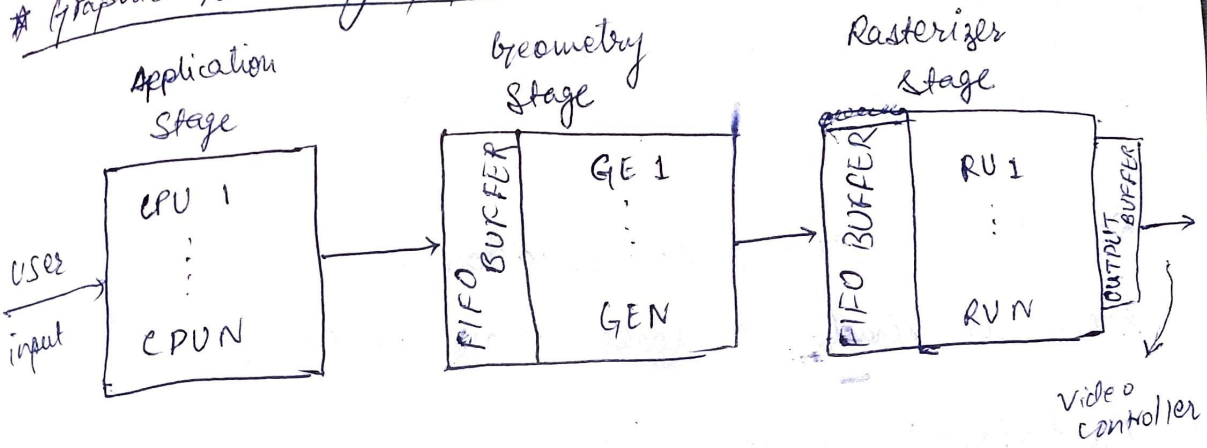
VR

UNIT III

* Graphic Rendering pipeline.



* Graphic ~~Modeling~~ rendering has 3 functional stage.

(a) Application Stage

*) done entirely by software in CPU.
*) reads world geometry database
*) reads user inputs by devices like mice, trackballs, trackers or sensing gloves.

Task & function:

*) In response to users input, application stage may change the view in simulation or change orientation of virtual objects.

(b) Geometry stage

*) input : application stage result
*) implemented in either software or hardware
*) ~~consists of model transformation (eg~~
~~eg translation~~
~~rotation~~
~~scaling~~

Task & function:

*) consists of - model tranformation (translation, rotation, scaling),
- mapping
- clipping
- scene projection
- lighting computation

*) lighting computation -

computes surface color based on type & no. of simulated light source in, lighting model, atmospheric effects (eg fog) etc.

*) Result of lighting computation -

Scene is shades & looks more realistics.

© Rasterizing stage:

*) done in hardware to gain speed

*) input: result of geometry stage.

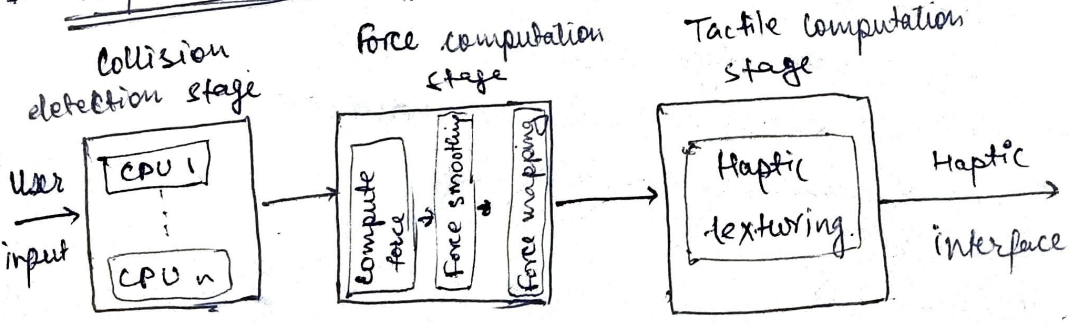
*) Task / function → colour, texture obtained from geometry stage

→ converts vertex info into pixel info needed by video display.

→ perform antialiasing in order to smooth out the jagged appearance of polygon edges.

---

* Haptic Rendering:



3 stages

(a) Collision Detection Stage:

*) input: physical characteristics of 3D objects are loaded from database.

eg- smoothness, weight,
surface temperature
surface compliance

*) function:
performs collision detections to determine which virtual objects collide.

*) Unlike Graphic pipeline, only colliding structures in the scene are passed down to subsequent pipeline stage.

(b) Force computation stage:

*) input: result of collision detection stage

*) function:
compute collision forces based on physical simulation models

*) simplies model is based on Hooke's law, where the contact force has spring ~~not~~ like dependence on the degree of surface deformation.

*) Other models involve damping & friction forces, are more realistic,
also more computationally intensive.

*) Includes - force smoothing
- force mapping.

(c) Haptic texturing

*) input: result of force computation stage.

