SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

School of Computing

Department of Computer Science and Engineering

UNIT - III

AUGMENTED AND VIRTUAL REALITY – SCSA3019

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.

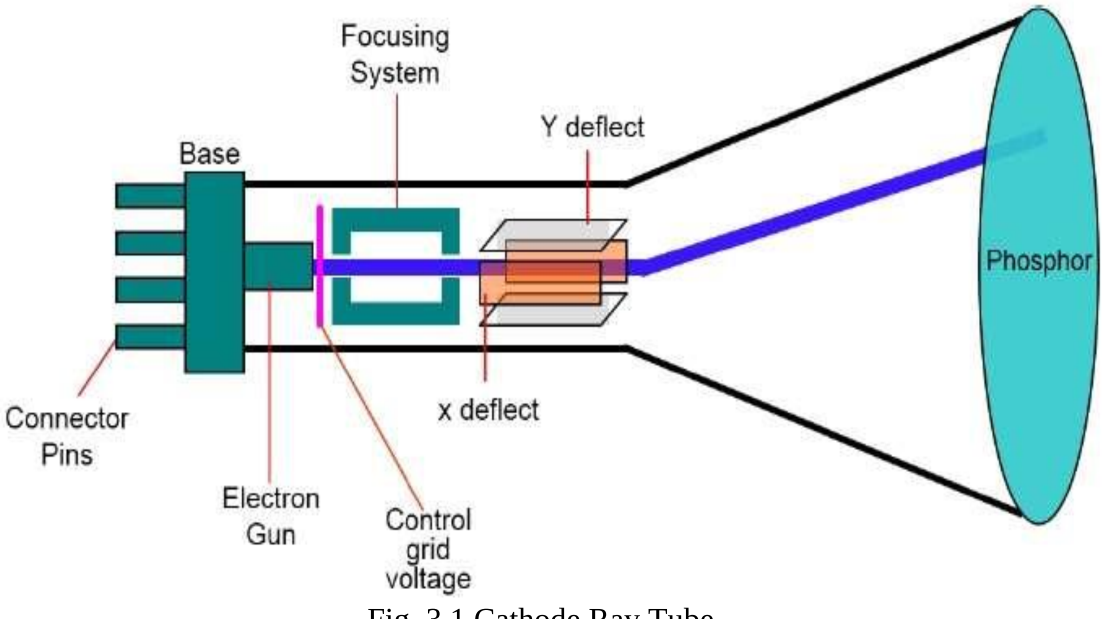


Fig. 3.1 Cathode Ray Tube

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.

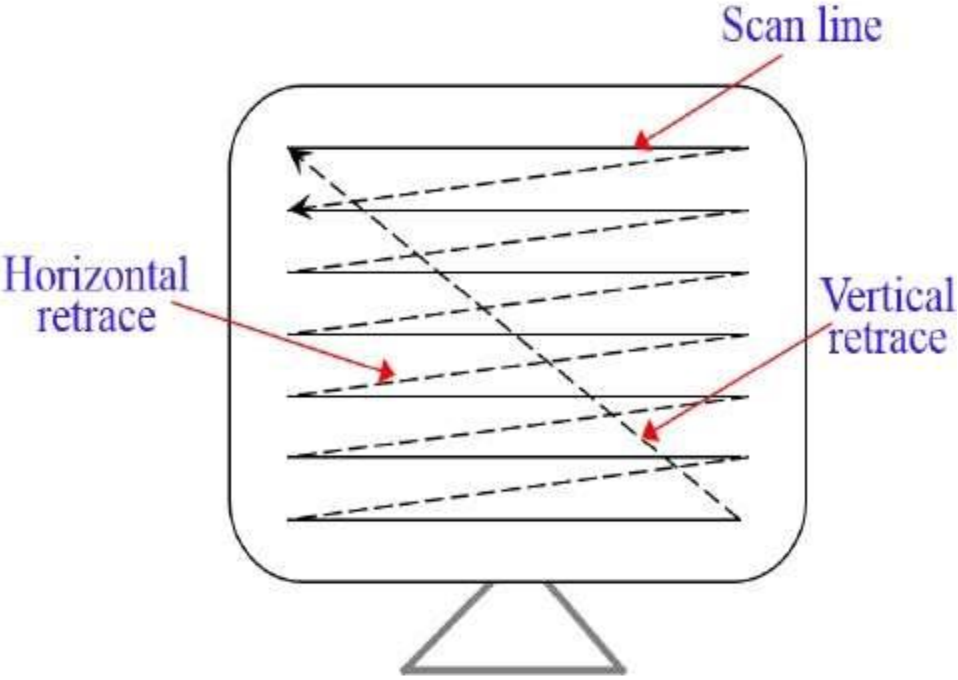


Fig. 3.2 Raster Scan

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.

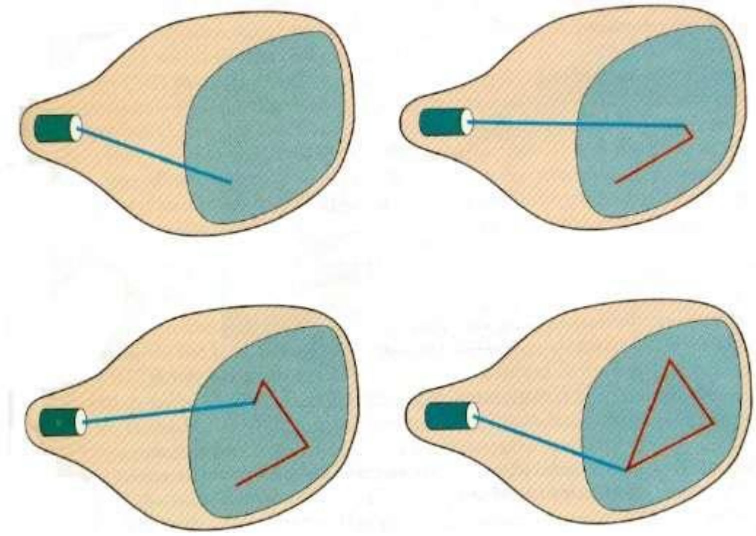


Fig. 3.3 Vector Scan

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. Trust me; the two yellow lines have the same length; that is exactly true for the two line segments in the right figure.

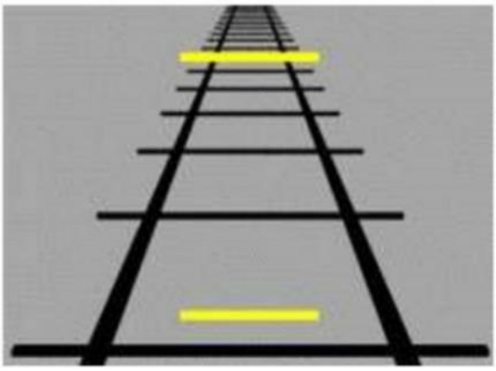


Fig. 3.4 Stereoscopy hardware equipment

` 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)

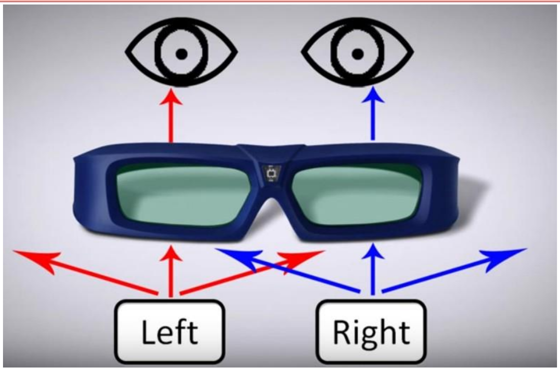


Fig. 3.5 Stereoscopy hardware equipment

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.

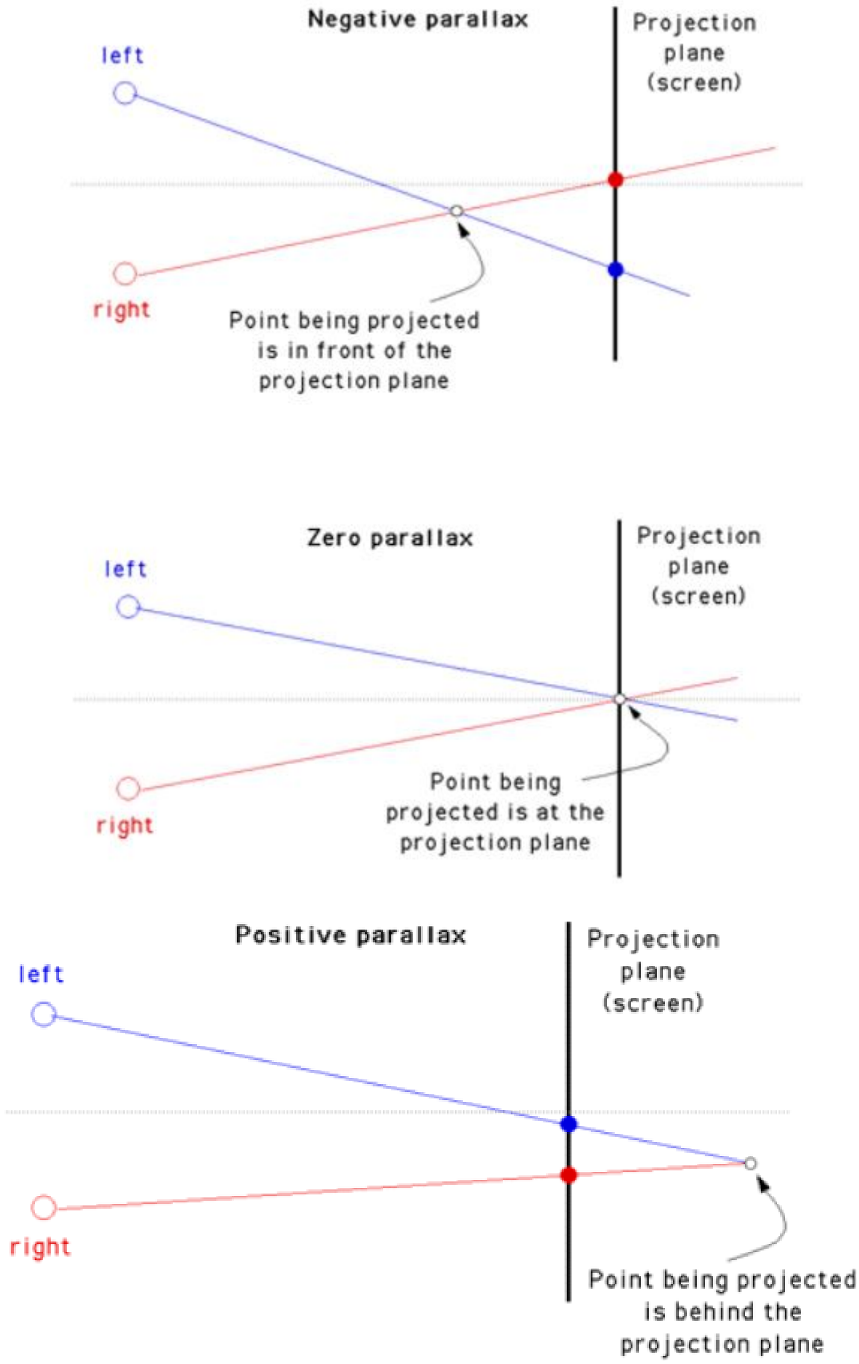


Fig. 3.6 Parallax - Types

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

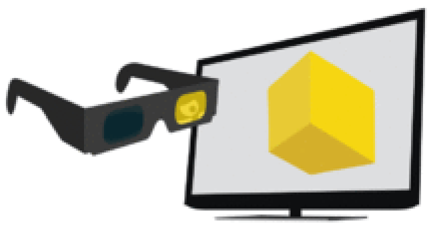


Fig. 3.7 Functional principle of active shutter 3D systems

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)

A wonderful 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)

A 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

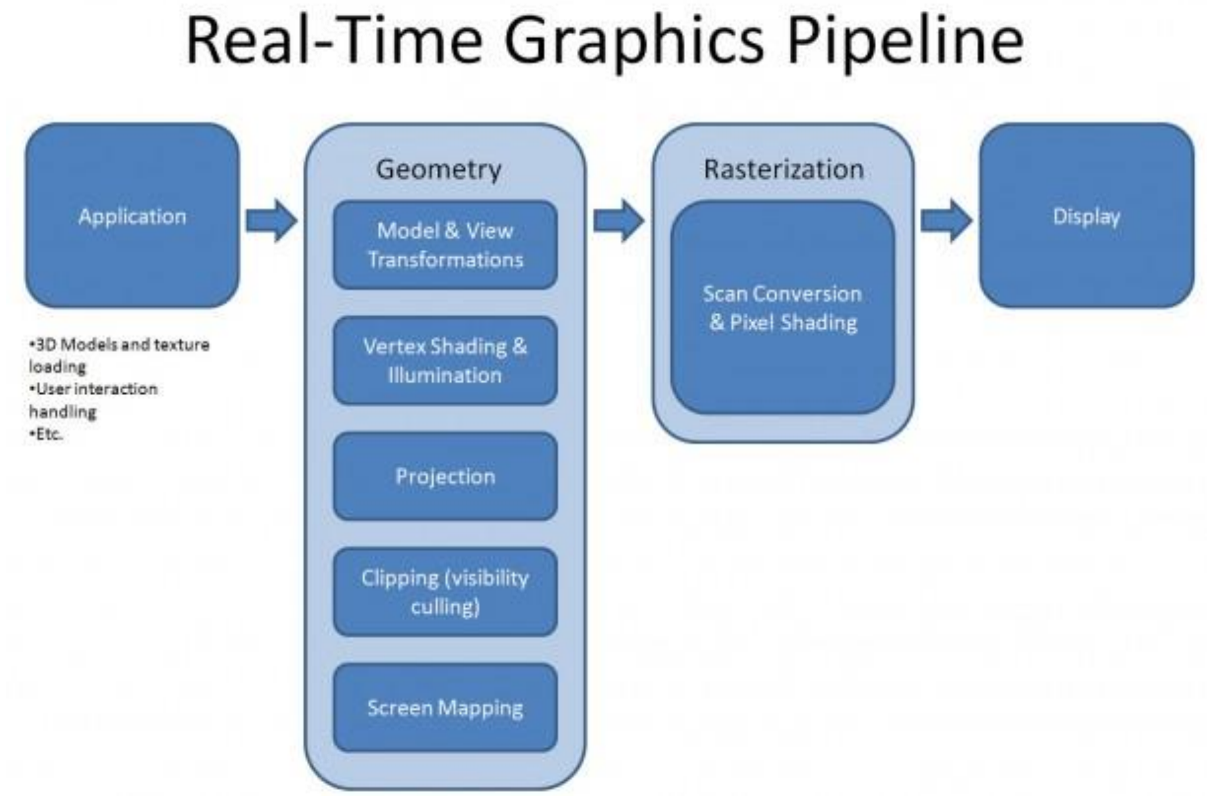


Fig. 3.10 Real-Time Graphics Pipeline

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: How do I get a 3D representation of my scene and turn it into a 2D representation? 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. I'll do my best to make it simple. 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 you take the picture.

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).

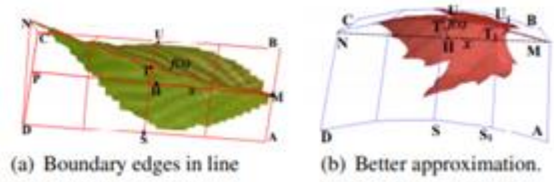


Fig. 3.10. Leaf modeling

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:

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

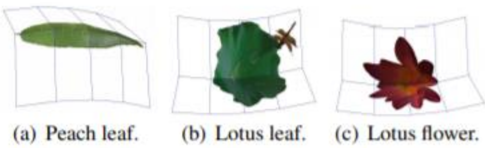


Fig. 3.11. A leaf model with different texture images.

The proposed 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.

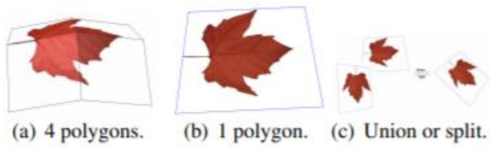


Fig. 3.12.Leaf models change base on quadric function

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.

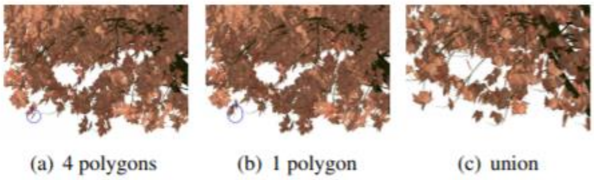


Fig. 3.13 Rendering results when a leaf represented with different polygons.

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.

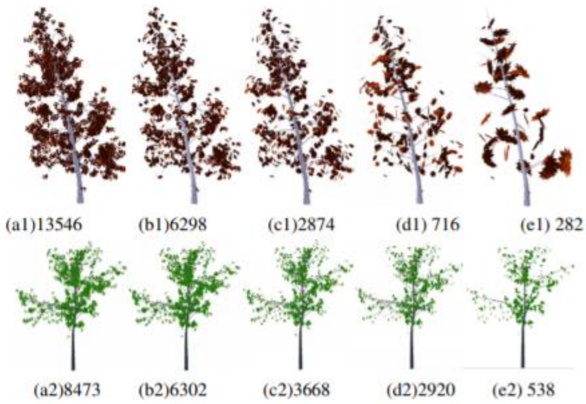


Fig. 3.14 LOD series of tree models

REAL-TIME FOREST RENDERING

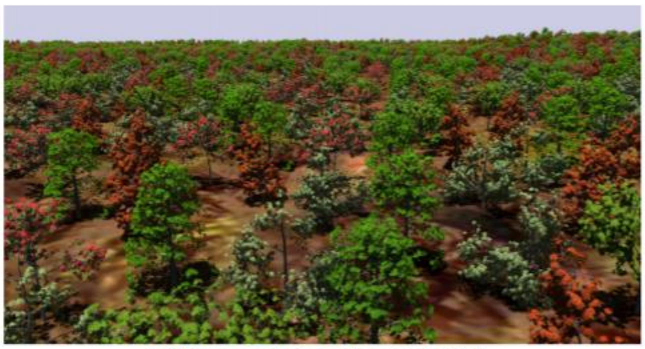
PSSM is employed to produce real-time and anti-aliasing shadows. Figure 3.15 shows the rendering results of different single trees used in our system.

(a) (b) (c)

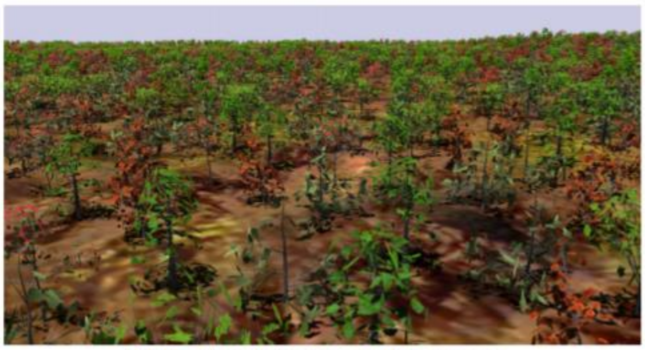
Fig. 3.15 Some single trees rendered with real-time and antialiasing shadows. (a) A simple tree. (b) A black popular with complex branching structure. (c) An apple tree with fruit and dense foliage.

Forest Scene Layout

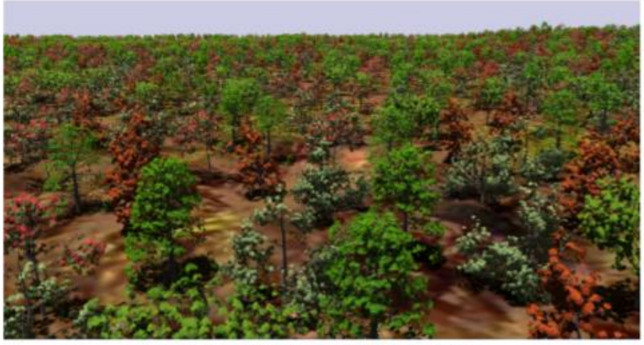
With some LOD tree models created with the method presented above, a large forest scene can be constructed as follows. A digital terrain model (DTM) covers an area of 262144 square meters. In the digital terrain model, there are 5 instances which produce 7446 tree positions with uniform distribution. 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. 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.

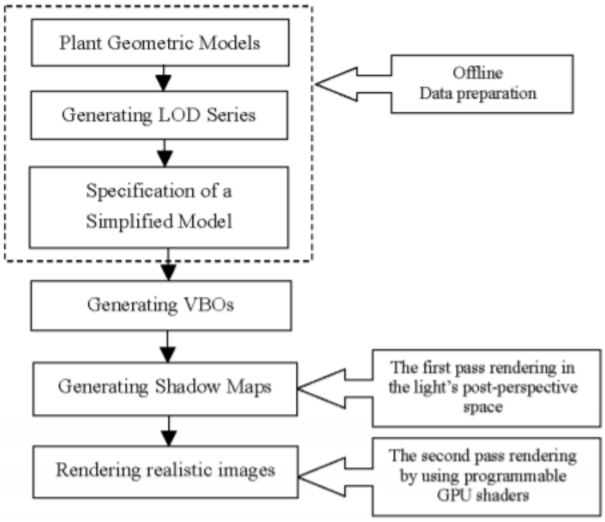


Fig. 3.17 Data processing flow chart

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."

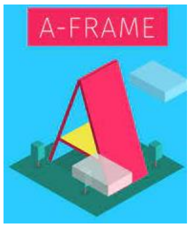
SketchUp

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

VIII. MultiGen

MultiGen Paradigm

- 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.

QUESTIONS

Part A

1. Identify the use of Computer Graphics in Virtual Reality
2. Quote the use of Stereoscopy in virtual reality
3. Interpret shutter system with respect to stereoscopy
4. How do you apply rendering in 3d models in real-time?
5. Differentiate between real-time rendering and offline-rendering.
6. Summarize the need to manage of Large Scale Environments?
7. Define VR Interaction Framework.
8. Connect CRYENGINE and Amazon Lumberyard.
9. Compare and Contrast Vega and Java 3D.
10. Infer the impact of virtual reality in human life in your own words.

Part B

1. Articulate how CRT, Raster Scan and Vector Scan used in creating an Virtual Reality Scenario.
2. Categorize the different stereoscopic technologies along with the different software used for it.
3. Design the steps involved in Real Time Rendering for Large-Scale Forest Scenes.
4. Identify and explain the list of tools in Virtual Reality.
5. Identify any visualization development toolkit for real-time simulation. Explain the same in detail.

School of Computing

Department of Computer Science and Engineering

UNIT - IV

AUGMENTED AND VIRTUAL REALITY – SCSA3019

UNIT IV INTRODUCTION OF AUGMENTED REALITY

System Structure of Augmented Reality-Key Technology in AR-- AR software development - AR software. Camera parameters and camera calibration. Marker-based augmented reality. Pattern recognition. AR Toolkit

Augmented Reality – Introduction

Augmented Reality (AR) is a general term for a collection of technologies used to blend computer generated information with the viewer's natural senses. A simple example of AR is using a spatial display (digital projector) to augment a real world object (a wall) for a presentation. Augmented reality technology was invented in 1968, with Ivan Sutherland's development of the first head-mounted display system. However, the term 'augmented reality' wasn't coined until 1990 by Boeing researcher Tim Caudell. The term "augmented reality," as well as the first true device of this kind, was created back in 1990 by Boeing researcher Tom Caudell and his colleague David Mizell. Augmented reality was first achieved, to some extent, by a cinematographer called Morton Heilig in 1957. He invented the Sensorama which delivered visuals, sounds, vibration and smell to the viewer.

Just two years later, Louis Rosenberg created Virtual Fixtures, the first AR system that was used by the U.S. Air Force. The device made use of a heads-up display (HUD) connected to two physical robot arms that the user could move through an upper-body exoskeleton that acted as a controller. The user saw the computerized robot arms in his visor, together with other computer-generated virtual overlays that simulated objects, barriers or guides existing in the real world. Today, in less than 30 years, AR technology has made a huge leap forward both in terms of performance and usability as well — so much that these clunky early models look like hilarious sweded movie cardboard equivalents of the modern devices! Augmented reality (AR) is a technology that lets people superimpose digital content (images, sounds, text) over a real-world environment. AR got a lot of attention in 2016 when the game Pokémon Go made it possible to interact with Pokémon superimposed on the world via a smartphone screen.

Augmented reality has been a hot topic in software development circles for a number of years, but it's getting renewed focus and attention with the release of products like Google Glass - Wearers communicate with the Internet via natural language voice commands Augmented reality is a technology that works on computer vision based recognition algorithms to augment sound, video, graphics and other sensor based inputs on real world objects using the camera of your device. It is a good way to render real world information and present it in an interactive way so that virtual elements become part of the real world. Augmented reality displays superimpose information in your field of view and can take you into a new world where the real and virtual worlds are tightly coupled. It is not just limited to desktop or mobile devices. Google Glass, a wearable computer with optical head-mounted display, is a perfect example. A simple augmented reality use case is: a user captures the image of a real-world object, and the underlying platform detects a marker, which triggers it to add a virtual object on top of the real-world image and displays on your camera screen.

Real-World Examples

AR applications can become the backbone of the education industry. Apps are being developed which embed text, images, and videos, as well as real–world curriculums. Printing and advertising industries are developing apps to display digital content on top of real world magazines. With help of AR, travelers can access real-time information of historical places just by pointing their camera viewfinder to subjects. AR is helpful in development of translation apps that can interpret text in other languages for user. Location based AR apps are major forms of AR apps. Users can access information about nearest places relative to current location. They can get information about places and choose based on user reviews. With the help of Unity 3d Engine, AR is being used to develop real-time 3D Games.

The Opportunity

It is estimated that 2.5 billion AR apps will be downloaded annually and will generate revenue of more than $1.5 billion by 2015. This is because AR apps will not be limited to conventional mobile apps. There will be new markets like Google Glass which will open more forms of development and use.

Development

To develop augmented reality apps … First - need to choose development tools. There are two major forms of augmented reality, marker- based AR and marker-less AR. A marker-based AR works on concept of target recognition. The target can be 3D object, text, image, QR Code or human-face called markers. After detection of the target by AR engine, you can embed the virtual object on it and display it on your camera screen. Qualcomm Vuforia SDK is our recommended framework to develop native apps. Marker-less AR, also known as location-based AR, uses GPS of mobile devices to record the device position and displays information relative to that location. Some of the examples of marker-less AR are apps like Layar and Wikitude that let you view information of nearby restaurants and other establishments.

