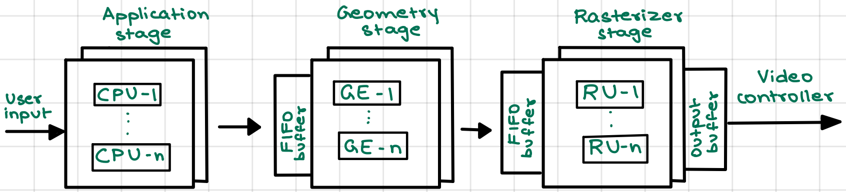
UNIT-3

GRAPHICS RENDERING PIPELINE



→ The first stage is the application stage, which is done entirely in software by the CPU.

- It reads the world geometry database as well as the user's i/p mediated by devices such as mice, trackballs, trackers or sensing gloves.
- In reponse to the user's i/p the appⁿ stage may change the view to the simulation or change the orientation of virtual objects.

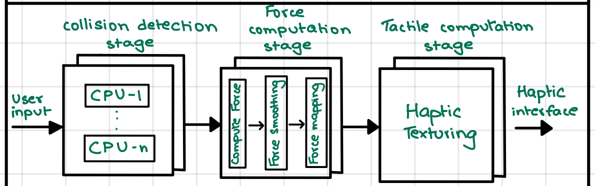
→ The appⁿ stage results are fed to the geometry stage, which can be implemented in s/ware or h/ware.

- It consists of model transformations (translation, rotation, scaling), lighting computations, clipping, scene projection & mapping.
- The lighting substage calculates the surface color based on the type & no. of simulated light sources in the scene, the surface material properties, atmospheric effects (like fog), etc.
- The lighting computations make the scene more realistic.

→ Rasterizing stage:

- Done in h/ware to gain speed.
- This stage converts the vertex info o/p by the geometry stage (such as color & texture) into pixel info needed by the video display.
- An imp function of this stage is to perform antialiasing in order to smooth out the jagged appearence of polygon edges.
- Antialiasing subdivides pixels into subpixel regions w/ an assigned color.
- The more subpixels used, better the image quality.

THE HAPTICS RENDERING PIPELINE



First stage:

- The physical characteristics (weight, smoothness, surface temp) of the 3D objects are loaded from the database.
- Collision detection is performed to determine which virtual objects will collide, if any.
- Unlike the graphics pipeline, here only the colliding structures in the scene are passed down to subsequent pipeline stages.

Second stage:

- Computes the collision forces based on various physical models. The simplest model is based on Hooke's Law.
- The more object contacts & the more complex the force shading, the higher is the chance of the pipeline becoming force limited.
- Force smoothing adjusts the direction of the force vector to avoid sharp transitions.
- Force mapping projects the calculated force to the characteristics of the particular haptic display system.

Third stage: Haptic Texturing

- Renders the touch feedback component of the simulation.
- Its computed effects, such as vibrations or surface temp, are added to the force vector & sent to the haptics output display.

GRAPHICS ACCELERATORS

→ Allow the real-time rendering of VR scenes on a PC.

→ Here, we're interested in cards that adhere to the openGL graphics std, perform stereo rendering, & provide the synchronization signal necessary for use of active glasses. These are typically cards produced by 3rd-party vendors, designed to retrofit mid-to-high-end PCs.

eg → ATI Fire GL2

- It is an eg. of a traditional dual-processor architecture.
- uses a geometry chip to perform the geometry-stage pipeline computations & a rasterizer chip for pixel-related computations & display.
- The chips are masked by a very large coolers.

