**VR Note for Exam**

**1 VR: A Brief History and Some Basic Concepts**

1. **What is Virtual Reality?**
   1. Def: a human-computer interface involves:
      1. **real-time simulation**
      2. **interactions**
      3. **multiple sensorial channels**

visual, auditory, tactile, smell, and taste.

* 1. four levels of virtual reality:
     1. **Passive**: user does not have much control.
     2. **Exploratory**: can move in a virtual world but cannot modify content.
     3. **Interactive**: explore and interact within a virtual world.
     4. **Collaborative**: Multiple users may interact with each other.

1. **Human Computer Interfaces**

concern with how to capture the user’s interaction instructions, the computer can analyze them and then generate its reactions back to the user.

* 1. **Vision-**Main difficulty: traditional computer vision technology focuses on pixels, it is difficult to recover high-le vel semantics from pixels.
  2. **Sound**
  3. **Gesture**
  4. **Tactile Sensing**
  5. **Other Senses**: smell.

1. **Types of VR Systems**
   1. Immersive VR: HMD, the CAVE system.
   2. Non-immersive VR
      1. Unlike immersive VR, non-immersive VR systems do not provide 360 degrees of vision.
      2. Low cost
      3. additional equipment: stereo glasses.
   3. Augmented VR (or AR)
      1. the use of transparent glasses onto which data can be projected on it.
      2. two common types: a pair or a single piece of lens; the flight helmet.
   4. Telepresence
      1. Visualize a remote site and to perform some tasks through controlling a remote machine or robot.
      2. two-way communication between the user and the remote robot.
      3. Applications: Space exploration, Remote surgery, Danger operations.
2. **Position Sensors-**tracking approaches
   1. **2D Mouse**
   2. **Mechanical Tracker**
   3. **Ultrasonic Tracker**
   4. **Magnetic Tracker**
3. **A Typical Immersive System-**Include: HMD, motion sensor, position tracking devices, 3D graphics workstation.
4. **Latency**: (sensor) move head-the signal is generated- (system) receives the signal-processes the signal- (computer) receives the information-processes it-updates the database- (the image generator) renders an updated image.

This would cause uncomfortable feeling.

**2 Input Devices**

1. Basic of 3D Tracking
   1. **Tracker**: measure the position or orientation of an object.
   2. **Degrees of Freedom (DOFs)**
      1. the number of independent motions to be tracked.
      2. Examples devices of different DOFs:
         1. **2 DOFs** (2D mouse)
         2. **3 DOFs** (accelerometers or gyroscopes)
         3. **6 DOFs** (most mobile phones)
   3. **Tracking Accuracy**
      1. Accuracy: the similarity of the object’s actual position and the measurement by the tracker. operating range:distance at which the accuracy is acceptable.
      2. Jitter: the changes in tracker output.
      3. Drift: the inaccuracy that grows as time passes. Require periodically resetting the error by a secondary tracker.
      4. Latency: the time delay between the moment of action and that of receiving the signal.

Cause simulation sickness-methods to reduce latency: faster communication lines; Reduce the processing delay of each step; synchronizing the tracker measurement, communication, rendering, and display loops.

* 1. **Tracker Update Rate**
     1. the number of measurements that the tracker reports every second
     2. The larger this update rate is, the better the dynamic response of the simulation.
     3. single tracking system needs to support more trackers to track multiple objects -> the sampling rate need to be reduced.

1. **Types of 3D Trackers**
   1. Mechanical Trackers
      1. Adv:
         1. General: Accurate, Small latency.
         2. Mechanical Mocaps 机械动作捕捉: No line-of-sight restriction, small drift, unlimited workspace.
      2. Dev:
         1. limited range of motion
         2. heavy weight can lead to fatigue
         3. interfere with motion and reduce freedom.
   2. Magnetic Trackers
      1. **AC Magnetic Trackers-**Problems: Eddy currents distort magnetic field and accuracy is degraded.
      2. **DC Magnetic Trackers:**
         1. eddy currents to disappear-Short time delay between the excitation of the transmitter.
         2. **Problems:** Ambient Error 环境误差; Metal Interference.
      3. **Adv:**
         1. Accurate, but not as accurate as mechanical trackers
         2. light weight
         3. No line-of-sight restriction
      4. Limitation: Affected by metallic objects/surfaces.
   3. Ultrasonic Trackers (non-contact)
      1. Reason for the low update rates: the need to wait for previous echoes to die out.
      2. Adv:
         1. No interference with metal
         2. Small and light weight
         3. Cheap
      3. Dev:
         1. Line-of-sight restriction
         2. Accuracy depends on sound velocity
         3. Susceptible to noise
   4. Optical Trackers (non-contact)
      1. **Outside Looking In**
         1. fix the sensor, put markers on user.
         2. Example: MotionAnalysis.
         3. Tracking sensitivity is degraded as: **distances**
            1. **among the markers decrease**
            2. **between the user and the camera increases**
            3. **between adjacent cameras increases**
         4. **Optical Motion Capture System Pipeline:** 
            1. **Calibration**
            2. **Motion Capture**
            3. **3D Position Reconstruction**
            4. **Fitting the Skeleton**
            5. **Post Processing**
         5. **Adv: High precision**
         6. **Dev: Occlusion problem**
      2. **Inside Looking Out**
         1. attach the sensor to the object, put markers on the walls.
         2. **Example: HiBall Tracker from 3rd Tech.**
         3. **Adv:** high precision and large tracking space.
   5. Inertial Trackers
      1. Three types of inertial sensors:
         1. **Gyroscope:** angular velocity, degrees per second.
            1. Three mutually orthogonal axes, xyz.
            2. accurate in the short term, but suffer from drift in the longer term.
         2. **Accelerometer:** linear acceleration, m/s2.
            1. unreliable in the short term, accurate in the long term.
         3. **Magnetometer:** magnetic field strength, uT.
            1. Depend on its latitude and longitude.
            2. Affected by metallic objects.
      2. Adv:
         1. Unlimited range of tracking
         2. No line-of-sight constraints
         3. Low jitter
      3. Drawbacks
         1. Rapidly accumulating errors or drift.
         2. Gyroscope bias -> orientation error increases with time.
         3. Accelerometer bias

Solution: periodically reset the error using another type of tracker.

