SNS COLLEGE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)
Approved by AICTE & Affiliated to Anna University
Accredited by NBA & Accrediated by NAAC with 'A++' Grade,
Recognized by UGC saravanampatti (post), Coimbatore-641035.

Department of Biomedical Engineering

Course Name: 19BMT401 – Virtual Reality in Medicine

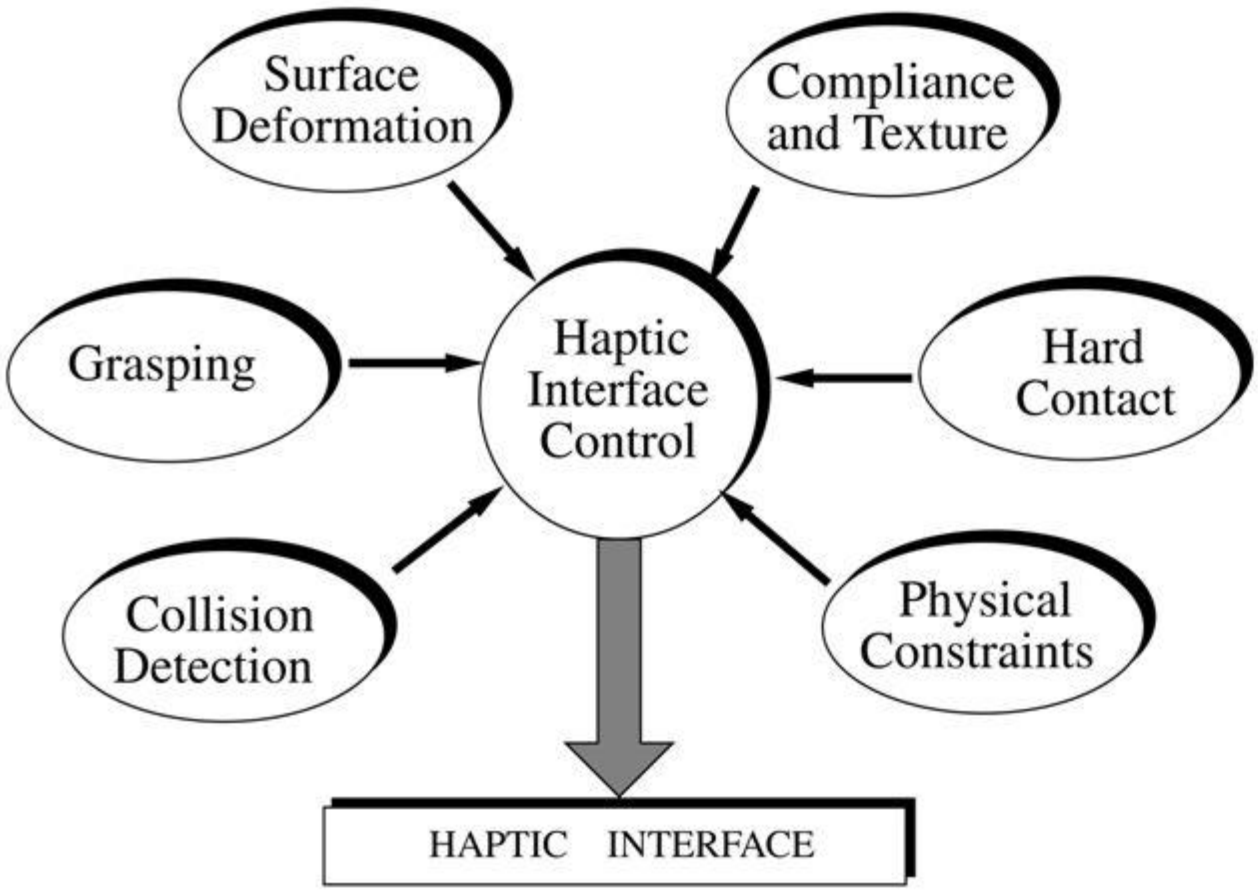
IV Year : VII Semester

Unit II –MODELING

Topic : MODEL MANAGEMENT

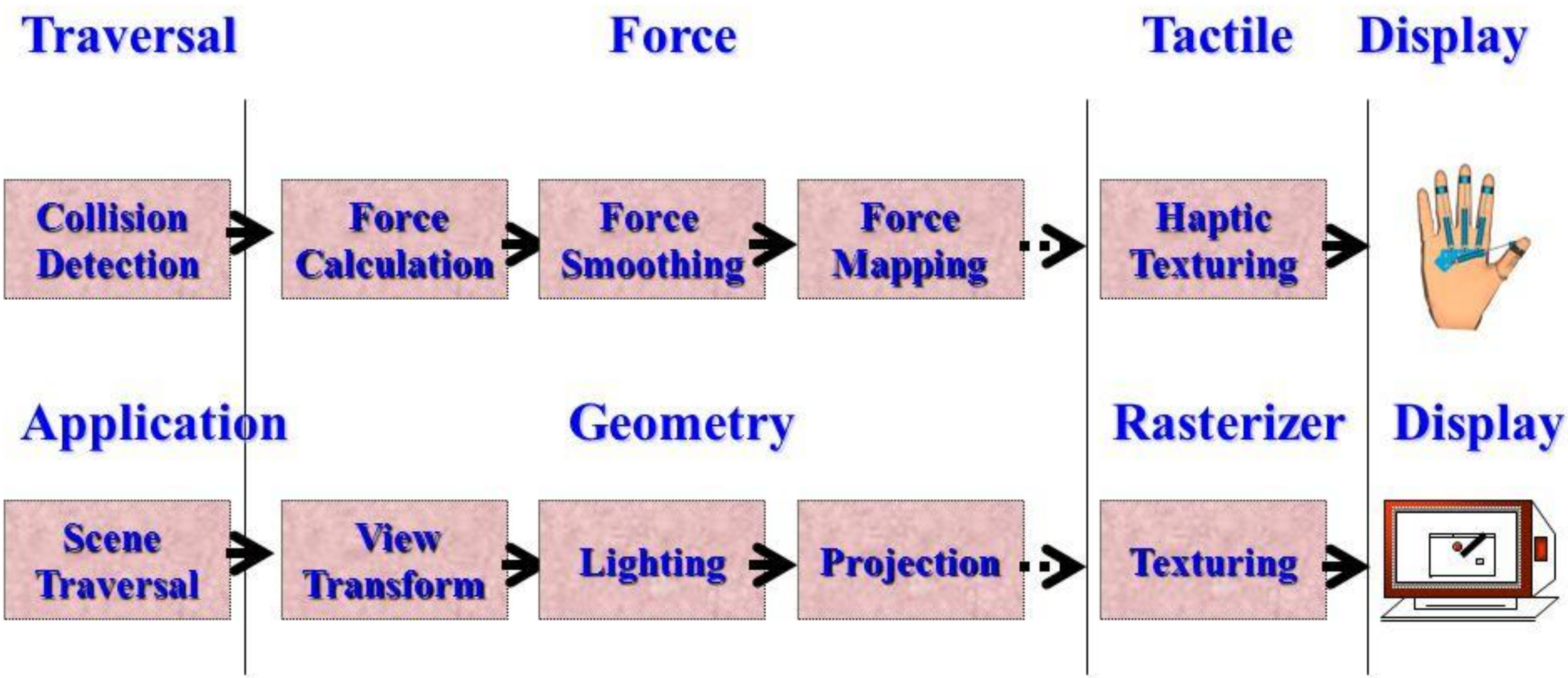
Virtual Reality Modeling

The VR physical modeling:



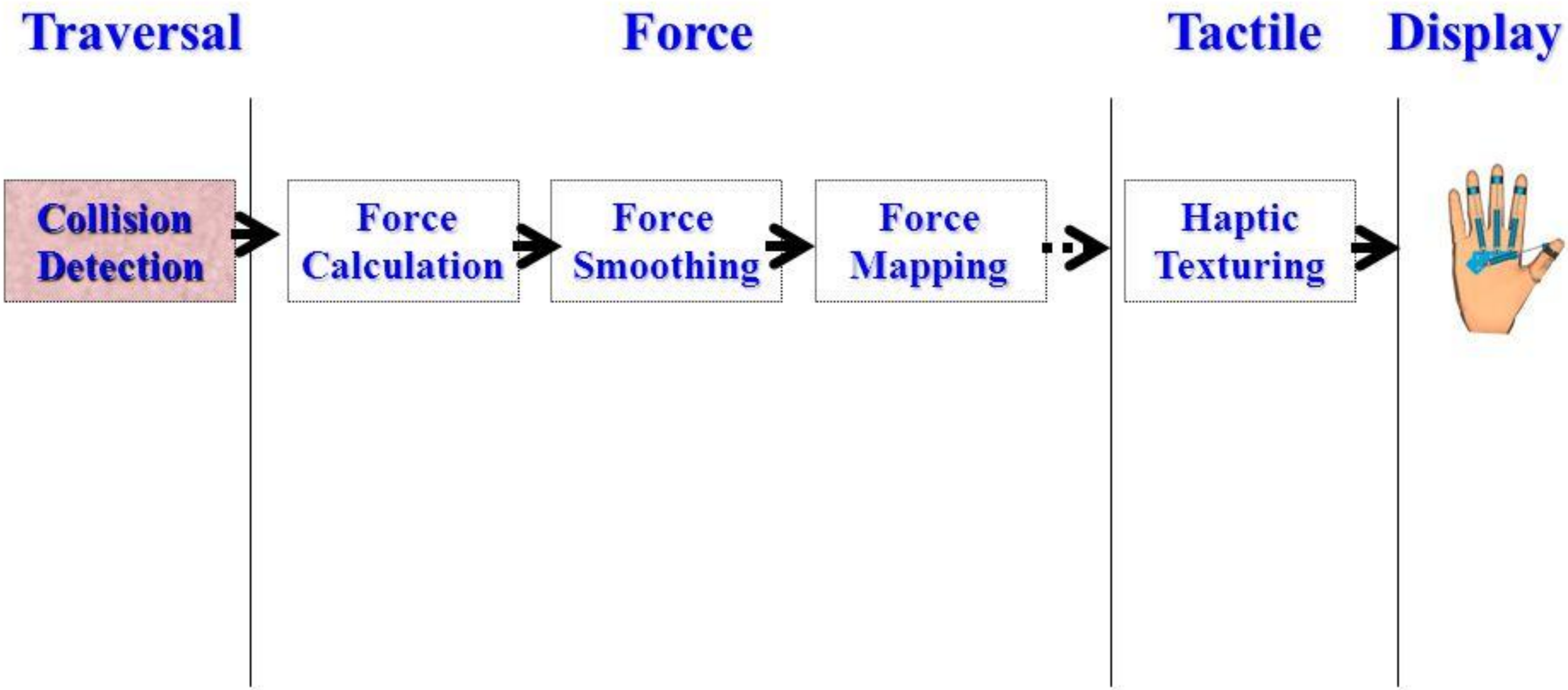
from (Burdea 1996)

The Haptics Rendering Pipeline (revisited)



adapted from (Popescu, 2001)

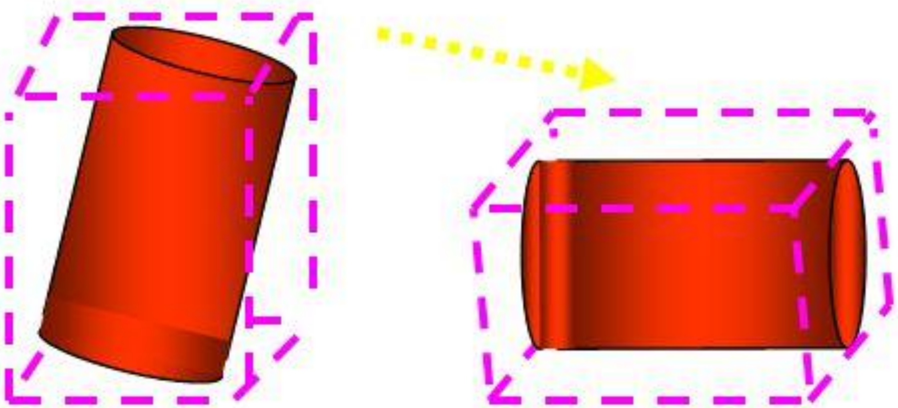
The Haptics Rendering Pipeline



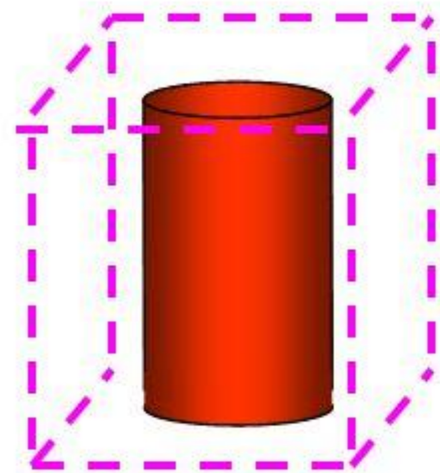
Collision detection:

- ✓Uses ***bounding box*** collision detection for fast response;
- ✓Two types of bounding boxes, with fixed size or variable size (depending on enclosed object orientation).
- ✓Fixed size is computationally faster, but less precise