GRAPHICS BENCHMARKS

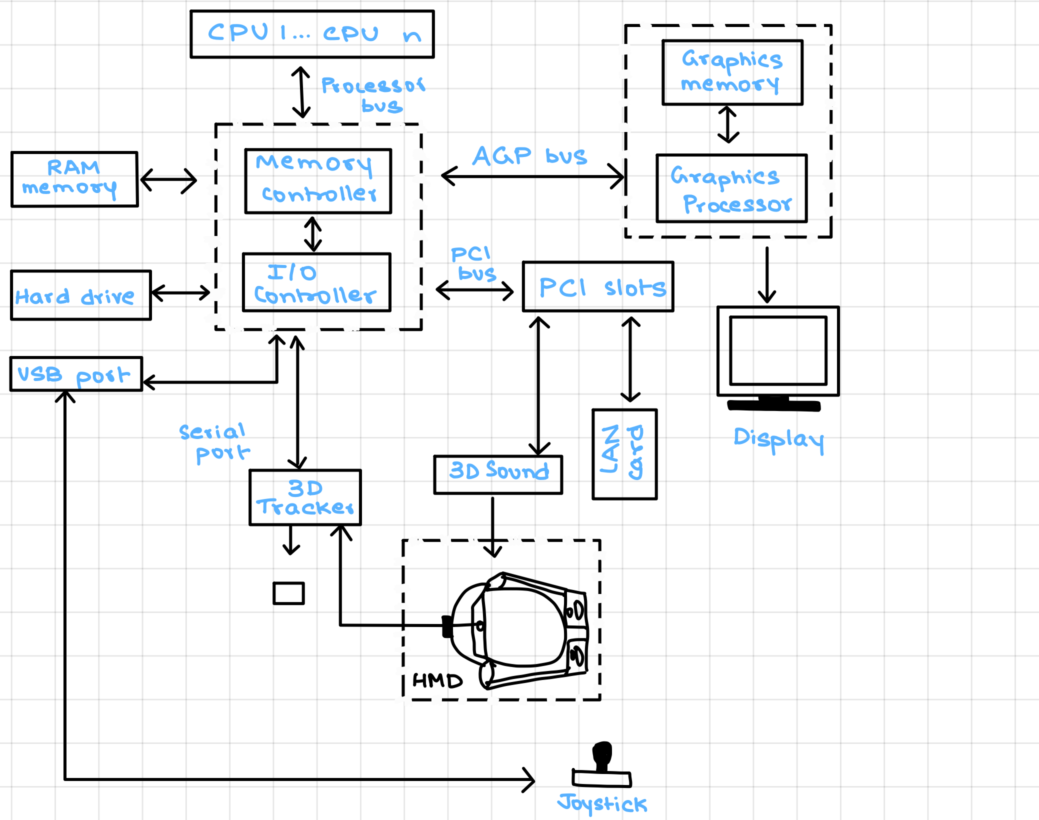
→ A productive way to rank graphics cards is to see how they perform in terms of frames/sec (fps) when running a given appⁿ.

→ Each horizontal bar corresponds to progressively faster CPUs feeding the graphics pipeline.

→ The size & resolution of the o/p display are maintained constant throughout the tests.

→ This shows that there is a need to have a uniform way to judge graphics hardware & not rely exclusively on manufacturer specification sheets.

PC VR - PC GRAPHICS ARCHITECTURE



WORKSTATION-BASED ARCHITECTURES

→ The largest computing base after the PC in deployed units is the workstation.

→ Its advantage over the PC is greater computing power due to the ability to use superscalar architectures, larger disk space, & faster communication.

→ The primary market for workstations is still general computing rather than VR.

eg → The Sun Blade 1000 Architecture

- Has Expert3D graphics rendering 6 million triangles/sec.
- Has 2 900-MHz UltraSpark III 64-bit processors, each of which is twice as fast as the previous -gen UltraSpark II CPUs.
- Their performance & low latency are helped by the presence of primary & 2dary caches.
- The data bus is wide & fully independent of the address bus.

DISTRIBUTED VR ARCHITECTURES

-> 1 adv workstations had over PCs until recently was the Unix OS support for multiple displays.

-> Many displays require multiple views to be rendered by the VR engine. In order to do so, the VR engine needs to have several graphics pipelines working cooperatively to provide the visual feedback needed by the simulation.

A distributed VR engine is one that uses 2 or more rendering pipelines. Such pipelines can perform graphics or haptics computations, & be located on a single computer, on several comps, or on multiple remote comps integrated in a single simulation.

Multipipeline Synchronization

- Whether multiple graphics pipelines are part of a single computer or of multiple cooperating ones, the o/p images need to be synchronized.
- This is especially true for tiled displays, where lack of synchronization can lead to disturbing visual artifacts.

Colocated Rendering Pipelines

- A system w/ " " " may consist of a single comp w/ a multipipeline graphics accelerator, side-by-side comps, or a combi-nation of these.
- Such systems aim to balance cost & performance.

Distributed Virtual Environments

- A virtual environment is said to be distributed if it resides on 2 (or more) networked comps, which share the total simulation computational load.
- The networked comps, called nodes, allow multiple users to interact w/ each other & w/ the objects in the shared virtual world.

UNIT-4

SCENE GRAPH

→ It is a hierarchical organization of objects in the virtual world, together w/ the view to that world.

→ The scene graph represents a tree structure, w/ nodes connected by branches.

→ The topmost node in the hierarchy is the root, which is the parent node to the whole graph.

→ The external nodes (leaves) have no children. They represent visible objects w/ their properties (geometry, appearence, behavior).

→ Internal nodes typically represent transformations, which position their children objects in space in relation to the root node or other objects. Any transformation applied to a given internal node will affect all its children.