* 1. 3D Navigation/Manipulation Devices
     1. **Navigation**: interactive change of view in the virtual environment.
     2. **Manipulation**: selecting an object for modification.
     3. **Example**: **Logitech Magellan, MicroScribe 3D.**

1. **Gesture Devices**
   1. measure the real-time positions of the user's fingers.
   2. allow gesture-based interactions with the virtual environment.
   3. Example: **The 5DT Data Gloves, Didjigloves.**

**3 Output Devices: Graphical Displays**

1. **The Human Visual System**
   1. **Color Perception**
      1. rods and three kinds of cones
         1. Rods for night vision
         2. Cones for color vision
            1. L-cone: red.
            2. M-cone: green.
            3. S-cone: blue.
   2. Eye is most sensitive to green.
   3. **Depth Perception:**
      1. Near objects:
         1. **Convergence** 汇聚: allow us to focus on a nearby object.
         2. **Accommodation** 调节
         3. **Disparity 视差:** 
            1. location difference of objects in the two eyes.
            2. estimate the depth of the object.
         4. **Parallax 视差运动:** when viewer moves his head horizontally, nearby objects appear to move more than distant ones.
      2. Objects far away (depth cue):
         1. **Occlusion:** nearby objects tend to occlude distant objects.
         2. **Perspective:** distant objects appear smaller.
         3. **Haze:** distant objects tend to be grayer.
         4. **surface texture:** similar to perspective.
   4. Important factors that determine the immersiveness of a VR system:
      1. depth perception
      2. large FOV
      3. high-resolution images
2. **HMD Image Formation**
   1. Lens in the HMD magnifies the image to a virtual image, which is larger and further away from the viewer.
   2. **convergence and accommodation conflict**: eyes converge on this object, but remain focusing on the virtual image plane.
3. **Personal Graphics Displays**
   1. Head-mounted displays (HMDs):
      1. Example: Oculus Quest 2, Oculus Quest Pro, PlayStation VR2.
      2. AR glasses concern how to project computer information on the glass: Microsoft Hololens, Google Glass, Apple Vision Pro.
   2. Hand-held displays: **Virtual binoculars SX.**
   3. Autostereoscopic monitors
      1. **Passive Autostereoscopic Displays:** user’s head is not tracked, **DTI 2018XL Virtual Window.**
      2. **Active Autostereoscopic Displays:** track the user’s head motion, **Ecomo 4D Displays.**
4. **Large-Volume Displays**
   1. allow several users to view a stereo image of the virtual world.
   2. Can be further classified:
      1. monitor-based: **PV290.**
      2. projector-based
         1. **Digital Projectors:**
            1. LCD

Adv: more light efficient; produce sharper images.

Dev: Pixellation (pixels of an image can be easily seen); more bulky; Possibility of “dead pixels”.

* + - * 1. DLP

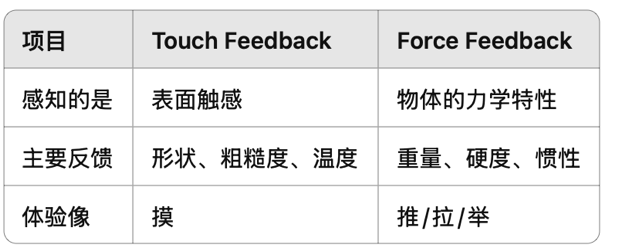
Adv: produce smoother images; produce images with higher contrast.

* + - 1. Example:PanoWall Projector Arrays, V-Dome.
  1. Compared to personal displays: improves users’ freedom and natural interaction.
  2. **Polarization:** a pair of polarized projectors + a pair of inexpensive polarized glasses; glasses have lenses polarized differently for the right and left eyes.

1. **Volumetric Displays:** A display of 360° view of objects

**4 Output Devices (Sound and Haptics)**

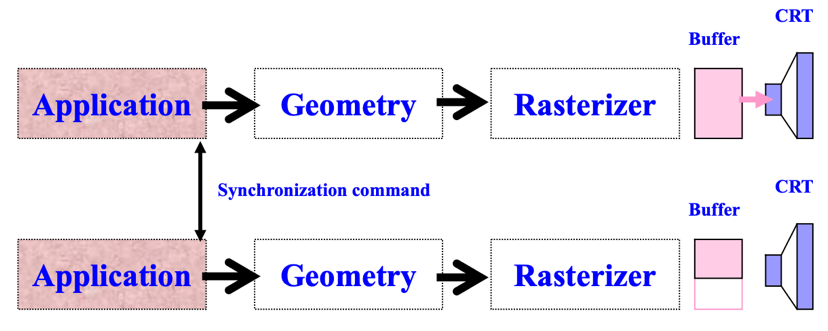
1. **Sound Properties**
   1. a pressure wave, travels through vibration 振动.
   2. Ordinary wave properties and behaviors: **reflection 反射, refraction 折射, diffraction 衍射.**
   3. Key function of the cochlea 耳蜗: convert sound vibrations into electrical signals.
2. **3D Sound**
   1. Difference between stereo sound and 3D sound: in 3D sound, the sound changes according to the location and orientation of the user’s head.
   2. Cues to determine the 3D location of sound source:
      1. **Azimuth cues**
         1. the horizontal angle
         2. Lower frequency sound, Interaural Time Difference (ITD) dominates the identification of angular location-the source is not far from the user.
      2. **Elevation cues**
         1. the vertical angle
         2. Outer ear is important.
      3. **Distance cues:** the distance.
   3. **Head-Related Transfer Function (HRTF)**
      1. Each person has his own HRTF signature.
      2. **Convolution:** Determine HRTF, take input sound, apply the transform, play back the output sound to the user make user have the sensation of hearing that sound coming from the original source.
      3. **Spatial Recognition Rate:** drops when someone else’s HRTF is used.
3. **Haptic Feedback**
   1. **Touch feedback:** real-time information on contact surface geometry, roughness and temperature.
   2. **Force feedback:** real-time information on virtual object surface properties, weight and inertia (惯性).



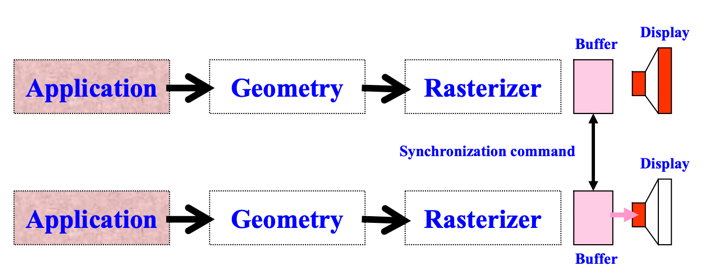
* 1. **Tactile Feedback Interfaces**
     1. Tactile interfaces provide: Vibrotactile feedback 振动触觉反馈, Temperature feedback.
     2. Example: **iFeel Mouse, The CyberTouch Gloves, The Temperature Feedback Gloves.**
  2. **Force Feedback Interfaces**
     1. Provide feedback forces.
     2. Example: Joysticks 操纵杆, Haptic arms 触觉机械臂.