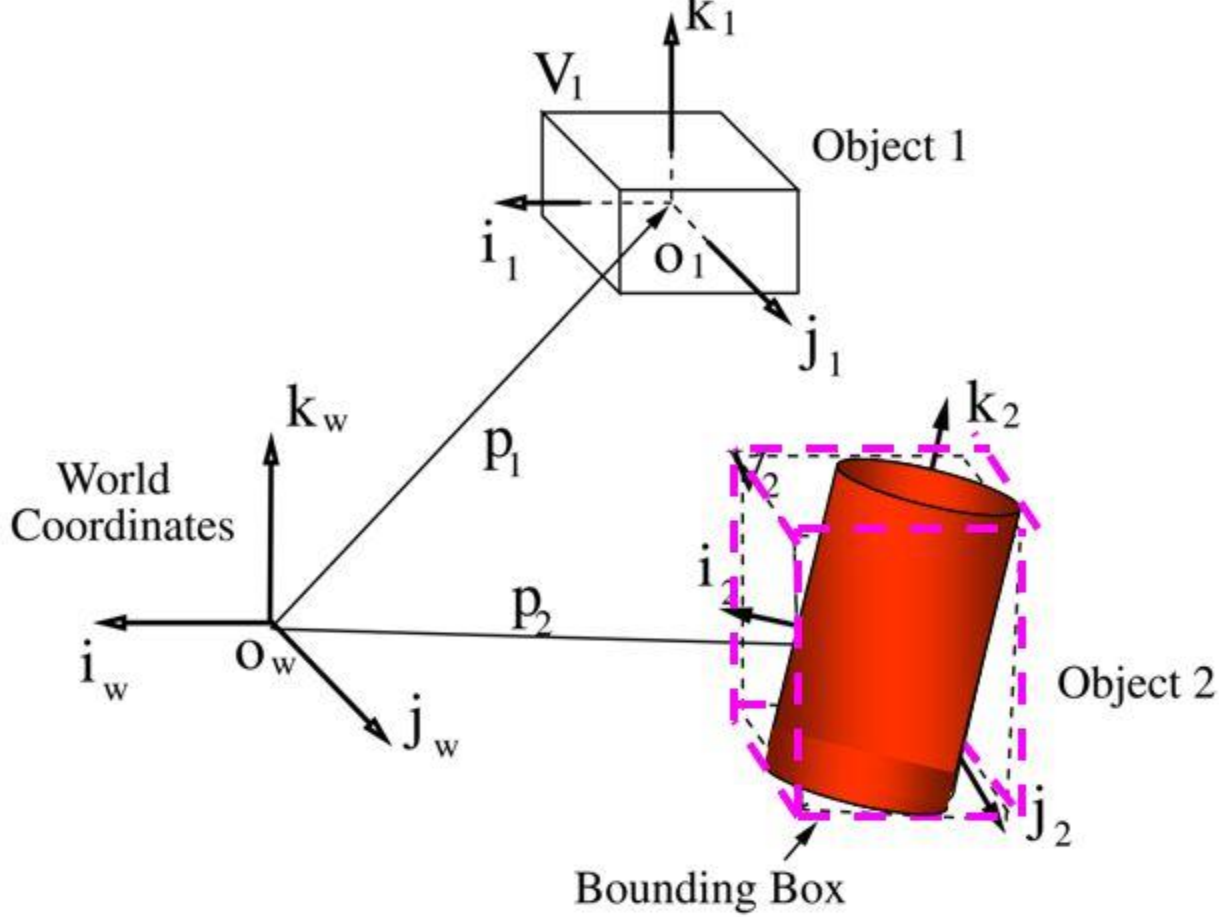
Variable size Bounding Box



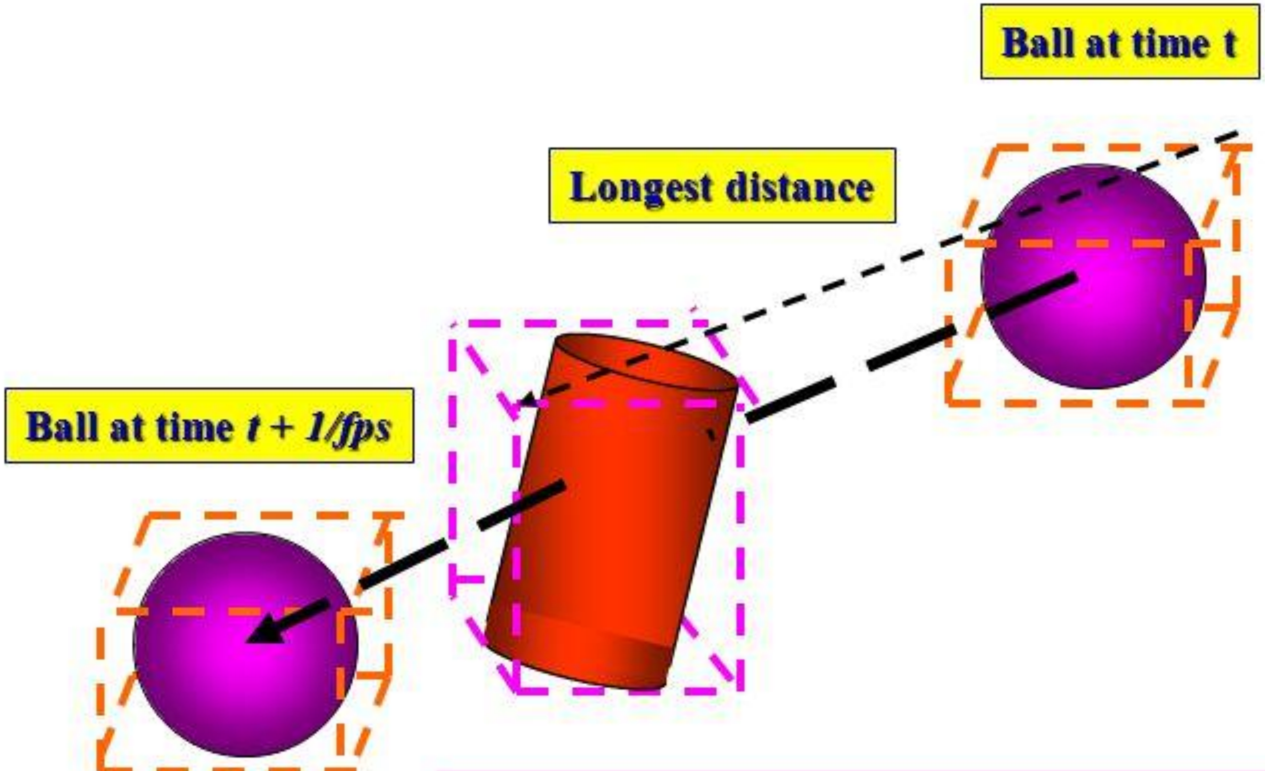
Fixed size Bounding Box



Collision Detection



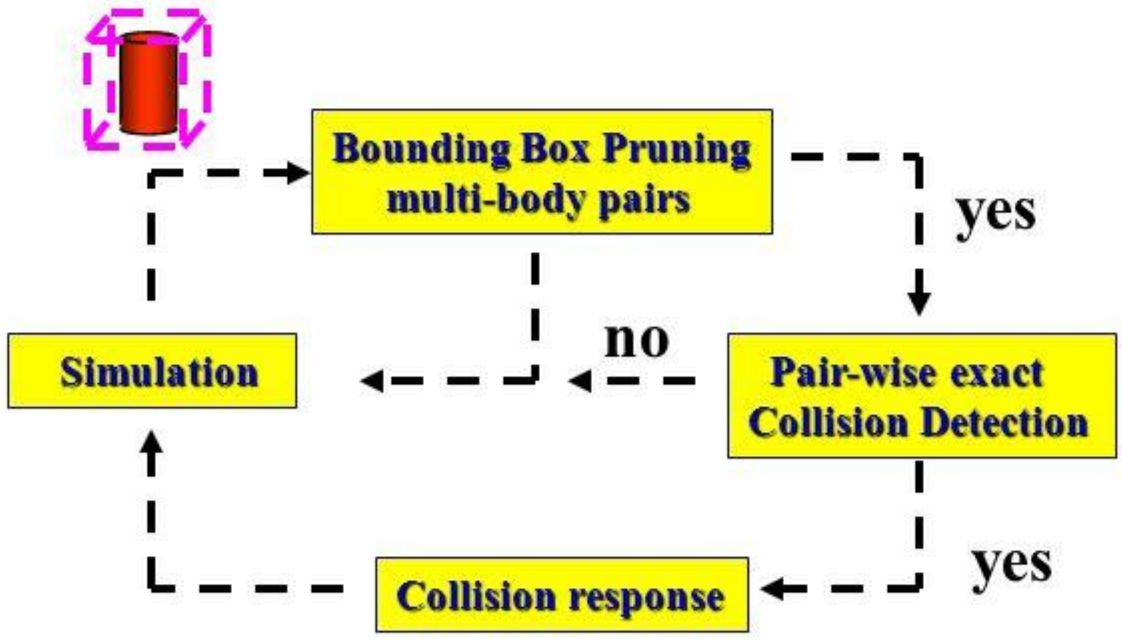
Undetected collision



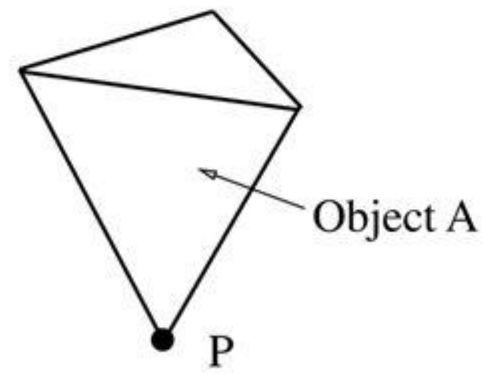
If longest distance $d < v \cdot 1/fps$ then collision is undetected (v is the ball velocity along its trajectory)

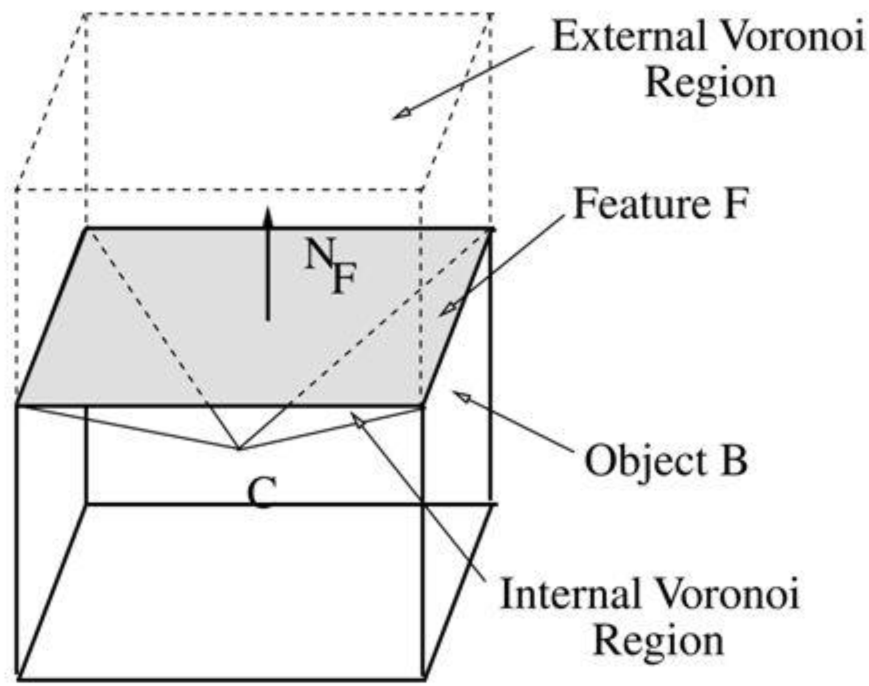
Two-stage collision detection:

✓For more precise detection, we use a two-stage collision detection: an ***approximate*** (bounding box) stage, followed by a slower ***exact collision detection*** stage.



Exact collision detection

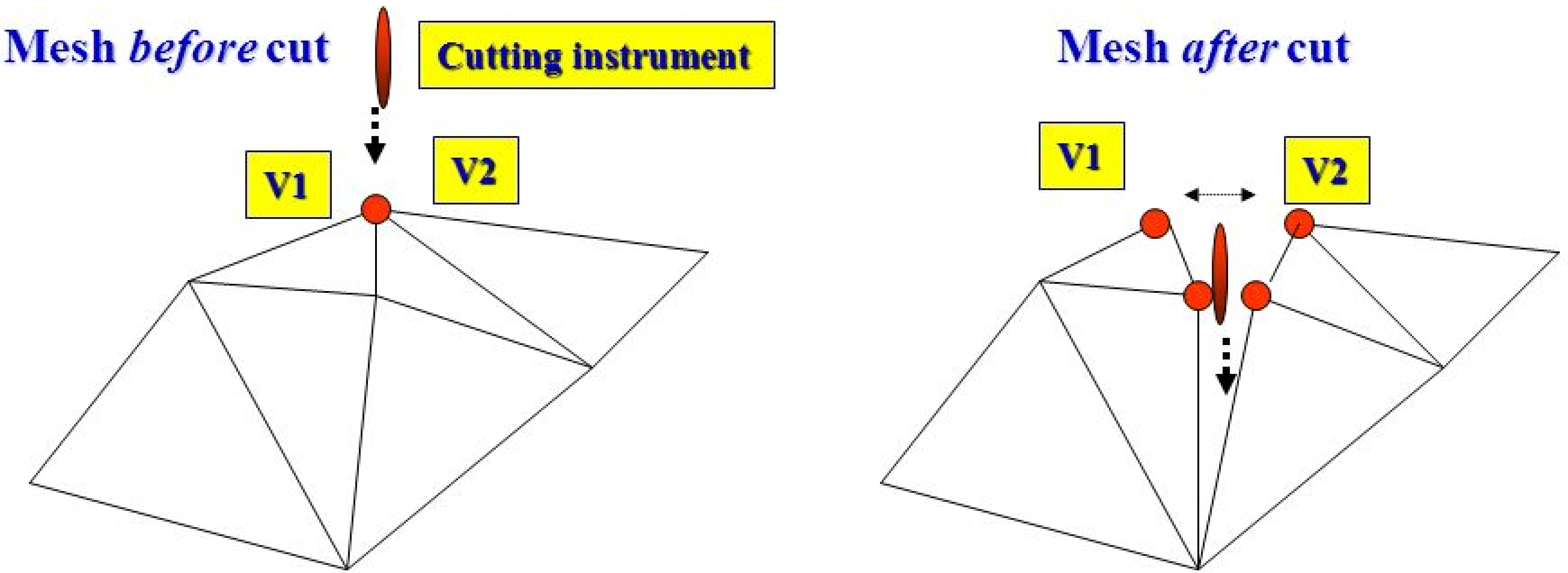




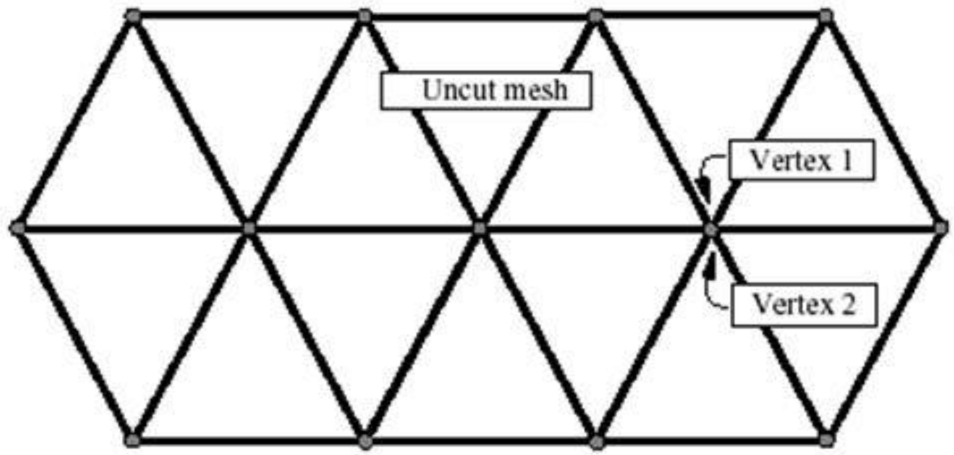
Surface cutting:

✓An extreme case of surface "deformation" is surface cutting. This happens when the contact force exceed a given threshold;

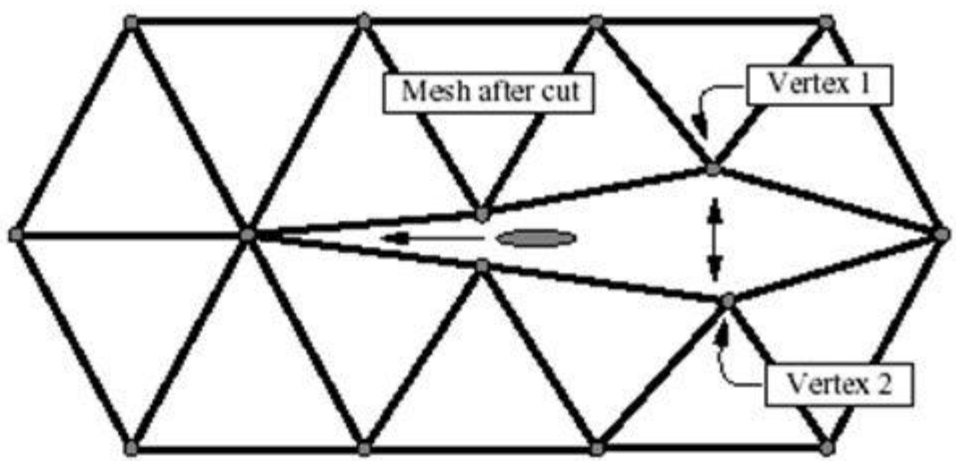
✓When cutting, one vertex gets a ***co-located twin***. Subsequently the twin vertices separate based on spring/damper laws and the cut enlarges.



Collision response – surface deformation

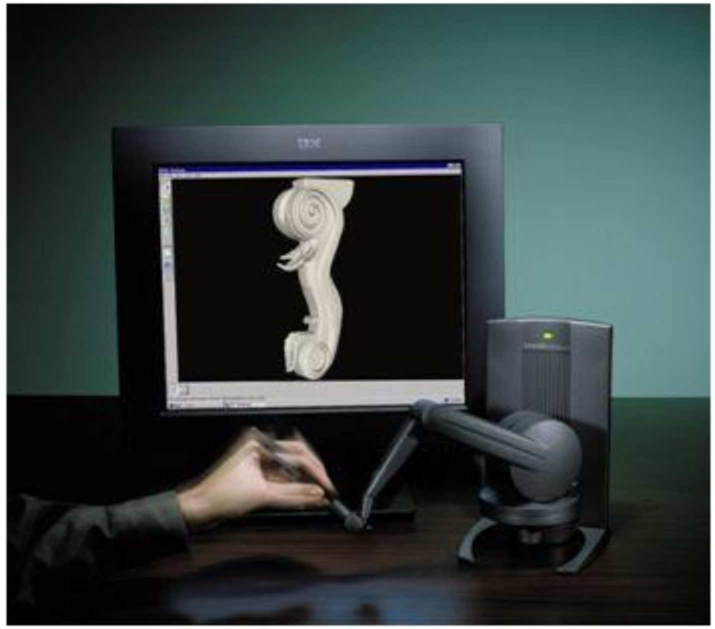


a)

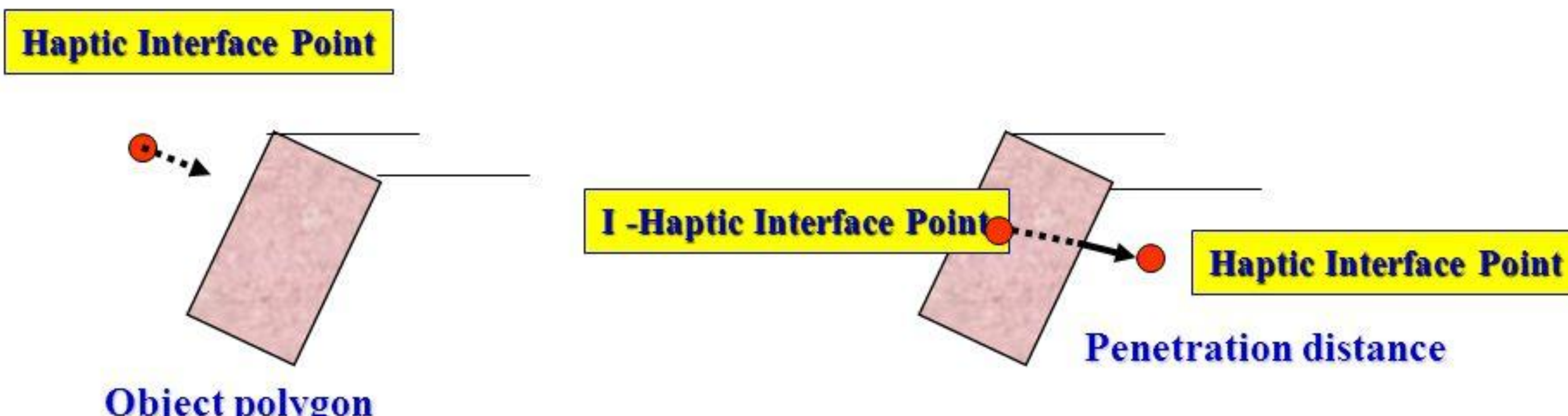


The Haptics Rendering Pipeline

Traversal	Force			Tactile	Display
Collision Detection →	**Force Calculation** →	Force Smoothing →	Force Mapping ⇢	Haptic Texturing →	



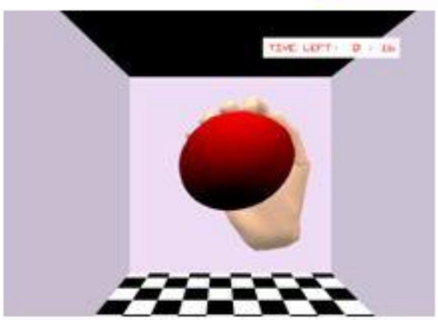
Haptic interface

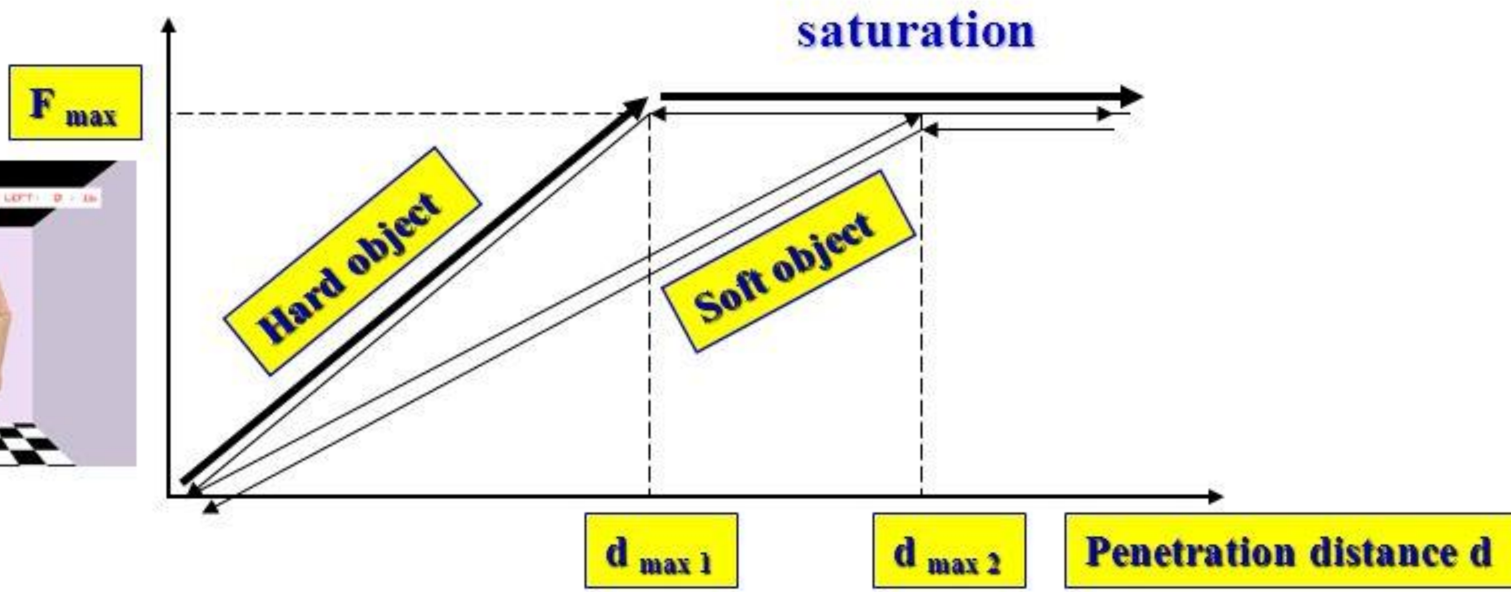


Force output for homogeneous elastic objects

$$F = \begin{cases} K \cdot d, & \text{for } 0 \le d \le d_{max} \\ F_{max} & \text{for } d_{max} < d \end{cases}$$