→ Scene graphs are not static, but change to reflect the current state of the virtual world. Such changes may be due to the user's i/p or to object intelligent behavior.

→ The scene graph updates happen once every frame & affect the node attributes. This process is done recursively, from the root node down to the leaves & back to the root node.

→ Once the scene graph is updated, it is ready for processing by the graphics pipeline.

SCENE ILLUMINATION

→ Determines the light intensities on the object surface.

Can be classified into:

① Local illumination:

- Treats the interactions b/w objects & light sources in isolation, neglecting the interdependences b/w objects.
- eg → Gouraud shading, which shades objects based on light intensity interpolation.

② Global illumination:

- Models the interreflections b/w objects & shadows, resulting in a more realistic-looking scene.
- 1 approach is to model object radiosity. This method considers that each object in the scene (including virtual light sources) emits an energy flow in the form of light. The amt of energy radiated by the objects per unit surface & unit time determines its radiosity.

TEXTURE MAPPING

→ Method of improving the image realism w/out the need for additional surface polygons.

→ The location of an object vertex is stored at modeling time in object coordinate space (x, y, z).

→ The modeler also stores the location in parametric space.

→ Then, during the geometry stage of the graphics pipeline the object is shaded, such that each pixel is given a color based on vertex color & interpolated light intensities (Gouraud shading) or interpolated normals.

→ Texturing is a technique performed in the rasterizing stage of the graphics pipeline in order to modify the object model's surface properties such as color, specular reflection, or pixel normals.

METHODS OF CONTROLLING ANIMATION

① **Full explicit** / explicitly-declared control
- Simplest type of control in which the animator describes all the events which could occur in an animation (Object level & Frame level).

② **Procedural Control** - A set of rules are used to animate the objects. Animator specifies the initial rules, procedure to process & later runs simulations. Rules are often based on physical rules of the real world expressed by mathematical equations.

③ **Constraint-based C** - This method specifies an animation sequence which needs to be followed.

④ **Control by tracking live action** - Control is achieved by examining the motions of objects in the real world wherein live action can be tracked to generate trajectories of objects in the course of an animation.

⑤ **Actors** - using actors is a high-level form of procedural control. An actor is a small program invoked once per frame to determine the characteris-tics of some object in the anim-ation.

⑥ **Kinematics** & **Dynamics** control-

Kinematics ↑ Refer to the position & velocity of points

Dynamics ↑ Takes into account the physical laws that govern kinematics.

This method uses the laws of physics to generate motion of pictures & objects. Dynamic constraints are used to simulate the dynamic behavior of such a system.

⑦ **Physically based Animation** -
Animation of motion of physically based models of cloth, plasticity, & rigid-body are described using simulations.
This sort of animation ties together the work on constraints, dynamics, procedural control & actors.

BASIC RULES OF ANIMATION

① SQUASH & STRETCH

It gives the sense of weight & volume to draw an object. Squashing & stretching are the 2 most basic animated reactions a drawn object can exhibit.

② ANTICIPATION

It is the physical preperation for an action. The moment of anticipation informs the audience that an action is about to take place.

③ STAGING

It is done to direct the audience's attention to the intended objects in a scene

④ STRAIGHT AHEAD & POSE-TO-POSE

"Straight-ahead" animation is a continuous process in which the animator draws each scene a single frame at a time. In pose-to-pose animation the animator plans out the key poses in the scene.

⑤ FLOW THROUGH & OVERLAPP-ING ACTION

A part of the object will keep moving for a while even if the base or centre of mass has stopped.

The main mass of an object in motion changes while the extraneous parts take time to adjust.

⑥ SLOW-OUT & SLOW-IN

The smoothness of an animation is governed mostly by how many frames the animation contains. More frames give a slower & smoother animation.

⑦ ARCS

Natural movements tend to follow the trajectory of an arc, as such, it is important for animators to consider implied "arcs" for each movement.

⑧ SECONDARY ACTION

Makes a scene more interes-ting to the viewer & further help enforce the illusion of reality.

⑨ TIMING

(No. of frames in a given scene) dictates the overall speed of the animation. The timing affects how real the movements seem.

⑩ EXAGGERATION

Remains true to reality, but presents it in a more extreme form.

⑪ SOLID DRAWING

Applies to objects drawn in a 3D space. Solid drawing techniques- weight, balance, anatomy, lighting & more.

⑫ APPEAL

Appeal in a cartoon character corresponds to what would be called charm in an actor.

UNIT-2

MECHANICAL TRACKER

→ It consists of a serial or parallel kinematic structure composed of links interconnected using sensorized joints.