*) function:
- renders touch feedback component of simulation.

*) its computed effects eg vibrations or surface temperature are added to force vector and sent to haptic output display.

* PC Graphic Architecture → PC VR Engine _ Intel Co.

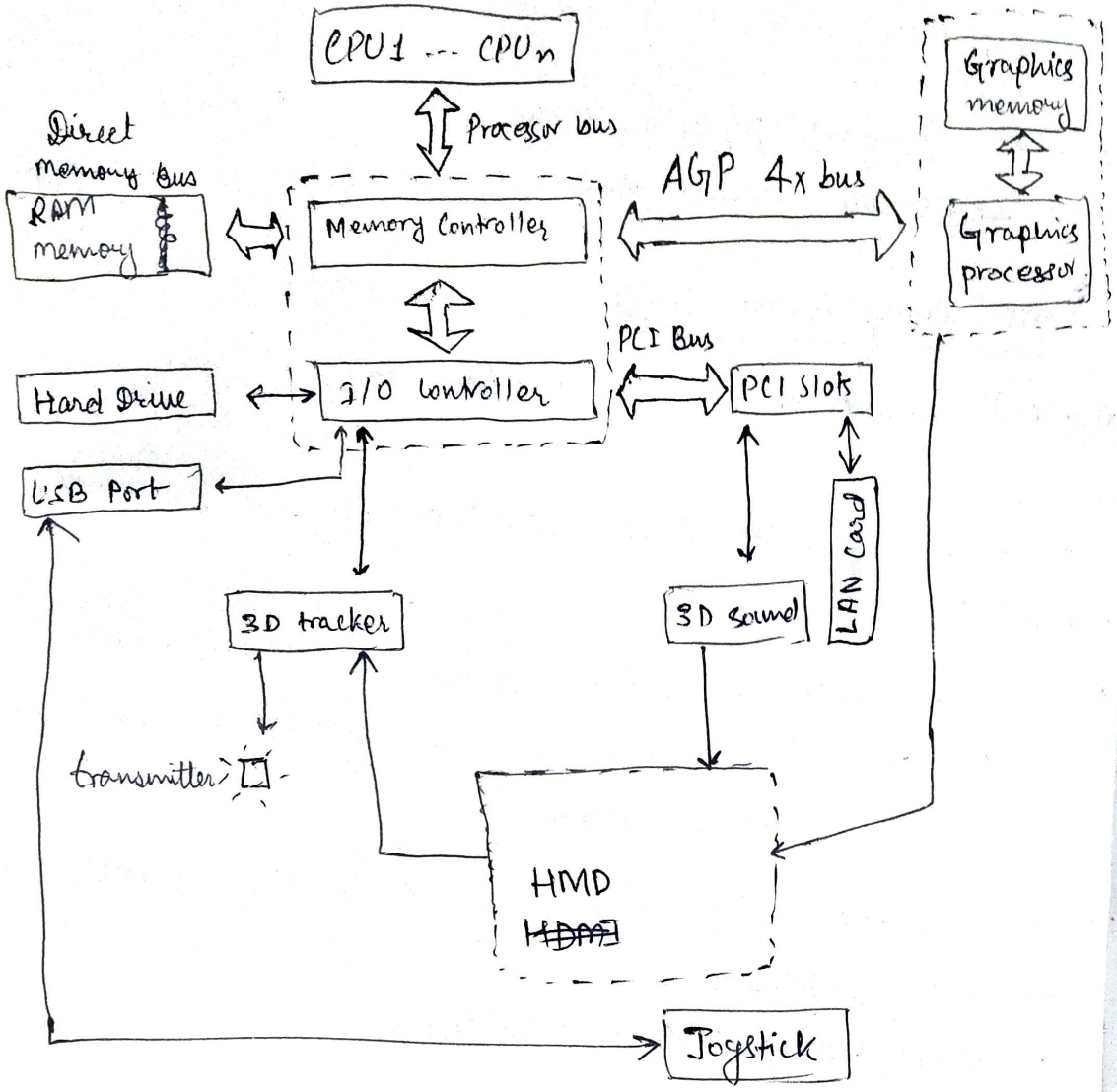


Fig: PC VR Engine. Adapted from Intel Co.

*) In order for a PC to be a VR engine, it needs to be integrated with specialized UO devices used in simulation.

*) input: The user provides input through
- 3D tracker
- joystick

*) feedback: The user receives 3 types of feedback
a) video feedback on HMD
b) haptic " " joystick
c) audio " " headphone connected to 3D sound card

*) tracker is connected to serial port-
*) joystic communicates through USB port
*) HMD connects to graphics card RGB output port-
*) 3D sound card plugs directly in PCI ~~slot~~ bus.

*) AGP bus :

- Intel AGP (Accelerated Graphics Port) bus solves the PCI bandwidth problem.
- AGP bus operates at much higher bandwidth than PCI bus
- AGP bus transfers texture & other graphics data directly from system RAM to video memory on graphics card.