where F_{max} is that haptic interface maximum output force

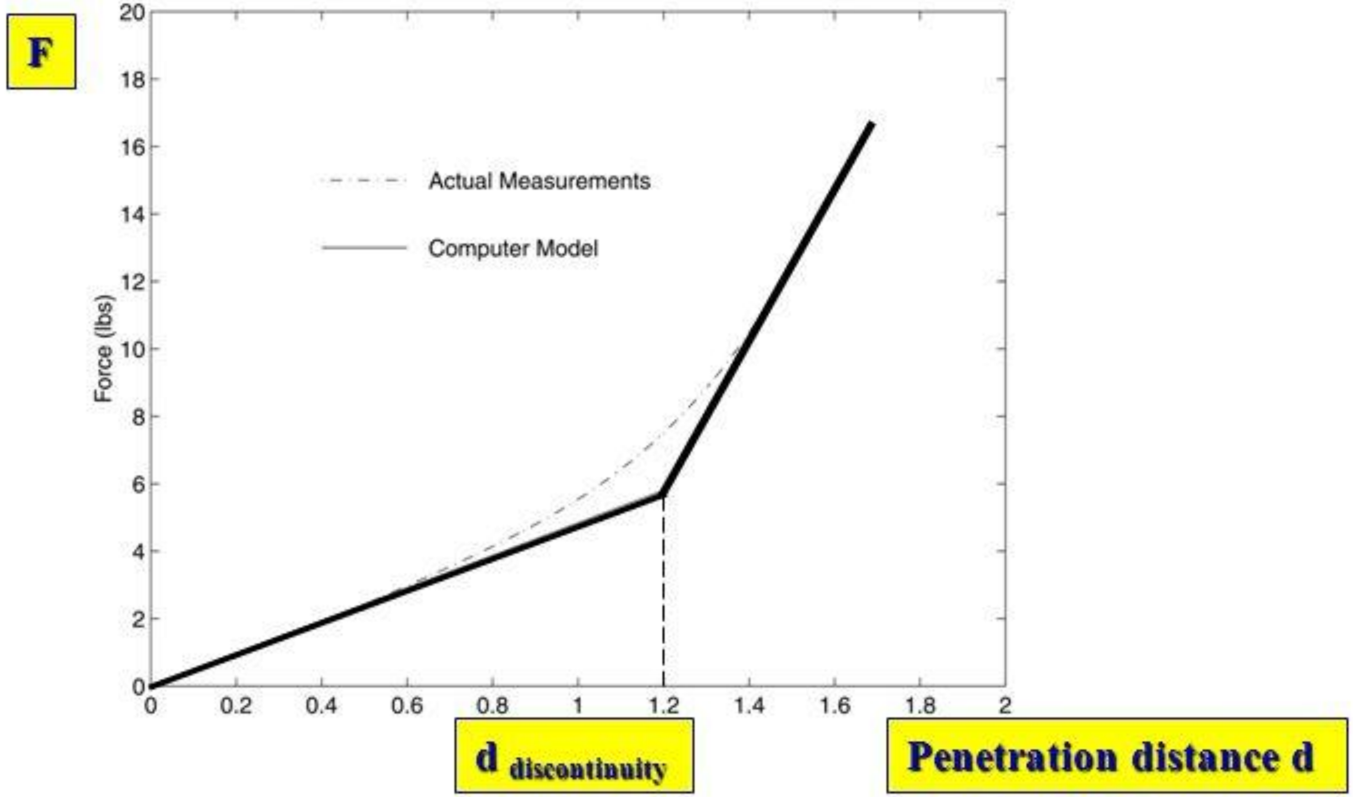


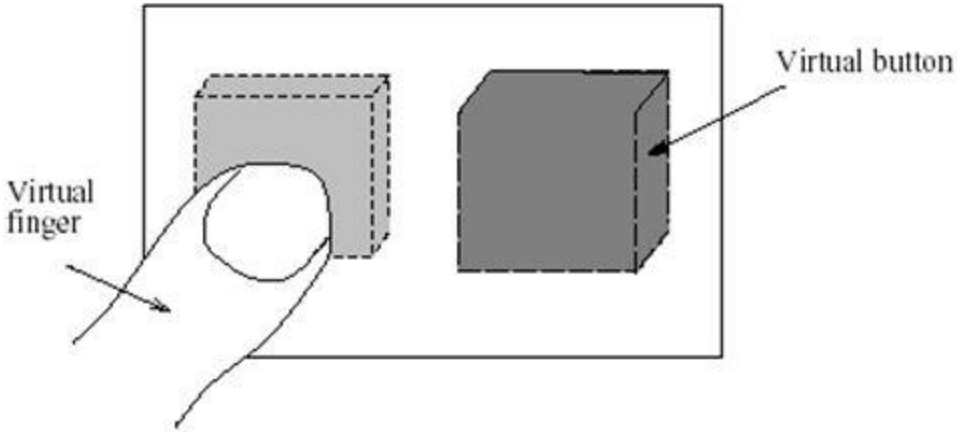


Force Calculation –
Elastic objects with harder interior

$$F = \begin{cases} K_1 \cdot d, & \text{for } 0 \le d \le d_{discontinuity} \\ K_1 \cdot d_{discontinuity} + K_2 \cdot (d - d_{discontinuity}), & \text{for } d_{discontinuity} \le d \end{cases}$$

where $d_{discontinuity}$ is object stiffness change point

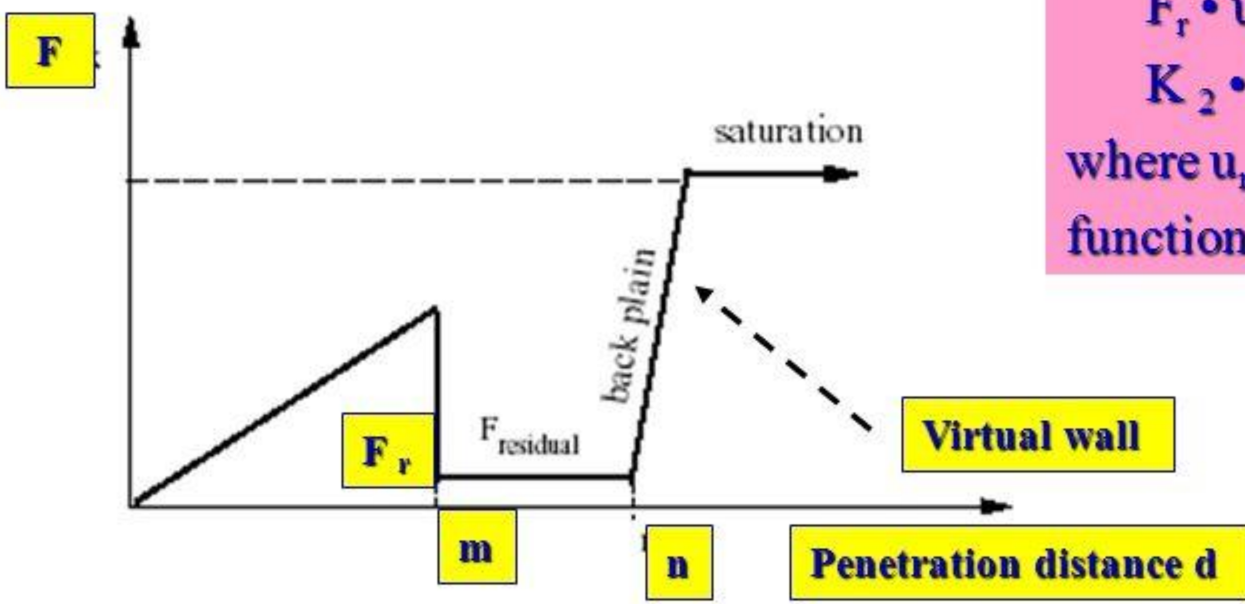




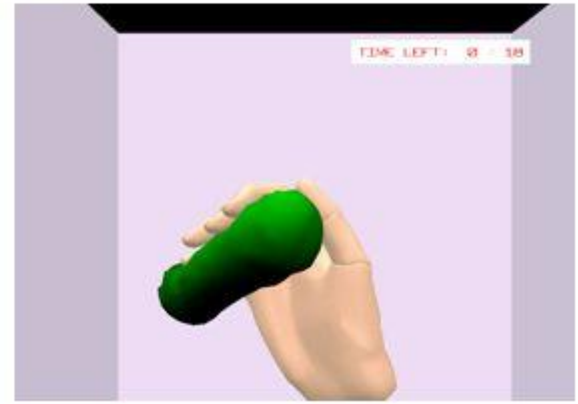
Force Calculation – Virtual pushbutton

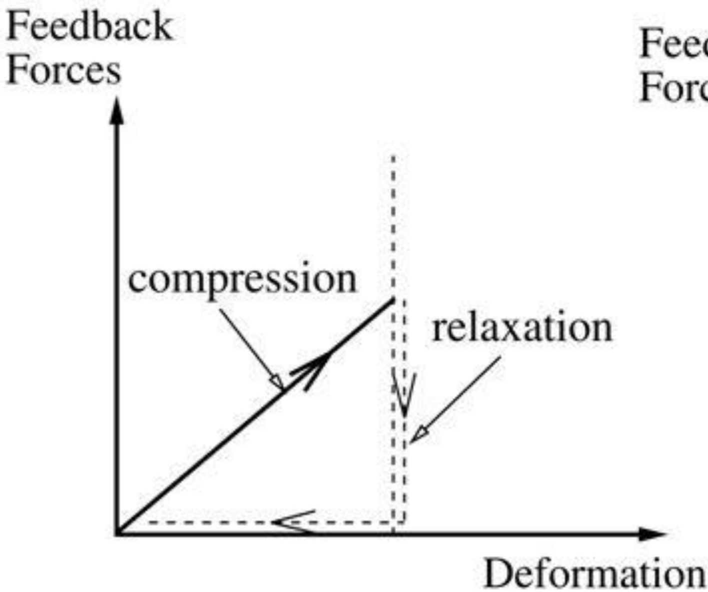
$$F = K_1 \bullet d\,(1 - u_m) + F_r \bullet u_m + K_2 \bullet (d - n)\,u_n$$

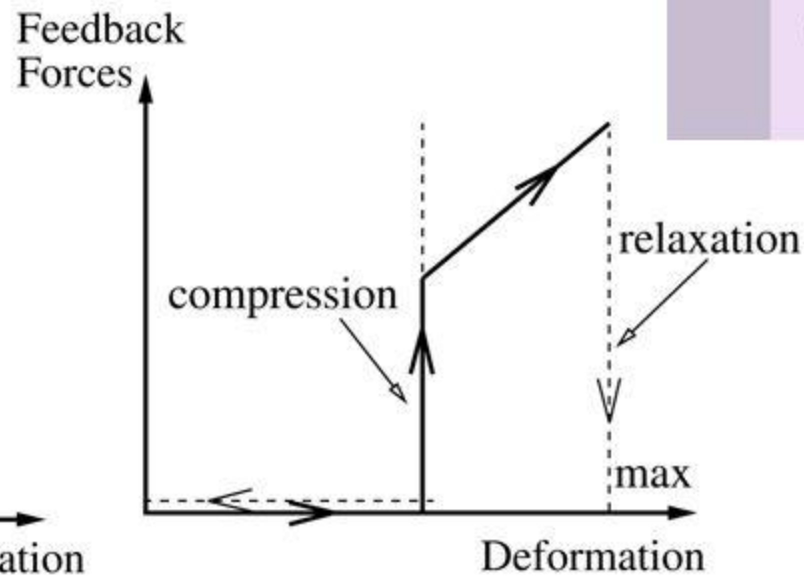
where u_m and u_n are unit step functions at m and n



Force Calculation – Plastic deformation







m

m n

$F_{initial} = K \cdot d$ for $0 \le d \le m$

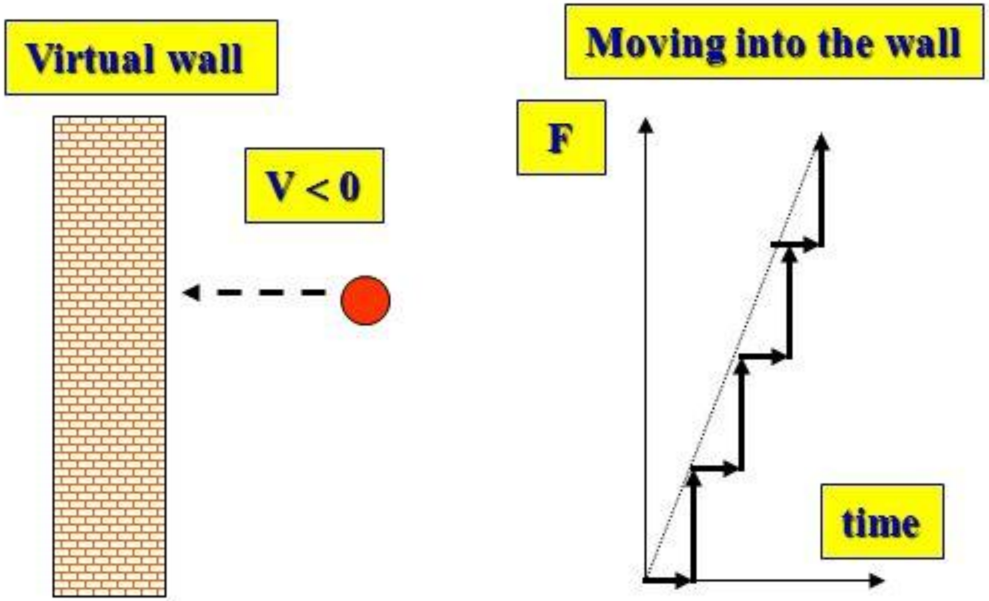
$F = 0$ during relaxation,

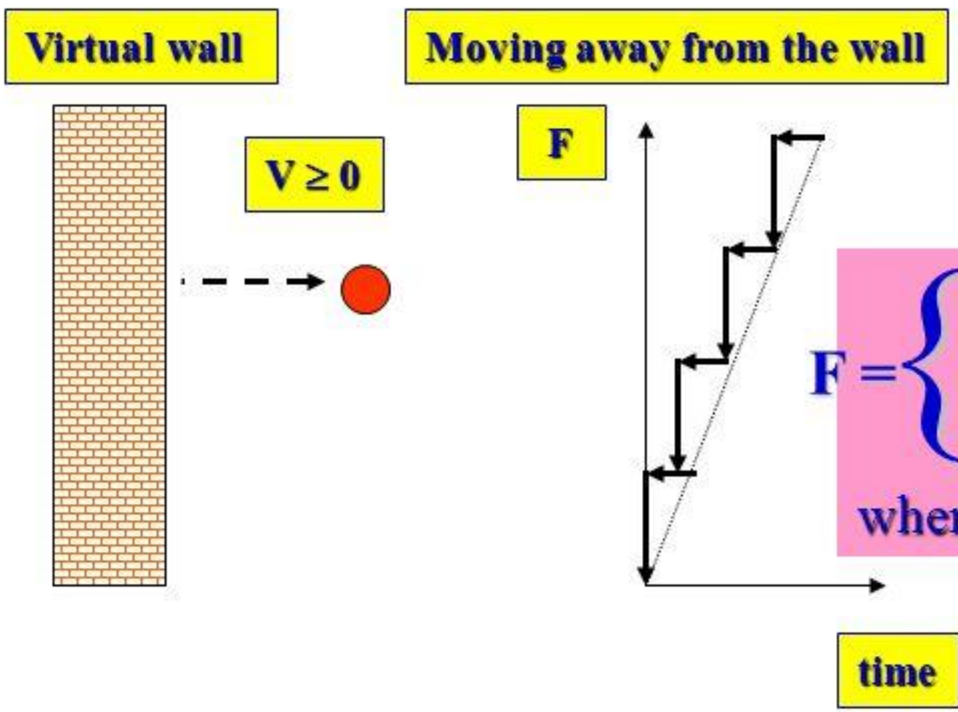
$F_{subsequent} = K_1 \cdot d \cdot u_m$ for $0 \le d \le n$

$F = 0$ during relaxation,

where u_m is unit step function at m

Force Calculation – Virtual wall



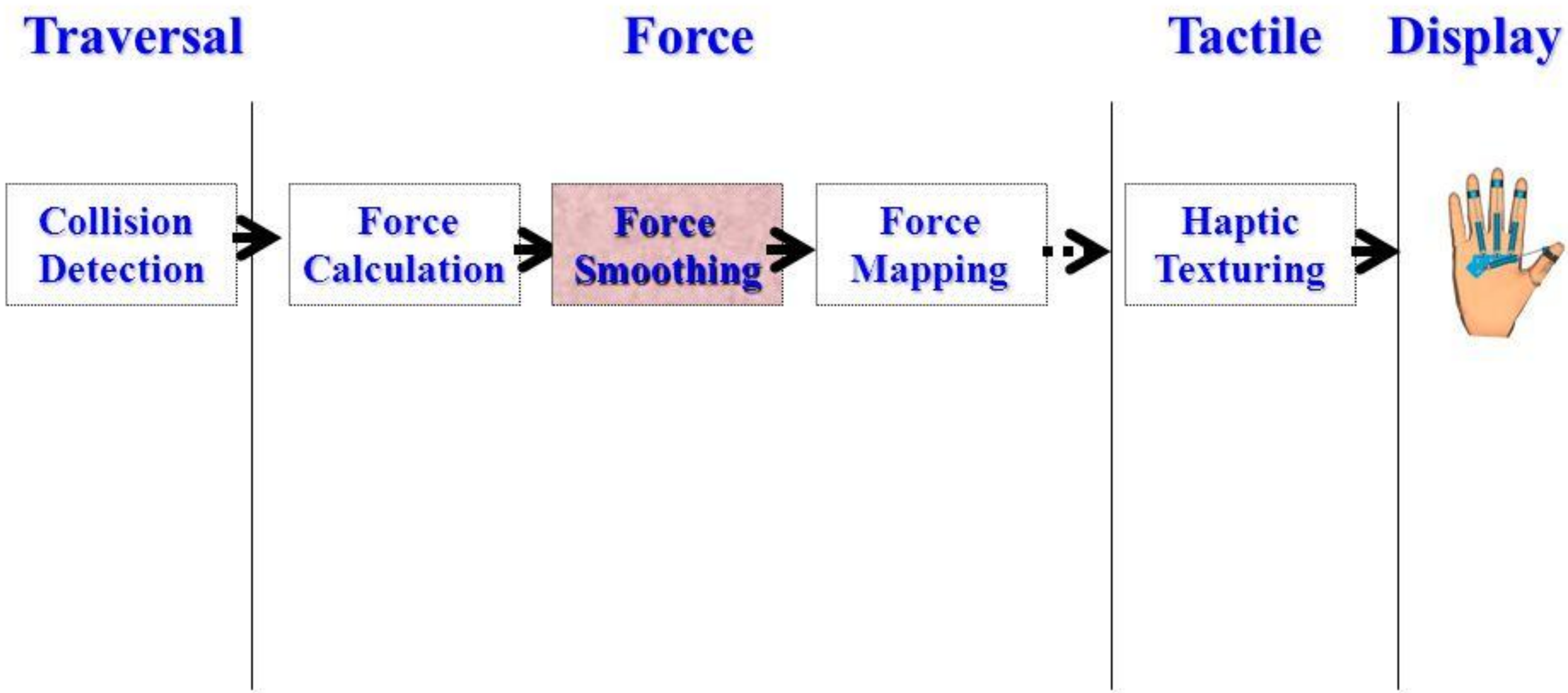


Generate energy due to sampling time- To avoid system instabilities we add a damping term