Barriers - need to cross

Although going forward AR seems to have a huge potential market, there are some factors which could slow down mass adoption of augmented reality. Some of the factors are:

- Public Awareness and reach of Mobile AR
- Technological Limitations
- Addressing Privacy Issues
- Mobile Internet Connectivity in Emerging Markets

AR can be considered a technology between VR and telepresence. While in VR the environment is completely synthetic and in telepresence it is completely real, in AR the user sees the real world augmented with virtual objects. A telepresence robot is a remote-controlled, wheeled device that has wireless internet connectivity. Typically, the robot uses a tablet to provide video and audio capabilities. TelePresence robots are commonly used to stand in for tour guides, night watchmen, factory inspectors and healthcare consultants. Examples of telepresence include remote manipulation of probes in the deep sea, working with dangerous chemicals, controlling operations on a space probe, or even manipulating surgical instruments just a few feet away.

I. System Structure of Augmented Reality

Augmented reality is achieved through a variety of technological innovations; these can be implemented on their own or in conjunction with each other to create augmented reality. They include:

General hardware components are the processor, the display, the sensors and input devices. Typically a smartphone contains a processor, a display, accelerometers, GPS, camera, microphone etc. and contains all the hardware required to be an AR device. Displays – while a monitor is perfectly capable of displaying AR data there are other systems such as optical projection systems, head-mounted displays, eyeglasses, contact lenses, the HUD (heads up display), virtual retinal displays, EyeTap (a device which changes the rays of light captured from the environment and substitutes them with computer generated ones),Spatial Augmented Reality (SAR – which uses ordinary projection techniques as a substitute for a display of any kind) and handheld displays.

Sensors and input devices include – GPS, gyroscopes, accelerometers, compasses, RFID, wireless sensors, touch recognition, speech recognition, eye tracking and peripherals. Software – the majority of development for AR will be in developing software to take advantage of the hardware capabilities. There is already an Augmented Reality Markup Language (ARML) which is being used to standardize XML grammar for virtual reality. There are several software development kits (SDK) which also offer simple environments for AR development. ARML - is a data standard to describe and interact with AR scenes. ARML consists of XML grammar and ECMAScript (European Computer Manufacturers Association). XML grammar is used to describe the location and appearance of virtual objects in the scene. ECMAScript binding is used to allow dynamic access to the properties of the virtual objects, as well as event handling. ARML focuses on visual augmented reality (i.e. the camera of an AR-capable device serves as the main output for augmented reality scenarios). ECMAScript is a general-purpose programming language, standardised by Ecma International according to the document ECMA-262. It is a JavaScript standard meant to ensure the interoperability of web pages across different web browsers. ECMAScript is commonly used for client-side scripting on the World Wide Web, and it is increasingly being used for writing server applications and services using Node.js. Ecma International is a nonprofit standards organization for information and communication systems.

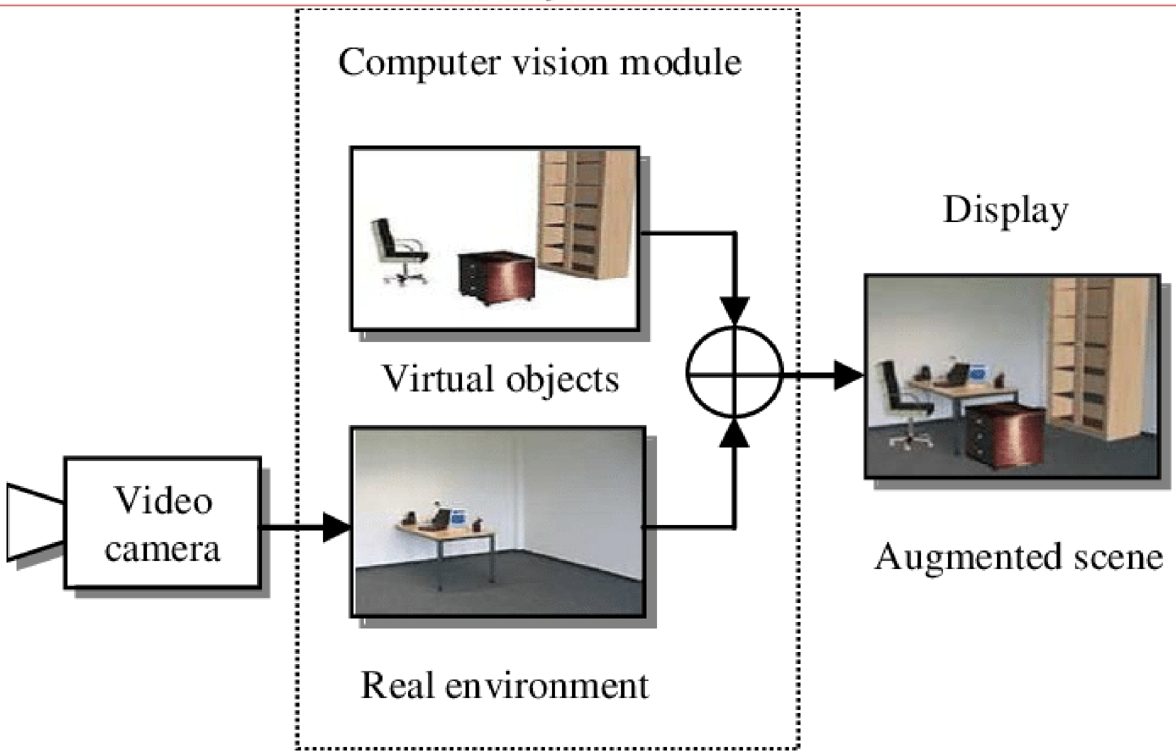


Fig.4.1 System Structure of AR

The blend of directed-perception (from physical world) and computer-mediated perception needs to be in the real-time - to provide a great AR experience, --- Sensors. Sensors connect the physical world to the computer-mediated box. It can be a camera, accelerometer, GPS, compass or microphone. Sensors make the first building-block of AR architecture. Sensors can be classified into two categories: (1) The one measuring a physical property of the environment that is not applicable to a human sense. e.g. Geo-location. (2) The ones capturing a physical property, directly detectable by human sensing capabilities. e.g. Camera. Some data of context analyzer and even some data of sensors is transferred to the brain of the system, the MAR Execution Engine. The main job of the engine is to verify if the conditions established by the AR designer and expressed in the MAR scene are met. It is a relatively complex part of AR architecture and has an orchestration role receiving messages from sensors, end-users, services, and rendering to the end-user. The results produced by the MAR execution Engine could be presented to the end-user by using specific displays, such as screens, loud-speakers, vibration devices. The user can also interact with the system using UI components.

User

AR technology is to provide artificial stimuli to cause the users to believe that something is occurring in the virtual world. Take Tesco store finder created by Junaio AR browser as an example: the purpose of this application is to provide the users with the awareness of Tesco's location, opening time, website and further information. Users are playing an important role in what takes place in AR architecture. Normally, AR architecture could happen in the mind of one user. But sometimes it could be two or many. Construct 3D - a geometric construction AR system aimed to promote students spatial capability due to the fact that they could view geometric entities in different sides. Two or many users could use an electric pen to modify all geometric entities, find the geometric relationship and work on the construction together. The concept of collaborative augmented reality is based upon two or many AR users. AR tennis is another application engaged with two users: two players could sit across the table from one another and hold their phone to view a virtual court overlaying the real world background and play the tennis game. Many AR researches focus on the normal users without particular group of people. Some of research designed AR application for children with autism and cognitive disability. 'user' - refer to an individual who manipulates and controls, the immediate intended beneficiary of an AR system. For example, doctors could use augmented reality as a visualization aid and possibly collect 3-D dataset of a patient in real time during the surgery. AR system brings the benefits for both doctors and patient. However, the AR user should be the surgeon who watches and controls the AR system rather the patient.

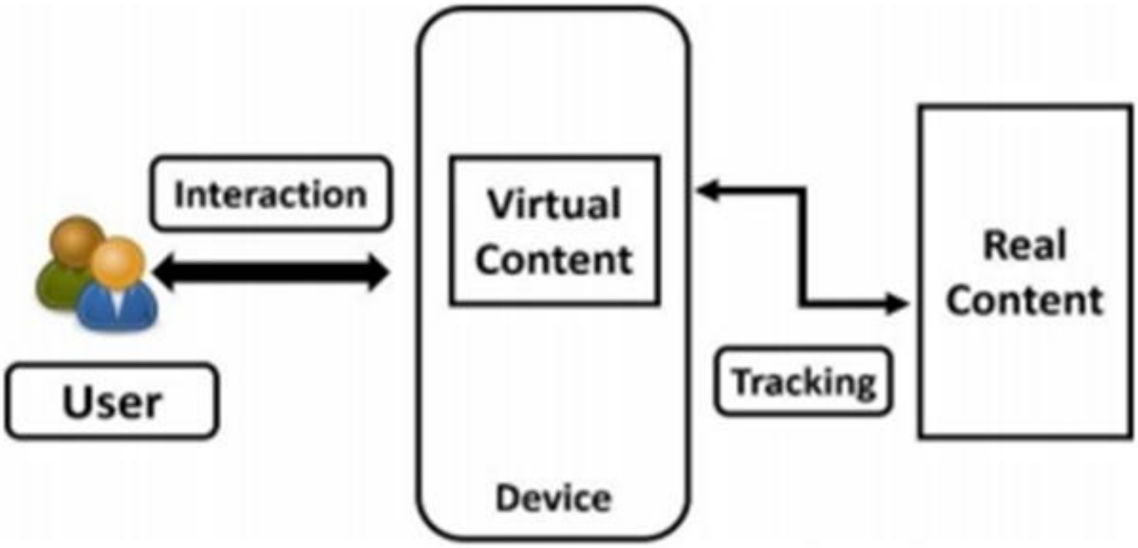


Fig. 4.2 Augmented Reality Architecture comprised by six elements

Interaction

Interaction is composed of two components including inter- and action. Inter means the state between or among things. Action means there is an influence and something that has been done. Therefore, interaction can be simply speaking that one entity does something and the other entity responds in some way. In the user-based augmented reality (AR) system, the process of interaction in this research mainly concentrates on a trigger caused by users and the response of AR system which can occur between users and AR device or users and virtual content. For example, if people try to use Tesco store finder by Junaio AR browser to find the location of the nearest Tesco, they have to launch a Tesco App or channel first. And then, people may adjust the position of the mobile device (e.g. Iphone or Ipad) to see the overlaid virtual bubble. The action of adjusting the AR device's physical position can be described as the interaction between users and AR device interaction. This action will end in a response of identifying the virtual target store on the device. Then, if people want to get more information about the particular Tesco store (e.g. opening time, distance or phone number), they need to click a virtual icon that visually indicates the store's information. After that, some more overlaid information will be presented in another pop-up slide. The action of clicking the virtual Tesco bubble means users and virtual content interaction. The response of new pop-up page implies the process of interaction has been done. However, in some of the particular AR scenario, the process of interaction between users and AR device or users and virtual content doesn't exist anymore. National geographic demonstrated a spectacular view of national geographic content on a large screen that visitors could see themselves along with the augmentations in their world. Users do not interact with the augmented reality device (the big screen) and the virtual national geographic content in this scenario.

Device

The term of device means the carrier or object could acquire physical world information and provide the compelling augmentation. It could be the mobile device, desktop computer, and big screen with projector or etc. three hardware functions in all AR device - sensors, processors and displays. Sensors recognize the state of the physical world which the AR system needs to be deployed. For example, a camera, one of the most popular AR sensors, could capture the physical world image and provide information to the AR users. GPS or other compass system aims to identify the location and orientation of user. A processor processes the sensors' information and generates the signals required to drive the display. Very often, the AR system will rely upon not just the processor on the device, but the processor on the server as well. A display will show the coexistence that users could sense the combination of physical world and virtual world. Based on these requirements of functions, the smart mobile phone or tablet seems to be the appropriate AR device compromising by a camera to capture, processors to process and a screen to display. The mobile device held by one hand could run different applications, which is moveable, easy to use and accessible from anywhere and anyplace.

Virtual content

Virtual content means the digital information presented by AR device, which plays the most important role in the AR architecture. The modality of virtual content could be 3D animation, 2D image, text, website, audio information or even vibration. AR users will not concern too much on devices, but will be attracted by different virtual content. Participants often strongly express curiosity to what a digital device could provide but rarely if ever affection to the device itself. A key feature of virtual content is that the virtual information can be changed dynamically. Going back to the Tesco example, when users use the app and move around, the virtual content (Tesco bubble) will pop up automatically. An icon visually indicating a real-world store can be clicked and more Store information will be presented. This additional information like the opening time website or the video instruction replaces the previous virtual bubble. The content of virtual information has been changed easily and completely.

Examples

Scanning a QR code using phone's camera provides additional information (so, AR) on screen. Google Glass and other head-up displays (HUD) like Vuzix Waveguide Lens put Augmented Reality directly into the glasses. These glasses could be used as reminders for patients undergoing medication. Real time battlefield data could be available to soldiers wearing these. We are also familiar with the various filters on Snapchat and Instagram, an aspect of AR. In the Netherlands, an application called Layar is available for download, which uses the phone's camera and GPS capabilities to gather information about our surroundings. We could point at a building and enquire about its history, whether it's on sale and more. AR Defender 2 is a mobile game for iOS users that helps you attain amazing experience, by turning any real world area into a virtual battlefield. Niantic has already achieved a lot by developing Ingress and Pokémon Go. Apps like Augment, is helpful for designers that allows users to upload 3D models and visualize them in a physical space.

Real content

Real content is the real-world information directly presented by device without any rendering, which includes geographic location, physical objects and real-world environment. Taking Tesco app example again, AR devices not only display the virtual Tesco bubble (virtual content), but present the users' location information surrounding the real-time environment, which means real content. However, while users look through the AR device, the real content will be more or less hidden. For example, Word lens AR translator generates the virtually translated words, which replace the real-world words. Users have to remove the AR device if they need to see the original words.
They cannot see both the virtual and real content simultaneously. Obstruction of real content is the intrinsic risk of augmented reality.

Tracking

Tracking describes the way of generating virtual content based on the real content, comprising three different features: synchronicity, antecedent and partial one to one. Due to the changes of the real content, an AR virtual counterpart has to be updated synchronously. For example, Word lens is an AR translation application that scans foreign text and displays the test translated in real time. Once the user changes his or her point of view to another word, the displayed translation on the device rapidly changes in the same time. If the process of generating virtual information is delayed for a long time, viewers are unable to obtain the useful information. The feature of antecedent means the real content (physical text) exists or happens before the virtual content (the translated digital word). If the virtual content is created before the real-world content, the virtual element is meaningless because it has no real world interpretation. Partial one to one describes another tracking feature of augmented reality. There is one and only one real content to correspond with the virtual content. However, there might be one or more than one piece of virtual information to correspond with the real-world content. That means the real content can be superimposed to different modality of virtual content. AR users could be one or many who might have interacting with the device or virtual content by adjusting or clicking. Virtual content is the additionally computer-generated information displayed on the AR device via an array of tracking transformation based on real-world counterpart. The AR architecture could bring benefits to AR designers and provide a more explicit basis on which to articulate AR criteria, classification and function.

Why Augmented Reality is Important

Development of AR technology is set to revolutionize industries from retail to military to education to tourism and transform the way that is been interacted with the digital world every day. Augmented reality has many uses in different fields like in archaeology, architecture, visual arts, commerce, education, video games and military training etc. Some applications of AR are - AR is being used to aid research in archaeology. AR can be used to recreate different structures and overlay them on the real environment so that researchers can study it correctly.

Importance

AR applications in smartphones include Global Positioning System (GPS) to locate the person's location and its phone's inbuilt compass to find device orientation. Augmented reality can be used in the field of tourism to enrich visitor's experience during visits like the Eiffel tower has an AR app that can show - throughout history when it was being built. and the list goes on. that's why AR and VR companies have raised more than $3 billion in 2017 in funding, thus 2018 has been doubled the year when AR goes mainstream, $45 billion globally 2019. The augmented reality (AR) and virtual reality (VR) market is set to grow by USD 162.71 billion, progressing at a CAGR of almost 46% during 2021-2025. It is sure that in the coming years it will change the way that technology is being looked and improve the integration of technology in our daily lives.

II. Key Technology in Augmented Reality

The Augmented Reality Technology is an important branch of Virtual Reality Technology. On the basis of virtual reality technology, augmented reality technology uses computer graphics technology and visualization technology to superimpose virtual images generated by computer operations to real pictures. Various technologies are used in augmented reality rendering, including optical projection systems, monitors, handheld devices, and display systems, which are worn on the human body. A head-mounted display (HMD) is a display device worn on the forehead, such as a harness or helmet-mounted.

- Intelligent display technology,
- 3d registration technology and
- intelligent interaction technology

constitutes the core technology circle of AR and play an important role in the development of AR.

Intelligent display technology

According to relevant data, more than 65% of the information acquired by human beings comes from their own vision, which has become the most intuitive way for human beings to interact with the real environment. With the development of intelligent display technology, augmented reality becomes a possibility, which is pushed to a new height by the various kinds of display devices generated based on intelligent display technology. Specifically, there are three main categories of display devices that occupy an important position in the field of AR technology today. First, helmet display (HMD *head mounted display*) was born in 1968. The optical perspective helmet display developed by Professor Ivan Sutherland makes it possible to superimpose simple graphics constructed by computers on real scenes in real time. In the later development, optical perspective helmet-mounted display and video perspective helmet-mounted display constitute the backbone of helmet-mounted display. Second, handheld device display, relying on the augmented reality technology of handheld display, handheld device display is very light, small, especially the popularity of smart phones, through video perspective to the use of augmented reality technology to present. Third, other display devices, such as PC desktop displays, match the real-world scene information captured by the camera to a three-dimensional virtual model generated by the computer and are ultimately displayed by the desktop display.

3D registration technology

As one of the most critical technologies in the augmented reality system, 3d registration technology enables virtual images to be superimposed accurately in the real environment. The main flow of 3d registration technology has two steps. First, determine the relationship between the virtual image, the model and the direction and position information of the camera or display device. Second, the virtual rendered image and model are accurately projected into the real environment, so the virtual image and model can be merged with the real environment. There are various ways of 3d registration, such as

- the registration technology based on hardware tracker,
- the 3d registration technology based on computer vision,
- the 3d registration technology based on wireless network and
- the mixed registration technology, among which the former two are the most popular.

For the three-dimensional registration technology based on computer vision, it sets the reference point to realize the determination of the direction and position of the real scene by the camera or the display.

Intelligent interaction technology

Intelligent interactive technology is closely related to intelligent display technology, 3d registration technology, ergonomics, cognitive psychology and other disciplines. In AR systems, there are a variety of intelligent interactions, including hardware device interactions, location interactions, tag-based or other information-based interactions. With the development of intelligent interaction technology, augmented reality not only superimposes virtual information to real scenes, but also realizes the interaction between people and virtual objects in real scenes. This interaction is based on the fact that people give specific instructions to the virtual object in the scene, and the virtual object can make some feedback, thus enabling the audience of the augmented reality application to achieve a better experience.

III. AR software development and AR software

AR Software

AR software works in conjunction with devices such as tablets, phones, headsets, and more. These integrating devices contain sensors, digital projectors, and the appropriate software that enables these computer-generated objects to be projected into the real world. Once a model has been superimposed in the real world, users can interact with it and manipulate the model. These solutions have additional uses aside from placing a 3D model into the real world. AR is commonly used for entertainment purposes—specifically gaming. This software can also be used to display contextual information. Users can point the hardware's camera display at an object to display valuable data.

Why Use AR Software?

As AR is still a young technology, it provides certain advantages to businesses that other software cannot offer. The following are just a few of the benefits to use AR software in the business.

Product view – AR technology allows potential customers to view and interact with the product or service before purchasing. This can enable them to make better-informed decisions.

Enhance content – AR technology allows users to embed various types of data onto content. People can point their device at a real-life object to learn whatever kind of information is necessary, instead of needing to search for it elsewhere.

Training – AR solutions enable users to train employees more thoroughly than they can get through documentation and meetings. This software allows for trainees to learn job responsibilities by fully visualizing them, instead of just reading about job duties.

Productivity – This software enables users to improve workflow and processes at their business. This is particularly true for manufacturing-based organizations. Factory line workers can spot potential dangers quicker, along with accessing necessary resources.

Engage your audience – Consumers are inundated with print and television advertisements for various products and services, to the point where they don't pay much attention to them. Inserting augmented reality into advertisements will catch the eye of your target demographic.

Who Uses AR Software?

AR software can be utilized by users in a number of different fields, such as:

Retail – Users in the retail industry can leverage AR technology so consumers can virtually test out products before they make a purchase. For example, AR retail applications allow users to upload a photo of themselves and visualize what a particular piece of clothing would look like on their body. Shoppers could also use these kinds of applications to visualize what a piece of furniture would look like in their house.

Education – AR technology is being increasingly used in the classroom to supplement lessons. For example, if a teacher was doing a lesson on astronomy, AR software could project a map of the solar system so students could visualize what they are learning about.

Repair and maintenance – Employees performing manual labor can wear AR glasses to help with repair and maintenance jobs. AR software can be used to project valuable data and inform the user where a certain part is supposed to go.

Medical – Doctors, particularly surgeons, can use AR technology for training purposes. All the documentation and videos out there are not realistic enough to prepare a surgeon for what surgery is really like. AR technology can help trainee surgeons visualize what the actual act of surgery would be like.

Kinds of AR Software

However, the following are some of the main types of AR software on the market now:

AR visualization software – This type of software enables organizations to create immersive experiences for consumers to interact with. AR visualization software users can upload 3D content and scale the image, adjust the color, and incorporate the additional details needed to give the best user experience possible.

AR content management system (CMS) – An AR CMS lets users bulk upload raw 3D content that will eventually become the basis for AR experiences. This content can be managed and edited within the platform.

AR SDK – These tools allow users to build digital objects that will blend into the real world that will eventually become fully fledged AR experiences.

AR WYSIWYG editor software – This software enables users with limited to no coding background to create customized AR experiences. These tools have drag-and-drop capabilities that let users upload 3D objects and drop them directly into previously designed scenes.

AR game engine software – These solutions give game developers the framework for creating AR video game experiences. Using AR game engine software, users can create and edit 3D characters that can interact with the real world.

AR training simulator software – AR training simulator software leverages AR technology to train employees for certain jobs.

Industrial AR platforms – These solutions are typically used by organizations in the industrial field. These tools include interactive AR content that improves these employees' productivity, effectiveness, and safety.

AR Software Features

Content management – Many AR solutions, regardless of the specific category they fall into, provide users with the ability to store and manage their content. This can range from raw 3D content that will serve as the basis of an AR experience to content that has already been designed.

Editing – AR solutions should allow for users to edit the 3D model they upload into the platform. Users can scale the image, adjust the color, and incorporate any additional details needed.

Hardware integration – In order to provide the intended AR experience for a consumer, the software must integrate with devices that support AR software. This includes glasses, Android and Apple mobile phones, and tablets.

Drag-and-drop – Some AR development solutions are designed to be user-friendly for those with little to no coding experience. Tools like this offer a WYSIWYG editor, which allows users to upload 3D objects and insert them into previously designed scenes so that they eventually become AR experiences.

Additional AR Features

Analytics – Some AR tools, such as products in the AR visualization software space, will provide analytics capabilities for users. This lets businesses see how consumers interact with the 3D object within AR mobile applications, which should be supported on both Apple and Android devices.

Upload content – AR software products allow businesses to upload 3D content necessary for their specific business purposes. This is particularly relevant for AR training simulators, as businesses need to ensure the software will support the content needed for trainees to learn the job at hand.

Trends Related to AR Software

AR advertising — Various brands are beginning to introduce augmented reality into their promotions. AR can enhance a user's experience with your brand. Entertainment companies will likely avail themselves of this technology, so they can bring various elements of a show or movie to life.

Health care — Not only can AR technology help to train surgeons, but it can assist them once they are already well-versed in their work. Some surgeons have already used AR while operating on human hearts, so they can visually see the clogged vessels they are working on.

AR will likely continue to grow in the healthcare field, as it can help caregivers make the best-informed decisions in life-or-death situations.

Android and Apple mobile sales — Smart phones are among the devices that can support AR technology. As AR software becomes more and more common in the marketplace, mobile phone manufacturers will likely begin to compete to build the phone best equipped to support AR. Android and Apple phones will likely go head-to-head with each other.

Wearable AR — Developers have begun to set their sights on wearable AR technology, specifically glasses. And as this continues to grow, developers are working to make these glasses more ergonomic. This technology is anticipated to become smaller, more form-fitting, and better attuned to human senses.

Opening field of view — As AR glasses are on the rise, developers are also working to open up the field of view. Most glasses limit the field of view to about 45–50 degrees, compared to the human eye's 120 degree field of view. AR developers are working to close that gap.

Potential Issues with AR Software

Cost

One of the biggest factors that has hindered AR from becoming mainstream is the cost. It can be very expensive to purchase the hardware to support AR technology. Streaming the content is also very costly. Content for these solutions needs to be streamed in a very high resolution and rendered at a high refresh rate. This content also requires a large bandwidth for streaming. All these factors add up, making consumers wary of adopting AR.

Accessibility and education

Due to the cost, AR technology is not too accessible to the masses. Since very few people are exposed to this software, it is hard for them to conceptualize the wide-ranging uses that AR can offer. Unless developers change the user experience and the messaging around this technology, it will be difficult to get past this hurdle. There are two broad classes of AR apps: - marker-based apps and location-based apps. Marker-based apps use predefined markers to trigger the display of AR overlays on top of the image. Location-based apps use GPS, accelerometer, or compass information to display AR objects on top of physical ones.

- To choose an AR SDK, the most important criteria to consider are:
  - cost,
  - supported platforms,
  - image recognition and tracking support,
  - Unity support,
  - OpenSceneGraph support,
  - GPS, etc.

Augmented reality (AR) has become the new trend in the digital world and can hardly meet a person who is not familiar with it after the boom that Pokemon Go brought into the lives of the average mobile user. Though many people consider AR to be only an entertainment technology, it's actually widely used in multiple industries like healthcare, e-commerce, architecture and many others. The potential of AR is seamless and brands are already utilizing this technology in their business to provide a brand new user experience. Companies implement AR to create product demos, interactive advertising and provide real-time information to customers. It was proved that when people touch or interact with a product, they are more likely to buy it due to the emotional bond established. According to a Statista forecast, the market of augmented and virtual reality is expected to reach the size of

$215 billion in 2021 end. Being a rapidly growing market with huge potential, AR attracts both huge corporations like Google, Apple, Facebook, etc., as well as smaller businesses.

AR Software Development

There are many types of Augmented Reality applications exist. Before starting the development of a augmented reality app - choose between two broad categories: location apps and marker-based apps.

