance and interactive behavior of visualization in a JSON format, and generate web-based views using Canvas or SVG.
- JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax.
- It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa)
- SVG, which stands for Scalable Vector Graphics, is an XML-based vector image format for two-dimensional graphics with support for interactivity and animation.
- They can be created and edited with any text editor, as well as with drawing software.

Vega Visualization

- Vega Visualization is a declarative language that provides the tools to support custom visualizations of large datasets, high-level exploratory data analysis, as well as flexible combinations of data visualization designs and interaction techniques.

- The Vega specification is in JSON structure, making it easy to understand, create, and operate programmatically.
- Developers and big data analysts are equipped with JSON visualizer tools that readily support custom algorithms and advanced visualization techniques without the burden of complex geometric visualization details.
- Vega facilitates the use of data visualization across a variety of web applications with its toolkit for data visualization:
- Vega provides a framework for data visualization designs such as data loading, transformation, scales, map projections, and graphical marks.
- Interaction techniques can be specified using reactive signals that dynamically modify a visualization in response to input event streams.
- Vega treats user input, such as mouse movement and touch events, as first-class streaming data to drive reactive updates to data visualizations.
- Vega data visualizations can be rendered using either HTML5 Canvas, which can provide improved rendering performance and scalability, or SVG, which can be used for infinitely zoomable, print-worthy vector graphics.
- Vega supports a wide variety of dataset loaders, allowing interactive visualization of many different data formats, and single or multi process application development.
- Reduce risk and improve asset utilization with COTS products
- Commercial off-the-shelf or commercially available off-the-shelf (COTS) - products are packaged or canned (ready-made) hardware or software
- Increase productivity with a consistent, compatible, and easy-to-use programming interface
- Attain predictable performance results and reduce development cycles
- Spend less time on graphics programming issues and more on domain-specific problem-solving
- Optimize realtime performance easily
- Meet demanding budgets and development schedules

Improve maintainability and support of applications

Vega Special Effects

- Pre-defined animation sequences, designed to simulate the appearance of certain dynamic visual effects, are hard or even impossible to render using standard database techniques.
- The Vega Special Effects module creates visual effects through various real-time techniques, from shaded geometry for non-textured machines to complex particle animations with texture paging, for the ultimate in real-time 3D effects.
- Vega Special Effects comes bundled with a large number of existing effects:

• Volumetric smoke	• Tracer
• Billboard smoke	• Explosion
• Fire/Flames	• Debris
• Muzzle flash	• Rotor wash
• Flak	• Water explosion
• Missile trail	• Rotating blade

VIII. MultiGen

- MultiGen-Paradigm, a developer of realtime 3D graphics software solutions, announced the availability of version 1.1 of SiteBuilder 3D
- SiteBuilder 3D provides users with a solution to quickly and easily transform 2D map data into realistic, fully interactive 3D scenes.
- In addition, the company is announcing the initial release of ModelBuilder 3D, an optional authoring software toolset that gives users the power to generate 3D models of real-world buildings, objects and vegetation for incorporation into 3D scenes generated by SiteBuilder 3D.
- Both products facilitate the simple creation of 3D scenes from GIS and geospatial data, without requiring a high-level of technical or 3D modeling experience, and are a direct result of MultiGen-Paradigm's commitment to expanding the use of realtime 3D visualization to 3D GIS.
- Some of the significant new features delivered with SiteBuilder 3D v1.1
    - include the abilities to generate terrain from any feature theme, or themes,
    - that contain elevation data, to define and
    - navigate custom paths, and
    - to produce digital movie files directly from interactive sessions.
- In addition, this latest release delivers enhanced environmental effects, display a top-down or orthographic view of the 3D scene.
- The technology for ModelBuilder 3D is based on MultiGen Creator™, the widely adopted modelling and authoring system for realtime 3D commercial, urban, and military simulation applications.
- ModelBuilder 3D will allow users to enhance realism by providing them the ability to quickly generate and incorporate 3D models of real-world buildings, objects and vegetation.

IX. Virtools

- Virtools was a software developer and vendor, created in 1993 and owned by Dassault Systèmes since July 2005.
- They offered a development environment to create 3D real-time applications and related services, targeted at system integrators, game studios and corporate end-users.
- Since 2006, the software is called 3DVIA Virtools as part of Dassault Systèmes' 3DVIA brand.
- The last release was Virtools 5 (5.9.9.15).
- Dassault Systèmes no longer updates the software and has taken it down in March 2009.
- The development platform is used in the industry for virtual reality applications, video games (prototyping and rapid development), as well as for other highly interactive 3D experiences, in web marketing and virtual product maintenance.

- It was awarded the 2009 MITX Technology Award for the best use of video in support of a product launch.
- Virtools is one of the major development tools used to create Ballance.
    - (Ballance is a 3D puzzle video game for Microsoft Windows)
- a powerful 3D content creation toolkit.
- For map makers and modders, it can be used to create or modify Ballance configuration files and maps.
- Development and maintenance of Virtools has ceased, and the software is no longer available for purchase since 2014.

Virtools consists of the following parts

- ➢ an Authoring application
- ➢ a Behavioral Engine (CK2)
- ➢ a Rendering Engine
- ➢ a Software Development Kit (SDK)

- Virtools is not designed to create 3D models, but it could be forced into doing so.

Different versions

Virtools Dev 2.1

- The version used originally by game makers to create Ballance.
- Need - to make use of the behavior plugins found in Ballance.
- Unfortunately this version is no longer available on the Internet.

Virtools Dev 3.0

- This version can be used to create or modify Ballance NMO files.
- According to the feedbacks from some mappers, Virtools 3.0, compared with 3.5, gets better perfomance on Windows XP;
- however, due to unknown reasons, this version gets stuck at the licence interface on most systems.

Virtools Dev 3.5

- The version used by most mappers.
- It's also the highest version that can be used to modify NMO files without making the resulting files non-loadable by the game.

Virtools 4.x

- Virtools SA is acquired by 3DVIA prior to the release of this version.
- It gains the ability to import and export 3DVIA's 3DXML format, has better shader support, and comes with a lot other improvements.
- A revised version (4.1) was released in addition to the initial 4.0 release.

Virtools 5.0

- The final version of Virtools.
- It has many useful functions, but NMO files modified on this version can't be loaded by Ballance.
- It's been proved that, by re-saving files modified by Virtools 5.0 in Virtools Dev 3.5, the files can return loadable by Ballance.
- In addition, multiple fan-made games (including Ballance Remix and World's Hardest Game 3D) were made with Virtools 5.0.