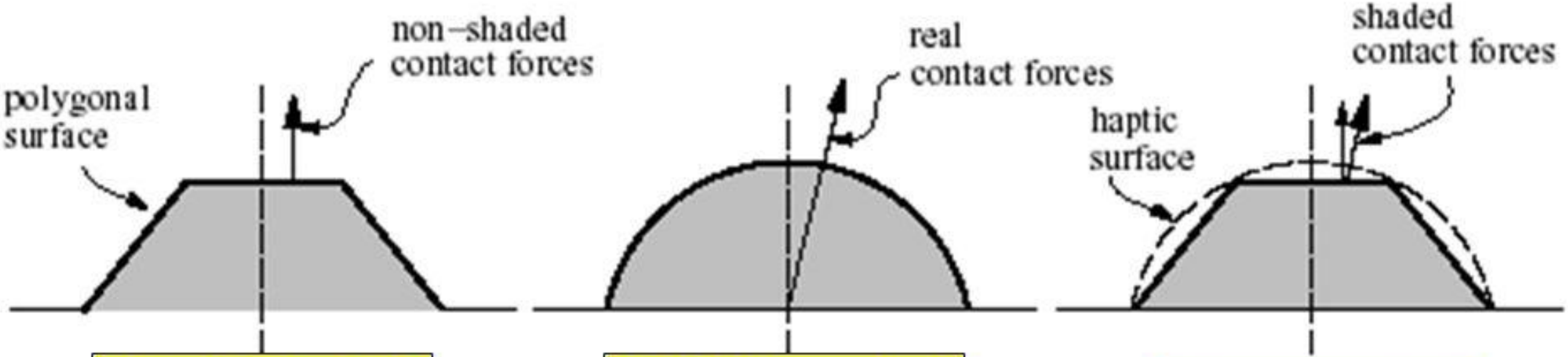
$$F = \begin{cases} K_{wall} \bullet \Delta x + B v, & \text{for } v < 0 \\ K_{wall} \bullet \Delta x, & \text{for } v \geq 0 \end{cases}$$

where B is a directional damper

The Haptics Rendering Pipeline



Force shading:



Non-shaded contact forces

Real cylinder contact forces

Contact forces after shading

$$F_{smoothed} = \begin{cases} K_{object} \cdot d \cdot \overline{N}, & \text{for } 0 \le d \le d_{max} \\ F_{max} \cdot \overline{N}, & \text{for } d_{max} < d \end{cases}$$

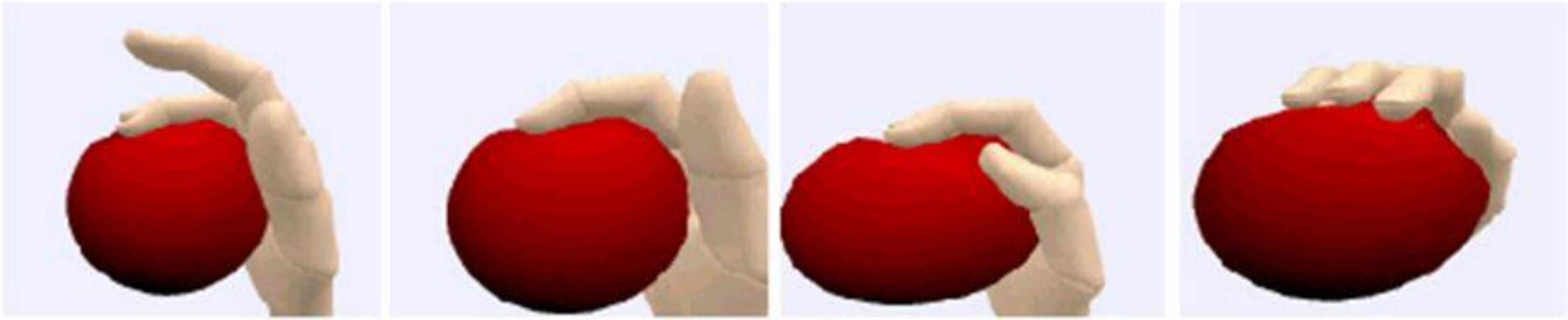
where $\overline{N}$ is the direction of the contact force based on vertex normal interpolation

The haptic mesh:

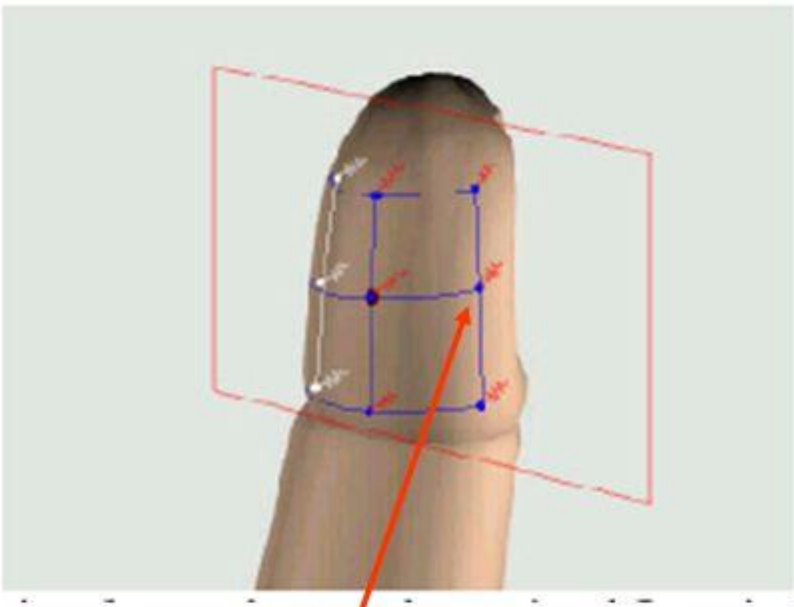
✓A single HIP is not sufficient to capture the geometry of fingertip-object contact;

✓The curvature of the fingertip, and the object deformation need to be realistically modeled.

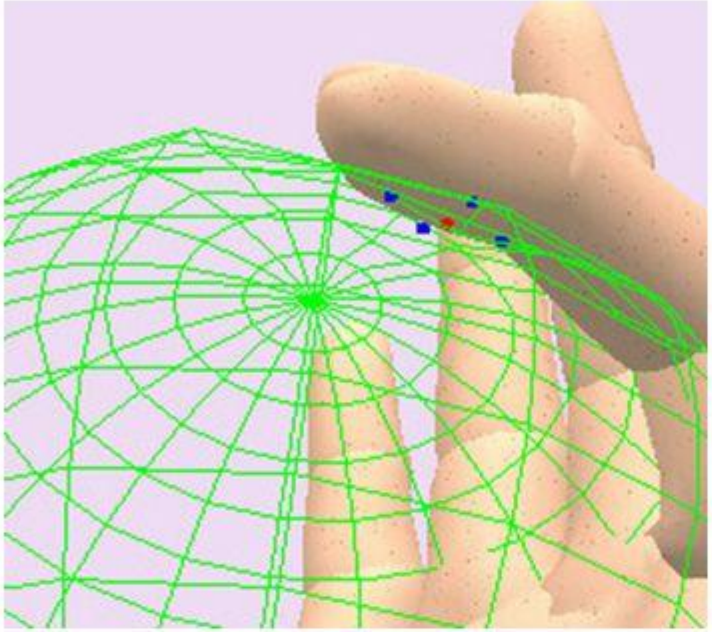
Screen sequence for squeezing an elastic virtual ball



Haptic mesh

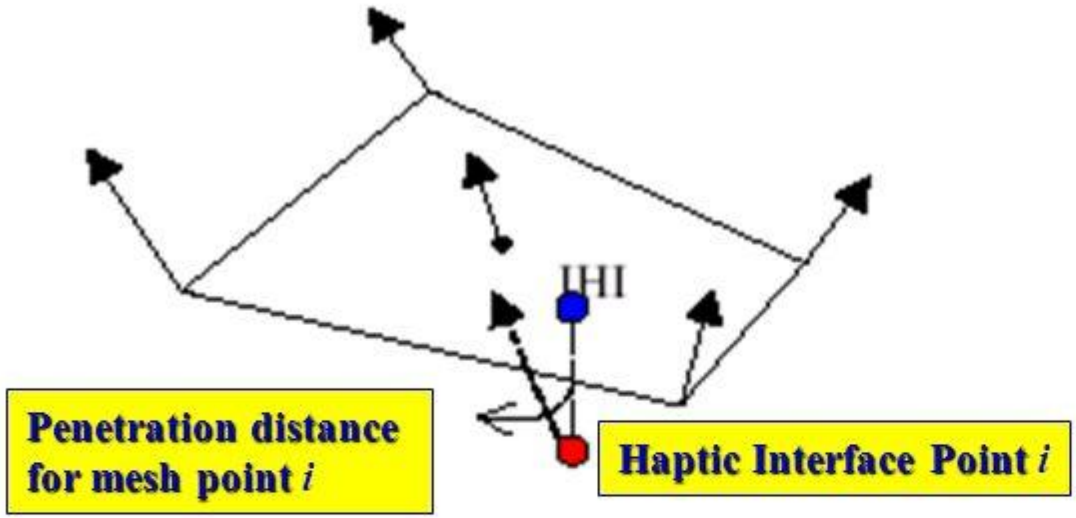


Mesh point *i*



Penetration distance for mesh point *i*

Haptic mesh force calculation



For each haptic interface point of the mesh:

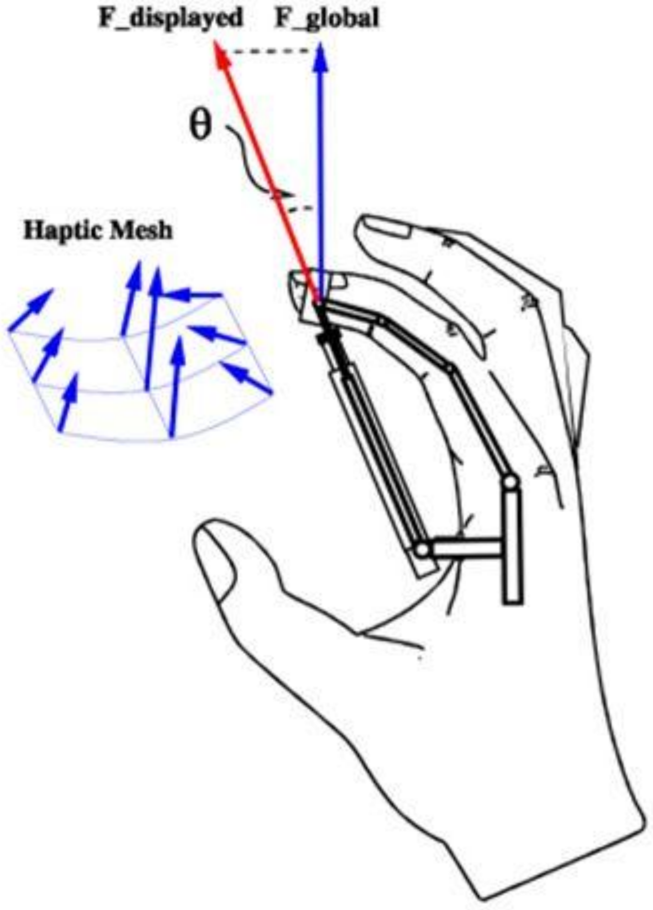
$$F_{\text{haptic-mesh } i} = K_{\text{object}} \cdot d_{\text{mesh } i} \cdot \overline{N}_{\text{surface}}$$

where $d_{\text{mesh } i}$ are the interpenetrating distances at the mesh points, $\overline{N}_{\text{surface}}$ is the weighted surface normal of the contact polygon

The Haptics Rendering Pipeline

Traversal	Force			Tactile	Display
Collision Detection →	Force Calculation →	Force Smoothing →	Force Mapping →	Haptic Texturing →	

Force mapping

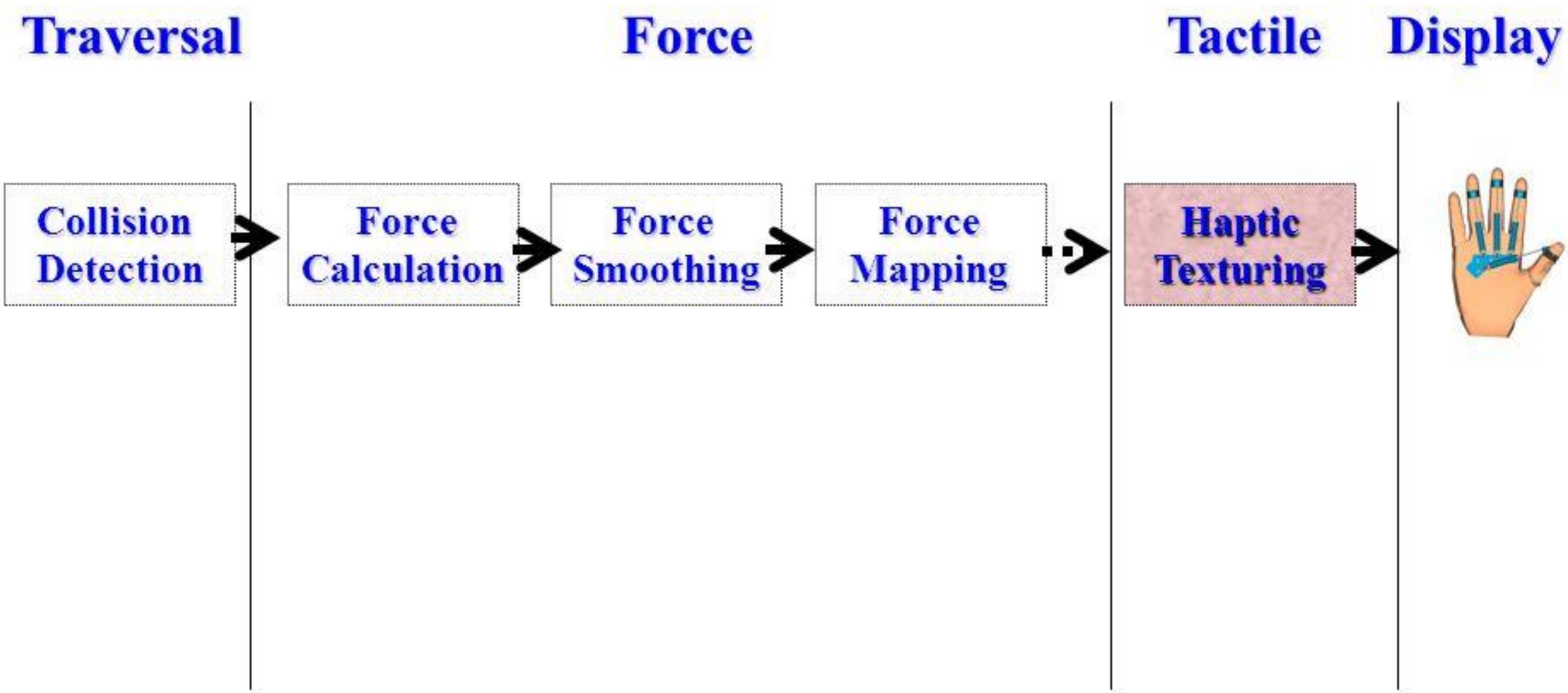


Force displayed by the Rutgers Master interface:

$F_{displayed} = (\Sigma F_{haptic\text{-}mesh}) \cdot \cos\theta$

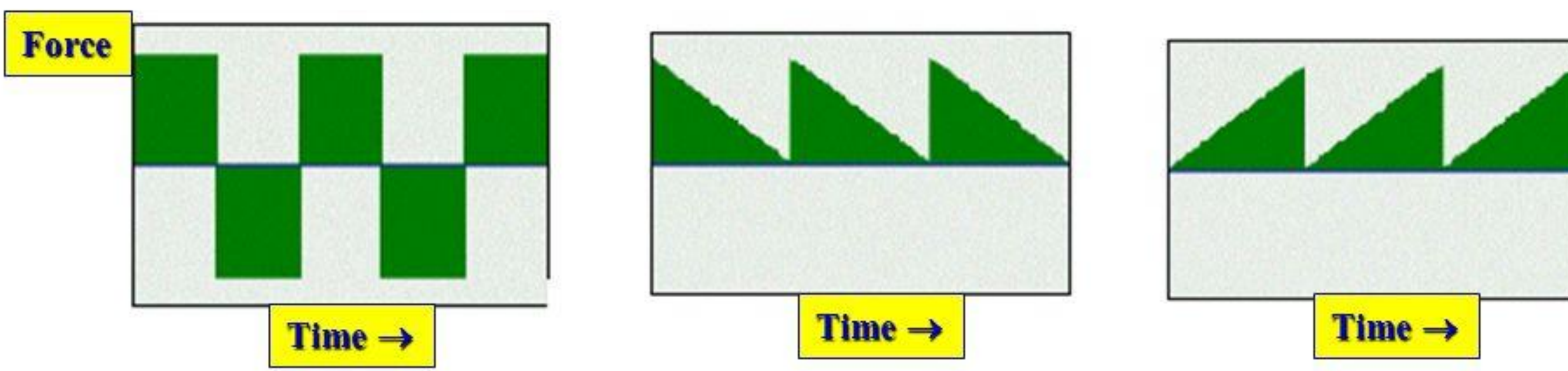
where θ it the angle between the mesh force resultant and the piston