Marker-based applications

Marker-based apps are based on image recognition. They use black and white markers as triggers to display AR content. To see the augmented component, you have to point the camera on a marker's position anywhere around you. Once the device recognizes the marker, an app overlays the digital data on this marker and you can see the augmented object. When building a marker-based app, the images or their descriptors should be provided beforehand to simplify the process of searching them when the camera data is being analyzed. In other words, the objects are already hard-coded in your app, so they are easier to detect. It's no wonder that the majority of AR apps are marker-based. They are especially popular in advertising.

Location-based applications

Location-based AR apps work without markers. They detect the user's position with the help of a GPS, an accelerometer, or a digital compass and overlay the augmented reality objects on top of real physical places. The most famous location-based app is surely Pokemon Go. These apps can send notifications to the user based on their location to provide them new AR content related to a given place. For example, an app could give recommendations about the best restaurants nearby, and show how to get there. As an additional example, an app could help you find your car inside a huge parking using GPS.Main Criteria to Choose an Augmented Reality SDKs. When it comes to choosing a development kit, it's easy to get frustrated by the number of tools available. In order to pick the SDK that best suits the project, make sure it supports all the features - app requires.

Main points to consider -

Cost

Pricing is the first distinguishing mark of an AR SDK. For those who want to try AR development for the first time, the best options are free open-source AR SDKs, which are open to contributions and can be extended with new features proposed by developers. Paid SDKs in most cases offer several pricing plans, depending on the user's needs. As it happens, free tiers have limited possibilities and are meant to be a "demo version" of the full product. Building a complex app with large, dynamic content will likely require a commercial license.

Platforms

If the plan is to develop app for iOS or Android, there won't be any problems when choosing an augmented reality toolkit, since nearly all of them support them. Meanwhile, the choice of tools that are compatible with Windows or macOS is rather small. Still, - can build your app for Windows computers or smart phones using augmented reality development kit, supporting the Universal Windows Platform (UWP).

Image recognition

This feature is a must-have for any AR app as it allows to identify objects, places and images. To this aim, smart phones and other devices use machine vision together with camera and artificial intelligence software to track images that can be later overlayed with animations, sound, HTML content etc.

3D recognition and tracking

3D image recognition and tracking is one of the most valuable features of any AR SDK. Due to the tracking, an app can "understand" and enhance the large spaces around the user inside of large buildings such as airports, bus stations, shopping malls, etc. Applications supporting it can recognize three-dimensional objects like boxes, cups, cylinders, toys etc. Currently, this technology is commonly used in mobile games and e-commerce.

Unity support

Unity is known to be the most popular and powerful game engine worldwide. Though it's usually used for developing computer games, it can also be utilized for making AR apps with powerful effects. Whether you are going to create a cutting-edge experience or extend a more traditional idea with new techniques, multipurpose tool like Unity allows you to implement both.

OpenSceneGraph support

OpenSceneGraph is an open source 3D graphic toolkit (application Programming interface). It's used by app developers in such domains as computer games, augmented and virtual reality, scientific visualization and modeling.

Cloud support vs local storage

When developing AR mobile applications, you have to decide whether user data will be stored locally or in the cloud. This decision is mostly driven by the number of markers you are going to create. If the plan is to add a large number of markers to the app, consider storing all this data in the cloud, otherwise the app will use much storage on the device. Furthermore, having an idea of the number of markers the app uses also matters because some augmented reality SDKs support a hundred markers while others support thousands. On the other hand, storing markers locally (i.e., on-device) enables users to run the augmented reality app offline, which could be convenient as Wi-Fi or mobile-data is not available.

GPS support (geolocation)

If the aim is to create a location-based AR application, geolocation is a fundamental feature that must be supported by the AR tool that is used. GPS can be used both in AR games like Pokemon Go as well as in apps made to overlay data on some nearby locations (for example to find the nearest restaurant).

SLAM support

SLAM means Simultaneous Localization and Mapping. It is an algorithm that maps the environment where the user is located and tracks all of their movements. AR apps containing this feature can remember the position of physical objects within some environment and position virtual objects accordingly to their position and users movements. SLAM has huge potential and can be used in many kinds of apps, not only AR apps. The main advantage of this technology is the ability to be used indoors while GPS is only available outdoors.

Augmented Reality SDK for Mobile Apps

- To create augmented reality app, list of popular tools are available on the market. These toolkits are considered to be the most relevant and appropriate based on the set of features they provide and their value for money. Some of them are free.

- Vuforia – best for Marker-based apps
- ARToolKit - best for Location-based apps
- Google ARCore - best for Marker-based apps
- Apple ARKit - best for Marker-based apps
- Maxst - best for Marker-based apps
- Wikitude - best for Marker-based apps

- Vuforia is a leading portal for augmented reality application development that has a broad set of features.

  Vuforia augmented reality SDK:
- Recognizes multiple objects including boxes, cylinders, and toys as well as mages.
- Supports text recognition including about 100,000 words or a custom vocabulary.
- Allows creating customized VuMarks, which look better than a typical QR-code.
- Allows creating a 3D geometric map of any environment using its Smart terrain feature
- Turns static images into full motion video that can be played directly on a target surface.
- Provides a Unity Plugin.
- Supports both Cloud and local storage.
- Supported platforms: iOS, Android, Universal Windows Platform, Unity. Pricing: free version, classic version - $499 one time, cloud - $99 per month and Pro version for commercial use.

- ARToolkit is an open-source tool to create augmented reality applications.

  Even though it's a free library, it provides a rather rich set of features for tracking, including:
- Unity3D and OpenSceneGraph Support.
- Supports both single and dual camera.
- GPS and compasses support for creation of location-based AR apps.

- Possibility to create real-time AR applications.
- Integration with smart glasses.
- Multiple Languages Supported
- Automatic camera calibration.
- Supported platforms: Android, iOS, Linux, Windows, Mac OS and Smart Glasses.
- Pricing: free

Google ARCore

- With two millions Android active users, Google could not miss the chance to give developers an opportunity to create AR apps on this operating system. That's how Google ARCore appeared.
- This toolkit works with Java/OpenGL, Unity, and Unreal.
- It provides features such as:
- Motion tracking - ARCore can determine the position and orientation of the device using the camera and spot the feature points in the room. That helps to place virtual objects accurately.
- Environmental understanding - Due to the possibility of detecting horizontal surfaces, virtual objects can be placed on tables or on the floor. This feature can be also used for motion tracking.
- Light estimation - This technology allows the app to match the lighting of the environment and to light virtual objects so they look natural within the surrounding space. With the help of smart light tracking developers can now create very realistic objects.
- Supporting devices: Currently: Google Pixel, Pixel XL, Pixel 2, Pixel 2 XL, Samsung Galaxy S7-S8+, Samsung A5-A8, Samsung Note8, Asus Zenfone AR, Huawei P20, OnePlus 5 ARCore is designed to work on devices running Android 7.0 and higher.
- Pricing: free

- With iOS11, Apple introduced its own ARKit, announced during Apple's Worldwide Developers Conference in June 2017.

- Here are the features of Apple's augmented reality SDK for iOS:
- Visual Inertial Odometry (VIO) allowing to track environment accurately without any additional calibration.
- Robust face tracking to easily apply face effects or create facial expressions of 3D characters.
- Tracking the light level of environment to apply the correct amount of lighting to virtual objects.
- Detecting horizontal planes like tables and floors, vertical and irregularly shaped surfaces.
- Detecting 2D objects and allows developers to interact with them.
- Integration with third-party tools like Unity and Unreal Engine.
- Devices: iPhone 6s and 6s Plus, iPhone 7 and 7 Plus, iPhone SE, iPad Pro (9.7, 10.5 or 12.9) – both first-gen and 2nd-gen, iPad (2017),iPhone 8 and 8 Plus, iPhone X
- Pricing: free

- MAXST has two SDKs available: a 2D SDK for image tracking and a 3D SDK for environment recognition.
- Here is the list of features of the 3D SDK:
- MAXST Visual Simultaneous Localization and Mapping for tracking and mapping environments. When you track the surroundings , the map is automatically extended beyond the first view along with the move of the camera. Maps can be also saved for the later uses.
- Saving files created with Visual Simultaneous Localization and Mapping to render 3D objects wherever you like on it to create more immersive AR experiences.
- QR and barcode scanning.
- Extended image tracking and Multi-target tracking - can track the target as far as the camera can see it and can also track up to 3 images at the same time.
- Tracking and placing digital objects in relation to the plane.
- Unity plugin integration.
- Supported platforms: Android, iOS, Mac OS and Windows.
- Pricing: free version, Pro-One time fee - $499, Pro-Subscription - $599 per year, Enterprise version.

wikitude

- Wikitude has recently introduced its SDK7, including support for simultaneous localization and Mapping.
- The tool provides currently the following features:
- 3D recognition and tracking.
- Image recognition and tracking.
- Cloud recognition (allows to work with thousands of target images hosted in the cloud).
- Location-based services.
- Smart glasses integration.
- Integration with external plugins, including Unity.
- Supported platforms: Android, iOS, Smart Glasses (currently Google Glass, The Epson Moverio BT-200, and the Vuzix M100).
- Pricing: Pro version - €2490 per year per app, Pro3D - €2990 per year per app, Cloud - €4490 per year per app, Enterprise version.
- Needless to say, augmented reality technology is trendy. Each new AR app launch causes waves of excitement.
- Therefore, savvy developers are trying to master this technology and launch their own AR apps.
- Now, developers have a wide choice of AR toolkits to create both marker-based and location-based apps.
- The first step to get started is picking up the augmented reality SDK most suited to comply with their requirements.
- Then compare features such as image and 3D recognition, storage possibilities, Unity and SLAM support, etc., for development teams to easily select the best toolkit for their future apps.

IV. Camera Parameters and Camera Calibration

Augmented Reality is used for a wide range of applications in computer vision such as computer-aided surgery, repair of complex machines, establishment modifications, interior or structural design. In Augmented Reality applications the user's view of the real world is enhanced by virtual information. This additional information is created by a computer which has a model of the real world and the model of some real world objects in which the user is located. These real objects are tracked, so the computer knows the location and rotation of them. The Augmented Reality system imagines a virtual camera in the virtual world which can see a range of virtual objects corresponding to real world objects. The additional virtual data is superimposed over these real objects. This visual enhancement can either have the form of labels, 3D rendered models, or even shading modifications. With the help of optical see through displays the user can see both the virtual computer-generated world on the screen and the real world behind it. In general these are displayed on an see through head mounted display (HMD) to get an Augmented Reality view.

These optical see through devices present a special challenge because the system has no access to the real world image data as at a video see through device. So the HMD represents a virtual camera for which several parameters must be accurately specified as well.

Necessary tasks needed for calibrating virtual cameras

- A virtual camera defines a projection of the virtual 3D world to the 2D image plane.
- As shown in figure 4.3 the user sees the computer generated 2D image appearing in his HMD about one meter in front of his face.
- The virtual world objects are registered in 3D.
- In order to see the right objects at the desired positions the virtual camera must provide the correct projection.
- Finding this projection is called camera calibration.
- Once the projection is found the viewing component of the Augmented Reality system uses it to represent the virtual world.

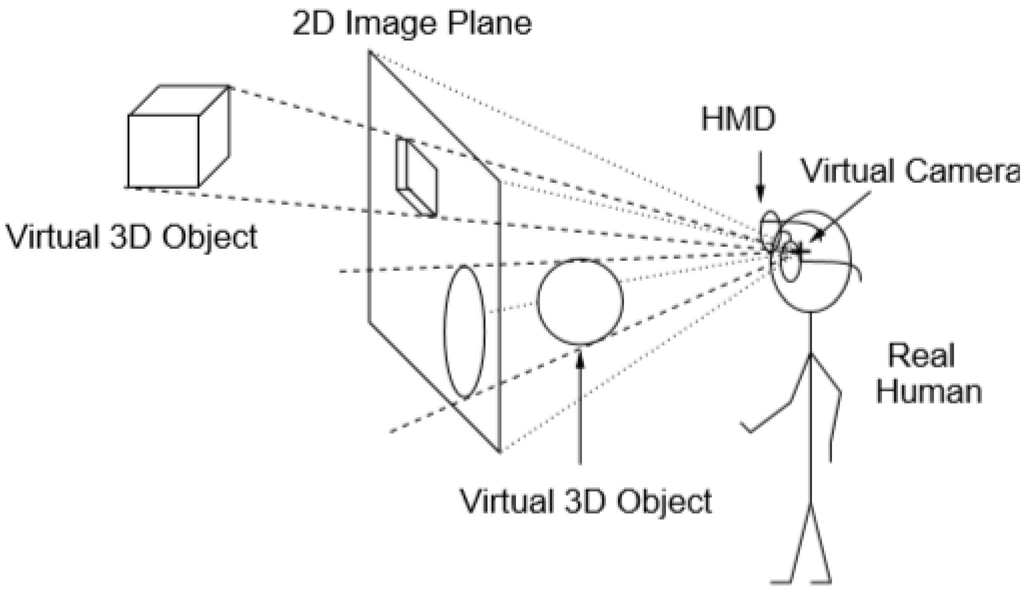


Fig. 4.3 The 3D virtual world is mapped to the 2D image plane

Virtual Camera / Camera Model

A camera maps the 3D virtual world to a 2D image. This mapping can be represented by a 3×4 projection matrix P. Each point gets mapped from the homogeneous coordinates of the 3D virtual world model to homogeneous coordinates of its image point on the image plane. In general this matrix has 11 degrees of freedom (3 degrees of freedom for the rotation, 3 more for the translation, and 5 from the calibration matrix K) and can be split up into two matrices P = KT. Whereas the matrix K holds the internal camera parameters, such as the focal length and aspect ratio. T is a simple transformations matrix which holds the external camera parameters that are the rotation and the translation. A finite camera is a camera whose center is not at infinity. Let e.g. the center be the origin of an Euclidean coordinate system, and the projection plane z = f. It is also called the image plane or focal plane. The distance f is called the focal length. The line from the optical camera center vertical to the image plane is called the principal axis or principal ray of the camera. The point where this line intersects the image plane is called the principal point or the image center. Furthermore the plane through the camera center parallel to the image plane is called the principal plane. Assume that the Augmented Reality system already knows the position Tmarker and the orientation Rmarker of the tracked HMD marker represented as the transformation F in figure 4.5.

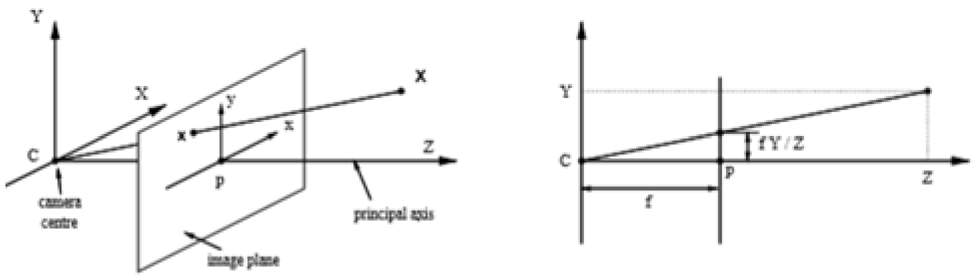
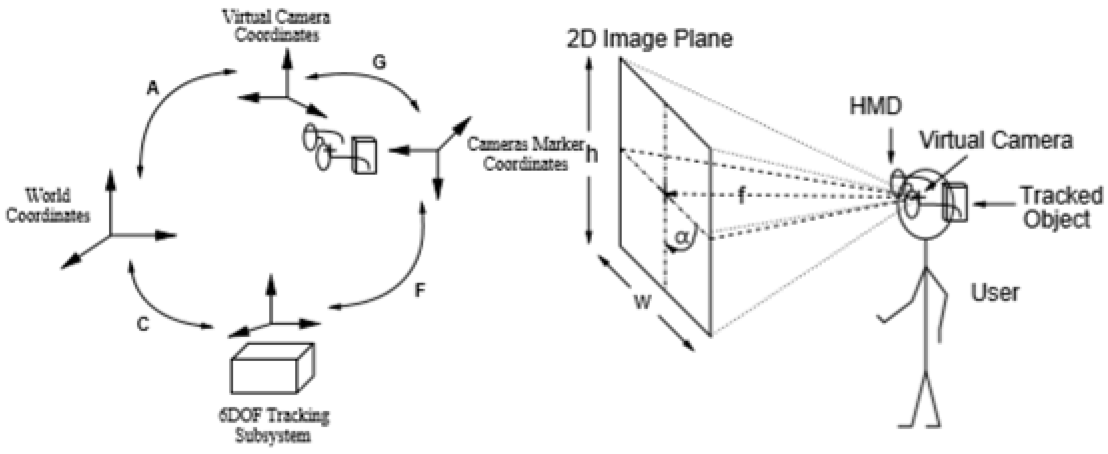


Fig. 4.4 Camera Geometry



(a) Extrinsic parameters and the relevant coordinate system transformations

(b) intrinsic parameters such as focal length f, aspect ratio w/h, and the skew represented as the angle α

Fig. 4.5 The Camera Calibration Parameters

In order to get the position and orientation of the virtual camera center the additional transformation G should be found. The position Tcamera2marker can be found by measuring with a ruler the distances in all three directions X,Y, and Z of the cameras bodies coordinate system to the virtual cameras center which should be approximately between the user's eyes. And the orientation Rcamera2marker can be found by measuring the angles χ, ρ, and σ between the X-, Y-, and Z-axis of the camera marker and the corresponding axis of the virtual camera coordinate systems. Constituting these angles, the desired transformation is

$$G = R_{camera2marker} [I|-T_{camera2marker}]$$

Now the viewing subsystem of the Augmented Reality system knows the pose of the virtual camera relative to the tracking subsystem's coordinate system. As the transformation C is also constant, because the tracking subsystem is rigidly fixed in the laboratory, it can be measured the same way as well. Thus the overall transformation matrix A that maps the virtual camera center to the world coordinate system is

$$A = GFC$$

where A is a 3 × 4 projection matrix that transforms world coordinates to camera coordinates. C is a 4×4 homogeneous transformation matrix that maps world to tracker coordinates. During the implementation phase it is assumed that the world and the tracker coordinate systems are equal, e.g. C = I. Furthermore F is also a 4 × 4 homogeneous transformation matrix that maps the coordinate system of the camera marker to the tracker coordinate system. At last G is the 3×4 projection matrix that defines the camera transformation relative to the coordinates of the camera marker. The matrix G is the desired projection matrix, as F is known to the Augmented Reality system by the tracking subsystem.

Calibration needed for -

In order to get an effective augmentation of the real world, the real and virtual objects must be accurately positioned relative to each other. The computer system contains in its virtual world a virtual camera which can see a range of several virtual objects. These are usually displayed on a head mounted display (HMD) to get an AR view. So the HMD represents a virtual camera for which several parameters must be accurately specified as well. If these parameters do not fit properly, the virtual picture might have a different size than the real one or even be distorted. Once all the parameters are found and adjusted correctly, the user can use the AR system to augment the reality. But may be some other user wants to use the same AR system as well. And maybe he has another interocular distance or he wears the HMDs lightly different than the person who first adjusted all the parameters. Even if this person puts on the HMD for another session again, the primarily adjusted parameters will not fit as good any more. So the procedure of calibrating the virtual camera has to be kept simple, in order to make it possible for users who know nothing about the mathematical background of calibration to adjust the HMD anytime fast and precise. Another problem has always been the accurate adjustment of different displays because different algorithms are necessary.

Calibration is the process of instantiating parameter values for mathematical models which map the physical environment to internal representations, so that the computer's virtual world matches the real world. These parameters include information about optical characteristics and pose of the real world camera, as well as informations about the environment, such as the tracking systems origin and pose of tracked objects.

Functional Requirements of the calibration service- describe the interactions between the system and its environment.

1. Accurate Alignment

The main goal of a successful calibration is an Augmented Reality system in which all virtual objects are optimal adjusted. The HMD user should see the real objects and the corresponding superimposed virtual objects accurate aligned. So the distance, size, and form of them should fit to their real counterparts. Even when the user moves through the tracked space the visual enhancement shall keep correctly aligned.

2. Easy to Use

In order to get a practical solution for the calibration of see through devices, the parameters need to get estimated in a user-friendly procedure. Thus the user interaction where the calibration points are measured shall be intuitive and not impose a great burden on the user.

Nonfunctional Requirements – describe the user visible aspects of the system that are not directly related with the functional behavior of the system

1. Performance

The performance of the calibration procedure primarily depends on the performance of the middleware. As the calibration service depends on DWARF the desired components can be executed distributed on different computers. Thus the system latency (delay) should be kept at a minimum. It is vitally to receive the relevant measurement data in real time as the user is allowed to move. Thus the measured parameters change in real time, too. The actual calculation of the calibration parameters does not need to be in real time as it will be done just once. But the updating of the viewing component should be completed within a few seconds.

2. Accurate Tracking

For an accurate alignment the Augmented Reality system needs to know the exact pose of the virtual camera respectively the tracked 6DOF (freedom of movement of a rigid body in three-dimensional space) marker of the HMD in real time. The pose of other objects, such as the position of the 3DOF calibration points, must be known, too. As the real world location of these objects may change by moving these, the virtual objects need the same pose change. This is solved by the ART track1 tracking subsystem which updates the virtual model of the Augmented Reality system in real time.

3. Reliability

The system should guide the user in a way that it is guaranteed to obtain good results. Additionally it should provide hints on how good the accuracy is at the moment.

4. Quality of Service

The goal is to find the optimal solution where the measurement deviation is minimized. Furthermore error estimates should be provided to other DWARF components in order to be able to reduce error accumulation.

Pseudo Requirements

Pseudo requirements are imposed by the client that restricts the implementation of the system. The only pseudo requirement that occurred for the calibration method is that the prototypical implementation has to be done in context with ARCHIE. So the main focus has been a good aligned ARCHIE application rather than a perfect calibration method.

Consequently the user interface controller of the calibration method needed to be written in Java depending on the object-oriented Petri net simulation framework called JFern .

NOTE: DWARF stands for Distributed Wearable Augmented Reality Framework. The name is an acronym representing the guidelines for the general system architecture. ARCHIE is the latest application. The acronym stands for Augmented Reality Collaborative Home Improvement Environment. The completion of the ARCHIE project provides new functionality to DWARF thereby making it more mature. Advanced Realtime Tracking (ART). For accurate position and orientation tracking, the infrared (IR)-optical Advanced Realtime Tracking subsystem ART track 1 is used with four cameras. Single Point Active Alignment Method (SPAAM) is a simple method to calibrate virtual cameras

V. Marker-based Augmented Reality

Marker-based augmented reality experiences require a static image also referred to as a trigger photo that a person can scan using their mobile device via an augmented reality app. The mobile scan will trigger the additional content (video, animation, 3D or other) prepared in advance to appear on top of the marker. Marker-based Augmented Reality uses a designated marker to activate the experience.

Popular markers include Augmented Reality QR codes, logos, or product packaging. The shapes or images must be distinctive and recognizable for the camera to properly identify it in various surroundings. There is another important factor of marker-based Augmented Reality. The marker-based AR experience is tied to the marker. This means that the placement of digital elements depends on the location of the marker. In most cases, the experience will display on top of the marker and move along with the marker as it is turned or rotated.

The key feature of Augment Reality in comparison to other image processing tools is that virtual objects are moved and rotated in 3D coordinates instead of 2D image coordinates. The main objectives of AR are analysis of changes in the captured camera frames and correct alignment of the virtual data into the camera scene based on the tracking results. In turn, a marker-based approach provides the accurate tracking using visual markers, for instance, binary markers (designed by ARUCO, METAIO, etc.) or with photo of real planar objects in camera scene.

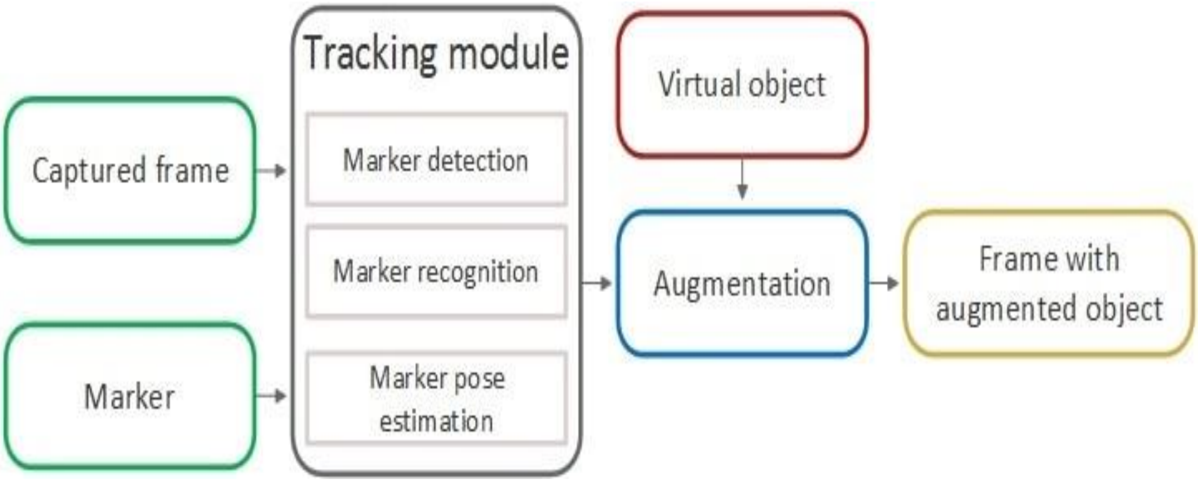


Fig. 4.6 AR System Flowchart

At first, the marker image should be there and then extract the consecutive camera frames. The tracking module in flowchart (Fig. in 4.6) is the core of the augmented reality system. It calculates the relative pose of the camera based on correctly detected and recognized marker in the scene. The term "pose" means the six degrees of freedom (DOF) position, i.e. the 3D location and 3D orientation of an object. The tracking module enables the system to add virtual components as a part of the real scene. And since the dealing is with camera frames in 2D coordinates system, it is necessary to use the projective geometry for virtual 3D object augmentation.

Detection and recognition

In the case of tracking by binary marker, the first necessary thing is to print the desired marker and place it in front of the camera. This requirement is an evident drawback of the tracking algorithm. The algorithm of detection is very simple and based on the marker nature:

- Application of adaptive thresholding to extract edges;
- Extraction of closed contours from binary image;
- Filtration of contours;
- Contours approximation and detection of quadrilateral shaped contours.

After above steps the marker candidates are stored for the further marker recognition. Each candidate is warped to the frontal view and divided on blocks. The task of recognition algorithm is to extract binary code from the marker candidate and compare it with the code of true marker. The most similar candidate is considered as a matched marker.

How to create marker-based augmented reality content?