**5 Computing Architectures for VR**

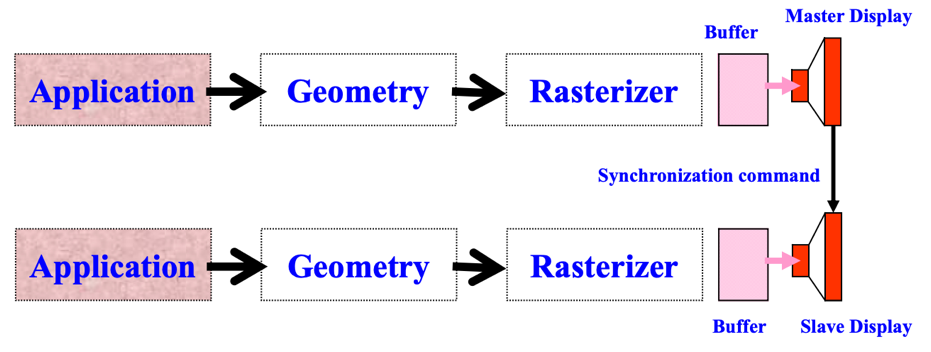
1. **VR System Architecture**
   1. Tasks:
      1. read the input devices
      2. update the virtual world.
      3. render the outputs
      4. feed it to the output devices
   2. A powerful computer architecture in VR engine needs to provide:
      1. Low latency
      2. Fast graphics rendering
      3. Haptics rendering
   3. **The Rendering Pipeline**
      1. **Rendering:** the process of converting 3D geometry models to 2D images.
2. **Graphics Rendering Pipeline**
   1. Application stage
      1. In CPU.
      2. The loop:
         1. Get user inputs
         2. Update the view
         3. Select the relevant objects, send them to the subsequent stages.
   2. Geometry stage
      1. In GPU.
      2. Tasks:
         1. Model transformations
         2. Shading computations
         3. Scene projection
         4. Clipping
      3. **Illumination**
         1. describes the intensity of light.
         2. Calculates the surface color based on:
            1. light sources
            2. lighting model
            3. properties of objects
            4. atmospheric effects
         3. **The Polygon Shading Model**
            1. **Flat Shading:** assigns same color to all pixels.
            2. **Gouraud Shading:** interpolates colors inside the polygons.
            3. **Phong Shading:** interpolates the vertex normal inside the polygons.
   3. Rasterizer stage
      1. Converts the vertex information into pixel information.
      2. Tasks:
         1. Break down geometry primitive into pixel fragments.
         2. Antialiasing 抗锯齿
         3. Texture mapping
   4. **Parallel Architectures**
      1. To speed up the performance.
      2. The **geometry stage**contains several geometry engines (GEs).
      3. The **rasterizer stage**also contains a number of rasterizer units (RUs).
      4. GEs write their results to the FIFO, RUs read the FIFO.
   5. **Graphics Pipeline Bottlenecks**
      1. **Testing for Pipeline Bottlenecks**
         1. CPU 100% -> **CPU-limited**
         2. the number of light sources or polygons reduces -> performance increases -> **transform-limited**
         3. the image resolution or the number of polygons reduces -> performance increases -> **fill-limited**
   6. **Graphics Pipeline Optimization:**
      1. **Optimizing the Application Stage:**
         1. Faster CPU
         2. Faster RAMs
         3. better compiler
         4. clever programming
      2. **Optimizing the Geometry Stage:**
         1. reduce the number of light sources
         2. reduce the scene complexity
      3. **Optimizing the Rasterizing Stage**
         1. reduce the image resolution
         2. reduce the number of polygons
3. **The Haptics Rendering Pipeline**
   1. **The Collision Detection Stage:** determine if there is a collision between the user and the object, the structures are passed to the next stage.
   2. **The Force Computation Stage:**
      1. **Computing force:** more objects collide, more complex.
      2. **Force smoothing:** adjusts the direction of the force to avoid sharp transitions.
      3. **Force mapping:** projects the computed force to the device.
   3. **The Tactile Computation Stage**
4. **GPU Architecture**
   1. CPU features:
      1. no native data parallelism
      2. Few arithmetic units
      3. low latency
      4. not high bandwidth
   2. Graphics rendering feature:
      1. Lots of data to process
      2. Lots of calculations
      3. Lots of parallelism
      4. Simple control
      5. Latency-tolerant
   3. GPU features:
      1. highly programmable
      2. high precision computations
   4. Advanced hardware graphics pipeline:
      1. programmable vertex processors
      2. programmable pixel processors
   5. **GPU Memory Addressing:** GPUs do not support direct memory addressing, only supports vector addressing. Reason is to allow efficient transfer of large data.
5. **Distributed VR Architecture**
   1. Use two or more rendering pipelines.
   2. **Side-by-side displays**
      1. in desktop VR workstations, or in large volume displays.
      2. One solution: a separate PC equipped with a graphics accelerator for each projector.
      3. Another solution: one PC with several graphics accelerator cards (one for each monitor).
   3. **Multi-pipeline Synchronization**
      1. Reduce the overall system latency
      2. Create consistent frame rates
      3. **Software Synchronization**



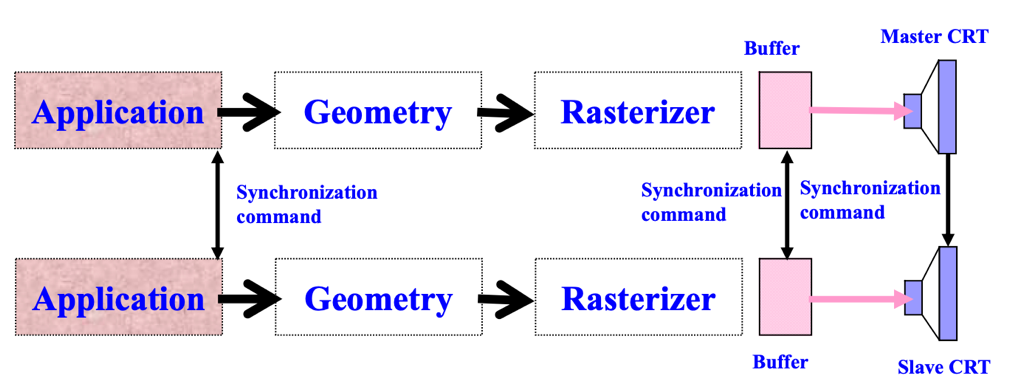
* + - 1. Synchronizing the application stages.
      2. Not adequate as it does not consider the possible lack of symmetry in the loads handled by each pipeline.
    1. **Frame Buffer Synchronization**