→ The dimensions of each link segment are known a priori & used by the direct kinematics computational model stored in the comp.

→ This model allows the determination of the position & orientation of 1 end of the mechanical tracker relative to the other, based on the real-time reading of the tracker joint sensors.

Advantages:

- Simple & easy to use
- Accuracy is fairly constant over the tracker work envelope, & depends essentially on the resolution of the joint sensors used.
- Unlike electromagnetic trackers, they are immune to interference from magnetic fields.
- They have very low jitter & the lowest latency of all tracking types.

A hybrid tracker is a system that utilizes 2 or more position measurement technologies to track objects better than any single technology would allow.

contd.

MAGNETIC TRACKER

→ It is a noncontact position tracker device that uses a magnetic field produced by a stationary transmitter to determine the realtime position of a moving receiver element.

→ whatever 3D measurement technique is used, it shouldn't be intrusive & shouldn't hinder the user's freedom of motion in the process of tracking him/her

→ In view of this requirement, noncontact 3D measurement techniques have replaced mechanical ones.

→ They use either magnetic fields, ultrasound, infrared cameras & LEDs

→ The transmitter consists of 3 antennas formed of 3 mutually orthogonal coils wound on a ferromagnetic cube.

OPTICAL TRACKERS

→ It is a non-contact position measurem-ent device that uses optical sensing to determine the real-time position/orientation of an object.

→ They require direct line of sight & are immune to metal interference.

→ Their update rates are much higher & their latency smaller than those of ultrasonic trackers as light travels much faster than sound. They are also capable of larger work envelopes.

ULTRASONIC TRACKERS

→ It is a noncontact position measurement device that uses an ultrasonic signal produced by a stationary transmitter to determine the real-time position of a moving receiver element.

→ They have 3 components — a transmitter, a receiver, & an electronic unit, similar to their magnetic counterparts. The difference is that the transmitter is a set of 3 ultrasonic speakers mounted about 30 cm from each other on a fixed frame.

→ Similarly the receiver is a set of 3 microphones mounted on a smaller rigid frame.

→ These trackers represent a cheaper alternative to the magnetic ones.

HYBRID INERTIAL TRACKERS

→ They are self-contained sensors that measure the rate of change of an object translation velocity.

→ Modern inertial trackers are solid-state structures that use microelectromechanical systems (MEMS) technology.

→ The rate of change in object orientation is measured by Coriolis-type gyroscopes.

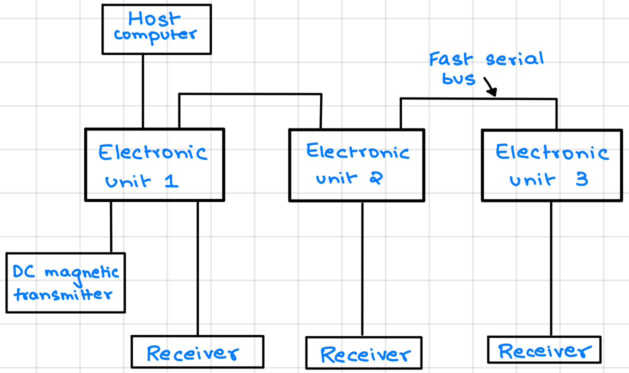
UNIT-1

FLOCK OF BIRDS

-> It is a magnetic tracker that uses DC magnetic fields.

-> It uses a distributed computing architecture to reduce computation time & maintain high tracker update rates.

-> There are 3 electronic units — Bird 1, 2 & 3 - interconnected w/ a fast serial bus, called the Fast Bird Bus.



CUBIC-MOUSE

-> Uses indexed motion — virtual objects are selected & deselected repeatedly. When the object of interest is deselected, it stays in place, while the user moves his or her arm back & then reselects the object & moves it using the tracker.

-> It consists of a plastic cube that houses a Polhemus Fastrack, 3 mutually perpendicular translating rods, & 6 appⁿ-programmable control buttons.

-> It is primarily designed for the manipulation of a single, large virtual model.

-> Users hold the mouse w/ their non-dominant hand & push the rods & buttons w/ their dominant hand.

TRACKBALLS

-> It is a class of interfaces that allow navigation / manipulation in relative coordinates.

eg -> Logitech Magellan

- It is a sensorized cylinder that measures 3 forces & 3 torques applied by the user's hand on a compliant element.
- Forces & torques are measured indirectly based on the spring deformation law.
- An alternate way to control VR objects is through force control — forces measured by the trackball are used to control forces applied by the VR object on the simulated environment.