Need - to create own augmented reality experience:

- a static trigger image
- digital content such as video or 3D object to feature on top of the chosen picture
- a software for combining the two pieces of content, (such as the Overly self-service AR creator)
- a mobile device with a compatible application to scan the marker and retrieve the AR content.

Getting the AR content straight

- what makes an excellent augmented reality marker?

An AR marker must be a photo specially created for our use. Make sure that the image is unique. Choose own design, a photo you have taken, or something from the business' library that no one else could easily access online. Using stock photos or Google images is not a good idea as someone else could be already using the same picture or will do so at a later stage and because of that it can show different content on the same marker. In short, there will not be ownership of that content. For the computer vision to detect the marker, use as many graphical elements and contrasts in the trigger photo as possible. The marker image is not a thing to be a minimalist about. The image recognition system thrives on different shapes, shadows, etc. Finally, fix the marker – where it is going to end up. Even if it looks great on the computer, ensure that it works when printed. Points to be noted: choose matte finish as glossy photos due to their reflection with light may be hard to read. AR markers are challenging to use for rounded objects such as cans or bottle labels, consider creating a bottle tag instead to ensure AR content looks good. Bottles can work, but usually it takes knowledge to create such things.

Environment considerations for marker-based augmented reality experiences

Even if the perfect marker is picked by design, there are still some other pointers that are key to the success of creating augmented experience. AR marker placement will be detrimental as the same trigger photo may not work at a bus stop as it does on a building facade or a small flayer. 50% rule for computer vision. Wherever the augmented reality marker is placed, consider that when people scan it with their mobile device, it must take up at least 50% of the camera screen — no less. So incase if a scene is intended to create in outdoor, consider how high up the banner is going to be. The higher it is, the bigger it has to be. If you're placing a marker on an A4 poster, consider how close people are going to be able to get to it, because for A4 one meter probably is optimal. If big banner ad is considered, ensure that the space for the AR content to be retrieved isn't limited. Consider that people need to be able to get the trigger photo within their camera screen. So if it is on a larger size, you should make sure that the environment surrounding it is vast enough for people to step back and scan the marker. If you place a large-scale ad on a busy junction, it may not be the best idea to add AR to it, because there may not be an appropriate space for people to stand to scan it or step back to get the image within their screen.

Outdoor AR markers are weather dependent

While an outdoor ad that has been lit up will work for 24/7, its AR functionality will not deliver the scans in all weather conditions. One of the challenges is night time, and if the AR trigger photo is consumed by night, you won't be able to see it nor scan it. Another point to consider when placing a marker outdoors is sunlight or shade. Both of these can affect if computer vision is capable of detecting your poster. Therefore, avoid placing AR markers on banners where there are drastic sunlight and shadow changes throughout the day.

Combining the AR content with a marker in 5 steps

- The platform - will most probably be web-based and require registration but once it is done, the steps should be similar for all. Upload the static design (trigger image) that you want to bring to life. Then - asked to upload the content (based on the platform, the type of content can be uploaded). Choose the content size and its location in respect to the marker, add any CTAs if possible
- Preview the look on web and press publish
- In a few seconds, the system will be updated, and
- can take the mobile phone to test the marker and
- share it.

Marker recognition can be local or cloud-based, it means that marker databases can be stored on device and recognition also happens on device. The databases can also be stored on a cloud and recognition happens on a server, phone is only sending point clouds to server. Device-based recognition can happen immediately, but if cloud recognition is used, then it will take a while longer for the content to be downloaded from the server. Usually it takes a couple of seconds before the user can see any augmented reality experience.

Pros

- If the marker image is prepared correctly, marker-based AR content provides quality experiences and tracking is very stable, the AR content doesn't shake
- Easy to use, detailed instructions are not required for people who use it for the first time

Cons

- When the mobile camera is moved away from the marker, AR experience disappears and the trigger photo has to be scanned again. It is possible to use extended tracking, but in most cases, extended tracking makes things worse.
- Scanning will not work if markers reflect light in certain situations (can be challenging with large format banners in ever-changing weather conditions)
- Marker has to have strong borders/contrast between black and white colors to make tracking more stable.
- Smooth color transition will make recognition impossible.

VI. Pattern Recognition

At the age of 5, most children can recognize digits and letters – small characters, large characters, handwritten, machine printed, or rotated – all easily recognized by the young. In most instances, the best pattern recognizers are humans, yet unaware to understand how humans recognize patterns. Pattern recognition is the automated recognition of patterns and regularities in data. Techniques for finding patterns in data have undergone substantial development over the past decades. Pattern recognition analyzes incoming data and tries to identify patterns. While explorative pattern recognition aims to identify data patterns in general, descriptive pattern recognition starts categorizing the detected patterns. Hence, pattern recognition deals with both of these scenarios, and different pattern recognition methods are applied depending on the use case and form of data. Consequently, pattern recognition is not *one* technique but rather a broad collection of often loosely related knowledge and techniques. Pattern recognition capability is often a prerequisite for intelligent systems. The data inputs for pattern recognition can be words or texts, images, or audio files. Hence, pattern recognition is broader compared to computer vision that focuses on image recognition. Automatic and machine-based recognition, description, classification, and grouping of patterns are important problems in a variety of engineering and scientific disciplines, including biology, psychology, medicine, marketing, computer vision, and artificial intelligence.

Pattern?

In 1985, Satoshi Watanabe defined a pattern "as the opposite of a chaos; it is an entity, vaguely defined, that could be given a name". In other words, a pattern can be any entity of interest that one needs to recognize and identify: It is important enough that one would like to know its name (its identity). Therefore, patterns include repeated trends in various forms of data. For example, a pattern could be a fingerprint image, a handwritten cursive word, a human face, or a speech signal. A pattern can either be observed physically, for example, in images and videos, or it can be observed mathematically by applying statistical algorithms.

Fig. 4.7 Examples of patterns: Sound wave, tree species, fingerprint, face, barcode, QR-code, handwriting, or character image

Recognizing a pattern?

Given a pattern, its recognition and classification can consist of one of the following two tasks: Supervised classification identifies the input pattern as a member of a predefined class (Descriptive). Unsupervised classification assigns the input pattern to a undefined class (Explorative). The recognition problem is usually posed as either classification or categorization task. The classes are either defined by the system designed (supervised classification) or are learned based on the similarity of patterns (in unsupervised classification). Pattern recognition is constantly evolving, driven by emerging applications that are not only challenging but also more computationally intensive.

Goal of pattern recognition

The goal of pattern recognition is based on the idea that the decision-making process of a human being is somewhat related to the recognition of patterns. For example, the next move in a chess game is based on the board's current pattern and buying or selling stocks is decided by a complex pattern of financial information. Therefore, the goal of pattern recognition is to clarify these complicated mechanisms of decision-making processes and to automate these functions using computers.

Definition of pattern recognition

Pattern recognition is defined as the study of how machines can observe the environment, learn to distinguish various patterns of interest from their background, and make logical decisions about the categories of the patterns. During recognition, the given objects are assigned to a specific category. Pattern Recognition, as it is a constantly evolving and broad field. An early definition of pattern recognition defines it as "a classification of input data via extraction of important features from a lot of noisy data" (1978, Thomas Gonzalez). In general, pattern recognition can be described as an information reduction, information mapping, or information labeling process. In computer science, pattern recognition refers to the process of matching information already stored in a database with incoming data based on their attributes.

Pattern Recognition and Artificial Intelligence (AI)

Artificial Intelligence (AI) refers to the simulation of human intelligence, where machines are programmed to think like humans and mimic their actions. Most prominently, fields of artificial intelligence aim to enable machines to solve complex human recognition tasks, such as recognizing faces or objects. Accordingly, pattern recognition is a branch of Artificial Intelligence.

Pattern Recognition and Machine Learning

Today, in the era of Artificial Intelligence, pattern recognition and machine learning are commonly used to create ML models that can quickly and accurately recognize and find unique patterns in data. Pattern recognition is useful for a multitude of applications, specifically in statistical data analysis and image analysis. Most modern use cases of pattern recognition are based on artificial intelligence technology. Popular applications include speech recognition, text pattern recognition, facial recognition, movement recognition, recognition for video deep learning analysis, and medical image recognition in healthcare.

How does Pattern Recognition Work?

Historically, the two major approaches to pattern recognition are Statistical Pattern Recognition (or decision-theoretic) and Syntactic Pattern Recognition (or structural). The third major approach is based on the technology of artificial neural networks (ANN), named Neural Pattern Recognition. No single technology is always the optimal solution for a given pattern recognition problem. All three or hybrid methods are often considered to solve a given pattern recognition problem.

Statistical Pattern Recognition

Statistical Pattern Recognition is also referred to as StatPR. In statistical pattern recognition, the pattern is grouped according to its features, and the number of features determines how the pattern is viewed as a point in a d-dimensional space. These features are chosen in a way that different patterns take space without overlapping. The method works so that the chosen attributes help the creation of clusters. The machine learns and adapts as expected, then uses the patterns for further processing and training. The goal of StatPR is to choose the features that allow pattern vectors to belong to different categories in a d-dimensional feature space.

Syntactic Pattern Recognition

Syntactic Pattern Recognition, also known as SyntPR, is used for recognition problems involving complex patterns that can be addressed by adopting a hierarchical perspective. Accordingly, the syntactic pattern approach relies on primitive subpatterns (such as letters of the alphabet). The pattern is described depending on the way the primitives interact with each other. An example of this interaction is how they are assembled in words and sentences. The given training samples develop how grammatical rules are developed and how the sentences will later be "read". In addition to classification, structural pattern recognition also provides a description of how the given pattern is constructed from the primitive subpatterns. Hence, the approach has been used in examples where the patterns have a distinct structure that can be captured in terms of a rule-set, such as EKG waveforms or textured images. The syntactic approach may lead to a combinatorial explosion of probabilities to be examined, requiring large training sets and very large computational efforts.

Template-matching

Template matching is one of the simplest and earliest approaches to pattern recognition. Matching is a generic operation that is used to determine the similarity between two entities of the same type. Therefore, template matching models try to discover similarities in a sample based on a reference template. Hence, the template matching technique is commonly used in digital image processing for detecting small sections of an image that match a template image. Typical real-world examples are medical image processing, quality control in manufacturing, robot navigation, or face recognition.

Neural network pattern recognition

AI pattern recognition using neural networks is currently the most popular method for pattern detection. Neural networks are based on parallel subunits referred to as neurons that simulate human decision-making. It can be viewed as massively parallel computing systems consisting of a huge number of simple processors with many interconnections (Neurons). The most popular and successful form of machine learning using neural networks is deep learning, which applies deep convolutional neural networks (CNN) to solve classification tasks. Today, neural network pattern recognition has the edge over other methods because it can change the weights repeatedly on iteration patterns. In recent years, deep learning has proven to be the most successful method to solve recognition tasks.

Hybrid pattern detection

After going through all the pattern recognition techniques, it is evident that no algorithm is always the most efficient for any use case. Therefore, combinations of various machine learning and pattern recognition algorithms lead to the best results or enable the implementation of efficient and optimized pattern detectors. Consequently, many pattern recognition projects are based on hybrid models to enhance the performance of the pattern recognizer for the specific use cases, depending on the type and availability of data. For example, deep learning methods achieve outstanding results but are computationally intensive, while "lighter" mathematical methods usually are more efficient. Also, it is common to apply methods for data pre-processing before applying AI pattern recognition models. Using the hybrid model will enhance the performance of the entire application or detection system.

Process of finding patterns in data

The design of pattern recognition systems essentially involves

- data acquisition and preprocessing,
- data representation, and
- decision making.

The pattern recognition process itself can be structured as follows:

- Collection of digital data
- Cleaning the data from noise
- Examining information for important features or familiar elements
- Grouping of the elements into segments
- Analysis of data sets for insights
- Implementation of the extracted insights

Pattern Recognition examples

Stock market prediction

- Using pattern recognition for stock market prediction applications is a classical yet challenging task with the purpose of estimating the future value of a company stock or other traded assets. Both linear and machine learning methods have been studied for decades. Only lately, deep learning models have been introduced and are rapidly gaining in popularity.

Optical character recognition

- Optical character recognition (OCR) is the process of classification of optical patterns contained in a digital image. The character recognition is achieved through image segmentation, feature extraction, and classification.

Text pattern recognition

- Machine learning based pattern recognition is used to generate, analyze, and translate text. Hence, patterns are used to understand human language and generate text messages. Accordingly, text recognition on words is used to classify documents and detect sensitive text passages automatically. Therefore, text pattern recognition is used in the Finance and Insurance industries for fraud detection.

Handwriting recognition

- Handwriting recognition is used to compare patterns across handwritten text or signatures to identify patterns. Various applications are involved in the computer recognition of pen-input handwritten words. However, handwritten word recognition and spotting is a challenging field because handwritten text involves irregular and complex shapes.

Face recognition and visual search

- Image recognition algorithms aim to detect patterns in visual imagery to recognize specific objects (Object Detection). A typical image recognition task is image classification, which uses neural networks to label an image or image segment based on what is depicted. This is the basis of visual search, where users can easily search and compare labeled images.

Voice or speaker recognition

- Voice recognition systems enable machines to receive and interpret dictation or are able to carry out spoken commands and interact accordingly. Speech recognition is based on machine learning for pattern recognition that enables recognition and translation of spoken language.

Emotion recognition systems

- Machine learning in pattern recognition is applied to images or video footage to analyze and detect the human emotions of an audience. The goal is to indicate the mood, opinion, and intent of an audience or customers. Hence, deep learning is applied to detect specific patterns of facial expressions and movements of people. Those insights are used to improve marketing campaigns and customer experience.

Benefits of Pattern Recognition

- Pattern recognition methods provide various benefits, depending on the application. In general, finding patterns in data helps to analyze and predict future trends or develop early warning systems based on specific pattern indicators. Further advantages include:
- Identification: Detected patterns help to identify objects at different angles and distances (for example, in video-based deep learning) or identify hazardous events. Pattern recognition is used to identify people with video deep learning, using face detection or movement analysis. Recently, new AI systems can identify people from their walk by measuring their gait or walking pattern.
- Discovery: Pattern recognition algorithms allow "thinking out of the box" and detecting instances that humans would not see or notice. Algorithm patterns can detect very fine movements in data or correlations between factors across a huge amount of data. This is very important for medical use cases; for example, deep learning models are used to diagnose brain tumors by taking images of magnetic resonance imaging.
- In information security and IT, a popular pattern recognition example is the use of pattern matching with an intrusion detection system (IDS) to monitor computer networks or systems for malicious activity or policy violations.
- Prediction: Forecasting data and making predictions about future developments play an important role in many pattern recognition projects, for example, in trading markets to predict stock prices and other investment opportunities or to detect trends for marketing purposes.
- Decision-making: Modern machine learning methods provide high-quality information based on patterns detected in near real-time. This enables decision-making processes based on reliable, data-based insights. A critical factor is the speed of modern, AI pattern recognition systems that outperform conventional methods and enable new applications. For example, medical pattern recognition, to detect risk parameters in data, providing doctors critical information rapidly.
- Big-Data analytics: With neural networks, it became possible to detect patterns in immense amounts of data. This enabled use cases that would not have been possible with traditional statistical methods. Pattern recognition is vital in the medical field, especially for forensic analysis and DNA sequencing. For example, it has been used to develop vaccines to battle the COVID-19 Coronavirus.

Pattern recognition algorithms can be applied to different types of digital data, including images, texts, or videos. Finding patterns enables the classification of results to enable informed decision-making. Pattern recognition can be used to fully automate and solve complicated analytical problems.

Interactive E-Learning System Using Pattern Recognition and Augmented Reality

The goal - is to provide students with realistic audio-visual contents when they are leaning. The e-learning system consists of image recognition, color and polka-dot pattern recognition, and augmented reality engine with audio-visual contents. When the web camera on a PC captures the current page of textbook, the e-learning system first identifies the images on the page, and augments some audio-visual contents on the monitor. For interactive learning, the e-learning system exploits the color-band or polka-dot markers which are stuck to the end of a finger. The color-band and polka-dot marker act like the mouse cursor to indicate the position in the textbook image. Appropriate interactive audio-visual contents are augmented as the marker is located on the predefined image objects in the textbook. This was applied to the educational courses in the school and obtained satisfactory results for real applications.

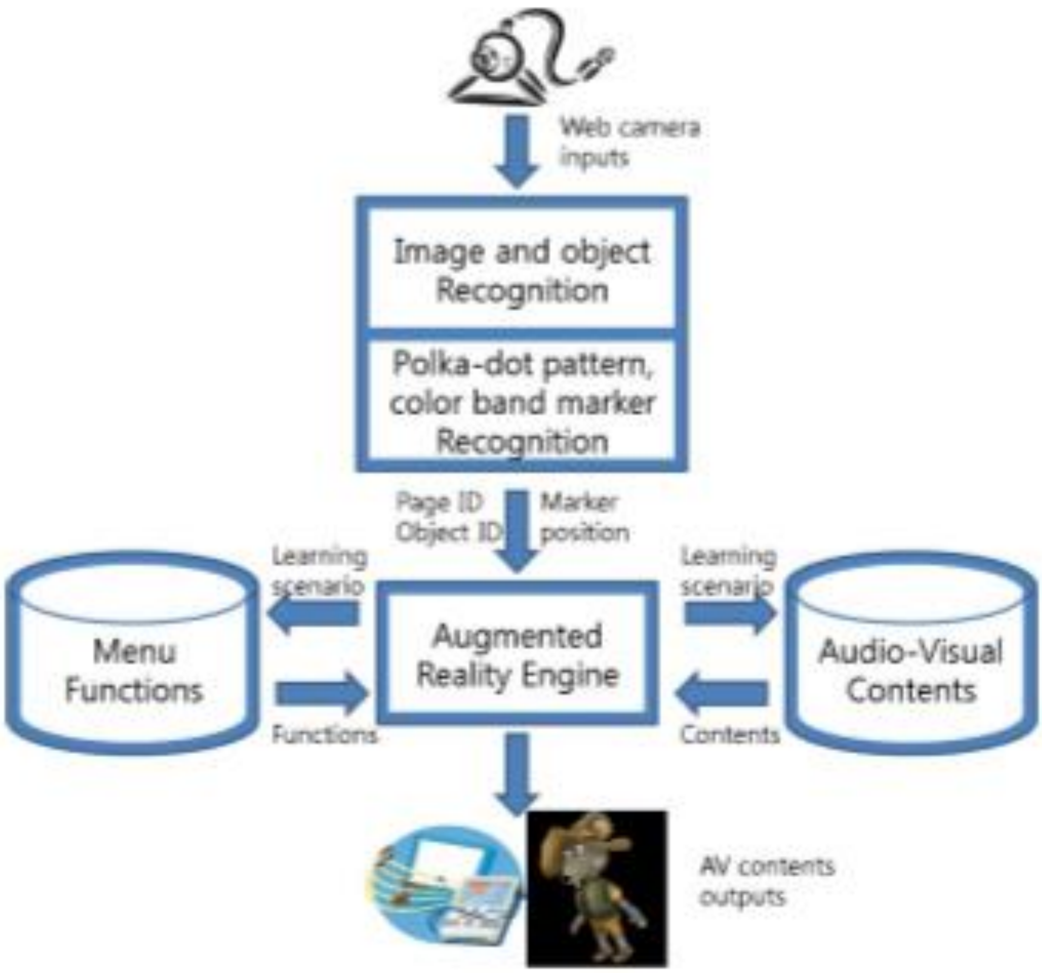


Fig. 4.8 Structure of e-learning system

The image and marker recognition enables students to learn interactively according to the predefined learning scenarios and audio-visual contents.

The e-learning system consists of image/object recognition, polka-dot pattern recognition, color-band marker recognition, augmented reality engine, audio-visual contents, and some learning scenarios of textbooks. The learning scenarios are the predefined processes when or where to augment the contents. The scenarios combine the educational contents with information technologies to maximize the learning efficiency. And the augmented reality engine realizes the scenarios. Fig. 4.8 shows the structure of e-learning system. A web camera connected to the computer focuses on the textbook. The students study watching the textbook and the captured video frame where some audiovisual contents are augmented. When video frames from web camera are given, the recognition modules identify the image and objects on the textbook, and polka-dot or color-band marker. Database of images and objects in the textbook in advance - is available. The image/object recognition module identifies the current text page and objects that the student is studying. Using the identified pages and objects from recognition module, the system knows where the objects are located in the video frame. Then, some audio-visual contents are augmented on the computer monitor according to the predefined educational scenarios. The augmented reality engine matches the scenarios to information from the recognition modules, and plays the audio-visual contents automatically. Some interactive learning actions are possible by the polkadot or color-band marker. The marker is a kind of computer mouse, and indicates the location in the video frame. If the marker is located on the specific objects or menu bars, the object-based interactions are performed based on the educational scenarios and contents. The related visual contents are displayed on the marker even though the marker is moving. Some interactive actions, such as dragging the virtual object, scrubbing-based reaction, and menu selection, are also defined in the e-learning system. For the usefulness of e-learning system, many educational contents and scenarios are produced for the real school courses. In addition, the authoring tool is developed to produce the educational scenarios and interactions easily, since the system is designed for general purposes. Thus, any contents providers and educational organizations can exploit the e-learning system for their interactive learning courses.

Two markers are designed using polka-dot pattern or color-band. The markers are put on the fingers as bands, and act like the computer mouse. The markers indicate their locations in the video frame, which enables the students to interact according to the objects in the textbook. When the marker is located at a specific object or menu in the textbook, the corresponding audio-visual contents are augmented on the computer, or the predefined menu function is performed. And some interactive functions such as dragging and scrubbing object are defined to support various learning actions.

Polka-dot Pattern Recognition

The polka-dot patterns are rare in the usual textbook, and well recognized both in the grayscale and color images. The polka-dot band for a finger is used as a computer mouse. The polka-dot marker is exploited for interactive augmentation of contents and menu selection. To detect polka-dot pattern exactly in real-time, fast filters of integer operations, hierarchical searching, and edge information are used. Fig. 4.9 shows two array patterns of polka-dot markers. The array patterns are empirically selected by the polka-dot recognition algorithm. Since the marker on a finger is subject to be rotated and slanted at the camera viewpoint, the array pattern of dots should be invariant to the perspective variations. According to the recognition algorithm, the optimal array pattern was selected. The hexagonal array is the best pattern that is invariant to the perspective distortions of camera viewpoints.

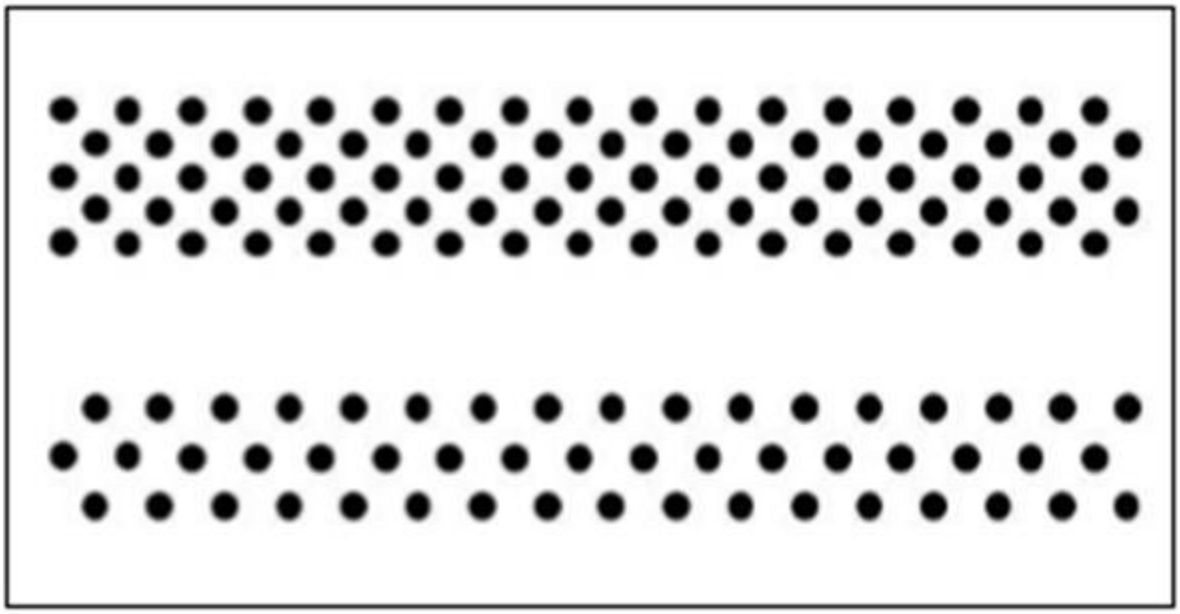


Fig. 4.9 Polka-dot patterns for interactive learning

The basic algorithm of polka-dot pattern recognition is the high pass filter in the horizontal and vertical directions. The high pass filter first finds the area where the grayscale pixel values are regularly varied with black and white pattern. For fast operation in marker detection, the search range is restricted based on the motion vector of previously detected polka-dot marker. The motion vector of polka-dot marker enables us to predict the next location. The next position of marker can be predicted; first detect the marker in the restricted search range. If the polka-dot marker is not detected in the restricted search range, the search range is expanded and the marker is found again. Finally, the detected marker is examined by edge information. Since the high pass filters can detect the complex textures or characters in the textbook as polka-dot markers, edge information is exploited to reduce the false positive errors. Since the characters or other complex textures usually have some line-edge properties unlike the polka-dot patterns, the false positive errors are decreased by the edge information. Fig. 4.10 shows some results of polka-dot pattern recognition. It is shown that one or two independent polka-dot markers are detected in the video frame. In the usual personal computer environment, the recognition is performed at higher than 25 frames per second for 640x480 resolution.