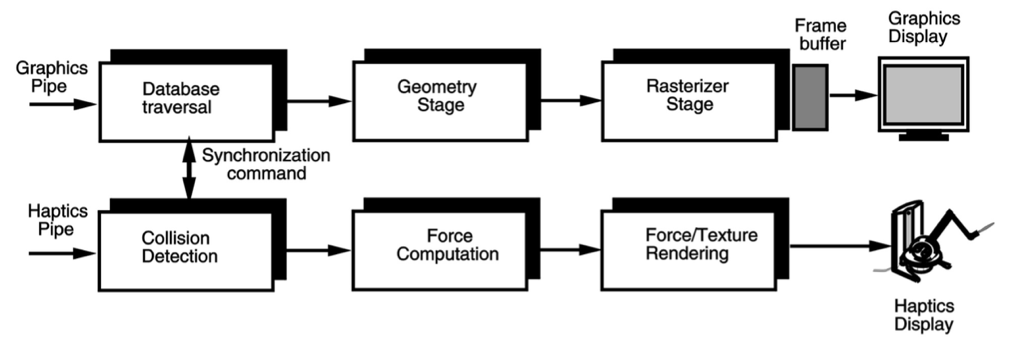
* + - 1. Synchronizing the buffer swapping.
    1. **Video Synchronization**



* + - 1. One of the displays becomes the master while the other(s) are slaves.
      2. Slaves follows the master.
    1. Best method: is to have software + buffer + video synchronization of the two (or more) rendering pipelines.



* 1. **Graphics and Haptics Pipeline Synchronization**
     1. Synchronization is done in the application stage.
     2. Two implementations:
        1. Force computation is done on host computer.
        2. Force computation is done by the processor embedded in the haptic interface controller.
     3. Decoupling the graphics and haptics pipelines is necessary since they have significantly different output rates.



1. **PC Clusters**
   1. Cannot be done by a single PC:
      1. Not enough PCIe slots.
      2. Not good throughput.
   2. two alternatives:
      1. Large, multi-pipeline workstations-drawbacks:
         1. High cost
         2. Each pipeline can only support one or two displays.
      2. PC cluster
         1. Reduce cost.
         2. Image tiling can be done by software.
         3. Need high-speed LAN to allow synchronization of the output graphics by a control server.
         4. LAN throughput becomes the main limiting factor
         5. Example: **Stanford U.**
   3. **WireGL:** A scalable system for clusters.

**6 Virtual Reality Modeling**

VR object modeling issues:

1. Geometric modeling:
   1. Types of object surface:
      1. polygonal meshes
         1. Triangular Meshes: preferred because computationally efficient.
         2. Using linear functions
         3. Constructed using OpenGL or a model editor (Maya and AutoCAD); Created using a 3D digitizer (stylus), or a 3D scanner (tracker, cameras and laser); Purchased from some online geometry stores but **static**.
         4. CAD-based Models

Created using Maya/AutoCAD.

Each moving part is a separate file.

Files need to be converted to formats compatible with VR toolkits or game engines.

Adv-use of available models in existing applications.

* + 1. splines 样条曲线 (for curved surfaces)
       1. Spline-based Surfaces:

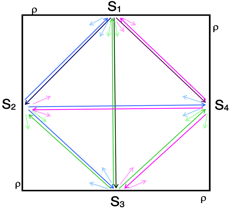
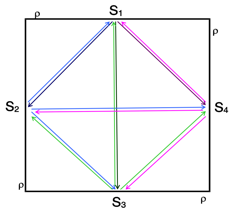
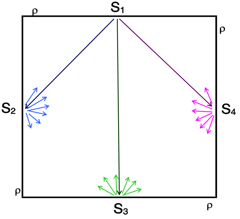
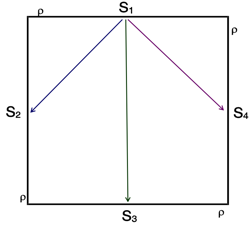
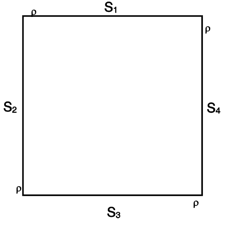
Compared to polygons: higher degree functions, use less storage and provide increased surface smoothness.

Parametric surfaces are extension of parametric splines with point coordinates.

* + 1. point based data

Conversion of scanner data: A scanner produces a dense **point cloud**; Using packages such as Geomagic Wrap to transform the point cloud data into surface data.

* 1. Object appearances:
     1. Scene illumination (local or global)
        1. **Local illumination**(flat shading, Gouraud shading, Phong shading):  considers objects individually.
        2. **Global illumination** considers all objects in the scene.



* + 1. texture mapping
       1. Performed in the rasterizer stage; uses a mapping function to map object parametric coordinates to texture space coordinates.
       2. Adv

Increase scene realism.

Provide better 3D spatial cues.

Reduce the number of polygons

* + - 1. The Texturing Functional Sub-Stages (Rasterizer)
         1. Map pixel location to object space location
         2. Use mapping function to find parameter space
         3. Use corresponder function to find texel
         4. Apply value transform function
         5. Modify illumination equation value



* + - 1. How to create textures: available online in texture “libraries”; from scanned photographs or using an interactive paint program.
      2. Image Texture:

A texel is a pixel in the texture image.

(Hardware operations during Texture Mapping)

the size of the polygon > the size of the texture - hardware performs **magnification**

the size of the polygon < the size of the texture - hardware performs **minification**

Use bilinear interpolation to assign colors to pixels

* + - 1. Multi-texturing

Several texels can be overlaid on one pixel.

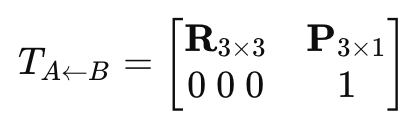
A texture **blending cascade**(混合级联) is made up of a series of texture stages.

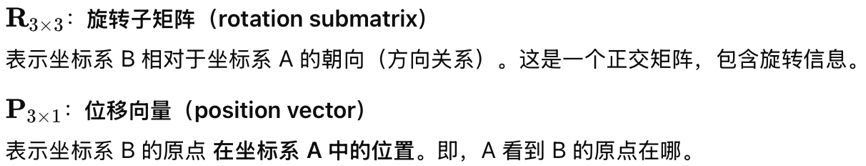
For Bump Mapping (凹凸贴图): Simulation of lighting effects caused by irregularities on object surface; Encodes surface irregularities as an image texture; No change in model geometry; No added computations; Part of the per-pixel shading operations of the NSR.

For lighting: Two traditional methods: vertex shading and light maps. **vertex shading-**each vertex is provided with some information to computer the vertex color; Lighting computation is real-time but require lots of polygons. **light maps-**compute high quality lighting effects in advance as 2D textures; Not real time, when objects move, light maps need to be recomputed.