-> Trackballs suffer from sensor coupling. (Translation + Rotation)

BIOMUSE

-> It is a bioelectric signal controller.

-> It accepts i/p of brain electrical activity, muscle signals & eye movement, captured through a headband & small skin sensors.

-> The o/p channel can be used to activate events in a computer system.

-> The BioMuse receives data from 4 main sources of electrical activity in the human body - EMG signals (muscle), EOG signals (eye), EKG (heart) & EEG (brain waves).

3D PROBES

-> Users felt a need for an i/o device that would be intuitive to use, inexpensive, & allow either absolute or relative position control of the simulation.

-> It consists of a small sensorized mechanical arm that sits on a support base, w/ a small footprint.

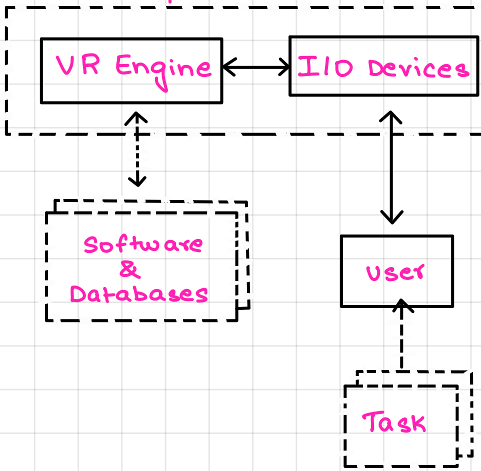
-> The probe has 6 joints.

-> Each rotary joint represents 1 degree of freedom, & thus the probe has 6 degrees of freedom, allowing simultaneous positioning & orienting of its tip.

-> The tip position wrt the base is obtained through direct kinematics calculations, based on sensor values & the length of the links.

5 COMPONENTS OF A VR SYSTEM

VR System Architecture



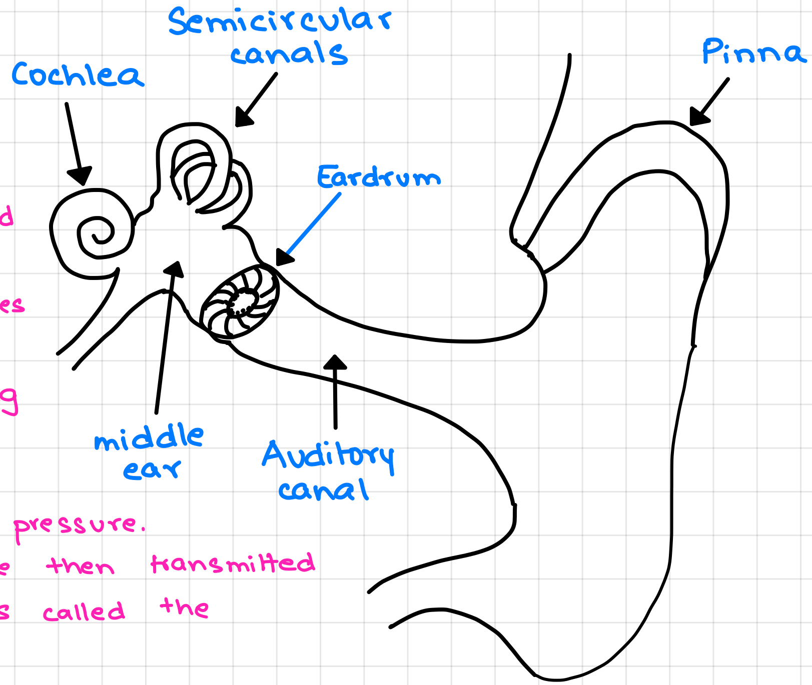
THE EAR

-> The Ear has 3 distinct parts – the outer ear, the middle ear, & the inner ear.

-> The outer ear helps direct a sound wave into the middle ear. It therefore acts as a filter & modifies the captured sound.

-> The pressure wave travels along the ear canal & eventually strikes the tympanic membrane (eardrum) which oscillates w/ the changes in pressure.

-> Movements of the membrane are then transmitted through a system of small bones called the ossicular system.



VR APPLICATIONS

① MEDICAL:

-> Virtual anatomy – focuses on anatomy teaching. Creating animations of 3D organ models based on the Visible Human Database is a precursor to developing a computerized anatomy teaching curriculum.

-> Triage & Diagnostics – once anatomical models are developed, they can be modified to add pathology such that students learn to diagnose abnormalities.

-> Surgery – training

-> Rehab

② ART, EDUCATION

-> video games

-> Interactive teaching

-> Diagnosing ADHD

③ ARMY / MILITARY TRAINING

-> Mine detection & defusion