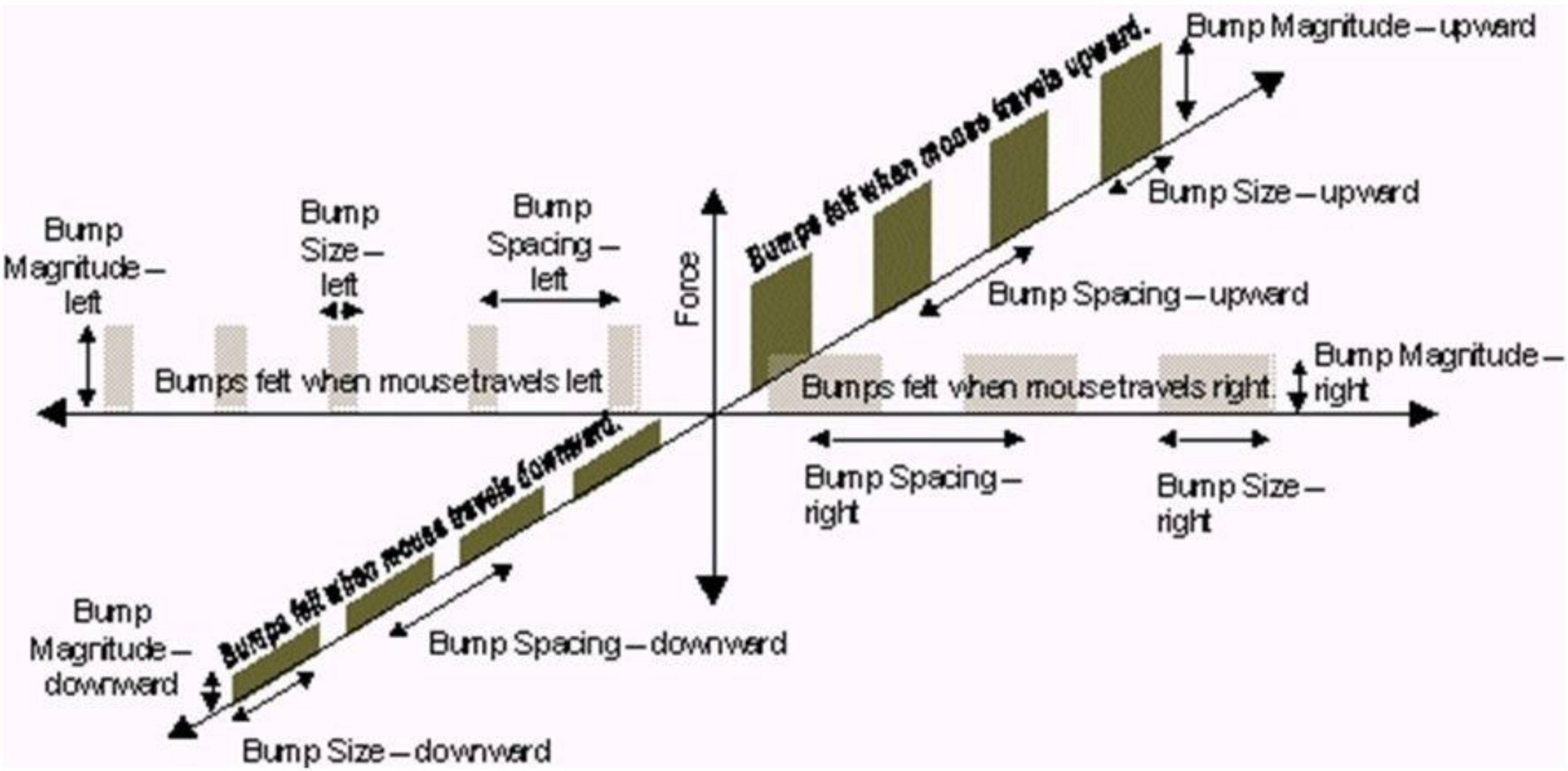
The Haptics Rendering Pipeline



Tactile mouse

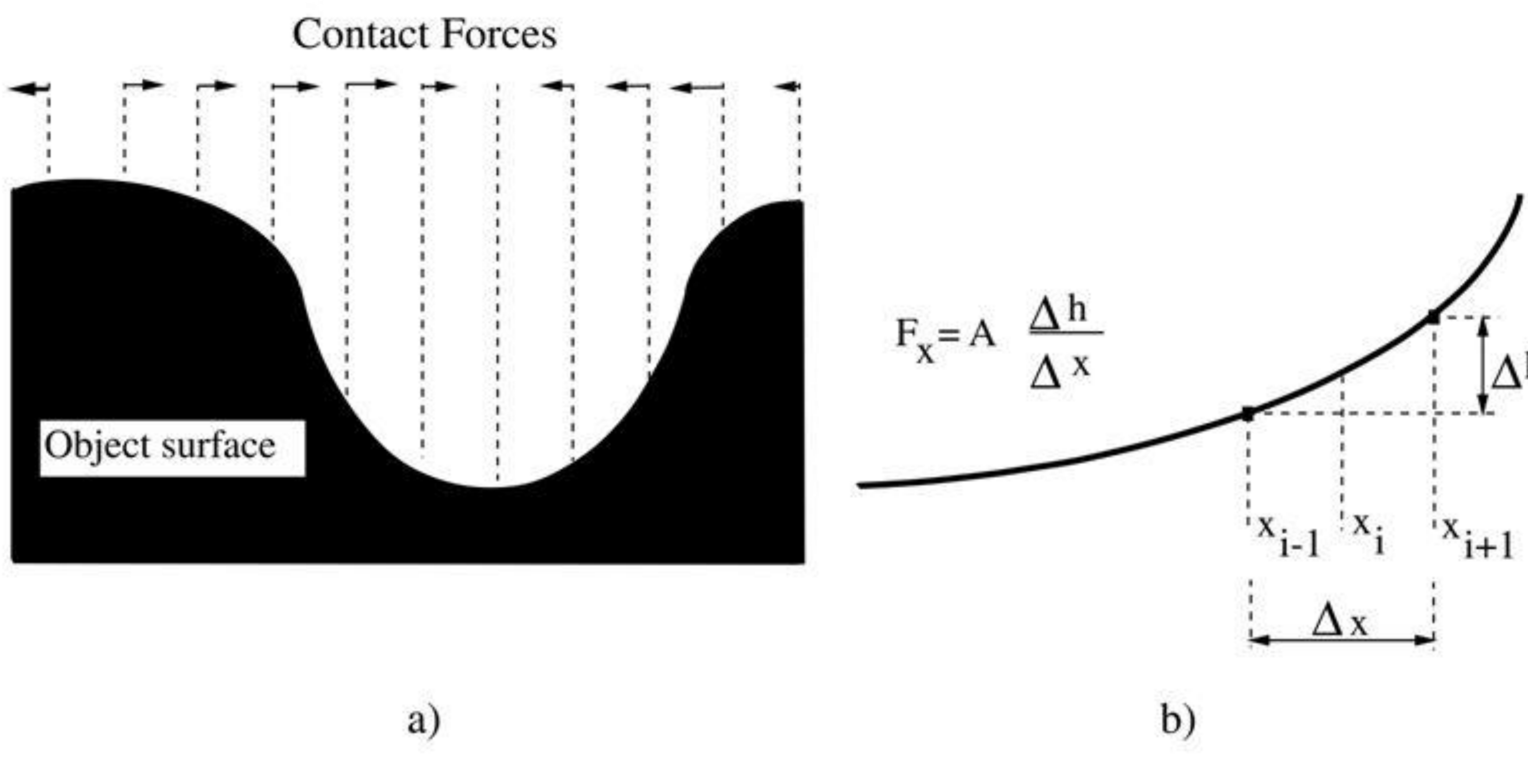
Tactile patterns produced by the Logitech mouse

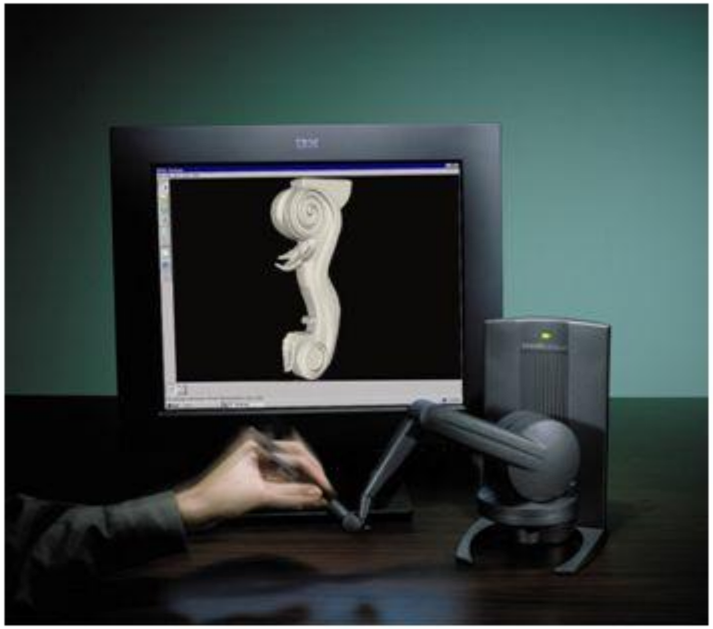




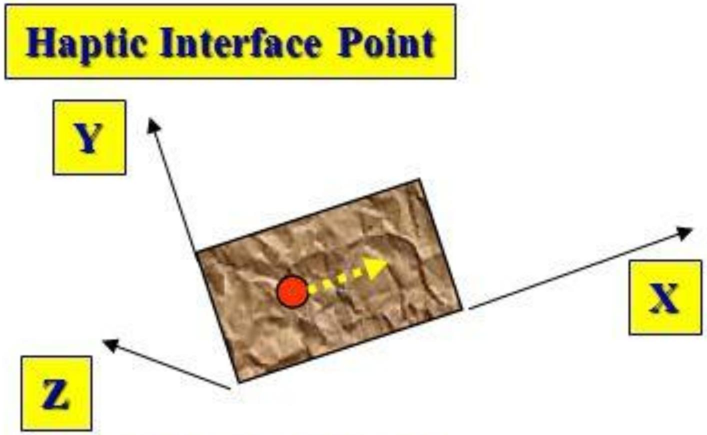
haptic mouse texture simulation

Surface haptic texture produced by the PHANToM interface





Haptic interface



Object polygon

$F_{texture} = A \sin(m\,x) \cdot \sin(n\,y)$,

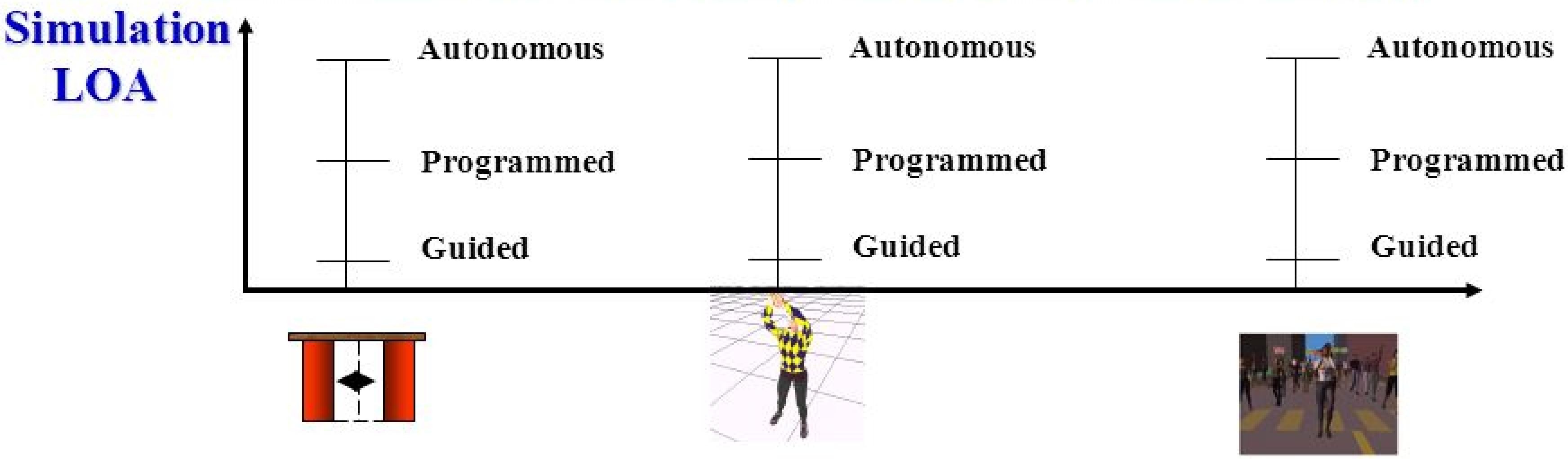
where A, m, n are constants:

- A gives magnitude of vibrations;
- m and n modulate the frequency of vibrations in the x and y directions

BEHAVIOR MODELING

- The simulation level of autonomy (LOA) is a function of its components
- Thalmann et al. (2000) distinguish three levels of autonomy. The simulation components can be either "guided" (lowest), "programmed" (intermediate" and "autonomous (high)

Simulation LOA = f(LOA(Objects),LOA(Agents),LOA(Groups))



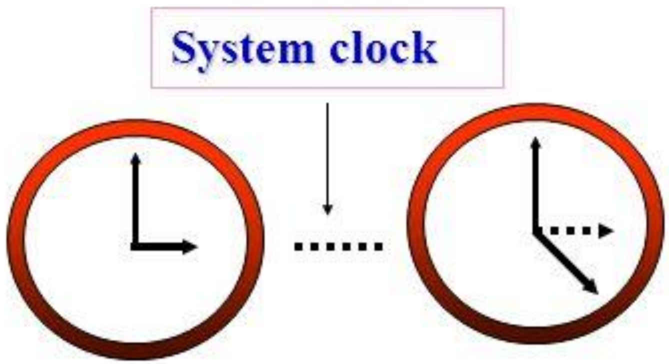
Interactive object **Intelligent agent** **Group of agents**

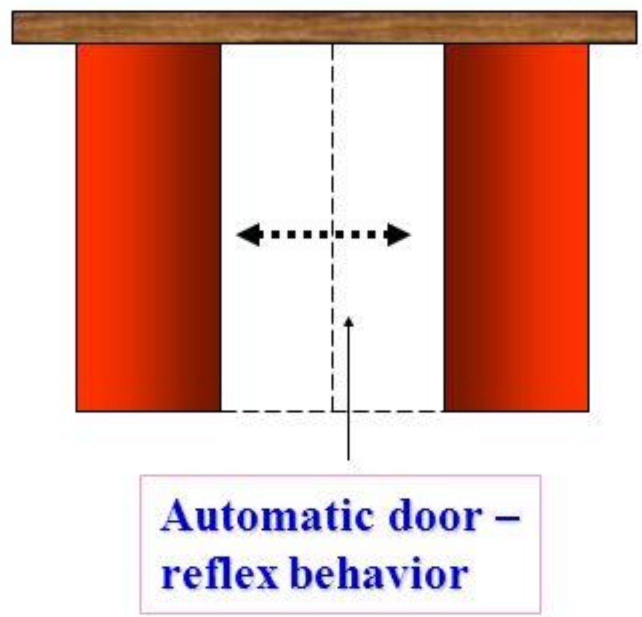
adapted from (Thalmann et al., 2000)

Interactive objects:

✓Have behavior independent of user's input (ex. clock);

✓This is needed in large virtual environments, where it is impossible for the user to provide all required inputs.



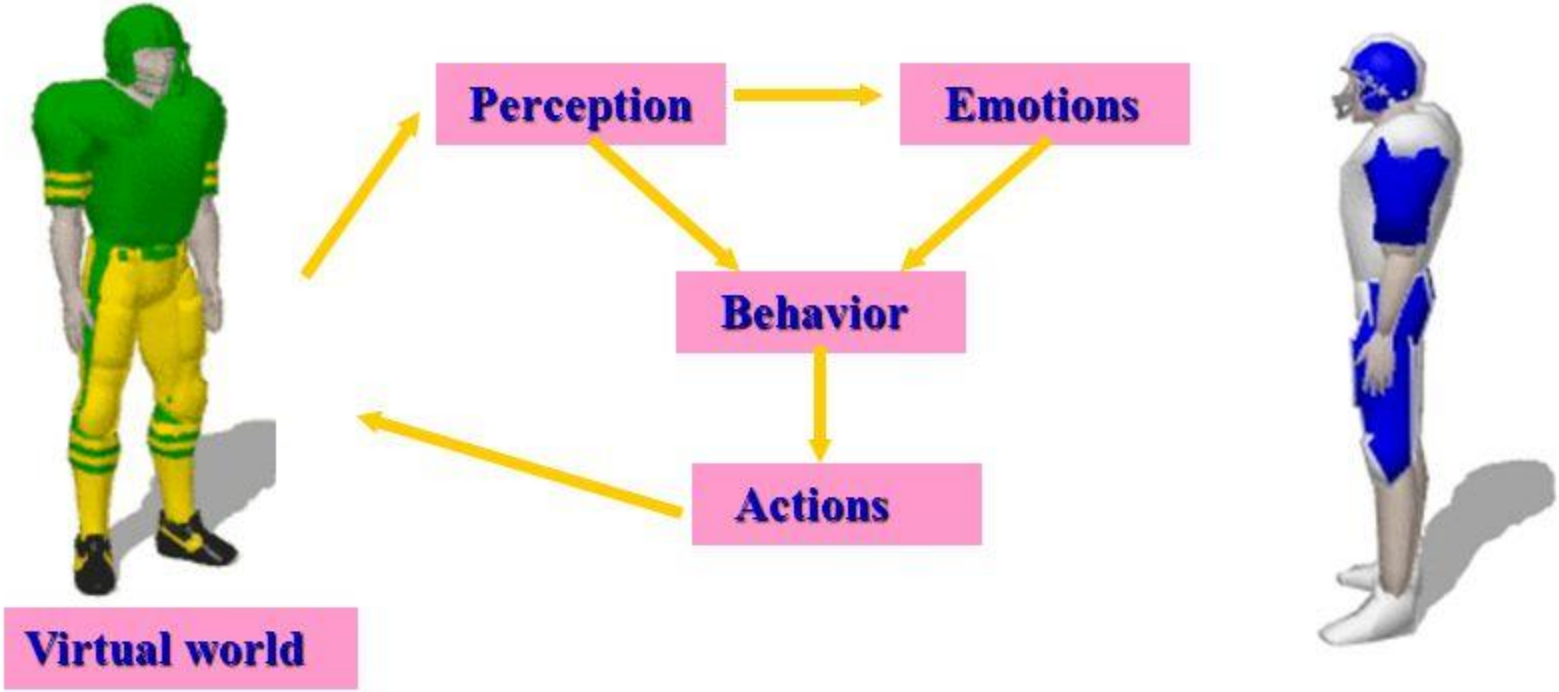


Interactive objects:

✓The fireflies in NVIDIA's ***Grove*** have behavior independent of user's input. User controls the virtual camera;

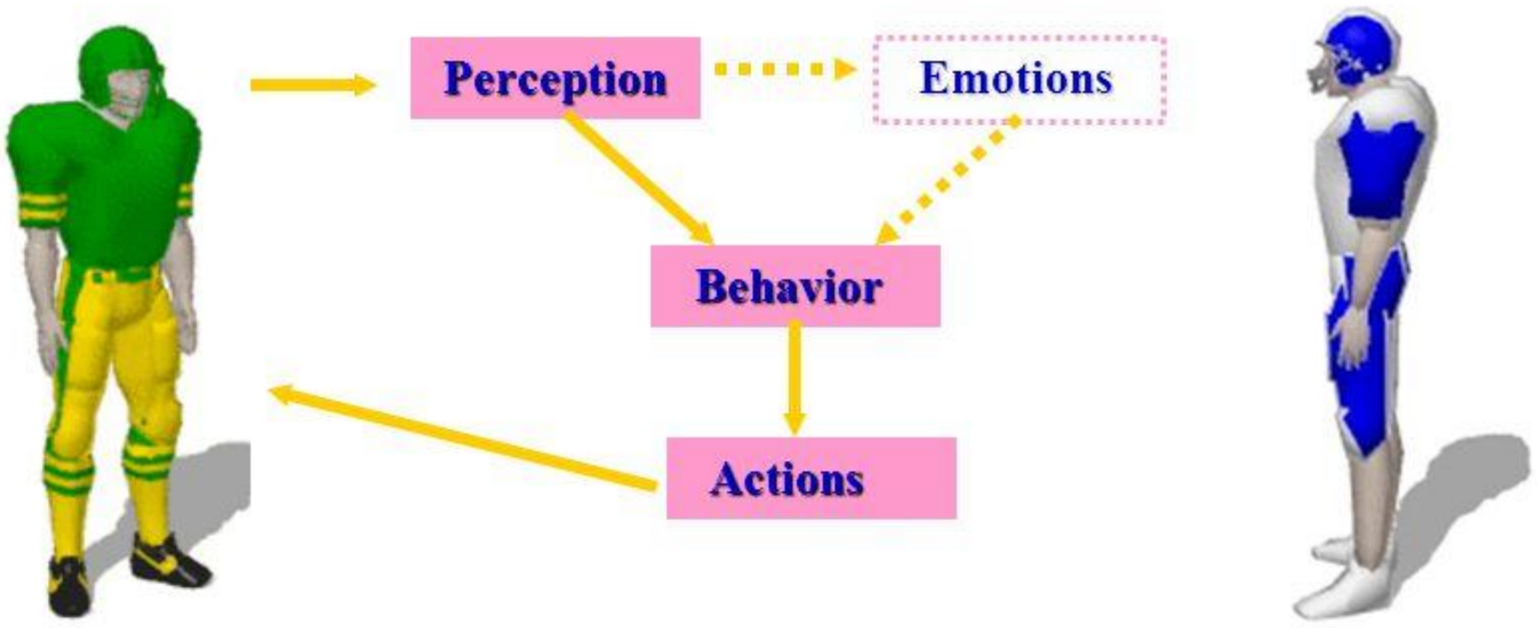
Agent behavior:

- ✓A behavior model composed of ***perception***, ***emotions***, ***behavior***, and ***actions***;
- ✓Perception (through virtual sensors) makes the agent aware of his surroundings.