**NSR:** producing high quality lighting in real-time; Requires a **normal map**; each pixel is provided with some information to compute the pixel color; allows diffuse, specular and spot light effects dynamically on per pixel.

1. Kinematical modeling (运动建模)
   1. Homogeneous transformation matrices (齐次变换矩阵)
      1. The general format of Homogeneous transformation matrices 从坐标系 B 到 坐标系 A 的变换





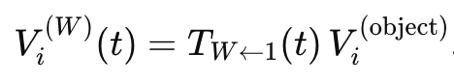
* + 1. Adv:

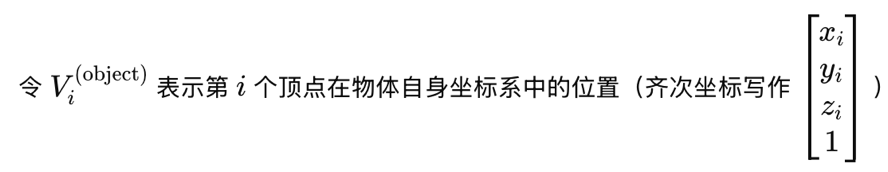
Treat both translations and rotations in the same way

Can be compounded

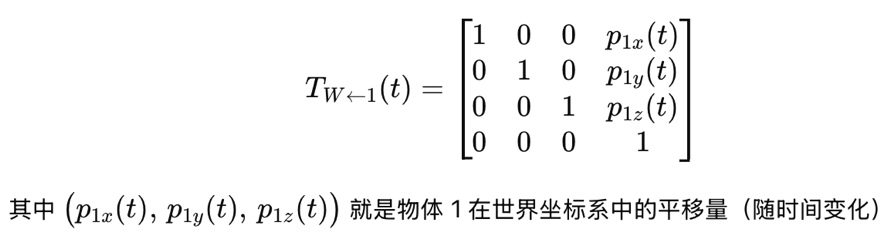
Easily invertible

* 1. Object position

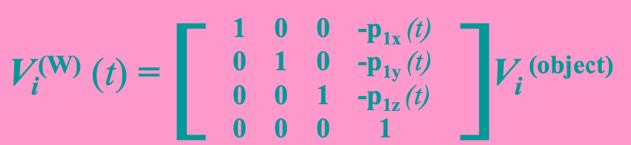




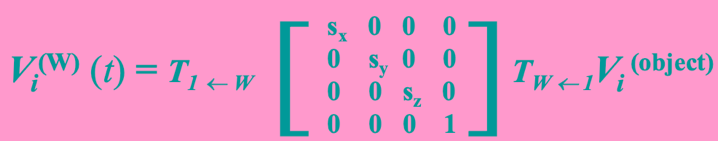
* + 1. Translation



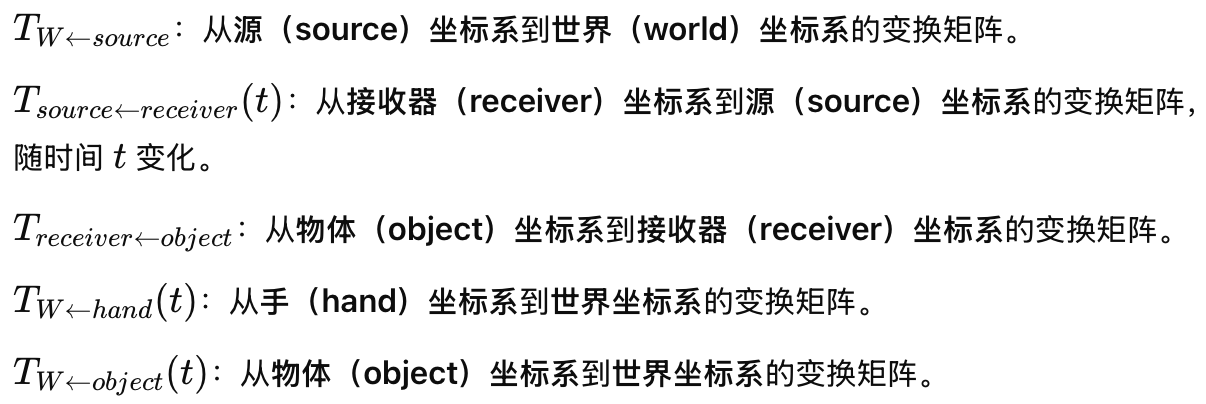
The object translates back to its initial position

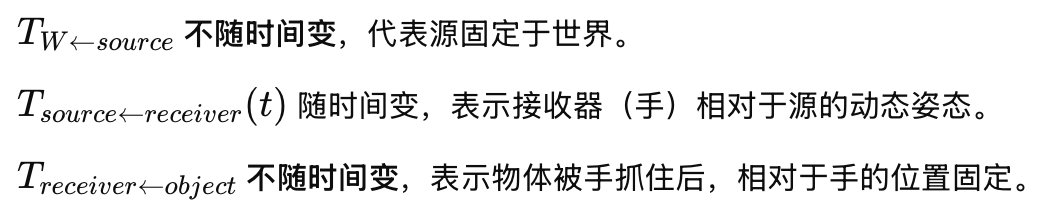


* + 1. the object needs to be scaled



* 1. Transformation invariants (不变量)