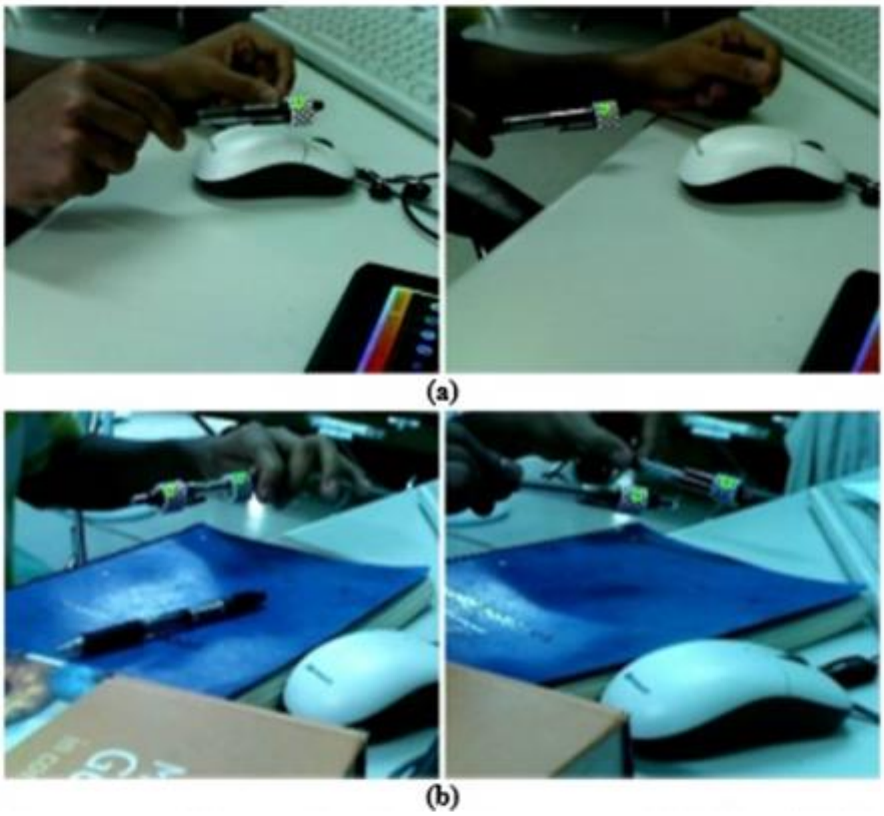


Fig. 4.10 Recognition Results of Polka-dot patterns. Multiple polka-dot markers are independently detected in the video frame. The light green squares mean the central position of polka-dot markers. (a) Single marker detection, (b) two markers detection

Color-band Recognition

Some interactions in the educational scenarios require two or more markers simultaneously to manipulate multiple objects. Since the polka-dot patterns have little distinct difference between them, it is difficult to operate the multiple markers independently. New multiple markers are needed to discriminate individually. Two color-band markers are designed which consist of three colors as shown in Fig. 4.11. The color-band markers are discriminated with each other and the polka-dot marker, thus, three markers are used simultaneously according to the educational scenarios and interaction.

Fig. 4.11 Color-band markers using three colors

The colors of the markers are selected from various experiments. The blue color is usually best recognized and most stable in the lighting variation. The blue band is located at the center of color-band marker, and is searched first. The other colors have been chosen since they are well discriminated with each other and the blue color. Two color-band markers are designed with different combinations of colors as shown in Fig. 4.11. The color-band markers are detected by finding blue color first. The hue components in HSV (Hue Saturation Value) color space are used for robust detection in various lighting conditions. When blue color pixels are detected, the shape and area of blue region are examined - whether the blue region satisfies the condition of marker or not. Then, the other colors (Green and Red, or Yellow and Purple) are searched around the blue region. The color range and the area of the color region is considered to confirm the color-band pattern. The order of colors and ratios of color areas are compared with the predefined criterions. Fig. 4.12 shows that two color-band markers are independently detected in a video frame. The color ranges of color-band markers are optimized according to the lighting environment. Note that the color ranges should be changed with respect to the lighting conditions. Thus, method should be devised to adjust the color ranges of markers automatically when the e-learning system is setup.

Fig. 4.12 Recognition results of color-band markers. Two markers are consistently detected when they are moving

RECOGNITION OF IMAGE AND OBJECT

Image recognition is designed for identification of current text page or objects. When the text page or objects are identified, the related audio-visual contents are automatically played on the PC. Since the pose information of objects is obtained in the captured image, the visual 3-D contents are augmented according to the poses of objects. Augmented reality (AR) toolkits have used geometric markers to be recognized in the images. The AR markers consist of black/white geometric shapes in the square. The AR markers are well recognized in the various image distortions, and they have been popular for interactivity of virtual systems. However, since the AR markers are directly printed on the textbook pages, they do not look good for text design. Here, the goal is to replace the AR geometric markers with image objects and to design a natural interface using the image objects.

Feature Extraction

Since the images are subject to be rotated, distorted by perspective viewpoints, and changed by scales, robust features invariant to the image variations are extracted. Scale-invariant features and some feature extraction algorithms are developed for image and objects recognition. The robust features called speeded up robust feature (SURF) can be exploited, which shows good recognition results and fast operation compared with SIFT. Since the e-learning system is also applied to the mobile devices like PDA (personal digital assistant) or mobile phone, SURF algorithm can be implemented with integer programming and optimized lookup tables. The first step of feature extraction is to detect the distinct points which are also invariant to image variations. The second step of feature extraction is to find a dominant orientation around the feature point. The orientation information normalizes the rotated images and objects. Thus, the images or objects are recognized in spite of rotational distortions. The last step of feature extraction is to describe the feature points as a vector structure. This descriptor recognizes the feature points. The square region around a feature point is selected for the descriptor. Note that the square is rotated by the dominant orientation before finding the descriptor. The size of square is related to the scale parameter. The square region is divided into 16 subregions, and 25 pixels (5x5) are sampled in each subregion

Feature Matching

The corresponding features are searched by the vector distance between descriptors. When features are extracted for image and object recognition, all pairs of features are examined by the vector distances. Then, the nearest (f1) and second nearest (f2) features are selected; and the nearest feature is matched with the features f. The only features that are on the same geometric relation are matched

Image and Object Recognition

With all pairs of matched features, the images or objects are recognized. The simplest method is to count the number of matched features. Without loss of generality, the image pairs that have the largest number of matched features are the same. There are some matching errors; homography is used to reduce matching errors. Since the homography reflects the geometric relations of features, it removes such mismatched features that satisfy the matching criterion without geometric correlation. Fig. 4.13 shows two example of image recognition. For real situations, the images or objects are occluded partially by the hand. As it is seen in Fig. 4.13, the images are well recognized under various image distortions, such as perspective distortion, luminance difference, scale difference, and occlusion. In Fig., the left images are the database images, and the right images are captured ones by the web camera. The images are well identified regardless of AR markers.

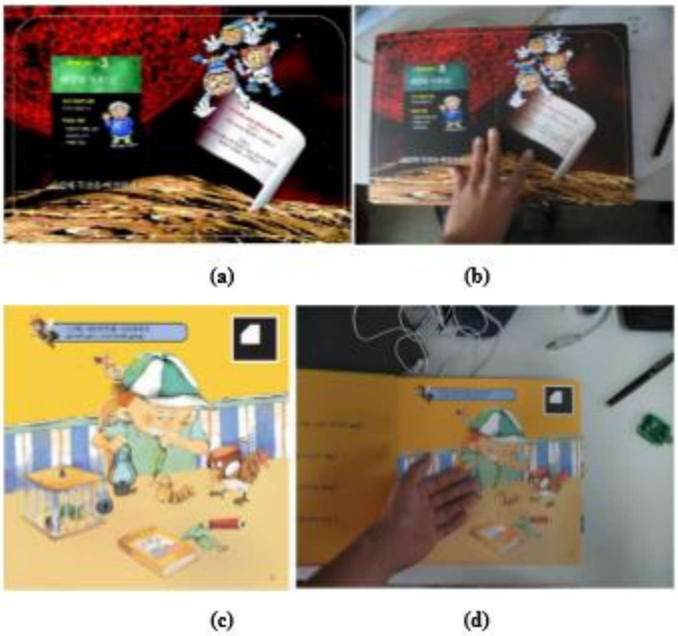


Fig. 4.13 Image recognition result (a) and (c) Original images recognized in the database regardless of AR markers, (b) and (d) Captured images by the webcam

Fig. 4.14 first shows that a moving graphic is augmented on the color-band marker. The left image (a) is the captured image by the web camera, and the right image (b) shows the augmented reality with graphic contents. The page ID is recognized by the image and objects in the text page. Then, the related audio-visual contents are augmented as the scenarios and student's interaction. The graphics are displayed above the marker so that the interactive augment reality is naturally performed. The augmented graphic objects also move as the marker moves.

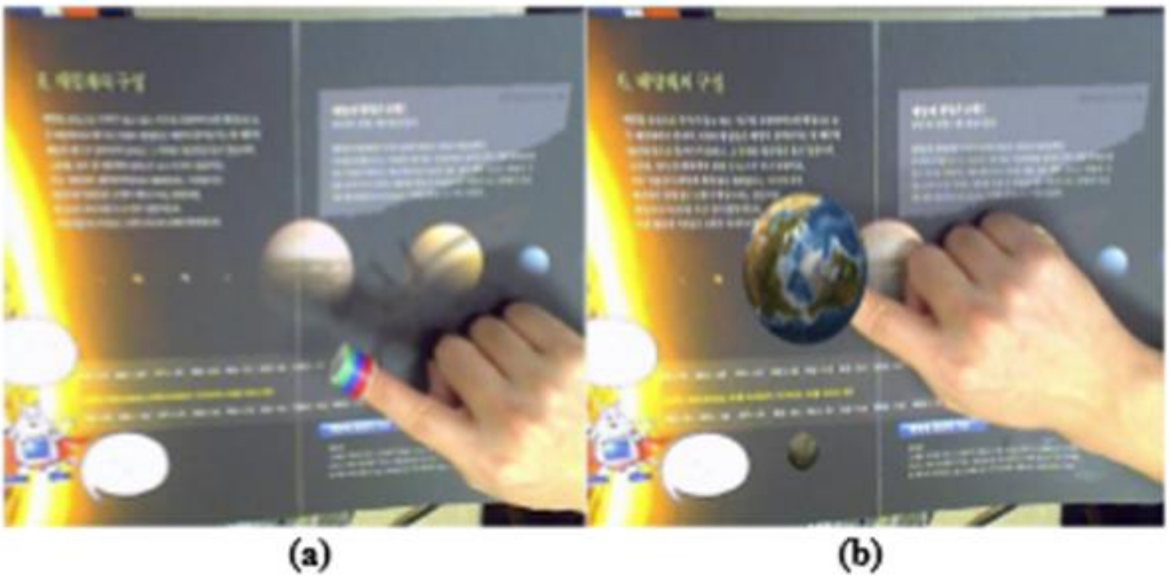


Fig. 4.14 Example of Augmented Reality using marker (a) a video frame captured by the web camera, (b) A visual content is augmented on the marker. The visual content is augmented on the marker, thus the marker is not seen on the monitor

Fig. 4.15 shows the commercial system and an exemplary image of interactive augmented reality. The interactive e-learning system using augmented reality was applied to the public elementary school in the courses of English and Science. This interactive augment reality made the students have more interest in learning. Therefore, the e-leaning system not only provides with audio-visual contents, but also improves the learning efficiency and concentration of students. It is expected that the e-learning system is very useful in the various educational courses.

(a) (b)

Fig. 4.15 Example of interactive augmented reality using image and marker recognition

VII. AR Toolkit

ARToolKit is an open-source computer tracking library. It is used for creation of strong augmented reality applications that overlay virtual imagery on the real world. Currently, it is maintained as an open-source project hosted on GitHub. ARToolKit is a very widely used AR tracking library with over 160,000 downloads on its last public release in 2004. In order to create strong augmented reality, it uses video tracking capabilities that calculate the real camera position and orientation relative to square physical markers or natural feature markers in real time. Once the real camera position is known a virtual camera can be positioned at the same point and 3D computer graphics models drawn exactly overlaid on the real marker. So ARToolKit solves two of the key problems in Augmented Reality; viewpoint tracking and virtual object interaction.

ARToolKit is a C and C++ language software library that lets programmers easily develop Augmented Reality applications. Augmented Reality (AR) is the overlay of virtual computer graphics images on the real world, and has many potential applications in industrial and academic research. One of the most difficult parts of developing an Augmented Reality application is precisely calculating the user's viewpoint in real time so that the virtual images are exactly aligned with real world objects. ARToolKit uses computer vision techniques to calculate the real camera position and orientation relative to marked cards, allowing the programmer to overlay virtual objects onto these cards. The fast, precise tracking provided by

ARToolKit should enable the rapid development of many new and interesting AR applications.

Features

- A multiplatform library (Windows, Linux, Mac OS X, SGI)
- A multi platform video library with:
    - multiple input sources (USB, Firewire, capture card) supported
    - multiple format (RGB/YUV420P, YUV) supported
    - multiple camera tracking supported
    - GUI initalizing interface
- A fast and cheap 6D marker tracking (real-time planar detection)
- An extensible markers patterns approach (number of markers)
- An easy calibration routine
- A simple graphic library (based on GLUT)
- A fast rendering based on OpenGL
- A 3D VRML support
- A simple and modular API (in C)
- Other language supported (JAVA, Matlab)
- A complete set of samples and utilities
- A good solution for tangible interaction metaphor
- OpenSource with GPL(General Public License) license for non-commercial usageTo develop an application - the source code for an existing example program: simpleLite.

The source code for this program is found inside ARToolKit installation in the directory examples/simpleLite/. The file is simpleLite.c. This program simply consists of a main routine and several graphics drawing routines. The functions which correspond to the six application steps described are shown in Table 1. The functions corresponding to steps 2 through 5 are called within the Idle() function.

Table 1: Function calls and code that corresponds to the ARToolKit applications steps.

ARToolKit Step	Functions
1. Initialize the video grabbing from the camera and load the marker(s)	setupCamera and setupMarker
2. Grab a video input frame	arVideoGetImage (called in mainLoop)
3. Detect the markers	arDetectMarker (called in mainLoop)
4. Calculate camera transformation	arGetTransMat (called in mainLoop)
5. Draw the virtual objects	Display
6. Close the application down	Quit

The most important functions in the program related to AR are main, setupCamera, setupMarker, mainLoop, Display, and cleanup. The GLUT library is used to handle the interaction with the operating system. GLUT, the OpenGL utility toolkit, is used to do things like open a window, and handle keypresses. However, GLUT is not required, and can be replaced with any library you like, e.g. MFC on Windows, Cocoa on Mac OS X, or QT (cross platform). The basics of a GLUT-based OpenGL application should be known before studying the code of simpleLite.c. OpenGL is a software interface to graphics hardware. This interface consists of about 150 distinct commands that you use to specify the objects and operations needed to produce interactive three-dimensional applications.

main

The main routine of simpleLite performs a number of setup tasks for the application. The first piece of AR specific code is near the top of main, where some variables that will be used to set up the application are declared:

```
char *cparam_name = "Data/camera_para.dat";
char *vconf = "";
char *patt_name = "Data/patt.hiro";
```

In this block - define the pathname of the camera parameter file the application will use, the video capture library configuration string, and the name of the marker pattern file the application will load and try to recognise. Next, first AR-specific function call:

```
// Hardware setup. //
if (!setupCamera(cparam_name, vconf, gARTThreshhold, &gARTCparam,
&gARHandle, &gAR3DHandle))
{
fprintf(stderr, "main(): Unable to set up AR camera.\n");
exit(-1);
}
```

setupCamera loads a file containing calibration parameters for a camera, opens a connection to the camera, sets some defaults (the binarization threshold in this case) and starts grabbing frames. It records its settings into 3 variables which are passed in as parameters. In this case, these parameters are stored in global variables. The next piece of code opens up a window to draw into. This code uses GLUT to open the window.

```
// Library setup. //   // Set up GL context(s) for OpenGL to draw into.
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
if (!prefWindowed)
{ if (prefRefresh) sprintf(glutGamemode, "%ix%i:%i@%i", prefWidth, prefHeight,
prefDepth,      prefRefresh);
    else
        sprintf(glutGamemode, "%ix%i:%i", prefWidth, prefHeight, prefDepth);
        glutGameModeString(glutGamemode);
        glutEnterGameMode(); }
        else
        {
        glutInitWindowSize(prefWidth, prefHeight);
        glutCreateWindow(argv[0]);
}
```

The code uses the value of a variable "prefWindowed" to decide whether to open a window, or whether to use fullscreen mode. Other variables prefWidth, prefHeight, prefDepth and prefRefresh - to decide how many pixels wide and tall, what colour bit depth to use, and whether to change the refresh rate of the display are simply held in static variables defined near the top of main.c in simpleLite. Next, with a window from GLUT, initialise the OpenGL part of the application. In this case, the ARgsub_lite library is used to manage the interaction between the ARToolKit video capture and tracking, and OpenGL.

```
// Setup ARgsub_lite library for current OpenGL context.
if ((gArglSettings = arglSetupForCurrentContext(gARHandle)) == NULL)
{ fprintf(stderr, "main(): arglSetupForCurrentContext() returned error.\n");
Quit(); }
debugReportMode(gARHandle, gArglSettings);
glEnable(GL_DEPTH_TEST);
arUtilTimerReset();
```

The third major part of ARToolKit initialisation is to load one or more markers which the camera should track. Information about the markers has previously been recorded into marker pattern files using the mk_patt utility (called "marker training"), so now these files can be loaded. In simpleLite, one marker is used, the default Hiro marker. The task of loading this marker and telling ARToolKit to track it - is performed by the function called setupMarker().

setupCamera and setupMarker

Before entering a real-time tracking and drawing state, the ARToolKit application parameters should be initialised. The key parameters for an ARToolKit application are: the patterns that will be used for the pattern template matching and the virtual objects these patterns correspond to. the camera characteristics of the video camera being used.

setupCamera begins by opening a connection to the video camera from which images for tracking will be acquired, using arVideoOpen(). The parameter vconf, passed to arVideoOpen is a string which can be used to request some video configuration other than the default. The contents of the vconf string are dependent on the video library being used. At this point it is found out from the video camera library how big the images it will supply will be, and what pixel format will be used:

```
static int setupCamera(const char *cparam_name, char *vconf, int threshhold,
ARParam *cparam, ARHandle **arhandle, AR3DHandle **ar3dhandle)
{ ARParam wparam; int xsize, ysize;
int pixFormat;   // Open the video path.
 if (arVideoOpen(vconf) < 0)
{ fprintf(stderr, "setupCamera(): Unable to open connection to camera.\n");
return (FALSE); }   // Find the size of the window.
if (arVideoGetSize(&xsize, &ysize) < 0)
return (FALSE);
fprintf(stdout, "Camera image size (x,y) = (%d,%d)\n", xsize, ysize);   // Get the
format in which the camera is returning pixels.
pixFormat = arVideoGetPixelFormat();
if (pixFormat < 0 )
{ fprintf(stderr, "setupCamera(): Camera is using unsupported pixel format.\n");
return (FALSE); }
```

Next – is to deal with the structures that ARToolKit uses to hold its model of the camera's parameters. These parameters are generated by the camera calibration process. The camera parameter file is loaded with the call to arParamLoad, with the path to the file being passed in a c-string as a parameter. Once the camera parameters are loaded, we adjust them to match the actual video image size being supplied by the video library, and then initialise a few necessary ARToolKit structures which depend on the camera parameters:

```
// Load the camera parameters, resize for the window and init.
```

```
if (arParamLoad(cparam_name, 1, &wparam) < 0)
{ fprintf(stderr, "setupCamera(): Error loading parameter file %s for camera.\n", cparam_name);
return (FALSE); }
arParamChangeSize(&wparam, xsize, ysize, cparam); fprintf(stdout, "*** Camera Parameter ***\n");
arParamDisp(cparam);
if ((*arhandle = arCreateHandle(cparam)) == NULL)
{ fprintf(stderr, "setupCamera(): Error: arCreateHandle.\n");
return (FALSE); }
if (arSetPixelFormat(*arhandle, pixFormat) < 0)
{ fprintf(stderr, "setupCamera(): Error: arSetPixelFormat.\n");
return (FALSE); }
```

setupCamera is completed by setting up some defaults related to the tracking portion of ARToolKit. These include debug mode, the labelling threshold, and the structure used to hold positions of detected patterns. Finally, the video library capturing frames are started and is ready to process -

```
if (arSetDebugMode(*arhandle, AR_DEBUG_DISABLE) < 0)
{ fprintf(stderr, "setupCamera(): Error: arSetDebugMode.\n");
return (FALSE); }
if (arSetLabelingThresh(*arhandle, threshhold) < 0)
{ fprintf(stderr, "setupCamera(): Error: arSetLabelingThresh.\n");
return (FALSE); }
if ((*ar3dhandle = ar3DCreateHandle(cparam)) == NULL)
{ fprintf(stderr, "setupCamera(): Error: ar3DCreateHandle.\n");
return (FALSE); }
if (arVideoCapStart() != 0)
{ fprintf(stderr, "setupCamera(): Unable to begin camera data capture.\n"); return (FALSE);
}
return (TRUE); }
```

The second major part of ARToolKit setup is to load pattern files for each of the patterns to be detected. In simpleLite, only one pattern is tracked, the basic "Hiro" pattern. setupMarker creates a list of patterns for ARToolKit to track, and loads the Hiro pattern into it. Loading multiple patterns can be seen in the simpleVRML example,

```
static int setupMarker(const char *patt_name, int *patt_id, ARHandle *arhandle, ARPattHandle **pattHandle)
{ if ((*pattHandle = arPattCreateHandle()) == NULL)
{ fprintf(stderr, "setupCamera(): Error: arPattCreateHandle.\n");
return (FALSE); }
// Loading only 1 pattern in this example.
if ((*patt_id = arPattLoad(*pattHandle, patt_name)) < 0)
{ fprintf(stderr, "setupMarker(): pattern load error !!\n"); arPattDeleteHandle(*pattHandle);
return (FALSE); }
```

```
arPattAttach(arhandle, *pattHandle);
return (TRUE); }
```

mainLoop

This is the routine where the bulk of the ARToolKit function calls are made and it contains code corresponding to steps 2 through 5 of the required application steps. First a new video frame is requested using the function arVideoGetImage. If the function returns non-NULL, a new frame has been captured, and the return value points to the buffer containing the frame's pixel data, so it is saved in a global variable.

```
// Grab a video frame.
if ((image = arVideoGetImage()) != NULL)
{ gARTImage = image; // Save the fetched image.
gCallCountMarkerDetect++; // Increment ARToolKit FPS counter.
```

Every time a new frame has been acquired, it needs to be searched for markers. This is accomplished by a call to the function arDetectMarker(), passing in the pointer to the new frame, and an ARHandle. The ARHandle holds the ARToolKit marker detection settings and also stores the results of the marker detection.

```
// Detect the markers in the video frame.
if (arDetectMarker(gARHandle, gARTImage) < 0)
{ exit(-1); }
```

The results of the marker detection process can now be examined - to check whether they match the IDs of the marker(s) loaded earlier. Of course, in simpleLite - need to check for one marker, the Hiro marker. A value known as the marker confidence is used to make sure that the Hiro marker is obtained and not a marker with a different pattern.

```
// Check through the marker_info array for highest confidence //
visible marker matching our preferred pattern.
k = -1; for (j = 0; j < gARHandle->marker_num; j++)
{ if (gARHandle->markerInfo[j].id == gPatt_id)
{ if (k == -1) k = j; // First marker detected.
else
if (gARHandle->markerInfo[j].cf > gARHandle->markerInfo[k].cf) k = j;
// Higher confidence marker detected.
} }
```

At the end of this loop, if k has been modified, then the marker containing the Hiro pattern is found, so the last task to be performed by ARToolKit on the marker is to retrieve its position and orientation (its "pose") relative to the camera. The pose is stored in an AR3DHandle structure. If the marker is not found, note that fact- because if no markers are found, 3D objects should not be drawn in the frame.

```
if (k != -1)
{ // Get the transformation between the marker and the real camera into gPatt_trans.
err = arGetTransMatSquare(gAR3DHandle, &(gARHandle->markerInfo[k]),
gPatt_width, gPatt_trans);
gPatt_found = TRUE;
}
else
{ gPatt_found = FALSE;
```

}

Finally.. since there is a new video frame, request the operating system call our Display function: glutPostRedisplay();

Display

This program run two loops in parallel.. one (in mainLoop()) grabs images from the camera and looks for markers in them. The other loop displays images and 3D objects over the top of detected marker positions. or other AR-related content that is to be drawn. These two loops run separately, because the operating system separates drawing from other regular tasks, to work more efficiently. In simpleLite, the drawing all happens in the function named Display(). (This gets called by the operating system via GLUT).

In the display function, several steps are done:

- Clear the screen and draw the most recent frame from the camera as a video background.
- Set up the OpenGL camera projection to match the calibrated ARToolKit camera parameters.
- Check whether any active marker is present, and if so, position the OpenGL camera view for each one to place the coordinate system origin onto the marker.
- Draw objects on top of any active markers (using the OpenGL camera).

Step 1: Clear the screen and draw the most recent frame from the camera as a video background:

```
// Select correct buffer for this context.
glDrawBuffer(GL_BACK);
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
// Clear the buffers for new frame.
 arglDispImage(gARTImage, &gARTCparam, 1.0, gArglSettings); // zoom = 1.0.
gARTImage = NULL;
```

The video image is this displayed on screen. This can either be an unwarped image, or an image warped to correct for camera distortions. Unwarping the camera's distorted image helps the virtual 3D objects appear in the correct place on the video frame.

Step 2: Set up the OpenGL camera projection to match the calibrated ARToolKit camera parameters.

```
// Projection transformation.
arglCameraFrustumRH(&gARTCparam, VIEW_DISTANCE_MIN,
VIEW_DISTANCE_MAX, p); glMatrixMode(GL_PROJECTION);
 glLoadMatrixd(p);
glMatrixMode(GL_MODELVIEW);   // Viewing transformation. glLoadIdentity();
```

The call to arglCameraFrustumRH converts the camera parameters stored in gARTCparam into an OpenGL projection matrix p, which is then loaded directly, setting the OpenGL camera projection. With this, the field-of-view, etc. of the real camera will be exactly matched in the scene.

Step 3: Check for any active markers, and if so, position the OpenGL camera view for each one to place the coordinate system origin onto the marker.

```
if (gPatt_found)
{
// Calculate the camera position relative to the marker.
// Replace VIEW_SCALEFACTOR with 1.0 to make one drawing unit equal to 1.0 ARToolKit units (usually millimeters).
arglCameraViewRH(gPatt_trans, m, VIEW_SCALEFACTOR);
glLoadMatrixd(m);
```

arglCameraViewRH converts the marker transformation (saved in mainLoop) into an OpenGL modelview matrix. These sixteen values are the position and orientation values of the real camera, so using them to set the position of the virtual camera causes any graphical objects to be drawn to appear exactly aligned with the corresponding physical marker. The virtual camera position is set using the OpenGL function glLoadMatrixd(m).

Step 4: Draw objects on top of any active markers (using the OpenGL camera).

Finally, The last part of the code is the rendering of 3D object, in this example the OpenGL colour cube. This function is simply an example and can be replaced with any drawing code. Simply, at the time the draw function is called, the OpenGL coordinate system origin is exactly in the middle of the marker, with the marker lying in the x-y plane (x to the right, y upwards) and with the z axis pointing towards the viewer. If you are drawing directly onto a marker, remember not to draw in the -z part of the OpenGL coordinate system, or else your drawing will look odd, as it will be drawn "behind" the marker.

cleanup

The cleanup function is called to stop ARToolKit and release resources used by it, in a clean manner:

```
static void cleanup(void)
{ arglCleanup(gArglSettings);
arPattDetach(gARHandle);
arPattDeleteHandle(gARPattHandle);
arVideoCapStop();
ar3DDeleteHandle(gAR3DHandle);
arDeleteHandle(gARHandle); arVideoClose(); }
```

Cleanup steps are generally performed in reverse order to setup steps.