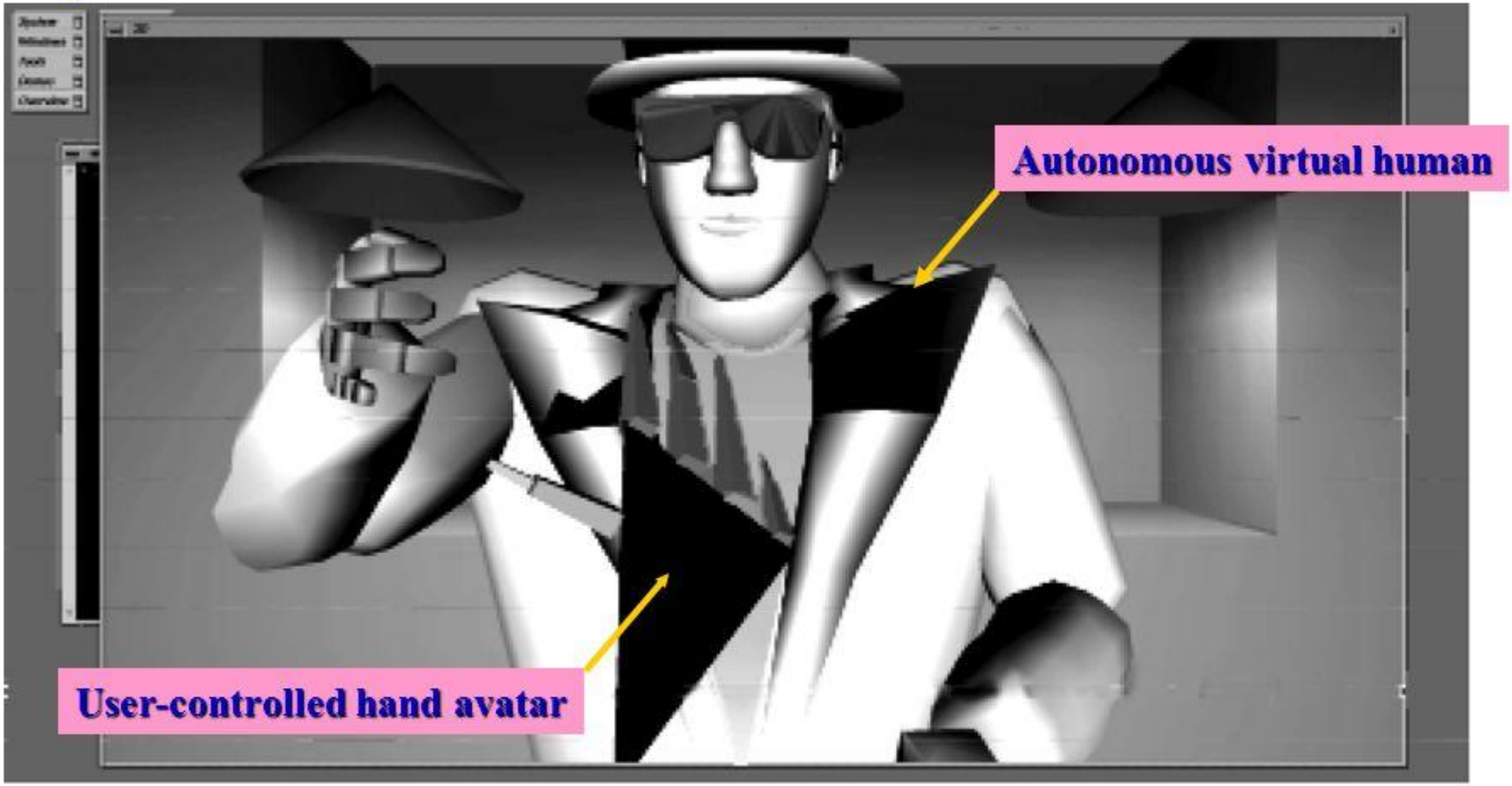


Reflex behavior:

- ✓ A direct link between perception and actions (following behavior rules ("cells");
- ✓ Does not involve emotions.



Object behavior



Another example of reflex behavior – "Dexter" at MIT [Johnson, 1991]: Hand shake, followed by head turn

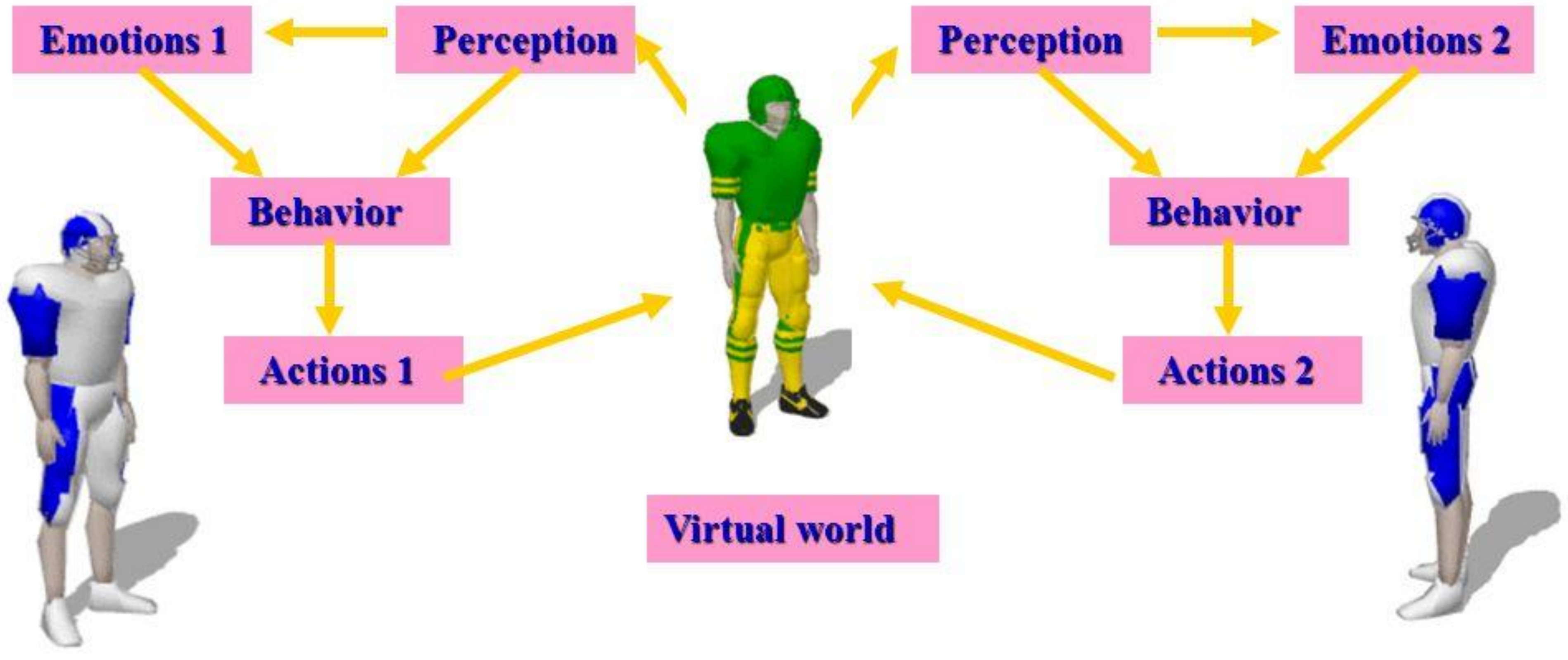
Agent behavior - avatars

If user maps to a full-body avatar, then virtual human agents react through ***body expression recognition***: example dance. Swiss Institute of Technology, 1999 (credit Daniel Thalmann)

Emotional behavior:

✓ A subjective strong feeling (anger, fear) following perception;

✓ Two different agents can have different emotions to the same perception, thus they can have different actions.



Crowds behavior

✓Crowd behavior emphasizes group (rather than individual) actions;

✓Crowds can have guided LOA, when their behavior is defined explicitly by the user;

✓Or they can have Autonomous LOA with behaviors specified by rules and other complex methods (including memory).

Political demonstration

Guided crowd

User needs to specify
Intermediate path points

VC 5.3

Autonomous crowd

Group perceives info on its environment and decides a path to follow to reach the goal

(Thalmann et al., 2000)

MODEL MANAGEMENT

✓It is necessary to maintain interactivity and constant frame rates when rendering complex models. Several techniques exist:

- Level of detail segmentation;
- Cell segmentation;
- Off-line computations;
- Lighting and bump mapping at rendering stage;
- Portals.

Level of detail segmentation:

✓ Level of detail (LOD) relates to the number of polygons on the object's surface. Even if the object has high complexity, its detail may not be visible if the object is too far from the virtual camera (observer).

Tree with 27,000 polygons

Tree with 27,000 polygons (details are not perceived)

Static level of detail management:

✓Then we should use a simplified version of the object (fewer polygons), when it is far from the camera.

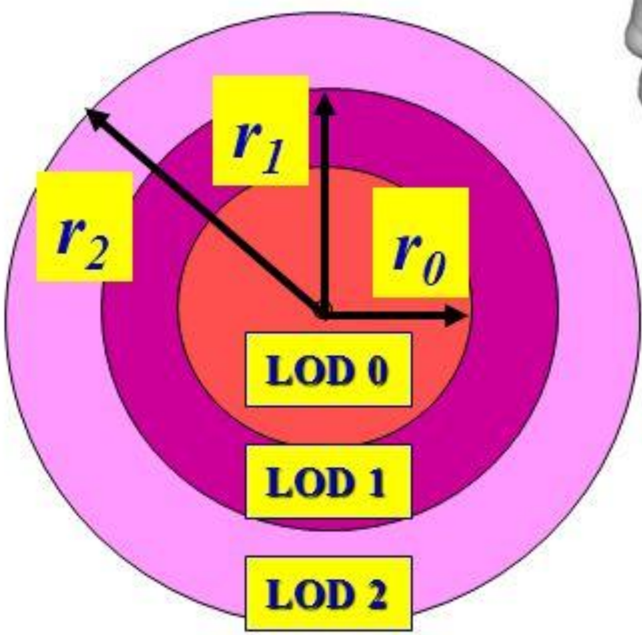
✓There are several approaches:

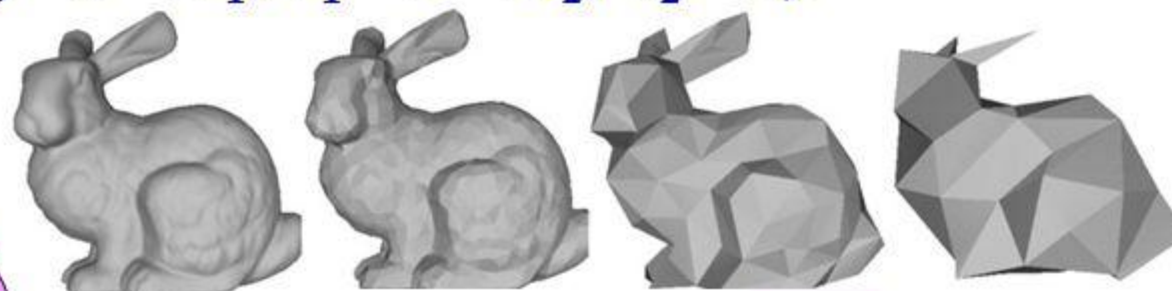
- Discrete geometry LOD;
- Alpha LOD;
- Geometric morphing ("geo-morph") LOD.

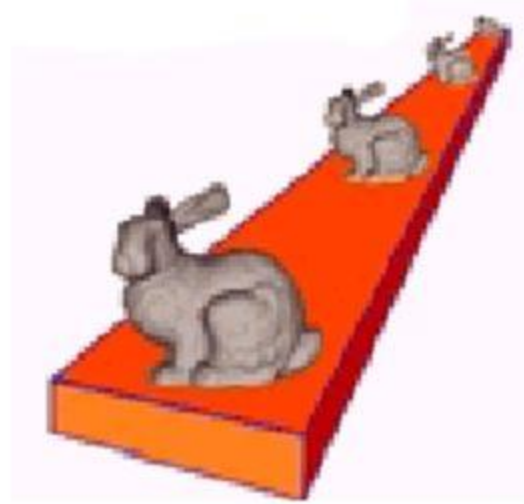
Discrete Geometry LOD:

✓Uses several discrete models of the same virtual object;

✓Models are switched based on their distance from the camera $(r < r_0; r_0 < r < r_1; r_1 < r < r_2; r_2 < r)$



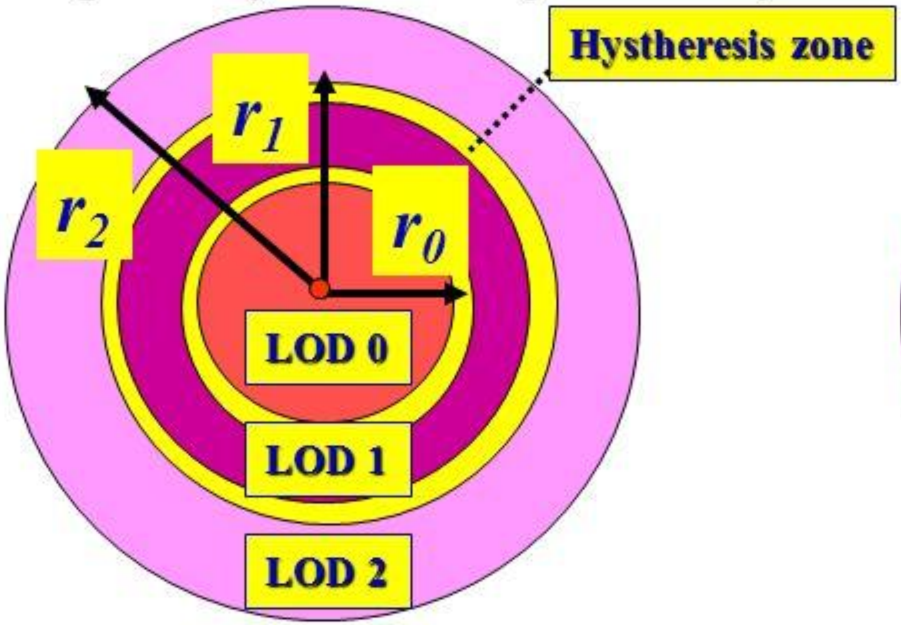


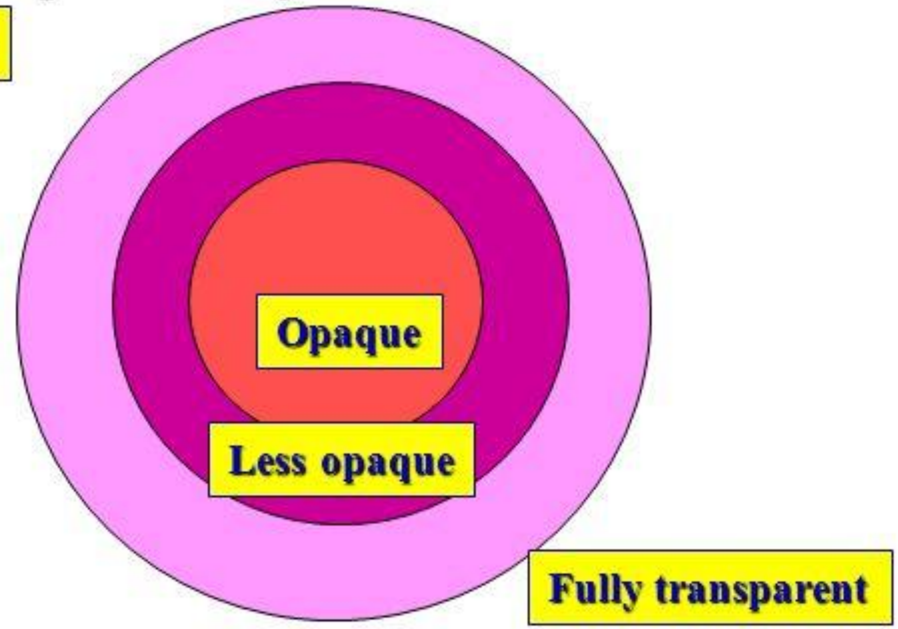


Alpha Blending LOD:

✓ Discrete LOD have problems on the $r_0 = r$, $r_1 = r$, $r_2 = r$ circles, leading to "popping". Objects appear and disappear suddenly. One solution is distance hystheresis. Another solution is model blending – two models are rendered near the circles;

✓ Another solution to popping is ***alpha blending*** by changing the transparency of the object. Fully transparent objects are not rendered.



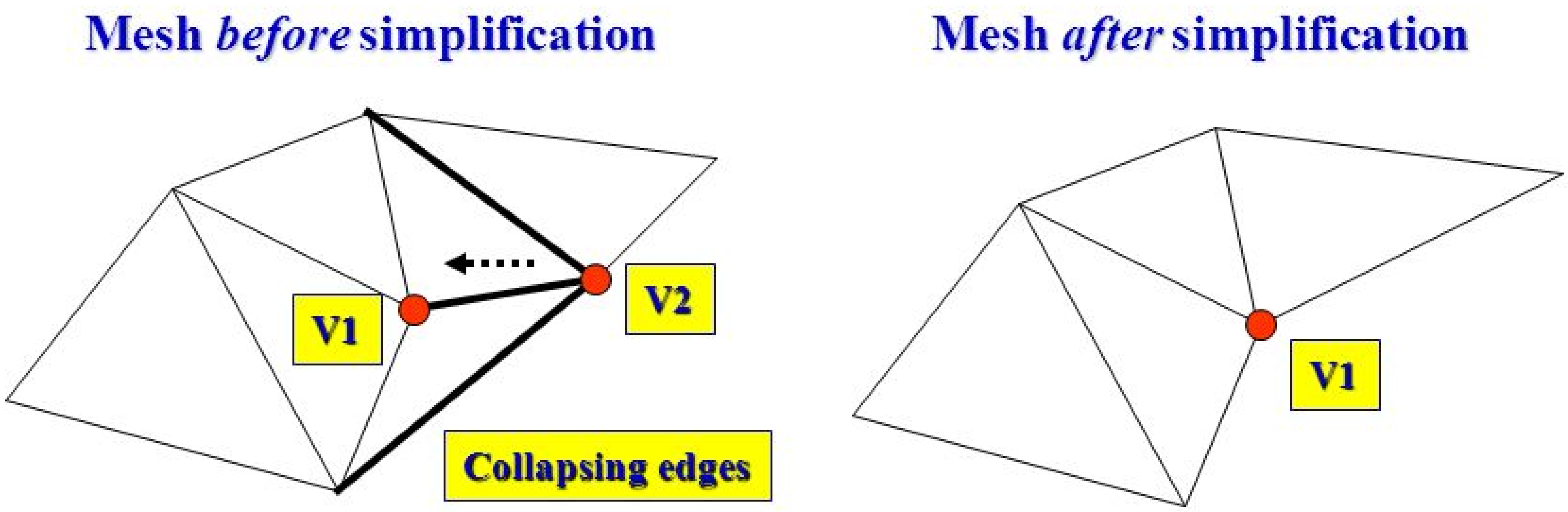


Geometric Morphing LOD:

✓Unlike geometric LOD, which uses several models of the same object, geometric morphing uses only one complex model.