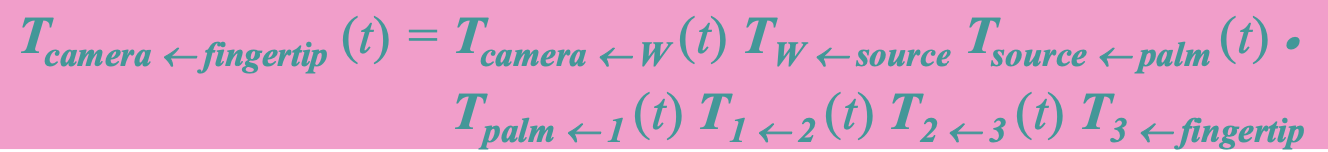
手在世界坐标系中的姿态：

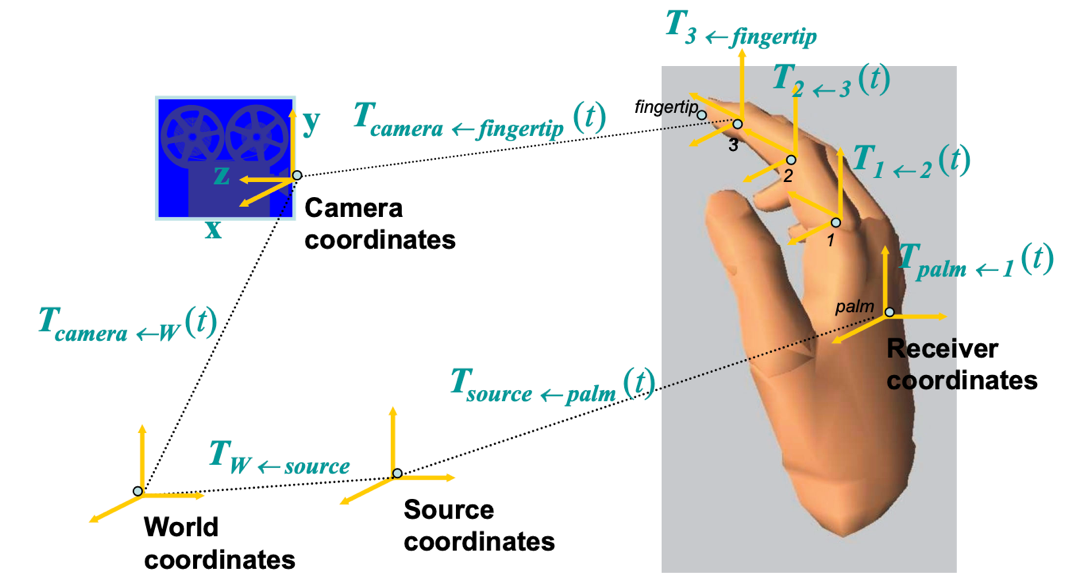


物体在世界坐标系中的姿态：



* 1. Object hierarchies (层级结构)





* 1. Viewing the 3D world

The Geometry Functional Sub-Stages (The rendering pipeline-geometry)

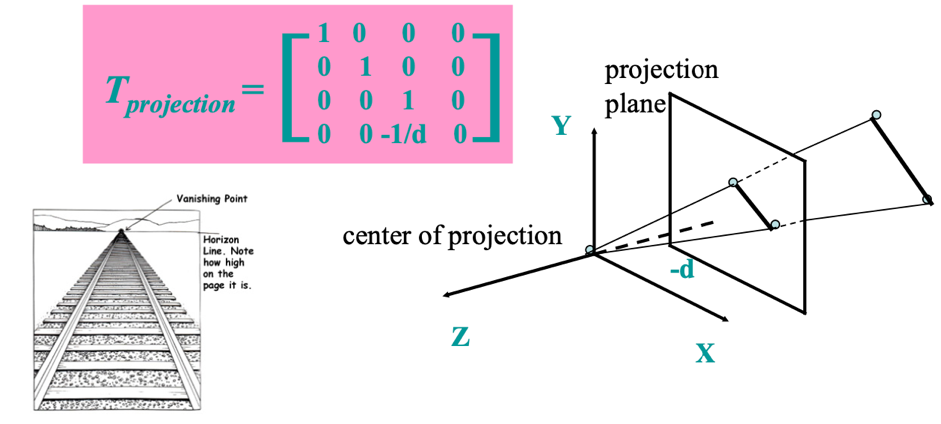
* + 1. Model & View Transformation

**Model transformations:** link object coordinates to world coordinates.

**view transformation**: maps world coordinates to camera coordinates.

* + 1. Lighting
    2. Perspective Transformation

the projection reference point is at the origin of the coordinate system, and the projection plane is at **–d**.



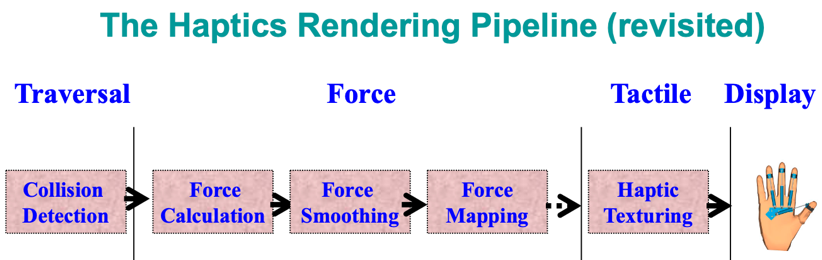
The portion of the virtual world seen by the camera is restricted by the front and back clipping planes.

* + 1. Clipping

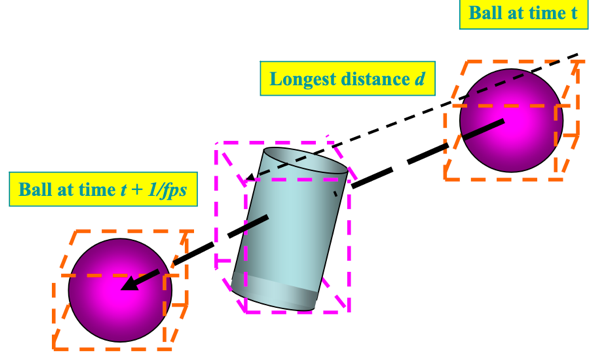
Why clipped? The view frustum is transformed to a unit cube through perspective transformation.

* + 1. Screen Mapping

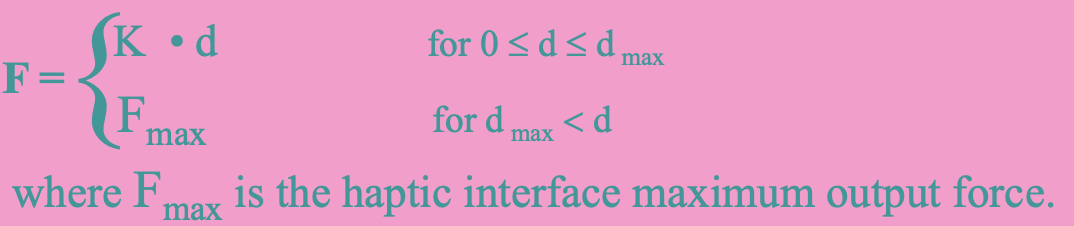
1. Physical modeling



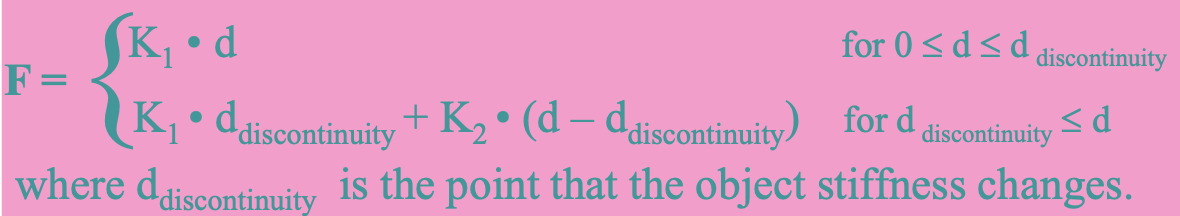
* 1. Collision Detection
     1. Use **bounding box**collision detection for fast response.
        1. Two types of bounding boxes: fixed size or variable size.
        2. Fixed size computationally faster, but less precise.
     2. Undetected Collision: Given the longest distance ***d*** between a moving object and an occluder, object moves more than ***d*** towards the occlude in a frame time.