Steps in cleaning up ARToolKit.

Cleanup step	!Functions to be called.
remove ARToolKit's connections to OpenGL	arglCleanup
release resources used in the marker tracking	arPattDetach, arPattDeleteHandle, ar3DDeleteHandle, arDeleteHandle
stop the video capture and close down the video path to free it up for other applications	arVideoCapStop, arVideoClose

QUESTIONS

Part – A

1. Illustrate Augmented Reality with an example.
2. List the applications of Augmented Reality.
3. Associate Augmented Reality with other technologies.
4. Assess the camera parameters involved in camera calibration.
5. Identify the major hardware and software components of AR system.
6. Distinguish marker-based and marker-less Augmented Reality.
7. Point out the importance of image quality in marker-based AR application.
8. Interpret how pattern recognition techniques are related to AR.
9. Name the platforms supported by AR Toolkit.
10. Compare and Contrast the advantages and disadvantages of AR Toolkit.

Part – B

1. Sketch down the system structure of Augmented Reality with a neat diagram and Explain.
2. Breakdown the techniques involved in Camera parameters and Camera Calibration.
3. Summarize the working behind Marker-based Augmented Reality.
4. Discuss the brief history of Augmented Reality Software development.
5. Design the steps involved in AR Toolkit to develop a simple application.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

School of Computing
Department of Computer Science and Engineering
UNIT - V

AUGMENTED AND VIRTUAL REALITY – SCSA3019

UNIT V APPLICATION OF VR IN DIGITAL ENTERTAINMENT

VR Technology in Film & TV Production.VR Technology in Physical Exercises and Games. Demonstration of Digital Entertainment by VR.3D user interfaces - Why 3D user interfaces. Major user tasks in VE. Interaction techniques for selection, manipulation and navigation.3DUI evaluation.

Applications of Virtual Reality in Entertainment include -

- Video games
- Virtual Museums
- Galleries
- Theatre
- Virtual theme parks
- Music VR experience

VR and the Entertainment Industry

One main industry benefiting from the virtual reality is the entertainment business. The human need to continually seek new and innovative ways to relax and energise leads many to the theatre or concert hall. The music industry uses VR to allow those in rural areas far from concert venues or those with certain diseases rendering exposure to crowds and loud noise painful to enjoy their favourite artists even when they cannot attend live. VR concerts give fans around the world a front row seat to some of the hottest bands for a fraction of the price of VIP admission. Virtual reality technology possesses the ability to bring dead artists back to life, at least in fans' earbuds or on a screen.

The Future of VR and the Entertainment World

As technology advances, the entertainment world will inevitably change with it. Instead of wearing funky cardboard red-and-green glasses to watch a 3D movie, theatre-goers will don VR headsets and truly immerse themselves in the action from scene one to the closing credits. The movement to bigger, better, more realistic action in movie and video game markets opens a world of opportunity for investors to collaborate on luxurious new entertainment venues. Many theatres have already added amenities such as luxurious-comfortable seating, improved snack selections and even table-side meal service. Introducing a VR component by adding viewing rooms replete with headsets or even movie seats that rock and slide along with the on-screen action will delight audiences. Staffed VR arcade rooms can give busy parents a break so they can catch a flick while their children play. If used correctly, VR technology holds the power to transform the entertainment world.

I. VR Technology in Film & TV Production

In recent years, VR virtual reality has gradually become a hot topic in society. As a medium integrating computer technology, imaging technology and human-computer interaction technology, VR brings both a new interactive narrative method in the evolution of media and new means of communication in the era of text change. The application of VR technology in the late stage of film and television has promoted the development of diversified creation and the development of film and television post production. At the same time, the application of VR technology has improved the efficiency of film and television production, reduced the cost of physical setting, and saved resources.

With the good word-of-mouth and display effects achieved by the broadcast of some science fiction movies around the world, people have begun to apply modern technologies such as IMAX and 4K to the production of film and television. Among them, the emergence of VR virtual technology has also officially become a new entry point for film and television art application technology after 3D technology. Many well-known film and television companies apply VR virtual technology in film and television production, which fully reflects the development prospects of VR virtual technology. In the exploration of VR image production and the corresponding industry talents, some domestic and foreign companies and universities have taken the lead in the industry. Although many Chinese film and television production companies use VR virtual technology, the application of VR in film is still in the state of exploration and development. Throughout the history of film development, film art and technology are closely linked. The development of virtual reality in the film field has created the glory of contemporary film. The combination of virtual reality technology and film art has changed the form and nature of film images, thereby affecting the artistic effect of films.

With the advent of the digital age, more and more films are using VR in the production process of movies, especially in surreal movies such as science fiction and magic. Virtual reality technology has made movies have an unprecedented shocking effect on the audience in terms of audiovisual experience, and even to a certain extent is more imaginative and artistically appealing than traditional movies. However, compared with Hollywood in the United States, the level of industrialization of domestic films is relatively backward, and the technological innovation and application capabilities are obviously insufficient. The full advantages of virtual reality technology have not been fully utilized.

The application of VR virtual reality in film and television post-production has made the film and television industry to diversify the creation of works, further promote the development of film and television post-production, and improve the overall level of post-production. At the same time, the application of VR technology improves the efficiency of film, reduces the cost of real scene setting, and saves resources.

VR technology is a three-dimensional virtual environment built using computer technology. This technology can make movies, animations and games into three-dimensional, so that the user experience will be improved. Users can have an immersive feel to movies and games. Under the technology that realizes visual 3D, the sound also achieves 3D effects. The surrounding sound also has a 3D feel. With the support of such technology, users are very fond of film and television works processed by VR technology, which also indirectly makes animation and digital imaging in colleges and universities. The synthesis profession has paid more and more attention to VR technology in the post-production of film and television. In fact, VR technology actually increases the human-computer interaction experience, making users more and more realistic about using machines, watching movies and playing games with an immersive feeling, such technology not only aroused People's interest in movie games and other activities can also make movie special effects and game feel more cool. All in all, the emergence of VR technology has directly subverted people's views on a series of activities such as watching movies and playing games. The previous activities were just the users watching, but the current activities are the users' feelings. Such an experience is deeply loved by people. The application of VR technology is very extensive. In addition to film post-production and 3D effects of games in the film and television industry, there are also some commonly used applications, such as virtual test makeup and dress-up applications on some shopping apps.

Film and Television Post Production

The post-production work of film is mainly to process the various materials that have been shot using unique technology, and cut multiple shots together into a complete film. In the production of film, post-production plays a vital role. It integrates the pre-production and improves the work efficiency to ensure the quality of film and television works. The post-production of film and television needs to wait for the completion of film and television production, and then use the computer production software to complete the editing and processing of the film. Film and television post-production is a relatively complicated link, which involves a lot of production processes. Use film and television post-production to add a few special effects, and edit and piece together the previously filmed video clips to finally present a complete film and television work.

The main steps of film and television post-production can be divided into the following three points:

(1) Editing of the lens

In the post-production of film and television works, the editing of the lens is the most basic. This part of the work is to get the various shots in the film and television works to be clipped and then pieced together, so that the lenses in the film and television works have a rough arrangement. Drawing on the guidance of film producers, getting regular cuts of film and television productions, and then reorganizing them, arranging the shots with no central idea or order at first into a logical, orderly and organized Storytelling footage are the parts of post- production of film. The director's ideas have a direct bearing on the pros and cons of a film and television work, and of course the editor's skills are also very important.

(2) Sound editing

An important component of film and television works is sound. In the post-production process, the sound should be processed while editing the lens, such as soundtrack and dubbing. At present, there are two kinds of dubbing methods. First, simultaneous recording, and when editing the sound later, the edited sound should match the edited picture, so that the dubbing can be complete and synchronized with the story picture and development in the work; Second, post-dubbing. This kind of dubbing does not require too much technical content. Usually, the lens is cut first and then dubbed by the dubbing staff. In the later stage, only the volume needs to be simply adjusted. The soundtrack of an excellent film and television work is often more attractive. It needs to be combined with the theme of the story to render the atmosphere of the entire play, and it will help to create a situation.

(3) Synthesis of special effects

In the post-production of film, a very important environment is the production of special effects. The development of special effects has a long history. Currently widely used is a film and television production technology that is gradually moving towards high-end. In the production process, in order to achieve the ideal artistic effect, some advanced special effects technology must be used, so 3D technology appears more and more frequently. At present, 3D technology is widely used in various film and television works. VR virtual reality technology is a three-dimensional virtual environment built using computer technology. This technology can make movies and animations into three-dimensional, give users a immersive feeling, and let the audience have a better viewing experience.

(1) VR virtual reality improves film efficiency

The design of VR virtual scene and the application of virtual shooting system have greatly improved the shooting efficiency, saved the crew transition time, and the shooting is not affected by weather and light, which shortens the shooting cycle. In addition, the creative space for post-production has been increased. The virtual scene can be adjusted later, which changes the traditional "shoot-and-make" mode. Although the post-production workload has increased relatively, using "post-front" can also compress. The post-production process can save the time of the virtual camera tracking and matching. At the same time, the on-site shooting departments can monitor the pre-compositing screen in real time to avoid scheduling confusion.

(2) VR virtual reality technology reduces movie production costs

The use of VR virtual reality technology to design virtual movie scenes can greatly save the material costs and labor costs of real scene scenes, and save resources. Because the physical setting is generally not recyclable, it is often demolished after shooting, which wastes resources and is not conducive to environmental protection. The virtual movie scene saves the achievement of physical setting.

(3) Problems in VR post-production in film and television

The design of VR virtual scenes and the application of the virtual preview shooting system have increased the technical staff of on-site shooting to a certain extent, which is relatively more complicated and demanding than the traditional shooting process, and the cost of post-production will also increase. However, as a whole, the production cost is reduced, but it is not suitable for all types of film and television production. VR virtual reality technology should be combined with the needs of the film itself, do not use VR technology as a gimmick for film and television. Only with a reasonable fusion of art and technology can VR virtual reality technology develop and progress in the film and television industry.

The main purpose of the production of film is to present an exquisite film and television work to the audience, so that the audience's demand for film and television work is fully satisfied. The application of VR virtual reality in film and television post-production has made the film and television industry to diversify the creation of works, further promote the development of film and television post production, and improve the overall level of post-production. Although the bottleneck of VR virtual reality in film and television post-production still exists, its advantages are also quite obvious. It improves the efficiency of film and television production, saves the setting cost, and avoids waste of resources. As long as the reasonable integration of art and technology, and according to the needs of the film itself, VR virtual reality technology can get better development and progress in the film and television industry.

II. VR Technology in Physical Exercises and Games

Video Games as Physical Education

Virtual reality technology, which turns users into active participants, is dramatically changing the way kids play video games. In a VR game, a user can play a sport or dance as part of the game — which means they actually move their body, not just their pupils and thumbs. The Valley Day School in Morrisville, became the first in the country to install a high-tech, state-of-the-art virtual gym, complete with camera sensors and stereo sound, 3D projectors and other gaming accessories.

The immersive VR setup transforms PE class into a "life-size video game" for the students. Improving the level of physical activity in children through VR-based games is welcome news for parents and educators because video games have been singled out as a main culprit responsible for the plummeting levels of physical activity among children. The Centers for Disease Control and Prevention noteg that 1 out of 6 children and teenagers in the country are obese. "In 2014, the CDC identified hours playing video games as one of the risk factors for low physical activity in the United States". But adding VR to the PE mix flips this scenario on its head. It is been stated that, "the inherent movement in virtual reality and augmented exercise make it possible that video games may soon be a positive contributor to physical fitness. And the popularity of video games as one of the world's most sustained and growing pastimes may make it a great ally for those that traditionally struggle with staying fit".

VR Exercise Games Tackle Obesity and Aid the Disabled

The general public is also looking more at using VR games to boost exercise, according to The New York Times. Some people have injuries or disabilities that prevent them from traditional forms of exercise, and anecdotal evidence suggests that VR video games with an exercise component can help them maintain their fitness levels. A 2019 study by the Journal of Special Education Technology called such games "a promising tool" to help kids achieve the recommended minimum daily amount of 30 minutes of moderate physical activity. Among the more popular games are Oculus Quest's Beat Sabre, Box VR, The Thrill of the Fight, SoundBoxing and Holopoint, to name a few. Another study cited in the journal's study noted that "physical activity is a key factor in preventing health problems that result from leading a sedentary lifestyle and can positively impact the health, fitness, and behavior of adults and youth."

The Virtual Reality Institute of Health and Exercise and the Kinesiology department at San Francisco State University have teamed up to develop the VR Health Exercise Tracker "built on hundreds of hours of VR-specific metabolic testing using research-grade equipment." The tracker collects metabolic data, including number of calories burned.

A good personal computer, VR headset (ranging in price from $350 to $800) and 6 square feet in which to move around are all a user needs to play these video fitness games, reported CNN. That means many institutions can easily make space for gamers and spur students to get off their mobile phones and tablets — and students won't even realize they are exercising.

Is Virtual Reality the Future of Exercise?

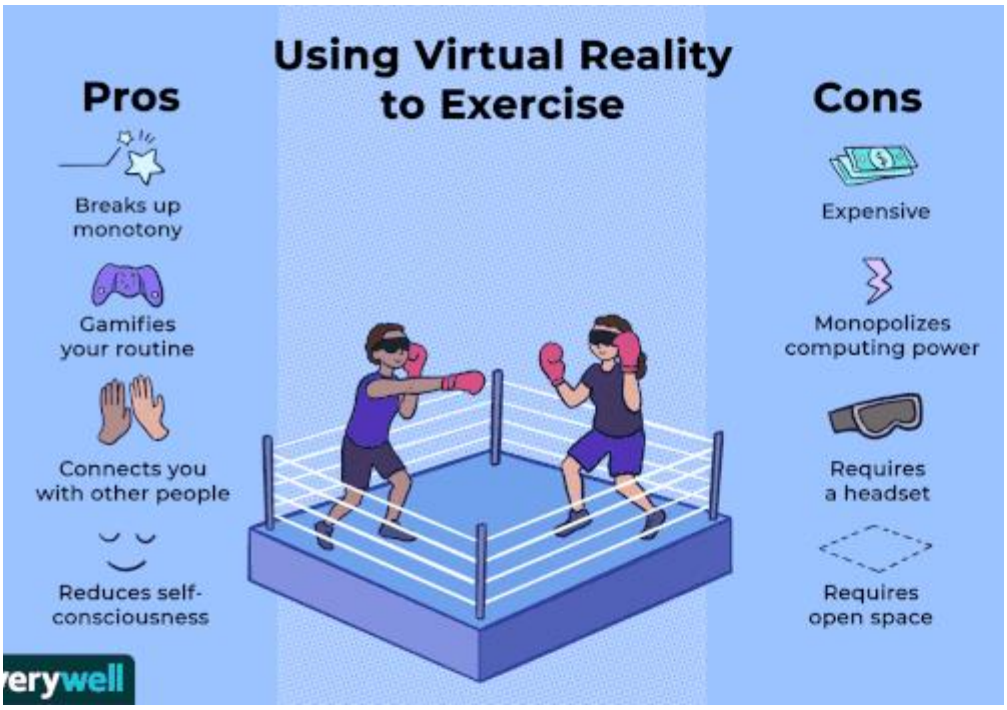


Fig. 5.1 Pros and Cons of using VR to Exercise

There's probably not a person on this planet who hasn't wanted to escape their reality in the middle of a tough workout. Huffing and puffing on a stationary bike (or some other piece of equipment) surrounded by hordes of other sweaty people counting down the minutes until they've met their goals, is rarely an exciting experience.

And while true virtual reality platforms—where a person's environment is completely replaced by a digital one—are still barely scratching the surface of the fitness world as of 2019, the industry is growing, and for good reason. "In 2018 VR is estimated to have grown 30%, largely due to the popularity of the PlayStation VR. The launch of (Facebook's) Oculus Quest (in 2019) is expected to be another huge leap forward," says Jordan Higgins, Head of Immersive Experience at U.Group, where its emerging tech incubator, ByteCubed Labs owns PRE-GAME PREP, a holographic football training system.

"Fitness applications are more supplemental than they are a primary form of exercise, but the novelty can do a lot to break up a monotonous workout routine." The idea being that you can be transported to a beautiful location like the Swiss Alps for your next stationary bike ride, rather than having to stare at the sweaty back of a fellow gym-goer. But it's not just the ability to "escape" the real world that makes virtual reality platforms an exciting new world for exercise. Due to the internet-connected nature of VR headsets and other devices, tracking and monitoring key health metrics also becomes more accessible for users. "New data points can be tracked for better insight into performance". "And combining immersive VR with other wearable sensors like the Apple Watch, you can really start to see a connected ecosystem that will drive the next generation of immersive fitness."

Pros of Using Virtual Reality for Fitness

As with any exercise trend, there are always benefits and drawbacks to jumping on the bandwagon. Given that virtual reality is more of a platform for different types of exercise than a form of exercise in-and-of-itself, it should be considered a generally safe and reasonable option for those who want to try it out. It should come as no surprise that it's going to be more appealing to those who are interested in gaming, technology, and tracking performance metrics than those who try to live their lives in a more disconnected, "off-grid" fashion. That said, everyone should consider the pros and cons before diving in headfirst.

Breaking Up Monotony

The most dedicated exercisers seem to have no problem hitting the gym every day, doing more-or-less the same workout, and continuing the trend ad nauseam. Trainers like to tout the importance of creating a habit, developing internal motivation, and looking for intrinsic rewards to help you continue to exercise. The reality is, though, that most people struggle with this no-nonsense approach. Doing the same workout day-in and day-out can lead to boredom and disenchantment with exercise. Due to the internet-connected nature of VR headsets and other devices, tracking and monitoring key health metrics also becomes more accessible for users.

"Gone is the drudgery of doing the same old workout routine in a crowded and stinky gym," says Mat Chacon, the CEO of VR company Doghead Simulations, and a VR-exercise advocate who went from fat-to-fit by working out in VR. "People can use VR to transport to any environment they desire—the moon, the beach, under the ocean, or wherever they want. Using VR, people can even load a 3D model of a boxing ring and practice ducks and slips and then do yoga in a Japanese zen garden without ever leaving their home." The endless opportunities for novelty in workouts, environment, and even the people you choose to interact within a connected, online world is one of the greatest benefits for those who often grow bored with the same old routine.

Gamification

The beauty of exercising in a 3D virtual environment is that you can move and interact within games themselves. These technologies, which essentially turn the workout experience into a virtual competition—with yourself or with others—can keep exercise fun while helping to distract you from the work you're actually doing. As of 2019, the number of full-fledged VR games designed specifically for exercise is relatively low, but there are games that, by their very nature, require full-body movement within the virtual environment, turning them into a workout. "I would argue the best, and most viral form of exercise mixed into gaming would be from the game Beat Saber," says Steve Kamb, the founder of NerdFitness, a website dedicated to helping nerds and gamers get fit. "Think of it like Guitar Hero, except it's immersive, and you're playing drums and dodging blocks, and getting a really solid cardiovascular experience." But Beat Saber isn't the only option—Kamb mentions Creed: Rise to Glory and First Person Tennis as other popular options—and as the interest in VR for exercise grows, developers will continue bringing more new games to the market.

"Gymtimidation" is a very real experience for people new to exercise or those who are trying a workout for the first time. For any number of reasons, you may feel self-conscious or out-of-place, whether it's because you're thinking about your body or fitness level, you don't know how to use the equipment or do the exercises, or you just don't know anyone at the gym. VR pretty much wipes out all those concerns. VR provides a support network and avatars actually give people a 'mask' to hide any insecurities and fears that might be preventing them from going to a more traditional gym or fitness class. People may also open up more than they might in a face-to-face setting or within a video chat.

Connection with Others

Given that virtual reality platforms and games are connected to the Internet, you can quite literally connect with other users all over the world. And not just other gamers. "You can interact with coaches in Brazil, meal planners in New York, or yoga instructors in Mumbai," Chacon says. "You can see each other and interact as though you're all in the same place at the same time. It's nice to high-five fellow VR workout participants and encourage each other to keep going."

No Waiting for Equipment

Most people use VR systems within the comfort of their own homes, which means they can simply log in and start exercising whenever they have time. This beats having to sign up early for a popular class or waiting in line for a treadmill or bench press during peak hours at the gym.

Cons of Virtual Reality and Exercise

Cost of VR Systems

The earliest iterations of VR devices, like most new technologies, were incredibly expensive, making purchase by the general public largely infeasible. But as technology has improved and more companies have entered the market, the cost of tethered and untethered systems continues to drop.

As of 2020, the Oculus Quest ranges from $399 to $499, the pared-down Oculus Go ranges from $199 to $249, PlayStation VR bundles start at $299, and HTC Vive prices start at more than $600. Of course, these are consumer-friendly headset systems that don't offer some of the features that more extensive VR systems include (some use full bodysuits and specialized treadmills), but they still aren't a price everyone can afford.

Certainly, you wouldn't want to throw down several hundred dollars for a VR system if you weren't completely confident you'd enjoy your exercise experience.

Computing Power

If you're electing to use computer-based VR systems, you need to make sure your computer's specs measure up, and if you're running VR through your home internet connection, you need to make sure you have enough bandwidth to load the graphics and keep the system running seamlessly. "You should make sure you have enough horsepower to run the VR software you like," says Jeanette DePatie, a certified fitness trainer who has spoken at tradeshows like CES about virtual reality and other technologies. "VR requires massive amounts of computing power, so you probably won't be able to run current or future games on your computer unless it's souped-up with the latest processors and graphics chips." - Jeanette DePatie

Wearing a Headset While Exercising

To replace your real-world environment with a virtual 3D environment, you have to wear a headset. Some of these headsets are tethered to an external system, which means you have to work around a physical tether, while other headsets are stand-alone. These stand-alone headsets are either connected wirelessly to an external system or are stand-alone units that enable you to interact in a virtual environment without having to stay within range of connected sensors.

Regardless, the headset is a requirement of virtual reality exercise, and it's not something that will appeal to everyone. Wearing a VR headset isn't necessarily that different from wearing a heavier pair of snowboard goggles, but the headset will naturally restrict your view of your real-world environment, and to stay secure, you'll need to secure them snugly to your head. If either of these requirements sounds uncomfortable, you may not want to test the waters of VR exercise.

All the Sweat

Even if you're comfortable with the idea of wearing a headset while exercising, you need to remember that you'll be wearing a headset while sweating. "I can't stress this enough—invest in cleaning wipes and a headset cover," says Higgins. The wipes keep your headset clean and hygienic for each successive workout, and the headset cover makes it easier to clean and more comfortable to wear.

Space Requirements

Wearing a headset while exercising can be a difficult while doing workout. If the view of your living room is blocked out and replaced with an empty boxing ring, you may inadvertently run into your coffee table or trip over your dog. Whenever possible, set up your VR system in a wide-open environment that's unlikely to be interrupted by other people or animals during your workouts.

Virtual Reality Gyms

If you're not ready to buy your own VR system, but you'd like to see what it's like to work out in a virtual environment, keep an eye on the gyms opening up in your area. The first-ever full-fledged VR gym, Black Box, opened in San Francisco in 2019, and entrepreneurs in other large cities are likely to follow suit, proving that VR really might be the future of exercise.

Running virtual:

The effect of virtual reality on exercise Research has shown that exercise among college aged persons has dropped over recent years. Many factors could be contributing to this reduction in exercise including: large workloads, the need to work during school, or perhaps technology use. A number of recent studies are showing the benefits of using virtual reality systems in exercise and are demonstrating that the use of such technology can lead to an increase in the number of young adults engaging in exercise. This study focuses on the effects that virtual reality has on heart rate and other bodily sensations during a typical work out. This study also analyses the participants' ability to pay less attention to their bodily sensations during exercise when using a virtual reality system. During this experiment, participants were exposed to two different conditions. Condition one being a traditional work out, riding an exercise bike at a middle tension level. Condition two was the same but the participant was wearing a virtual reality headset. The data collected led to the conclusion that working out while wearing a virtual reality headset will lead to a higher heart rate, and in turn can lead to burning more calories during a workout. The study also found participants who wore the virtual reality headset were able to remove themselves from their bodily sensations allowing them to workout longer. Virtual reality fitness can be a great way to build fitness confidence. (And having great workouts available from home is even more important now since many gyms still remain closed due to the COVID-19 pandemic.)

Turn Your Workout into a Game: VR and the Future of Fitness

Digital technology has already transformed fitness. Your smartphone counts your daily steps; lightweight, wireless-enabled watches and other "wearables" can monitor your heart rate and vital statistics; and gym equipment with built-in workout tracking and video monitors can take your workout to anyplace on the planet. Before you take a breather from your virtual at-home spin class or let your smart watch sync with your sleep app and hourly analytics, make room for virtual reality (VR). Not only will VR enhance the overall workout experience, but it will also address obstacles to exercise — a lack of motivation and our natural preference to conserve energy and avoid activity. Because VR is emerging as the next big computing and consumer platform (following the emergence of the internet and mobile devices), it should be no surprise to see VR driving innovation across industries and use cases, such as architecture, mental health and education.

The VR Opportunity

Ryan DeLuca and Preston Lewis spent 17 years growing a leading e-commerce site that provided information about sports and fitness and sold nutritional supplements. It became very successful, with more than 30 million fitness enthusiasts visiting it each month. In 2015, the duo decided to start a new venture combining their loves for fitness and technology. After three years, Black Box VR launched, a virtual reality gym. They sign up for the gym in January and stop going in March. They lapse. They stop showing up, even once a week. Even the most advanced scientists and doctors haven't figured out a solution to this problem.
The traditional gym is being changed into a game –
Black Box VR isn't the only one making moves into the VR fitness market.

- WalkOVR has created a system of sensors that attach to the knees, ankles and torso to record lower body movement, making it possible to run in virtual environments while staying put in reality. The product was designed with fitness in mind but is also compatible with games that are not necessarily fitness related.

- BoxVR received VR Fitness Insider's 2017 Best VR Fitness Game of the Year for its at-home VR boxing workout where the user punches to a rhythmic beat. Developed by fitness instructors, it's like a VR Tae-Bo video, minus the three easy payments of $19.99 plus shipping and handling.
- VirZOOM's "VZfit Sensor Kit" attaches to any stationary bike to turn it into a VR cycling experience. In the VR world, the cyclist can bicycle through real destinations or fly Pegasus through a canyon.