✓Various LOD are obtained from the base model through ***mesh simplification***

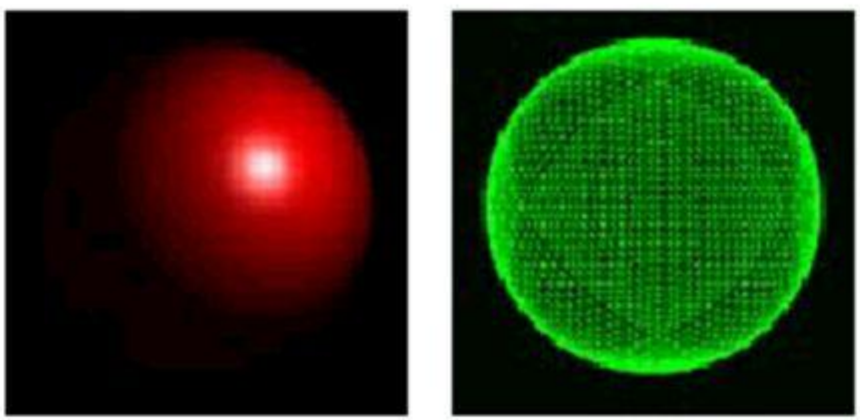
✓A triangulated polygon mesh: ***n*** vertices has ***2n*** faces and ***3n*** edges



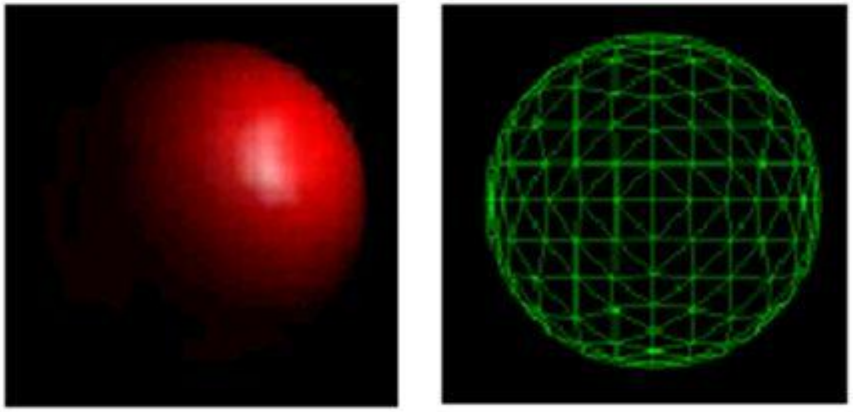
Single-Object adaptive level of detail LOD:

✓Used where there is a single highly complex object that the user wants to inspect (such as in interactive scientific visualization.

✓Static LOD will not work since detail is lost where needed- example the sphere on the right loses shadow sharpness after LOD simplification.



Sphere with 8192 triangles – Uniform high density

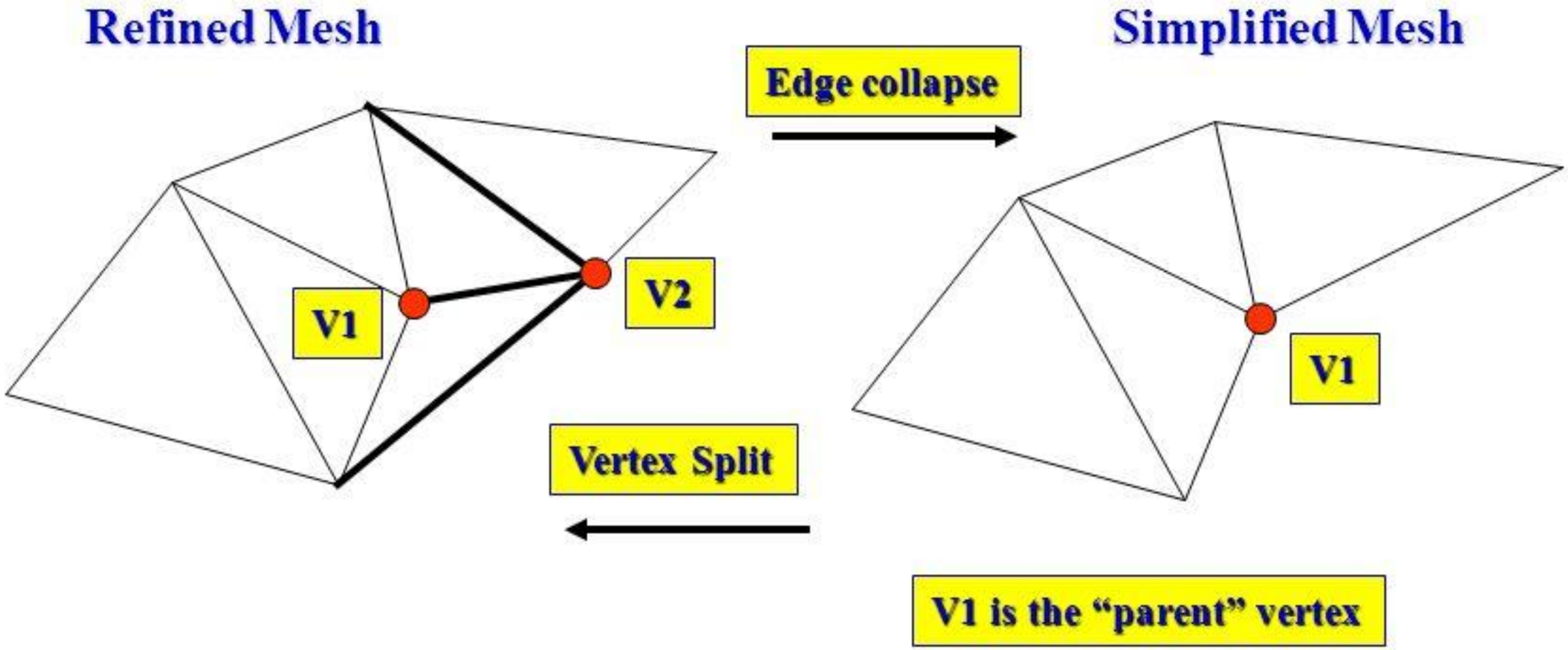


Sphere with 512 triangles – Static LOD simplification

(from Xia et al, 1997)

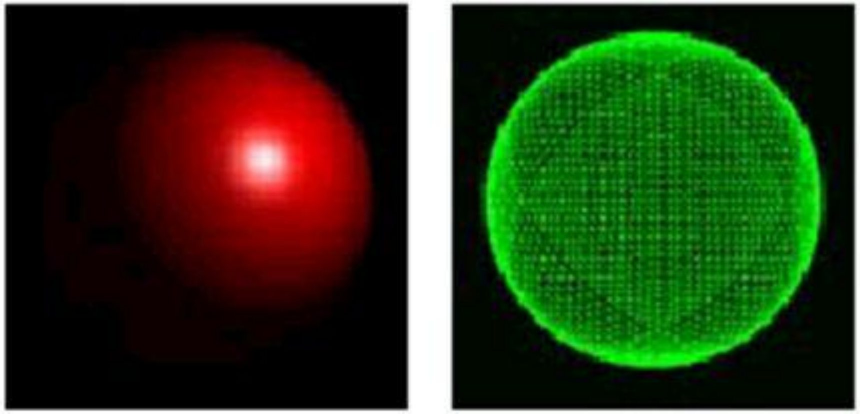
Single-object Adaptive Level of Detail

Sometimes edge collapse leads to problems, so vertices need to be split again to regain detail where needed. Xia et al. (1997) developed an adaptive algorithm that determines the level of detail based on distance to viewer as well as normal direction (lighting).

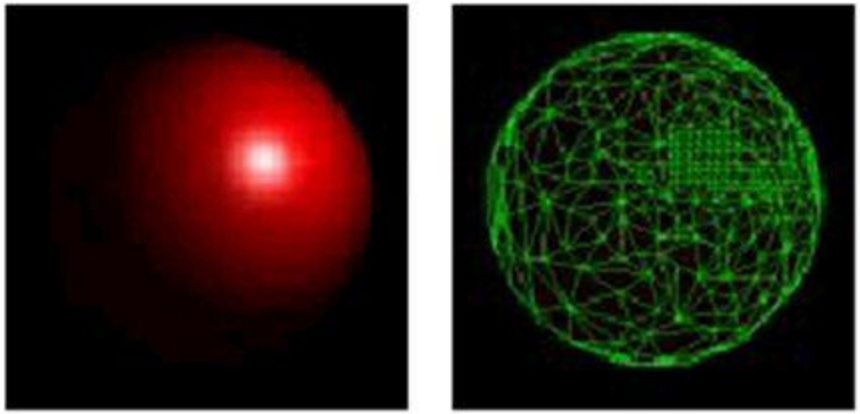


(adapted from Xia et al, 1997)

Single-object Adaptive Level of Detail



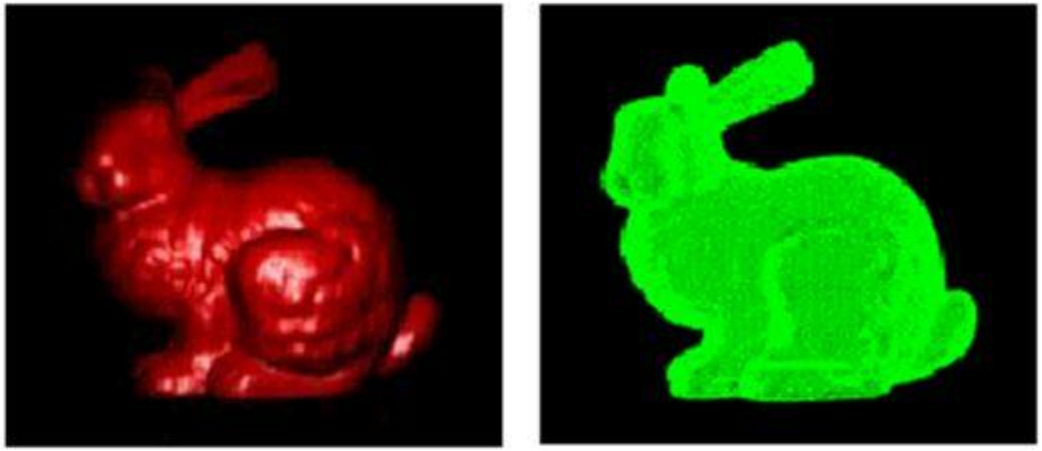
Sphere with 8192 triangles – Uniform high density, 0.115 sec to render



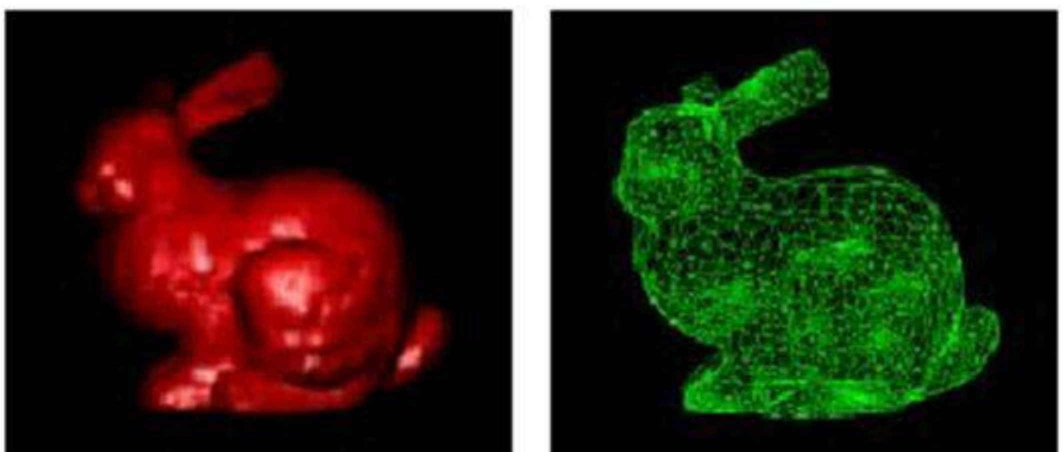
Sphere with 537 triangles – adaptive LOD, 0.024 sec to render (SGI RE2, single R10000 workstation)

(from Xia et al, 1997)

Single-object Adaptive Level of Detail



Bunny with 69,451 triangles – Uniform high density, 0.420 sec to render

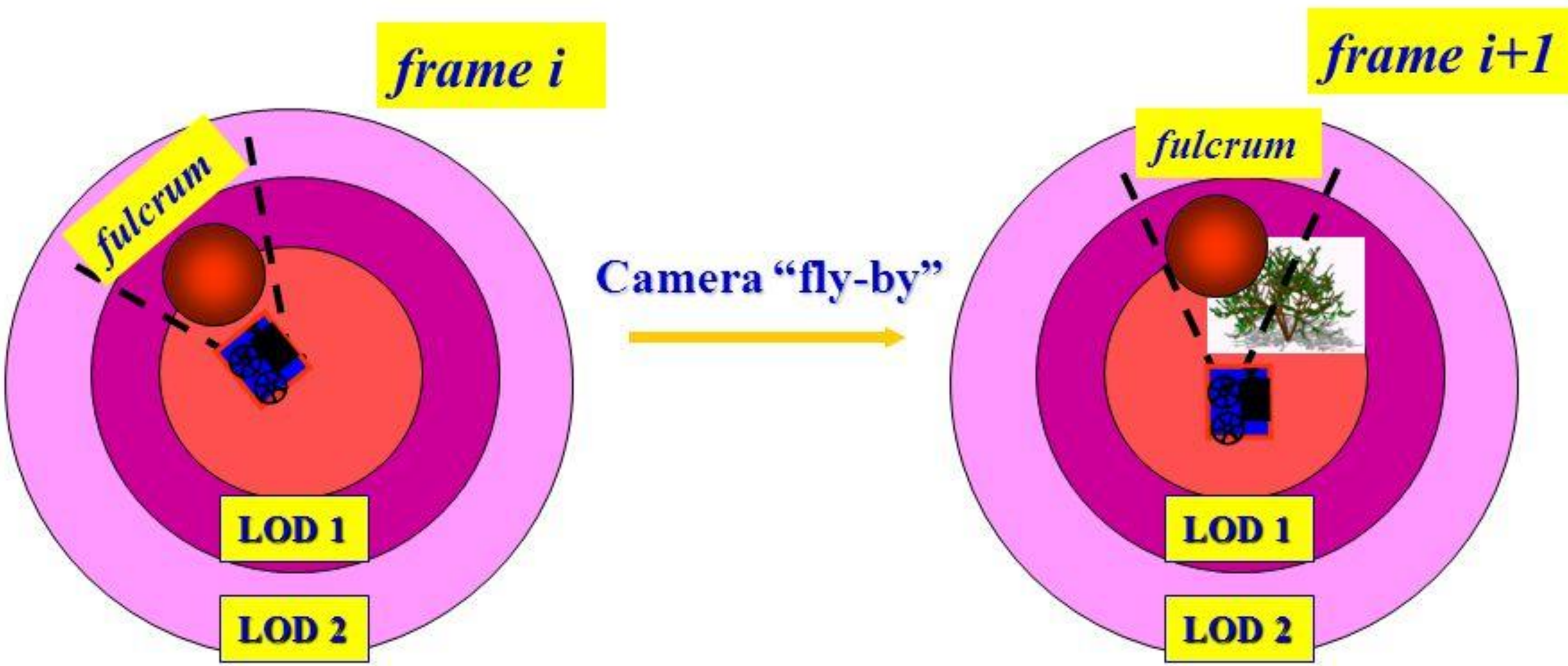


Bunny with 3615 triangles – adaptive LOD, 0.110 sec to render (SGI RE2, single R10000 workstation)

(from Xia et al, 1997)

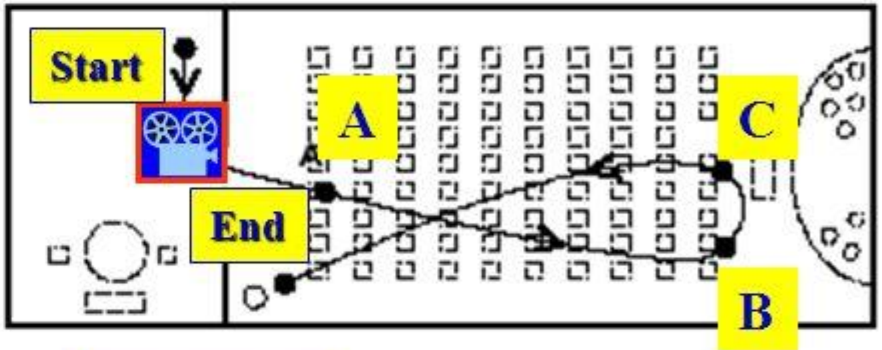
Static LOD:

✓ Geometric LOD, alpha blending and morphing have problems maintaining a constant frame rate. This happens when new complex objects appear *suddenly* in the scene (fulcrum).