* + 1. Two-stage Collision Detection: **approximate** (bounding box) **collision detection** stage -> **exact collision detection** stage.
    2. Collision Response-Surface Cutting:
       1. happens when the contact force exceeds a given threshold.
       2. When cutting, a vertex is duplicated. Then, the twin vertices separate based on laws and the cut enlarges.
  1. Force Calculation (Force)
     1. Force calculation for homogeneous elastic objects:



* + 1. Force calculation for elastic objects with harder interior:



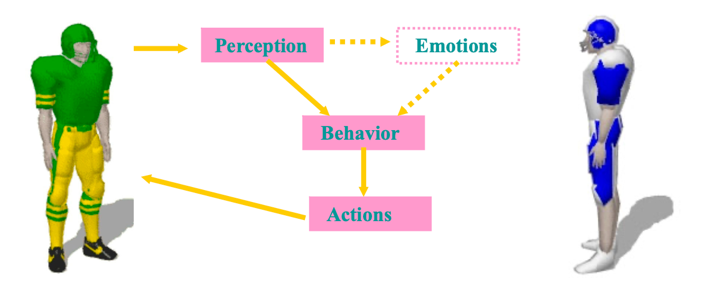
* 1. Force Smoothing-The Haptic Meshes 触觉网格
     1. A single HIP (haptic interaction point 触觉交互点) is not sufficient to capture the geometry of fingertip-object contact.
     2. The curvature of the fingertip and the object deformation need to be realistically modeled.
  2. Force Mapping (Force)
  3. Haptic Texturing (Tactile)

1. Object behavior modeling (intelligent agents)

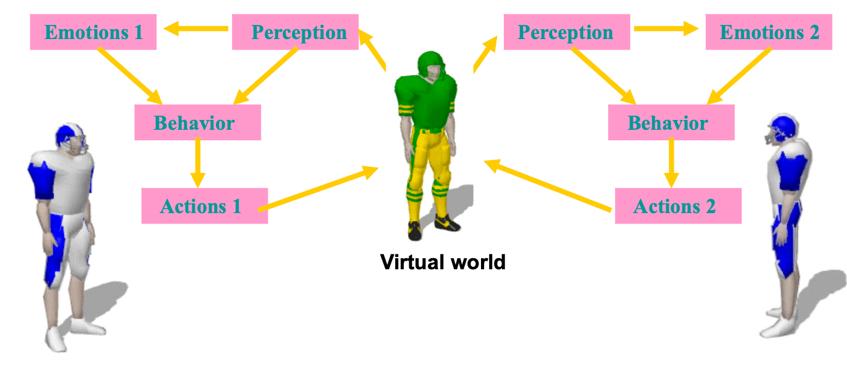
The degree of autonomy of a virtual environment (VE) depends on the autonomy of its components: interactive objects, agents, and crowds.

three levels of autonomy (LoAs) for each of the components: **guided < programmed** < **autonomous**.

* 1. Interactive objects:Have behavior independent of user’s input, e.g., clock.
  2. Agent Behavior:
     1. is composed of **perception, emotions, behavior,** and **actions.**
     2. Perception allows the agent to sense his/her surroundings.
     3. Reflex Behavior (Involuntary Motions 非自主运动): A direct link between perception, actions and behavior rules, without emotions.



* + 1. Emotional Behavior (Voluntary Motions):  Two different agents can have different emotions to the same perception, resulting in different actions.



* 1. Crowd Behavior
     1. group actions
     2. guided LoA when defined by user, autonomous LoA by rules.

1. Model management-Several techniques to maintain interactivity and constant frame rates when rendering complex models
   1. **Level of detail**(for modeling individual objects)
      1. is related to the number of polygons
      2. **Static LoD Management**
         1. use a simplified version when the object is far from the camera
         2. several approaches:
            1. **Discrete Geometry LoD:** Uses several discrete models, Suitable model is selected based on the distance;problems at *r = r0, r = r1, r = r2* circles, leading to “popping”.
            2. **Model Blending LoD:** two models of adjacent LoDs are rendered.
            3. **Geometric Morphing LoD:** only one complex model, Simplified models from the full resolution model through **edge collapses**; High resolution models from the base model through **vertex splits**.
      3. **Adaptive LoD:** 
         1. adjust the resolution depends on the viewer’s location and orientation.
         2. **Time-Critical Rendering:** try to finish rendering an image of as high quality as possible, within a given deadline.
         3. **Adaptive LoD Management-**time-critical rendering technique: selects LoDs of visible objects based on deadline.
   2. **Cell segmentation**(for modeling an environment).
      1. maintain **interactivity** and **constant frame rates**when rendering complex models.
      2. Used in architectural walkthroughs.
      3. **Database Management:** 
         1. minimize page faults and improve *fps* uniformity during walkthroughs.
         2. predict rotation and translation of the camera, prefetch the appropriate objects.
   3. **Lighting and bump mapping**at rendering stage
   4. **Portals** (for separating an environment into regions)

**7 Sense of Presence**

1. **Definition of** the sense of presence
   1. the feeling of being there in a virtual environment.
   2. senses include: visual, audio, tactile, motion and smell.
2. **Existence of Sense of Presence-**limitations with current VR technologies:
   1. Characters are not realistic enough.
   2. Incorrect sensory information
   3. Sensors poorly match human capabilities. 传感器与人类感官能力不匹配
   4. User must put on wires and gadgets.
3. Three physical variables to determine degree of the sense of presence:
   1. number of sensory channels
   2. how much can the sensory Information alter the sensor inputs
   3. ability to modify the environment
4. **How to Measure Presence?**
   1. Subjective measures

Common Questionnaires: be used to compare similar environments.

* 1. Psychophysical measures

relate the **physical magnitude** 物理量of a stimulus with the **observer’s subjective rating**

* 1. Objective measures
     1. Physiological measures 生理测量
     2. Performance measures: compare the task performance in real situation and in the virtual environment.
  2. **An Example**
     1. Subjective: Answer questionnaires afterwards
     2. Psychophysical: HMD screen resolution; HMD field of view; Quality of audio feedback.
     3. Physiological: Heart rate; Blood pressure.