Turning workouts into a game is a genius move — capitalizing on our human need for instant rewards and achievements from a game versus waiting days or even weeks to see physical results from a workout.

A key benefit of a VR workout is consistency and tracking. Entire markets are devoted to tracking workouts; you may be wearing one on your wrist right now. Virtual reality hardware is designed to track movements to enable the user to interact with the virtual environment. These sensors and accelerometers can track even the most minute movements, making it a very efficient medium for tracking a workout. The effort that was put into VR workout will have immediate in-game rewards and long-term health benefits. Sure, it may feel like as if the users are playing a game and having fun.

III. Demonstration of Digital Entertainment by VR

People spend a lot of time and money on video games, social networks, cinema, amusement parks, music concerts, and sports games. Most likely, virtual reality will not replace these entertainments, but it can make them more inclusive and immersive. In the last few years, using virtual reality for entertainment has been mainly experimental. Now the VR entertainment market is entering the commercial stage and boasting some profitable projects. Various virtual reality entertainment options are discussed that will even make proper investment decisions.

The Venture Reality Fund states that the total investment in the VR industry reached $2.3 billion in 2017. Almost half of all investments in virtual reality fall on the entertainment sphere. According to Kaleidoscope's research, in 2017, more than $1 billion was invested in VR entertainment. The Kaleidoscope's experts forecast that in the coming years, investors will have a significant choice of virtual reality projects that can bring income. An increase in the creation of more lifelike VR entertainment experiences will engage more consumers and boost the commercial success of the virtual entertainment industries.

The VR gaming market generates the highest income compared to other virtual reality entertainments. In 2017, the virtual reality gaming industry made $2.2 billion. At the same time, more than 35 games have earned over $1 million, which is an indicator of healthy competition and the potential of the VR gaming market. PlayStation, Nintendo Wii, and Xbox virtual reality games bring the highest revenue. The consoles attract hardcore players, who are willing to pay for expensive lifelike VR games. Not all players can afford to buy a high-end virtual reality headset and VR controller. However, budget virtual reality glasses, such as Google Cardboard Glasses, allow them to try VR apps with minimal costs. Also, arcades – location-based VR entertainment centers, help promote virtual reality games. In the VR arcade, a user can enjoy immersive experiences for a reasonable fee.

Virtual reality has an extensive application. With its help, traditional types of entertainment can take a new dimension.

Virtual reality is often blamed for leading people to isolation. On the other hand, it can also unite people in virtual worlds. VR worlds are online virtual reality social platforms where people can interact with each other. Communication in the virtual spaces is much like communication in the real world, but it provides almost unlimited possibilities in the choice of settings and ways of spending time. Neurons Inc conducted a survey of USA residents who have never used virtual reality and describe themselves as the late adopters. This study of social interaction revealed that 59% of the respondents consider virtual reality as the desired way of communicating with friends and family who are far away. Interest in VR worlds is fueled by the IT market giants, planning to build a strong VR community. For example, Facebook is now actively promoting its social media VR app Facebook Space. Microsoft also believes in the great future of virtual networks. That's why Microsoft acquired a popular social VR platform AltspaceVR. At the moment the prospective areas for the development of virtual reality worlds are more photorealistic three-dimensional avatars and the creation of exciting and detailed locations.

Theatre

The incredible success of the New York show Sleep No More prompted tremendous interest in the immersive theatre. An immersive or interactive theatre is a dramatic performance, where there is no traditional stage, and the audience is involved in the action. Now the producers of the interactive theatre shows are looking for ways to further immerse the audience into action. One such method is virtual reality. Technical capabilities of VR can accurately convey the main features of the immersive theatre, such as narrative, immersion, and interactivity.

Greenlight Insights research reveals that consumers are most interested in the VR theatre among all virtual reality entertainment. The survey covered more than 2,000 United States residents of different gender, age, and social standing. 66% of respondents were willing to visit the VR theatres. The combination of theatre and virtual reality can create a successful business model. The theatrical VR content supposes clear ways of monetization and can bring stable income for several years, unlike the VR movies, which become irrelevant very quickly due to the low involvement of viewers to the action.

Cinema

The location-based virtual reality entertainment industry is about to grow up to $825 million by 2021 according to Greenlight Insights' forecasts. Cinema can become a leader in this emerging market. Currently, the most successful adept of VR cinema is IMAX. Let's look at how IMAX runs VR movie theatres to reveal the significant factors for their success. It is crucial to choose proper hardware. For example, IMAX applies, among others, the StarVR headsets, which are not available for home use. Thus, it helps to attract VR-enthusiasts who want to try cutting-edge technology. Unique content can also be a prominent competitive advantage. For instance, Justice League VR – the blockbuster created in the Warner Bros. partnership attracted a mass audience in Imax VR centers. The popularity of multidimensional cinemas proves that viewers want to immerse themselves in the movie. Further development in this direction requires the creation of new forms of movie storytelling, which can involve a VR viewer in the action and make the VR movie experience truly immersive.

Museum

Usually, a museum is considered to be entertainment for academics and does not attract a wide audience. With the virtual reality technology, you can engage a large audience to visit the museum. Museums can use virtual reality apps for the location-based virtual reality entertainment. For example, the British Museum used VR devices to engage adults and children with their Bronze Age collections. Visitors in the virtual reality headset could walk through the ancient landscape and interact with the artifacts using a VR controller. Many museums create their virtual reality applications for desktops and mobile devices. After the launch of such virtual applications, museum representatives note an increase in the museum attendance.

Amusement Park

Amusement parks are designed to give people an unusual experience, entertainment, and exciting sensations. This market has very tough competition, with the increasing difficulty to amaze visitors. The combination of virtual reality and rides creates a unique experience and can highlight your amusement park amidst traditional entertainment. Theme parks are trying to carry visitors to an unusual setting. A virtual reality headset truly immerses a user in another world. Moreover, it is much cheaper than the creation of material objects and visual effects. You can create several temporary versions of virtual applications, for example, for Halloween or Christmas, and provide users with the most relevant experience. Usually, consumers enjoy VR attractions, though advanced users often expect more interaction. Also, do not forget that rides can cause vertigo or sickness. Therefore, it is necessary to take care that virtual reality headsets you use are high-quality and comfortable for visitors.

Gallery

Creating a VR application for a gallery is a long-term investment, which will help you gain customer attention and build your loyal audience. Let's see how you can use VR in the gallery business. The most apparent use of VR is a virtual tour around the gallery. Such virtual reality tours allow people to enjoy the art without having to stand in lines, pursue masterpieces, and sometimes even interact with them. In addition to this, galleries are experimenting with the development of complex interactive VR applications, for example, where the user can create compositions using some patterns. The Tretyakov Gallery in Moscow created such a VR app to engage their wide audience with art. Wearing virtual reality glasses, app users create their paintings in the manner of famous artists and can share them on social media. Most likely, the VR application itself will not bring profit. Still, it is a powerful marketing tool to increase awareness and attract a large number of visitors to your gallery.

Live Music Concerts

Competition in the music industry forces producers and musicians to endless experiments in search of means to keep up with the trend and be attractive to the audience. Let's look at how the music industry can make use of virtual reality. Today, major music festivals such as Coachella, Lollapalooza, Tomorrowland, Sziget Festival have their VR applications or 360-degree videos. Virtual reality solutions help to scale the festivals, even more, increase their audience and earn on the sale of VR music content. Top artists, for example, Paul McCartney, U2, Björk, Coldplay, Imagine Dragons present their live performances through VR. With the virtual reality headset, spectators are virtually transported to the best seats to immerse in the concert. The VR viewers pay for watching concerts. It brings income to both the artist and the platform hosting VR concerts. A promising direction of VR music performances is fully simulated virtual reality concert. It means that three-dimensional images of artists and the environment around them are created for the show. Thus, VR viewer can visit, for example, a recording studio of the favourite artist and get an entirely new immersive experience in a musical performance.

Live Sports Games

Sports games are the other live events that are popular in a 360-degree format. A person only needs to wear a virtual reality headset to become a VR viewer. Now, to make a profit from broadcasting VR sports games, it's enough to place advertising and take a fee for the possibility of viewing the game through a VR app. Few companies are already broadcasting virtual reality professional sports games. Facebook, for instance, is now streaming VR baseball and VR football games through the platform Oculus Venues. This virtual reality streaming service promises to bring the stadium experience to everybody's home. Of course, at the moment, the emotions that the VR football viewer feels are different from those that the real viewer experiences on the field. Actions on the sports field occur very quickly, so spherical cameras are not able to capture the image and correctly transmit it to the viewer. Further development of motion capture will significantly increase the realism of VR live sports games.

Hobby Lesson

Virtual reality hobby lesson is a perfect mix of education and entertainment, which is often called edutainment. Usually, virtual reality edutainment is used as a marketing tool to promote products and build brands. VR hobby lesson can be a training video in a 360-degree format. Or it can be an interactive VR application, which is more effective in immersing the user in a virtual environment. To watch a VR video, a user only needs to wear virtual reality glasses. And to interact with the VR app, a VR controller is required as well. Virtual reality hobby lessons are great for all age categories. But so far the market has very few proposals for children. If your business is related to children products, VR app in the form of a virtual reality game for kids can help you stand out against similar companies.

Games

New genres of VR games replicate traditional game genres. Therefore, innovative VR games, in particular using specific properties of virtual reality, such as haptic feedback or recognition of smells and flavors, can make a real breakthrough in the industry. The market of virtual reality games for kids is also poorly developed. For a pleasant game experience, a child needs a special small sized virtual reality headset for kids. The development of virtual reality games for kids is no more difficult than developing games for adults. However, the competition in this market is still low, which gives certain competitive advantages. Job Simulator, developed by the Owlchemy Lab, is the game, which managed to become a hit among children and adults. This game became a bestseller and earned more than $3 million in revenue. Job Simulator encourages players to experiment for completing funny tasks, teaches positive role models and is easy to play regardless of age.

On the whole, the combination of the emotional entertainment industry and virtual reality technologies creates products with extraordinary commercial potential. Entertainment and virtual reality industries stimulate each other's growth, attracting more and more consumers. Therefore, traditional venture capital funds and innovative accelerators invest money in VR entertainment.

IV. 3D user interfaces

On desktop computers, good user interface (UI) design is now almost universally recognized as a crucial part of the software and hardware development process. Almost every computing-related product touts itself as "easy to use," "intuitive," or "designed with your needs in mind." For the most part, however, desktop user interfaces have used the same basic principles and designs for the past decade or more. With the advent of virtual environments (VEs), augmented reality, ubiquitous computing, and other "off-the-desktop" technologies, three-dimensional (3D) UI design is now becoming a critical area for developers, students, and researchers to understand.

Modern computer users have become intimately familiar with a specific set of UI components, including input devices such as the mouse and keyboard, output devices such as the monitor, interaction techniques such as drag-and-drop, interface widgets such as pull-down menus, and interface metaphors such as the desktop metaphor. These interface components, however, are often inappropriate for the non-traditional computing environments and applications under development today. For example, a wearable-computer user may be walking down the street, making the use of a keyboard impractical. A head-mounted display in an augmented reality application may have limited resolution, forcing the redesign of text-intensive interface components such as dialog boxes. A virtual reality application may allow a user to place an object anywhere in 3D space, with any orientation—a task for which a 2D mouse is inadequate. Thus, these non-traditional systems need a new set of interface components: new devices, new techniques, new metaphors. Some of these new components may be simple refinements of existing components; others must be designed from scratch. Most of these non-traditional environments work in real or virtual 3D space, so these new interfaces are termed as 3D user interfaces.

V. Why 3D user interfaces

1. 3D interaction is relevant to real-world tasks:

Interacting in three dimensions makes intuitive sense for a wide range of applications because of the characteristics of the tasks in these domains and their match with the characteristics of 3D environments. For example, VEs can provide users with a sense of presence (the feeling of "being there"—replacing the physical environment with the virtual one), which makes sense for applications such as gaming, training, and simulation. If a user is immersed and can interact using natural skills, then the application can take advantage of the fact that the user already has a great deal of knowledge about the world. Also, 3D UIs may be more direct or immediate; that is, there is a short "cognitive distance" between a user's action and the system's feedback that shows the result of that action. This can allow users to build up complex mental models of how a simulation works, for example.

2. The technology behind 3D UIs is becoming mature:

User interfaces for computer applications are becoming more diverse. Mice, keyboards, windows, menus, and icons—the standard parts of traditional WIMP (Windows, Icons, Mouse, and Pointers) interfaces—are still prevalent, but non-traditional devices and interface components are proliferating rapidly. These include spatial input devices such as trackers, 3D pointing devices, and whole-hand devices that allow gesture-based input. Multisensory 3D output technologies, such as stereoscopic projection displays, head-mounted displays (HMDs), spatial audio systems, and haptic devices are also becoming more common.

3. 3D interaction is difficult:

With this new technology, new problems have also been revealed. People often find it inherently difficult to understand 3D spaces and to perform actions in free space. Although we live and act in a 3D world, the physical world contains many more cues for understanding and constraints and affordances for action that cannot currently be represented accurately in a computer simulation. Therefore, great care must go into the design of user interfaces and interaction techniques for 3D applications. It is clear that simply adapting traditional WIMP interaction styles to 3D does not provide a complete solution to this problem. Rather, novel 3D UIs based on real-world interaction or some other metaphor must be developed.

4. Current 3D UIs are either simple or lack usability:

There are already some applications of 3D user interfaces used by real people in the real world (e.g., walkthroughs, psychiatric treatment, entertainment, and training). Most of these applications, however, contain 3D interaction that is not very complex. More complex 3D interfaces (e.g., immersive design, education, complex scientific visualizations) are difficult to design and evaluate, leading to a lack of usability. Better technology is not the only answer—for example, 30 years of VE technology research have not ensured that today's VEs are usable. Thus, a more thorough treatment of this subject is needed.

5. 3D UI design is an area ripe for further work:

Finally, development of 3D user interfaces is one of the most exciting areas of research in human– computer interaction (HCI) today, providing the next frontier of innovation in the field. A wealth of basic and applied research opportunities are available for those with a solid background in 3D interaction. It is crucial, then, for anyone involved in the design, implementation, or evaluation of nontraditional interactive systems to understand the issues

VI. Major user tasks in VE

Interaction techniques for selection, manipulation and navigation

Selection and Manipulation

The quality of the interaction techniques that allow us to manipulate 3D virtual objects has a profound effect on the quality of the entire 3D UI. Indeed, manipulation is one of the most fundamental tasks for both physical and virtual environments: if the user cannot manipulate virtual objects effectively, many application-specific tasks simply cannot be performed. Therefore, 3D interactions with techniques for selecting and manipulating 3D objects is discussed -

The human hand is a remarkable tool; it allows manipulating physical objects quickly and precisely, with little conscious attention. Therefore, it is not surprising that the design and investigation of manipulation interfaces are important directions in 3D UIs. The goal of manipulation interface design is the development of new interaction techniques or the reuse of existing techniques that facilitate high levels of user-manipulation performance and comfort while diminishing the impact from inherited human and hardware limitations.

3D Manipulation

Software components maps user input captured by input devices, such as the trajectory of the user's hand and button presses, into the desired action in the virtual world (such as selection or rotation of a virtual object). There is an astonishing variety of 3D interaction techniques for manipulation—the result of the creativity and insight of many researchers and designers. They provide a rich selection of ready-to-use interface components or design ideas that can inspire developers in implementing their own variations of manipulation interfaces.

3D Manipulation Tasks

The effectiveness of 3D manipulation techniques greatly depends on the manipulation tasks to which they are applied. The same technique could be intuitive and easy to use in some task conditions and utterly inadequate in others. For example, the techniques needed for the rapid arrangement of virtual objects in immersive modeling applications could be very different from the manipulation techniques used to handle surgical instruments in a medical simulator.

In everyday language, manipulation usually refers to any act of handling physical objects with one or two hands. For the practical purpose of designing and evaluating 3D manipulation techniques, we narrow the definition of the manipulation task to spatial rigid object manipulation—that is, manipulations that preserve the shape of objects. This definition is consistent with an earlier definition of the manipulation task in 2D UIs as well as earlier human and motion analysis literature.

However, even within this narrower definition there are still many variations of manipulation tasks characterized by a multitude of variables, such as application goals, object sizes, object shapes, the distance from objects to the user, characteristics of the physical environment, and the physical and psychological states of the user. Designing and evaluating interaction techniques for every conceivable combination of these variables is not feasible; instead, interaction techniques are usually developed to be used in a representative subset of manipulation tasks. There are two basic approaches to choosing this task subset: using a canonical set of manipulation tasks or using application-specific manipulation tasks.

Canonical Manipulation Tasks

The fundamental assumption of any task analysis is that all human interactions of a particular type are composed of the same basic tasks, which are building blocks for more complex interaction scenarios. Consequently, if 3D manipulation is divided into a number of such basic tasks, then instead of investigating the entire task space of 3D manipulation, we can design and evaluate interaction techniques only for this small subset. The results can be then extrapolated to the entire space of 3D manipulation activities. This section develops one of the possible sets of canonical manipulation tasks.

Tasks

Virtual 3D manipulation imitates, to some extent, general target acquisition and positioning movements that is performed in the real world—a combination of reaching/grabbing, moving, and orienting objects. Virtual 3D manipulation also allows users to do that which is not possible in the real world, such as making an object bigger or smaller. Therefore, the following tasks are designated as basic manipulation tasks:

Selection is the task of acquiring or identifying a particular object or subset of objects from the entire set of objects available. Sometimes it is also called a target acquisition task. The real-world counterpart of the selection task is picking up one or more objects with a hand, pointing to one or more objects, or indicating one or more objects by speech. Depending on the number of targets, distinguish between single-object selection and multiple-object selection can be distinguished.

Positioning is the task of changing the 3D position of an object. The real-world counterpart of positioning is moving an object from a starting location to a target location. Rotation is the task of changing the orientation of an object. The real-world counterpart of rotation is rotating an object from a starting orientation to a target orientation. Scaling is the task of changing the size of an object. While this task lacks a direct real-world counterpart, scaling is a common virtual manipulation for both 2D and 3D UIs. Hence, this is included as a basic manipulation task. This breakdown of the tasks is compatible with a well-known task analysis for 2D GUIs and several task analyses for VEs. Although some also include object deformation (changing the shape of an object), which is not included because 3D object deformations are often accomplished via manipulation of 3D widgets using the canonical above. Additionally, selection processes might be preceded by an exploratory task. Sometimes users explore the physical characteristics of an object (such as texture or shape) before selecting it. This may, for example, occur when an object is occluded or an interface is used eyes-off, and the actual characteristics of the object are unknown before selection.

Parameters of Canonical Tasks

For each canonical task, there are many variables that significantly affect user performance and usability. For example, in the case of a selection task, the user-manipulation strategy would differ significantly depending on the distance to the target object, the target size, the density of objects around the target, and many other factors. Some of the task variations are more prominent than others; some are stand-alone tasks that require specific interaction techniques. For example, object selections within arm's reach and out of arm's reach have been often considered two distinct tasks. Therefore, each canonical task defines a task space that includes multiple variations of the same task defined by task parameters—variables that influence user performance while accomplishing this task. Each of these parameters defines a design dimension, for which interaction techniques may or may not provide support.

Application-Specific Manipulation Tasks

The canonical tasks approach simplifies manipulation tasks to their most essential properties. Because of this simplification, however, it may fail to capture some manipulation task aspects that are application-specific. Examples of such application-specific manipulation activities include positioning of a medical probe relative to virtual 3D models of internal organs in a VR medical training application, moving the control stick of the virtual airplane in a flight simulator, and exploring the intricacies of an object's surface such as a mountain range. Obviously, in these examples, generalization of the manipulation task does not make sense—it is the minute details of the manipulation that are important to capture and replicate.

Manipulation Techniques and Input Devices

There is a close relationship between the properties of input devices that are used to capture user input and the design of interaction techniques for a manipulation task: the choice of devices often restricts which manipulation techniques can be used. Here some of the important device properties are briefly reviewed that relate to manipulation techniques. Just like input devices, visual display devices and their characteristics (supported depth cues, refresh rate, resolution, etc.) can significantly affect the design of 3D manipulation techniques. Haptic displays could also have a pronounced effect on the user performance of manipulation tasks. The input devices are intimately linked to interaction techniques for manipulation.

Control Dimensions and Integrated Control in 3D Manipulation

Two characteristics of input devices that are key in manipulation tasks are, first, the number of control dimensions (how many DOF the device can control), and second, the integration of the control dimensions (how many DOF can be controlled simultaneously with a single movement). For example, a mouse allows for 2-DOF integrated control, and magnetic trackers allow simultaneous control of both 3D position and orientation (i.e., 6-DOF integrated control). Typical game controllers, on the other hand, provide at least 4-DOF, but the control is separated—2-DOF allocated to each of two joysticks, where each has to be controlled separately.

The devices that are usually best for 3D manipulation are multiple DOF devices with integrated control of all input dimensions. Integrated control allows the user to control the 3D interface using natural, well-coordinated movements, similar to real-world manipulation, which also results in better user performance. It was found that human performance was poor in multidimensional control. The recent studies suggest that this conclusion was due mostly to the limited input device technology that was available for multiple DOF input at the time when those experiments were conducted. Indeed, the input devices that were used did not allow users to control all degrees of freedom simultaneously. For example, in one experiment, subjects were required to manipulate two separate knobs to control the 2D position of a pointer.

Some 3D manipulation techniques also rely on more than one device with multiple integrated DOF. Such techniques usually employ two handheld devices and allow the user to complete a task by coordinating her hands in either a symmetric or an asymmetric fashion. These types of techniques are referred to as bimanual interactions. The reality of real-world 3D UI development, however, is that the device choice often depends on factors besides user performance, such as cost, device availability, ease of maintenance, and targeted user population. Therefore, even though 6-DOF devices are becoming less expensive and increasingly accessible, a majority of 3D UIs are still designed for input devices with only 2-DOF, such as a mouse, or those that separate degrees of freedom, such as game controllers.

Force versus Position Control

Another key property of input devices that significantly affects the design of interaction techniques is whether the device measures position or motion of the user's hand, as motion trackers and mice do (isomorphic control), or whether it measures the force applied by the user, as joysticks do (elastic or isometric control). In 6-DOF manipulation tasks, position control usually yields better performance than force control. Force control is usually preferable for controlling rates, such as the speed of navigation. Most 3D manipulation techniques assume that devices provide position control.

Device Placement and Form Factor in 3D Manipulation

The importance of device shape in manual control tasks has been known for a long time. Hand tools, for example, have been perfected over thousands of years, both to allow users to perform intended functions effectively and to minimize human wear and tear.

NAVIGATION

Navigation is a fundamental human task in the physical environment. The navigation tasks are faced mainly in synthetic environments: navigating the Web via a browser, navigating a complex document in a word processor, navigating through many layers of information in a spreadsheet, or navigating the virtual world of a computer game. Navigation in 3D UIs is discussed here.

Travel

Travel is the motor component of navigation—the task of moving from the current location to a new target location or moving in the desired direction. In the physical environment, travel is often a "no-brainer." Once we formulate the goal to walk across the room and through the door, our brains can instruct our muscles to perform the correct movements to achieve that goal. However, when our travel goal cannot be achieved effectively with simple body movements (we want to travel a great distance, or we want to travel very quickly, or we want to fly), then we use vehicles (bicycles, cars, planes, etc.). All vehicles contain some interface that maps various physical movements (turning a wheel, depressing a pedal, flipping a switch) to travel.

In 3D UIs, the situation is similar: there are some 3D interfaces where simple physical motions, such as walking, can be used for travel (e.g., when head and/or body trackers are used), but this is only effective within a limited space at a very limited speed. For most travel in 3D UIs, our actions must be mapped to travel in other ways, such as through a vehicle metaphor, for example. A major difference between real-world travel in vehicles and virtual travel, however, is that 3D UIs normally provide only visual motion cues, neglecting vestibular cues—this visual-vestibular mismatch can lead to cybersickness

Interaction techniques for the task of travel are especially important for two major reasons. First, travel is easily the most common and universal interaction task in 3D interfaces. Although there are some 3D applications in which the user's viewpoint is always stationary or where movement is automated, those are the exception rather than the rule. Second, travel (and navigation in general) often supports another task rather than being an end unto itself. Consider most 3D games: travel is used to reach locations where the user can pick up treasure, fight with enemies, or obtain critical information. Counter intuitively, the secondary nature of the travel task in these instances actually increases the need for usability of travel techniques. That is, if the user has to think about how to turn left or move forward, then he has been distracted from his primary task. Therefore, travel techniques must be intuitive—capable of becoming "second nature" to users.

Wayfinding

Wayfinding is the cognitive process of determining and following a route between an origin and a destination. It is the cognitive component of navigation—high-level thinking, planning, and decision-making related to user movement. It involves spatial understanding and planning tasks, such as determining the current location within the environment, determining a path from the current location to a goal location, and building a mental map of the environment. Real-world wayfinding has been researched extensively, with studies of aids like maps, directional signs, landmarks, and so on.

In virtual worlds, wayfinding can also be crucial. In a large, complex environment, an efficient travel technique is of no use if one has no idea where to go. Unlike travel techniques or manipulation techniques, where the computer ultimately performs the action, wayfinding techniques only support the performance of the task in the user's mind. Clearly, travel and wayfinding are both part of the same process (navigation) and contribute towards achieving the same goals. However, from the standpoint of 3D UI design, they are generally considered to be distinct. A travel technique is necessary to perform navigation tasks, and in some small or simple environments a good travel technique may be all that is necessary. In more complex environments, wayfinding aids may also be needed. In some cases, the designer can combine techniques for travel and wayfinding into a single integrated technique, reducing the cognitive load on the user and reinforcing the user's spatial knowledge each time the technique is used. Techniques that make use of miniature environments or maps fit this description, but these techniques are not suitable for all navigation tasks.

3D Travel Tasks

There are many different reasons why a user might need to perform a 3D travel task. Understanding the various types of travel tasks is important because the usability of a particular technique often depends on the task for which it is used. Experiments based on travel "testbeds" have attempted to empirically relate task type to technique usability. Travel tasks are classified as - exploration, search and maneuvering.

Exploration

In an exploration or browsing task, the user has no explicit goal for her movement. Rather, she is browsing the environment, obtaining information about the objects and locations within the world and building up knowledge of the space. For example, the client of an architecture firm may explore the latest building design in a 3D environment. Exploration is typically used at the beginning of an interaction with an environment, serving to orient the user to the world and its features, but it may also be important in later stages. Because a user's path during exploration may be based on serendipity (seeing something in the world may cause the user to deviate from the current path), techniques to support exploration should allow continuous and direct control of viewpoint movement or at least the ability to interrupt a movement that has begun. Forcing the user to continue along the chosen path until its completion would detract from the discovery process. Of course, this must be balanced, in some applications, with the need to provide an enjoyable experience in a short amount of time. Techniques should also impose little cognitive load on users so that they can focus cognitive resources on spatial knowledge acquisition, information gathering, or other primary tasks.