*) Transfer is mediated by memory control chip.
*) This allows simultaneous CPU-to-RAM traffic
*) AGP transfer rate is ~~over~~ 2 GB/sec using AGP 8x bus.
*) AT these high rates, it's possible to reserve a part of ~~ram~~ RAM to serve as secondary video memory.

~~*) This reduces overall~~

*) AGP allows graphics controller to send new read requests to RAM simultaneously receiving data from previous requests
*) This reduces overall system latency.
*) Second chip on Intel chipset is UO controller hub, which allows traffic to hard disks, serial ports, USB etc.

5 methods to construct Virtual Objects/shape/surface.

*) Using Toolkit Editor

*) Using Online 3D object Database - buy ready made 3D models. - 3 levels of details L, M, H

*) Importing CAD files - AutoCAD 3-D studio, standard programs in mech & archit designs.

*) Creating surfaces using 3D Digitiger - approach used when a given 3D virtual object is not a part of existing db.

*) " " " " Scanner)

- faster way to build 3D models, especially for large objects.

5 parameters involving Kinematic Modeling:

*) Kinematic modeling :-

↳ determines location of 3D object w.r.t world system of coordinates

↳ determines their motion in virtual world.

*) 5 parameter :- (H O T Object View)

(i) Homogeneous Transformation Matrix

↳ 4 X 4 matrix

↳ offers certain computational advantages:

*) treat both rotations & translation in uniform way

*) they can be compounded such that complex kinematics relationship can be ~~modeled~~ modeled.

(ii) Object Position:

↳ Object Surface Modeling uses (x, y, z), vertex coordinates

↳ expressed in an object system of co-ordinates

↳ This system of co-ordinates is attached to the

object center of gravity

*) When object moves in virtual world, its system of coordinates move with it.

*) Therefore, position & orientation of object vertices in the object system of coordinates remain invarient, regardless of object position in the scene.

(iii) Transformation Invarient

*) Position & orientation of 3D object in virtual world is mapped to position & orientation of 3D tracker attached to the sensing glove worn by the user.

(iv) Object Hierarchies:

*) groups of objects which move together as a whole but whose parts can also move independently.

*) A hierarchy implies atleast 2 levels of virtual objects. — parent ~~level~~ object : higher level objects
— child object : lower level objects.

(v) Viewing the 3D World:

*) Mapping the virtual objects to camera coordinates (eye space) is the first task in the geometry stage of rendering pipeline.

*) followed by lighting, perspective projection, clipping & screen mapping.

* 5 parameters involved in Physic based Modeling

(Note: 4 headings/parameters are same as in ~~Haptic~~ Haptic rendering)

(i) Collision Detection:

*) first stage in Haptic rendering

*) determines whether 2 or more objects are in contact with each other.

*) Can be considered as form of Haptic Clipping as only objects that collide are processed by haptic rendering pipeline.

(ii) Surface Deformation:

*) Collision detection is followed by collision response.

*) depends on characteristic of Virtual objects & an application being developed.

*) If objects in contact is nonrigid, then one form of collision response is Surface deformation

*) It changes the 3D object geometry & thus needs to be coordinated with graphics pipeline.

(iii) Force Computation:

*) for computation it takes into account –

→ type of surface contact

→ kind of surface deformation

→ object's physical & kinematic characteristics

*) The simplest surface contact is single-point contact
*) 2 types of objects → elastic virtual object
→ plastic virtual object.

(IV) Force smoothing & Mapping.

*) force shading changes direction of feedback force.
*) feedback force is produced during interaction b/w polygonal surface.
*) This is done in order to simulate contact with smooth curved surfaces.
*) Direction of ~~computed force~~ contact force — computed based on weighted normals of vertices of the polygon being contacted.

(V) Haptic Texturing.

*) last stage of haptic rendering pipeline.
*) haptic texture enhances the realism of ~~virtual~~ physical model of object surface
*) haptic texture can add new info about object eg slippery, cold smooth etc.

✡ Scene illumination & Texture Mapping from context of Object Visual Appreance.

(*) Object Visual Appearance

→ Modeling the geometry of a virtual object is the necessary first step in creating a realistic-looking 3D scene.

→ next step is to illuminate the scene such that object becomes visible.

→ Object Visual Appearance depends strongly on the
(i) type & placement of virtual light source,
(ii) object's surface reflectivity coefficient.

(*) Texture Mapping:

↳ Method of improving image realism without adding more surface polygons.

↳ the location of object vertex is stored in object coordinate space (x, y, z)

↳ texturing is a technique performed in the rasterizing stage of the graphics pipeline

↳ done in order to modify object model's surface properties such as color, specular reflection, pixel normals.

(*) Scene illumination:

↳ determines light intensities on the object surface.

↳ 2 types:
(i) Local scene illumination - treats interaction b/w objects and light source in isolation. neglects interdependence of objects
(ii) Global scene illumination - models interreflections b/w objects & shadows, results in more realistic looking scene.