1. **What Increases Presence?**
   1. High visual quality
   2. Large field of view
   3. Low latency
   4. Head tracking
   5. Multiple senses: Audio; Haptics.
   6. Interactivity
   7. Able to see the user’s own body parts
2. **What Decreases Presence?**
   1. Disjoint Senses
   2. High latency
   3. Poor interactivity
   4. Cables
   5. Low quality audio
   6. Can’t see the user’s own body parts

**8 VR Applications-Types of VR Applications**

1. Medical applications
   1. Adv:
      1. better train medical students and doctors
      2. Allow errors to be made, rare cases to be modeled
      3. Reduce medical costs
   2. **Training in EMR to Bio-terrorism:** 
      1. allows virtual characters to “die”; much cheaper to organize.
      2. Training scenarios – airport explosion.
   3. **Virtual Anatomy** 虚拟解剖
   4. **Anatomic Visualizer** 解剖可视化器
   5. **The Surgical Process**
      1. **Diagnosis:** Rutgers DRE Trainer, Virtual Colonoscopy
      2. **Anesthesia** 麻醉: HT Medical Systems
      3. **Surgery:** MIST VR.
      4. **Rehabilitation** 康复:Ankle 脚踝, Stroke Patient Trials at Rutgers, VR Psychological Rehabilitation, VR Cognitive Rehabilitation.

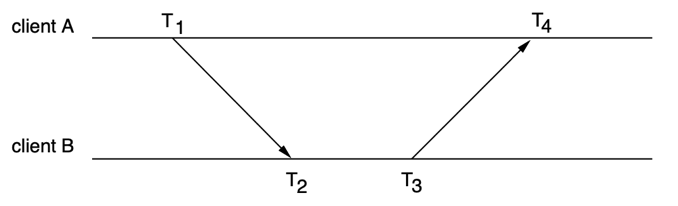
VR therapy advs: privacy, safer, cheaper.

1. Entertainment / arts / education
   1. **VR Education Applications:** Virtual Physics Laboratory.
   2. **VR Art Applications:** preservation of cultural heritage, new way to explore art.
   3. **VR Entertainment Applications:** Virtual Swimming cycling driving,Disney Quest.
2. **VR Military Applications-**Defense / aerospace
   1. Adv:
      1. remote training
      2. reduce training costs
      3. simulate missions ahead of execution
      4. visualize enemy weapon
      5. train soldiers to use new weapons
   2. **Army Use of VR:** 
      1. **Small Arms Trainer:** Individual and team training; analyze shooters’ performance.
      2. **FATS:** convert live weapons into simulated weapons.
3. Manufacturing
   1. **Virtual Prototyping**
   2. **Assembly Verification: Car Body Tolerances**
   3. **Ergonomic Analysis** 人体工程学分析
4. Robotics
   1. Teleoperation
   2. Try to alleviate problems:
      1. poor visibility
      2. large time delays:

**9 Distributed Virtual Environments**

1. **Distributed Virtual Environments**
   1. a shared VE allows remote users to interact with virtual objects.
   2. Collaborative Virtual Environment: supports multiple users to perform a task together in the virtual world.
   3. Applications: virtual museum, online games.
2. **Multi-Server DVEs**
   1. Emphasize on interactivity.
   2. In client-server system, server handle requests from clients, update object positions, distribute update messages.
   3. Many multi-server methods
      1. **User Partitioning:**
         1. partition the users to different servers
         2. simple but users of different servers cannot interact with each other.
      2. **region partitioning**
         1. divides the VE into multiple fixed regions. Each region is served by one server.
         2. Less popular region suddenly becomes popular may cause server get overloaded.
      3. **dynamic partitioning:** divides the VE into multiple regions dynamically during run-time. If the total amount of load of a region is too high, it is repartitioned into a smaller region, and vice versa.
3. **Motion Synchronization**
   1. Problem of DVE: network latency.
   2. Two types of events:
      1. discrete events: occurs after a large time gap, network latency does not affect.
      2. continuous events: occur with very small time gaps, network latency affects.
   3. **Synchronization**
      1. maintain the consistency of objects shared among different machines.
      2. Traditional approach: causal ordering 因果顺序- Events should be executed according to the order that they occur.
      3. **Object locking.**
      4. **Dead Reckoning** 推测导航:
         1. Both of machines run the same motion predictor for each moving object; The predictor predicts the position of the moving object based on the last update message; User specifies an error threshold value; error between the actual position and the predicted position > threshold, sends an update message to the receiver telling the receiver the latest object position with other information; error < threshold, no need to update.
         2. Prevent massive update messages and save band width.
4. **Motion Prediction**
   1. **Polynomial Predictors** 多项式预测器
      1. two popular polynomial predictors:
         1. FOP (First Order Polynomial) 一阶
         2. SOP (Second Order Polynomial) 二阶
      2. used with dead reckoning
      3. adv: simple and small computation overhead.
      4. Limitation: not accurate because of changing network latency.
   2. **Estimating Network Latency**

Simple method: send out a message to the other client and then read the return message. Limitation: requires a round-trip message to determine the latest network latency.



* 1. **Problems with Prediction**
     1. When objects started to move, prediction will be wrong.
     2. Objects continuously change its v and a.

**Tutorial 5 Marker-Based AR**

1. Comparison between CG, VR, and AR
   1. CG uses virtual camera to project virtual 3D world to a display.
   2. VR can be regarded as a stereo counterpart of CG.
   3. AR uses a real camera to project both real 3D world and virtual 3D content.
2. Marker Based AR
   1. General AR pipeline
      1. Initialize a 3D coordinate system
      2. Anchor digital content to the position
      3. Project digital content to the captured image.
   2. Challenges
      1. In the first step, how can we associate a coordinate system with the physical world?
         1. Markers associates a coordinate system with the physical world and tracking camera pose.
      2. In the third step, we need both extrinsic parameter and intrinsic parameter, how can we get them?
         1. The camera intrinsic parameter is derived via camera calibration.
   3. ArUco Lib and ArUco Marker
      1. Each detected marker in the image, you can get its ID and camera pose relative to it with this ArUco LIB.
      2. An ArUco marker: a wide black border + an inner binary matrix (determine its identifier (ID) and provide error correction)
   4. Camera Calibration
      1. Purpose: obtain the intrinsic parameter of the camera.
      2. Two calibration targets:
         1. **ChAruco Board**
         2. **ArUco Board**

**Tutorial 4**

裸眼立体显示器（Autostereoscopic Displays）

* 视差屏障（Parallax Barrier）
* 柱状透镜（Lenticular Lens）

 如果观看者不在最佳位置，可能出现的问题：

右眼可能看到本应给左眼看的图像，

这种现象称为逆向图像（inverse image）。

一只眼睛可能同时看到两个图像，

结果导致图像模糊（blurred image）。