To what extent should 3D UIs support exploration tasks? The answer depends on the goals of the application. In some cases, exploration is an integral component of the interaction. For example, in a 3D visualization of network traffic data, the structure and content of the environment is not known in advance, making it difficult to provide detailed wayfinding aids. The benefits of the visualization depend on how well the interface supports exploration of the data. Also, in many 3D gaming environments, exploration of unknown spaces is an important part of the entertainment value of the game. On the other hand, in a 3D interface where the focus is on performing tasks within a well-known 3D environment, the interface designer should provide more support for search tasks via goal-directed travel techniques.

Search

Search tasks involve travel to a specific goal or target location within the environment. In other words, the user in a search task knows the final location to which he wants to navigate. However, it is not necessarily the case that the user has knowledge of where that location is or how to get there from the current location. For example, a gamer may have collected all the treasure on a level, so he needs to travel to the exit. The exit may be in a part of the environment that hasn't yet been explored, or the user may have seen it previously. This leads to the distinction between a naïve search task, where the user does not know the position of the target or a path to it in advance, and a primed search task, where the user has visited the target before or has some other knowledge of its position.

Naïve search has similarities with exploration, but clues or wayfinding aids may direct the search so that it is much more limited and focused than exploration. Primed search tasks also exist on a continuum, depending on the amount of knowledge the user has of the target and the surrounding environment. A user may have visited a location before but still might have to explore the environment around his starting location before he understands how to begin traveling toward the goal. On the other hand, a user with complete survey knowledge of the environment can start at any location and immediately begin navigating directly to the target. Although the lines between these tasks are often blurry, it is still useful to make the distinction.

Many 3D UIs involve search via travel. For example, the user in an architectural walkthrough application may wish to travel to the front door to check sight lines. Techniques for this task may be more goal oriented than techniques for exploration. For example, the user may specify the final location directly on a map rather than through incremental movements. Such techniques do not apply to all situations, however. A map-based technique was quite inefficient, even for primed search tasks, when the goal locations were not explicitly represented on the map. It may be useful to combine a target-based technique with a more general technique to allow for the continuum of tasks discussed above.

Maneuvering

Maneuvering is an often-overlooked category of 3D travel. Maneuvering tasks take place in a local area and involve small, precise movements. The most common use of maneuvering is to position the viewpoint more precisely within a limited local area to perform a specific task. For example, the user needs to read some written information in the 3D environment but must position herself directly in front of the information in order to make it legible. In another scenario, the user wishes to check the positioning of an object she has been manipulating in a 3D modeling system and needs to examine it from many different angles. This task may seem trivial compared to large-scale movements through the environment, but it is precisely these small-scale movements that can cost the user precious time and cause frustration if not supported by the interface.

A designer might consider maneuvering tasks to be search tasks, because the destination is known, and therefore use the same type of travel techniques for maneuvering as for search, but this would ignore the unique requirements of maneuvering tasks. In fact, some applications may require special travel techniques solely for maneuvering. In general, travel techniques for this task should allow great precision of motion but not at the expense of speed. The best solution for maneuvering tasks may be physical motion of the user's head and body because this is efficient, precise, and natural, but not all applications include head and body tracking, and even those that do often have limited range and precision. Therefore, if close and precise work is important in application, other techniques for maneuvering, such as the object-focused travel techniques must be considered.

Additional Travel Task Characteristics

In the classification of the tasks above, they are distinguished by the user's goal for the travel task. Remember that many other characteristics of the task should be considered when choosing or designing travel techniques:

Distance to be traveled:

In a 3D UI using head or body tracking, it may be possible to accomplish short-range travel tasks using natural physical motion only. Medium-range travel requires a virtual travel technique but may not require velocity control. Long-range travel tasks should use techniques with velocity control or the ability to jump quickly between widely scattered locations.

Amount of curvature or number of turns in the path:

Travel techniques should take into account the amount of turning required in the travel task. For example, steering based on torso direction may be appropriate when turning is infrequent, but a less strenuous method, such as hand-directed steering (most users will use hand-directed steering from the hip by locking their elbows in contrast to holding up their hands), would be more comfortable when the path involves many turns.

Visibility of the target from the starting location:

Many target-based techniques depend on the availability of a target for selection. Gaze-directed steering works well when the target is visible but not when the user needs to search for the target visually while traveling.

Number of DOF required for the movement:

If the travel task requires motion only in a horizontal plane, the travel technique should not force the user to also control vertical motion. In general, terrain-following is a useful constraint in many applications.

Required accuracy of the movement:

Some travel tasks require strict adherence to a path or accurate arrival at a target location. In such cases, it’s important to choose a travel technique that allows for fine control and adjustment of direction, speed, or target location. For example, map-based target selection is usually inaccurate because of the scale of the map, imprecision of hand tracking, or other factors. Travel techniques should also allow for easy error recovery (e.g., backing up if the target was overshot) if accuracy is important.

Other primary tasks that take place during travel:

Often, travel is a secondary task performed during another more important task. For example, a user may be traveling through a building model in order to count the number of windows in each room. It is especially important in such situations that the travel technique be unobtrusive, intuitive, and easily controlled.

VII. 3DUI evaluation

One of the central truths of human–computer interaction (HCI) is that even the most careful and well-informed designs can still go wrong in any number of ways. Thus, evaluation of UIs becomes critical. In fact, the reason we can provide answers to questions such as those above is that researchers have performed evaluations addressing those issues. Some of the evaluation methods that can be used for 3D UIs, metrics that help to indicate the usability of 3D UIs, distinctive characteristics of 3D UI evaluation, and guidelines for choosing evaluation methods are discussed. The evaluation should not only be performed when a design is complete, but that it should also be used as an integral part of the design process.

Evaluation has often been the missing component of research in 3D interaction. For many years, the fields of VEs and 3D UIs were so novel and the possibilities so limitless that many researchers simply focused on developing new devices, interaction techniques, and UI metaphors—exploring the design space—without taking the time to assess how good the new designs were. We must critically analyze, assess, and compare devices, interaction techniques, UIs, and applications if 3D UIs are to be used in the real world.

Purposes of Evaluation

Simply stated, evaluation is the analysis, assessment, and testing of an artifact. In UI evaluation, the artifact is the entire UI or part of it, such as a particular input device or interaction technique. The main purpose of UI evaluation is the identification of usability problems or issues, leading to changes in the UI design. In other words, design and evaluation should be performed in an iterative fashion, such that design is followed by evaluation, leading to a redesign, which can then be evaluated, and so on. The iteration ends when the UI is "good enough," based on the metrics that have been set (or, more frequently in real-world situations, when the budget runs out or the deadline arrives!). Although problem identification and redesign are the main goals of evaluation, it may also have secondary purposes. One of these is a more general understanding of the usability of a particular technique, device, or metaphor. This general understanding can lead to design guidelines, so that each new design can start from an informed position rather than from scratch. For example, we can be reasonably sure that users will not have usability problems with the selection of items from a pull-down menu in a desktop application, because the design of those menus has already gone through many evaluations and iterations. Another, more ambitious, goal of UI evaluation is the development of performance models. These models aim to predict the performance of a user on a particular task within an interface. For example, Fitts's law predicts how quickly a user will be able to position a pointer over a target area based on the distance to the target, the size of the target, and the muscle groups used in moving the pointer. Such performance models must be based on a large number of experimental trials on a wide range of generic tasks, and they are always subject to criticism (e.g., the model doesn't take an important factor into account, or the model doesn't apply to a particular type of task). Nevertheless, if a useful model can be developed, it can provide important guidance for designers.

Terminology

Some important terms must be designed for understanding 3D UI evaluation. The most important term is usability. Usability encompasses everything about an artifact and a person that affects the person's use of the artifact. Evaluation, then, measures some aspects of the usability of an interface. Usability measures (or metrics) fall into several categories, such as system performance, user task performance, and user preference. There are at least two roles that people play in a usability evaluation. A person who designs, implements, administers, or analyzes an evaluation is called an evaluator. A person who takes part in an evaluation by using the interface, performing tasks, or answering questions is called a user. In formal experimentation, a user is sometimes called a subject. Finally, evaluation methods and evaluation approaches are distinguished. Evaluation methods (or techniques) are particular steps that can be used in an evaluation. An evaluation approach, on the other hand, is a combination of methods, used in a particular sequence, to form a complete usability evaluation.

Evaluation Metrics for 3D Interfaces

Three types of metrics for 3D UIs are - system performance metrics, task performance metrics and user preference metrics.

1. System Performance Metrics

System performance refers to typical computer or graphics system performance, using metrics such as average frame rate, average latency, network delay, and optical distortion. From the interface point of view, system performance metrics are really not important in and of themselves. Rather, they are important only insofar as they affect the user's experience or tasks. For example, the frame rate probably needs to be at real-time levels before a user will feel present. Also, in a collaborative setting, task performance will likely be negatively affected if there is too much network delay.

2. Task Performance Metrics

User task performance refers to the quality of performance of specific tasks in the 3D application, such as the time to navigate to a specific location, the accuracy of object placement, or the number of errors a user makes in selecting an object from a set. Task performance metrics may also be domain-specific. For example, evaluators may want to measure student learning in an educational application or spatial awareness in a military training VE. Typically, speed (efficiency) and accuracy are the most important task performance metrics. The problem with measuring both speed and accuracy is that there is an implicit relationship between them: I can go faster but be less accurate, or I can increase my accuracy by decreasing my speed. It is assumed that for every task, there is some curve representing this speed/accuracy trade-off, and users must decide where on the curve they want to be (even if they don't do this consciously). In an evaluation, therefore, if you simply tell your subjects to do a task as quickly and precisely as possible, they will probably end up all over the curve, giving you data with a high level of variability. Therefore, it is very important that you instruct users in a very specific way if you want them to be at one end of the curve or the other. Another way to manage the trade-off is to tell users to do the task as quickly as possible one time, as accurately as possible the second time, and to balance speed and accuracy the third time. This gives you information about the trade-off curve for the particular task you're looking at.

3. User Preference Metrics

User preference refers to the subjective perception of the interface by the user (perceived ease of use, ease of learning, satisfaction, etc.). These preferences are often measured via questionnaires or interviews and may be either qualitative or quantitative. The user preference metrics generally contribute significantly to overall usability. A usable application is one whose interface does not pose any significant barriers to task completion. Often, HCI experts speak of a transparent interface—a UI that simply disappears until it feels to the user as if he is working directly on the problem rather than indirectly through an interface. UIs should be intuitive, provide good affordances (indications of their use and how they are to be used), provide good feedback, not be obtrusive, and so on. An application cannot be effective unless users are willing to use it (and this is precisely the problem with some more advanced VE applications—they provide functionality for the user to do a task, but a lack of attention to user preference keeps them from being used).

For 3D UIs in particular, presence and user comfort can be important metrics that are not usually considered in traditional UI evaluation. Presence is a crucial, but not very well understood metric for VE systems. It is the "feeling of being there"—existing in the virtual world rather than in the physical world. How can we measure presence? One method simply asks users to rate their feeling of being there on a 1 to 100 scale. Questionnaires can also be used and can contain a wide variety of questions, all designed to get at different aspects of presence. Psychophysical measures are used in controlled experiments where stimuli are manipulated and then correlated to users' ratings of presence (for example, how does the rating change when the environment is presented in mono versus stereo modes?). There are also some more objective measures. Some are physiological (how the body responds to the VE). Others might look at users' reactions to events in the VE. Tests of memory for the environment and the objects within it might give an indirect measurement of the level of presence.

Finally, if a task is known for which presence is required, we can measure users' performance on that task and infer the level of presence. There is still a great deal of debate about the definition of presence, the best ways to measure presence, and the importance of presence as a metric. The other novel user preference metric for 3D systems is user comfort. This includes several different things. The most notable and well-studied is so-called simulator sickness (because it was first noted in flight simulators). This is symptomatically similar to motion sickness and may result from mismatches in sensory information (e.g., your eyes tell your brain that you are moving, but your vestibular system tells your brain that you are not moving). There is also work on the physical aftereffects of being exposed to 3D systems. For example, if a VE mis-registers the virtual hand and the real hand (they're not at the same physical location), the user may have trouble doing precise manipulation in the real world after exposure to the virtual world. More seriously, activities like driving or walking may be impaired after extremely long exposures (1 hour or more). Finally, there are simple strains on arms/hands/eyes from the use of 3D devices. User comfort is also usually measured subjectively, using rating scales or questionnaires.

Two well-developed VE evaluation approaches

1. Testbed Evaluation Approach

This approach empirically evaluate interaction techniques outside the context of applications (i.e., within a generic context rather than within a specific application) and add the support of a framework for design and evaluation, which is summarized here. Principled, systematic design and evaluation frameworks give formalism and structure to research on interaction; they do not rely solely on experience and intuition. Formal frameworks provide us not only with a greater understanding of the advantages and disadvantages of current techniques, but also with better opportunities to create robust and well performing new techniques based on knowledge gained through evaluation. Therefore, this approach follows several important evaluation concepts, elucidated in the following sections. Figure 5.2 presents an overview of this approach.

Initial Evaluation

The first step toward formalizing the design, evaluation, and application of interaction techniques is to gain an intuitive understanding of the generic interaction tasks in which one is interested and current techniques available for the tasks. This is accomplished through experience using interaction techniques and through observation and evaluation of groups of users. These initial evaluation experiences are heavily drawn upon for the processes of building taxonomy, listing outside influences on performance, and listing performance measures. It is helpful, therefore, to gain as much experience of this type as possible so that good decisions can be made in the next phases of formalization.

Taxonomy

The next step is to establish taxonomy of interaction techniques for the interaction task being evaluated. These are technique decomposition taxonomies. For example, the task of changing an object's color might be made up of three subtasks: selecting an object, choosing a color, and applying the color. The subtask for choosing a color might have two possible technique components: changing the values of R, G, and B sliders or touching a point within a 3D color space. The subtasks and their related technique components make up taxonomy for the object coloring task.

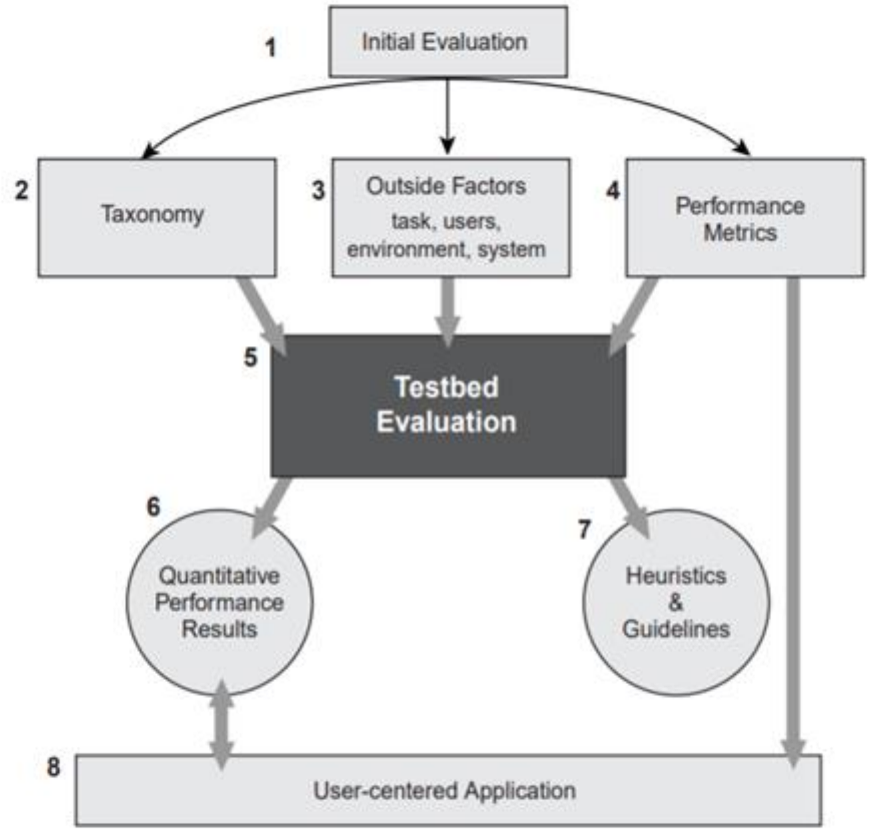


Fig. 5.2 Testbed evaluation approach

Ideally, the taxonomies established by this approach need to be correct, complete, and general. Any interaction technique that can be conceived for the task should fit within the taxonomy. Thus, subtasks will necessarily be abstract. The taxonomy will also list several possible technique components for each of the subtasks, but they do not list every conceivable component. Building taxonomies is a good way to understand the low-level makeup of interaction techniques and to formalize differences between them, but once they are in place, they can also be used in the design process. One can think of taxonomy not only as a characterization, but also as a design space. Since taxonomy breaks the task down into separable subtasks, a wide range of designs can be considered quickly, simply by trying different combinations of technique components for each of the subtasks. There is no guarantee that a given combination will make sense as a complete interaction technique, but the systematic nature of the taxonomy makes it easy to generate designs and to reject inappropriate combinations.

Outside Factors

Interaction techniques cannot be evaluated in a vacuum. A user's performance on an interaction task may depend on a variety of factors, of which the interaction technique is but one. In order for the evaluation framework to be complete, such factors must be included explicitly and used as secondary independent variables in evaluations. Bowman and Hodges identified four categories of outside factors. First, task characteristics are those attributes of the task that may affect user performance, including distance to be travelled or size of the object being manipulated. Second, the approach considers environment characteristics, such as the number of obstacles and the level of activity or motion in the VE. User characteristics, including cognitive measures such as spatial ability and physical attributes such as arm length, may also contribute to user performance. Finally, system characteristics, such as the lighting model used or the mean frame rate, may be significant.

Performance Metrics

This approach is designed to obtain information about human performance in common VE interaction tasks—but what is performance? Speed and accuracy are easy to measure, are quantitative, and are clearly important in the evaluation of interaction techniques, but there are also many other performance metrics to be considered. Thus, this approach also considers more subjective performance values, such as perceived ease of use, ease of learning, and user comfort. The choice of interaction technique could conceivably affect all of these, and they should not be discounted. Also, more than any other current computing paradigm, VEs involve the user's senses and body in the task. Thus, a focus on user-centric performance measures is essential. If an interaction technique does not make good use of human skills, or if it causes fatigue or discomfort, it will not provide overall usability despite its performance in other areas.

2. Sequential Evaluation Approach

The sequential evaluation approach is a usability engineering approach and addresses both design and evaluation of VE UIs. While some of the components are well suited for evaluation of generic interaction techniques, the complete sequential evaluation approach employs application-specific guidelines, domain-specific representative users, and application-specific user tasks to produce a usable and useful interface for a particular application. In many cases, results or lessons learned may be applied to other, similar applications (for example, VE applications with similar display or input devices, or with similar types of tasks), and in other cases (albeit less often), it is possible to abstract the results to generic cases. Sequential evaluation evolved from iteratively adapting and enhancing existing 2D and GUI usability evaluation methods. In particular, it modifies and extends specific methods to account for complex interaction techniques, nonstandard and dynamic UI components, and multimodal tasks inherent in VEs. Moreover, the adapted/extended methods both streamline the usability engineering process and provide sufficient coverage of the usability space. While the name implies that the various methods are applied in sequence, there is considerable opportunity to iterate both within a particular method as well as among methods. It is important to note that all the pieces of this approach have been used for years in GUI usability evaluations. Figure 5.3 presents the sequential evaluation approach. It allows developers to improve a VE's UI by a combination of expert-based and user-based techniques. This approach is based on sequentially performing user task analysis, heuristic (or guidelines based expert) evaluation, formative evaluation and summative evaluation, with iteration as appropriate within and among each type of evaluation. This approach leverages the results of each individual method by systematically defining and refining the VE UI in a cost-effective progression. Depending upon the nature of the application, this sequential evaluation approach may be applied in a strictly serial approach or iteratively applied many times. For example, when used to evaluate a complex command-and-control battlefield visualization application, user task analysis was followed by significant iterative use of heuristic and formative evaluation and lastly followed by a single, broad summative evaluation.

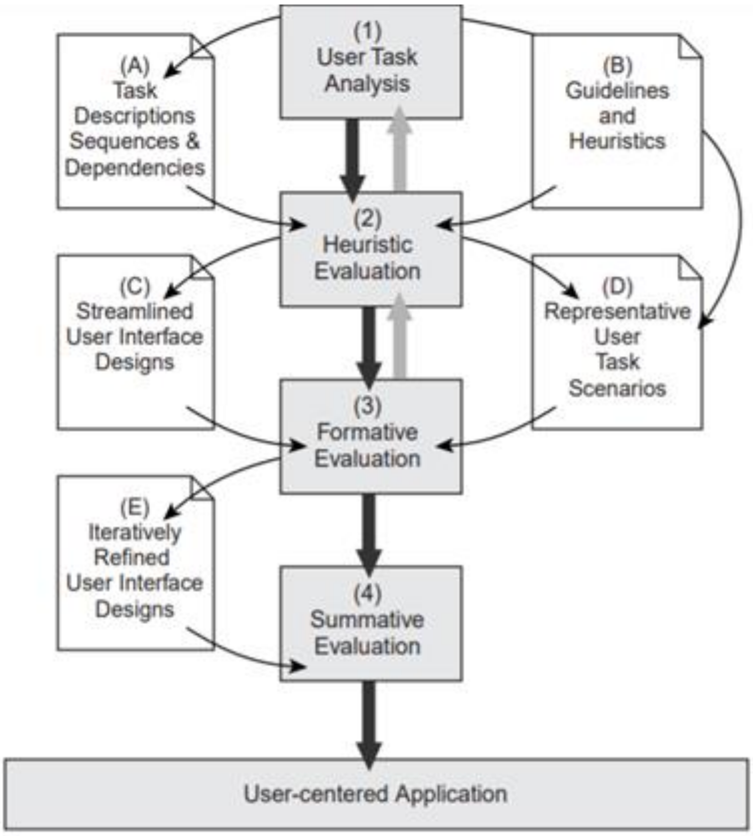


Fig. 5.3 Sequential Evaluation Approach

From experience, this sequential evaluation approach provides cost effective assessment and refinement of usability for a specific VE application. Obviously, the exact cost and benefit of a particular evaluation effort depends largely on the application's complexity and maturity. In some cases, cost can be managed by performing quick and lightweight formative evaluations (which involve users and thus are typically the most time consuming to plan and perform). Moreover, by using a "hallway methodology," user-based methods can be performed quickly and cost effectively by simply finding volunteers from within one's own organization. This approach should be used only as a last resort or in cases where the representative user class includes just about anyone. When used, care should be taken to ensure that "hallway" users provide a close representative match to the application's ultimate end users.

Following are some of the guidelines for those wishing to perform usability evaluations of 3D UIs. The first subsection presents general guidelines, and the second subsection focuses specifically on formal experimentation.

1. General Guidelines

Begin with informal evaluation.

Informal evaluation is very important, both in the process of developing an application and in doing basic interaction research. In the context of an application, informal evaluation can quickly narrow the design space and point out major flaws in the design. In basic research, informal evaluation helps you understand the task and the techniques on an intuitive level before moving on to more formal classifications and experiments.

Acknowledge and plan for the differences between traditional UI and 3D UI evaluation.

These differences must be considered when designing a study. For example, you should plan to have multiple evaluators, incorporate rest breaks into your procedure, and assess whether breaks in presence could affect your results.

Choose an evaluation approach that meets your requirements.

With respect to interaction techniques, there is no optimal usability evaluation method or approach. A range of methods should be considered, and important questions should be asked. For example, if you have designed a new interaction technique and want to refine the usability of the design before any implementation, a heuristic evaluation or cognitive walkthrough fits the bill. On the other hand, if you must choose between two input devices for a task in which a small difference in efficiency may be significant, a formal experiment may be required.

Use a wide range of metrics.

Remember that speed and accuracy alone do not equal usability. Also remember to look at learning, comfort, presence, and other metrics in order to get a complete picture of the usability of the interface

2. Guidelines for Formal Experimentation

Design experiments with general applicability.

If you're going to do formal experiments, you will be investing a large amount of time and effort, so you want the results to be as general as possible. Thus, you have to think hard about how to design tasks that are generic, performance measures to which real applications can relate, and a method for applications to easily reuse the results.

Use pilot studies to determine which variables should be tested in the main experiment.

In doing formal experiments, especially testbed evaluations, you often have too many variables to actually test without an infinite supply of time and subjects. Small pilot studies can show trends that may allow you to remove certain variables because they do not appear to affect the task you're doing.

Look for interactions between variables—rarely will a single technique be the best in all situations.

In most formal experiments on the usability of 3D UIs, the most interesting results have been interactions. That is, it's rarely the case that technique A is always better than technique B. Rather, technique A works well when the environment has characteristic X, and technique B works well when the environment has characteristic Y. Statistical analysis should reveal these interactions between variables.

Questions

Part A

1. Point out the applications of Virtual Reality in Digital Entertainment.
2. How is VR affecting the film industry? Interpret.
3. Summarize the problems encountered in VR post-production in film & television.
4. Compare and Contrast the advantages and disadvantages of using VR for fitness.
5. Do video games really count as exercise? Relate with an example.
6. Define 3d User Interface.
7. List the devices used for virtual reality and 3D interaction.
8. Discuss the tasks involved in Selection and Manipulation techniques for 3D environments.
9. State the purpose of evaluation of 3D user interface.
10. How has society benefitted from VR? Infer.

Part B

1. Identify the role played by VR technology in film and TV production. Explain in detail.
2. "VR in sports" – Analyze and Illustrate with real-time scenario.
3. Discuss the 3D user interaction techniques with virtual environment.
4. How will you assess the evaluation metrics for 3D interfaces?
5. Explain the applications of VR in Digital Environment.

REFERENCE BOOKS

1. Sherman, William R. and Alan B. Craig. Understanding Virtual Reality – Interface, Application, and Design, Morgan Kaufmann, 2002.
2. Fei GAO. Design and Development of Virtual Reality Application System, Tsinghua Press, March 2012.
3. Guangran LIU. Virtual Reality Technology, Tsinghua Press, Jan. 2011.
4. Burdea, G. C. and P. Coffet. Virtual Reality Technology, 2nd Edition. Wiley-IEEE Press, 2003/2006.
5. Doug Bowman, Ernst Kruijff, Joseph LaViola Jr., Ivan Poupyrev, "3D User Interfaces: Theory and Practice".