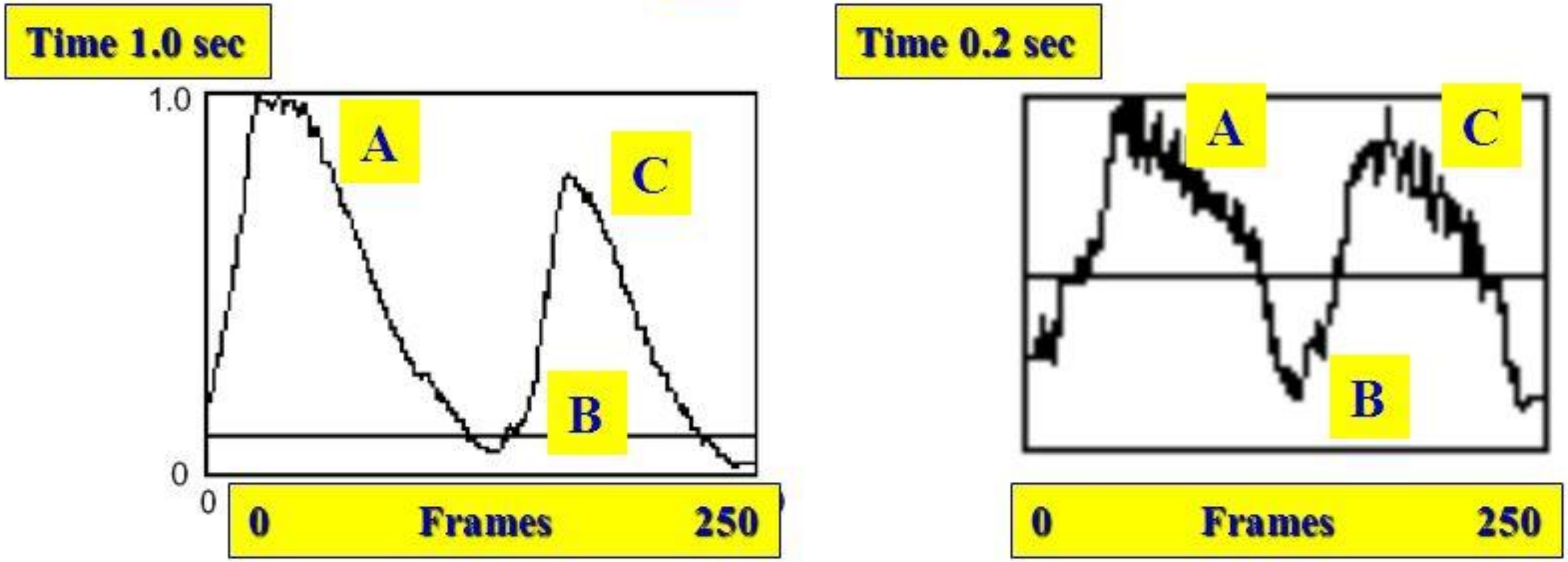


Architectural "walk-through" (UC Berkeley Soda Hall)

Camera path through auditorium



No LOD menegament, 72,570 polygons



No LOD management

Static LOD management

from (Funkhauser and Sequin, 1993)

Adaptive LOD Management-continued:

- ✓ An algorithm that selects LOD of visible objects based on a specified frame rate;
- ✓ The algorithm (Funkhauser and Sequin, 1993) is based on a benefits to cost analysis, where cost is the time needed to render Object ***O*** at level of detail ***L***, and rendering mode ***R***.
- ✓ The cost for the whole scene is

$$\Sigma \text{ Cost } (O,L,R) \leq \text{Target frame time}$$

- ✓ where the cost for a single object is

$$\text{Cost } (O,L,R) = \max (c1\text{Polygons}(O,L) + c2 \text{ Vertices}(O,L), c3 \text{ Pixels}(O,L))$$

c1, c2, c3 are experimental constants, depending on R and type of computer

Adaptive LOD Management:

✓Similarly the benefit for a scene is a sum of visible objects benefits;

Σ Benefit(O,L,R)

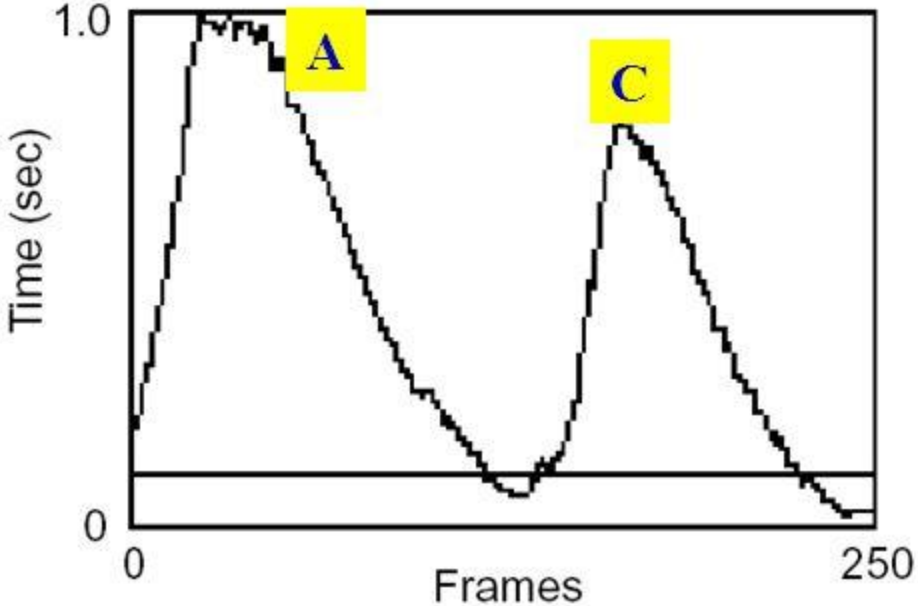
✓where the benefit of a given object is

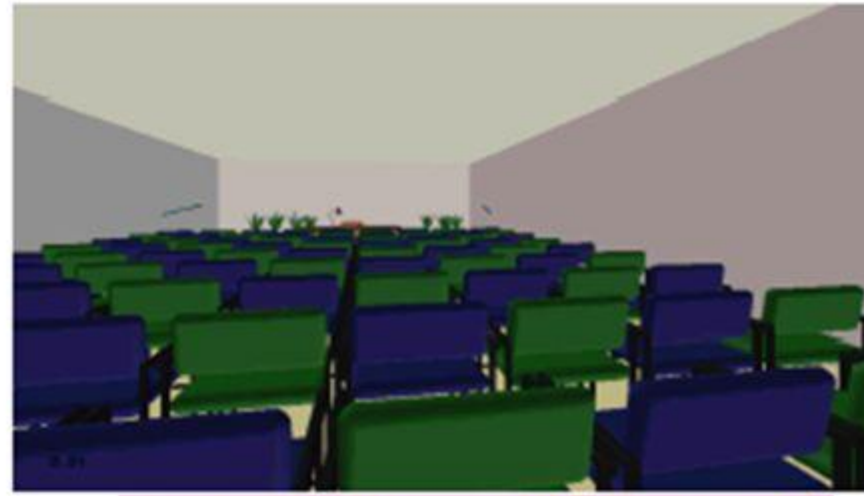
Benefit(O,L,R) = size(O) * Accuracy(O,L,R) * Importance(O) * Focus(O) * Motion(O) * Hysteresis(O,L,R)

✓ The algorithm tries to maximize each object's "value"

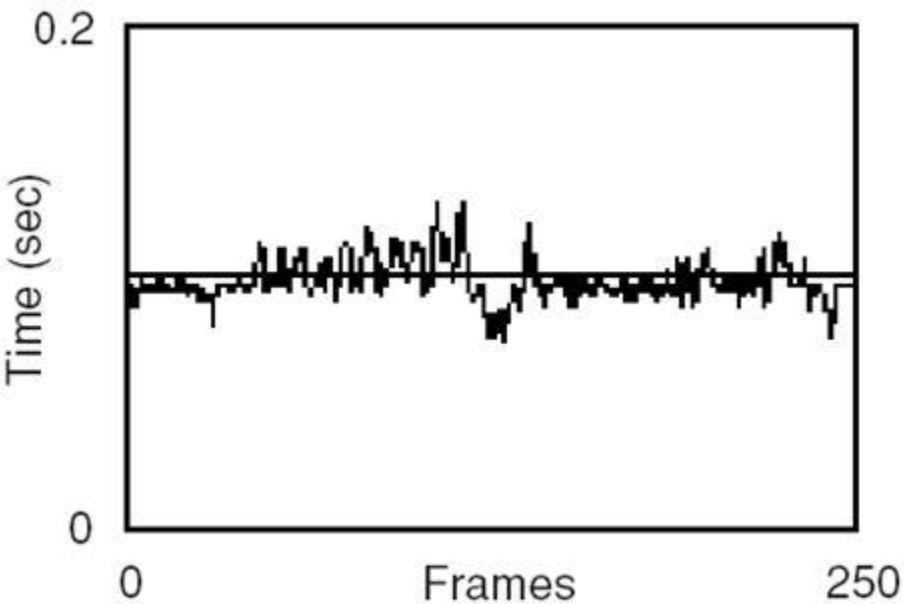
Value= Benefit(O,L,R)/Cost(O,L,R)

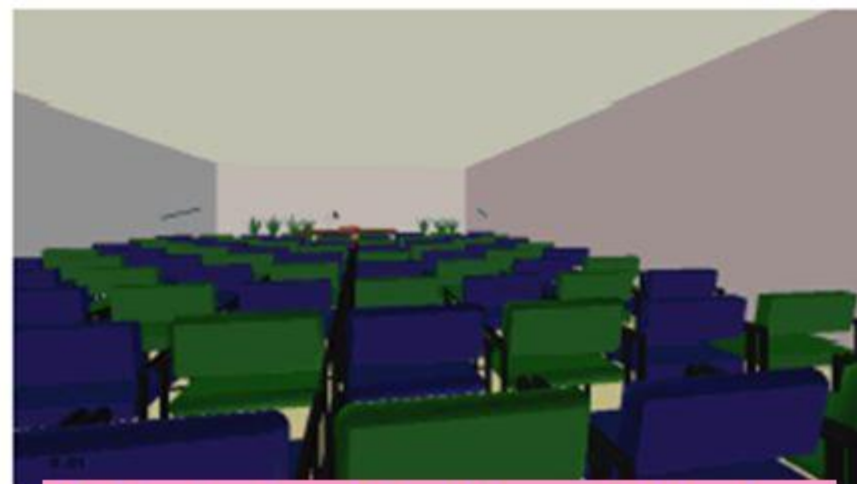
✓Objects with higher value (larger size) are rendered first





No detail elision, 72,570 polygons





Optimization algorithm, 5,300 poly.
0.1 sec target frame time (10 fps)

from (Funkhauser and Sequin, 1993)

Level of detail segmentation – rendering mode

No detail elision, 19,821 polygons

Optimization, 1,389 poly., 0.1 sec target frame time

Level of detail – darker gray means more detail

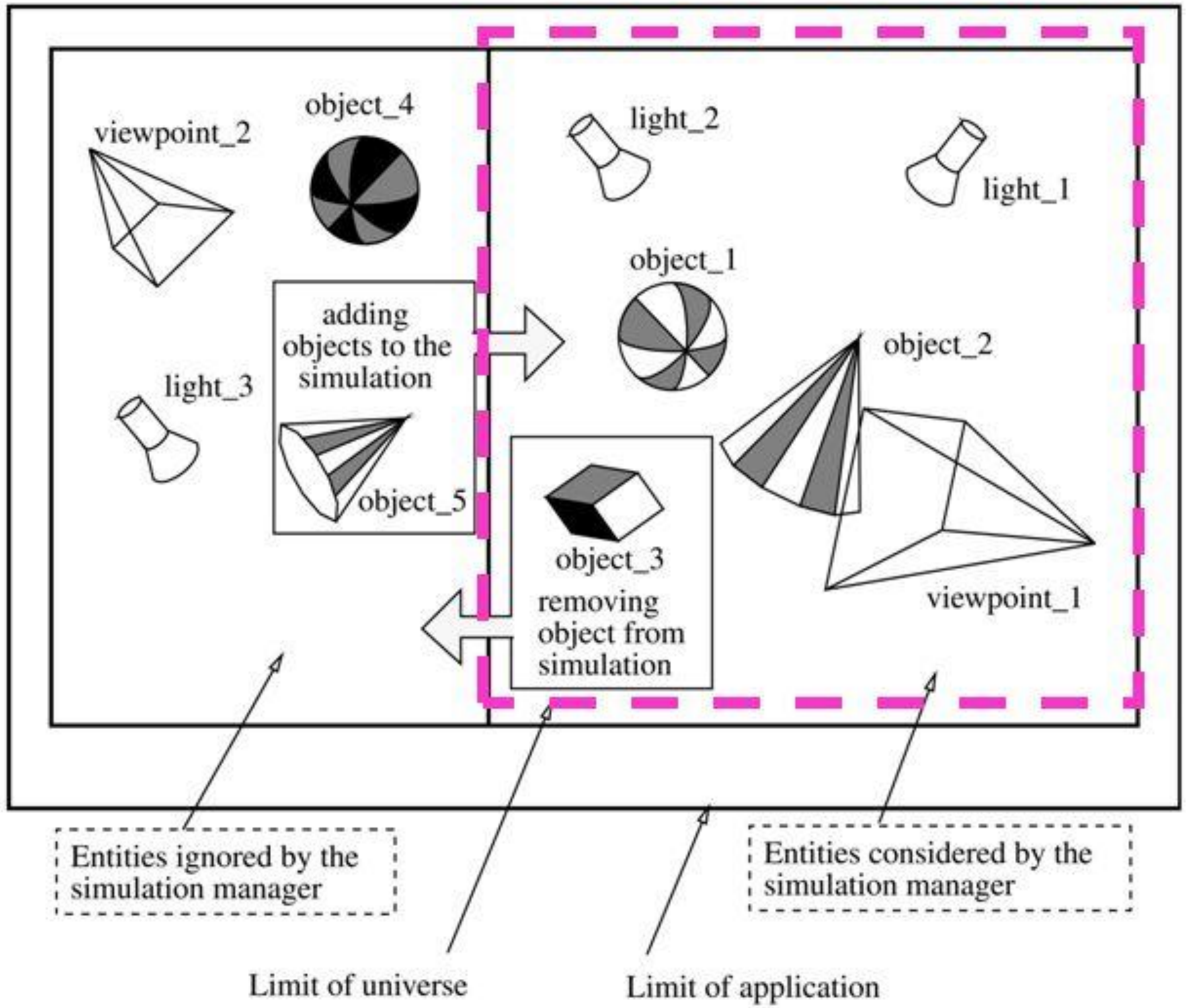
from (Funkhauser and Sequin, 1993)

Cell segmentation:

- ✓It is another method of model management, used in architectural walk-through;
- ✓To maintain the “virtual building” illusion it is necessary to have at least 6 fps (Airey et al., 1990)
- ✓Necessary to maintain ***interactivity*** and ***constant frame rates*** when rendering complex models.

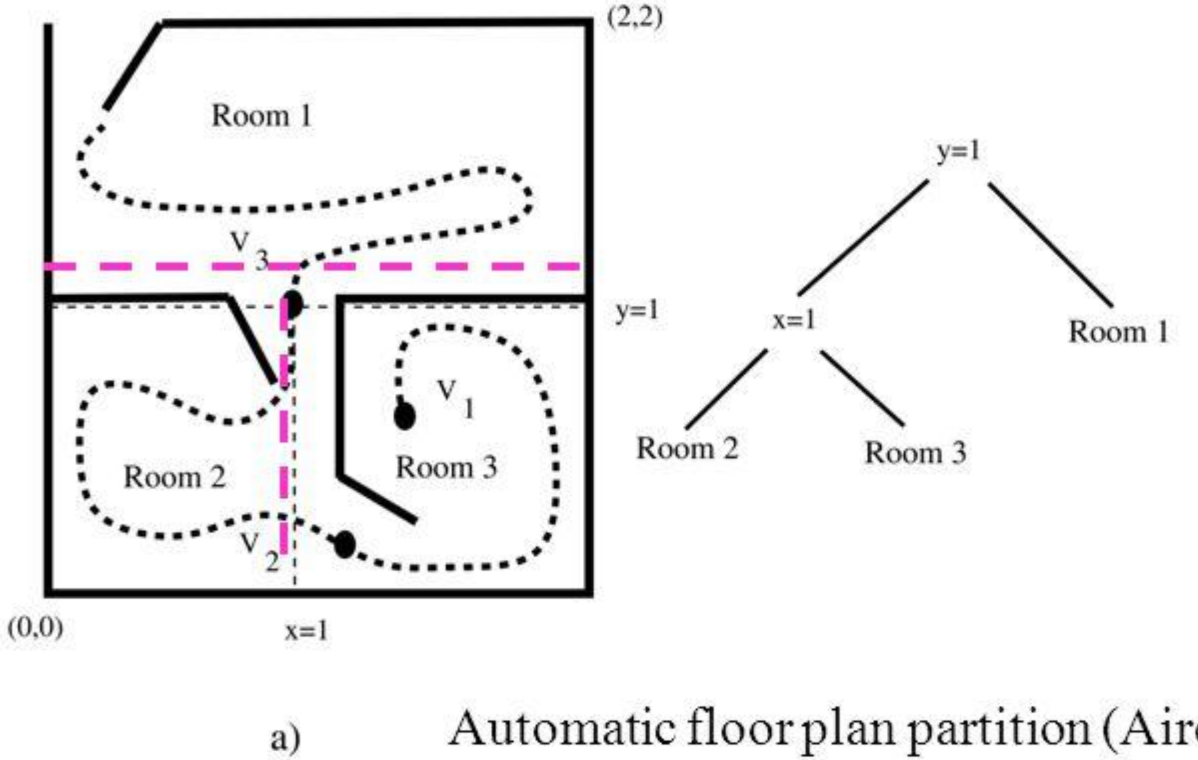
Model management

Only the current "universe" needs to be rendered



Cell segmentation – increased frame rate

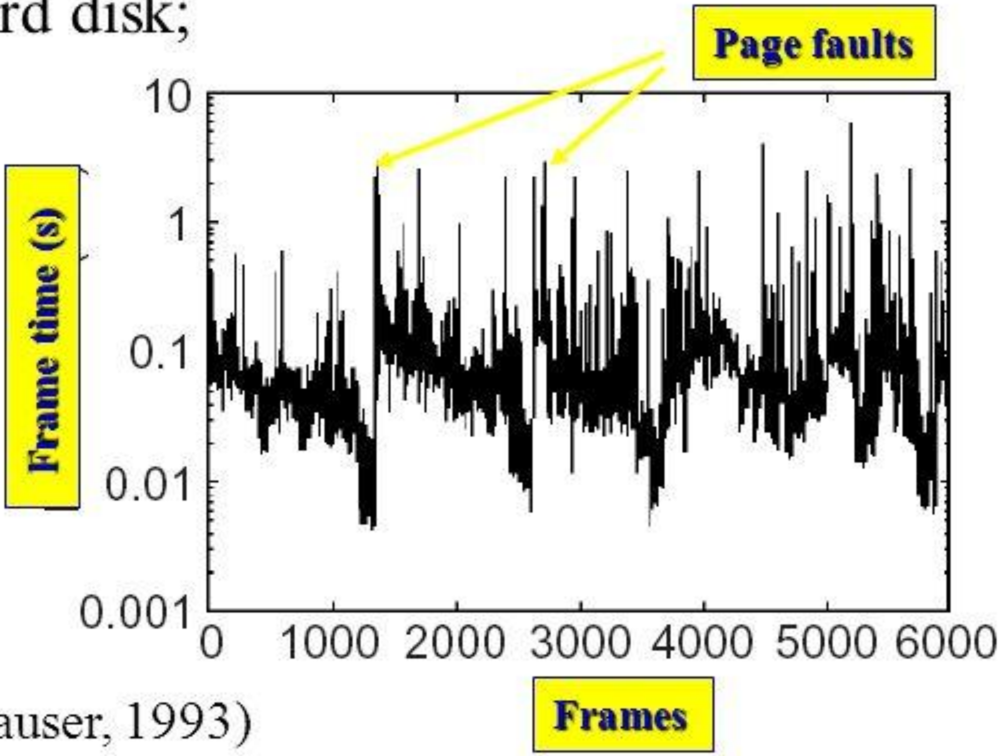
- ✓Buildings are large models that can be partitioned in "cells" automatically and off-line to speed up simulations at run time;
- ✓Cells approximate rooms;
- ✓Partitioning algorithms use a "priority" factor that favors occlusions (partitioning along walls)



Automatic floor plan partition (Airey et al., 1990)

Cell segmentation

✓Building model resides in a fully associative cache;
✓But cell segmentation alone will not work if the model is so large that it exceeds available RAM;
✓In this case large delays will occur when there is a page fault and data has to be retrieved from hard disk;

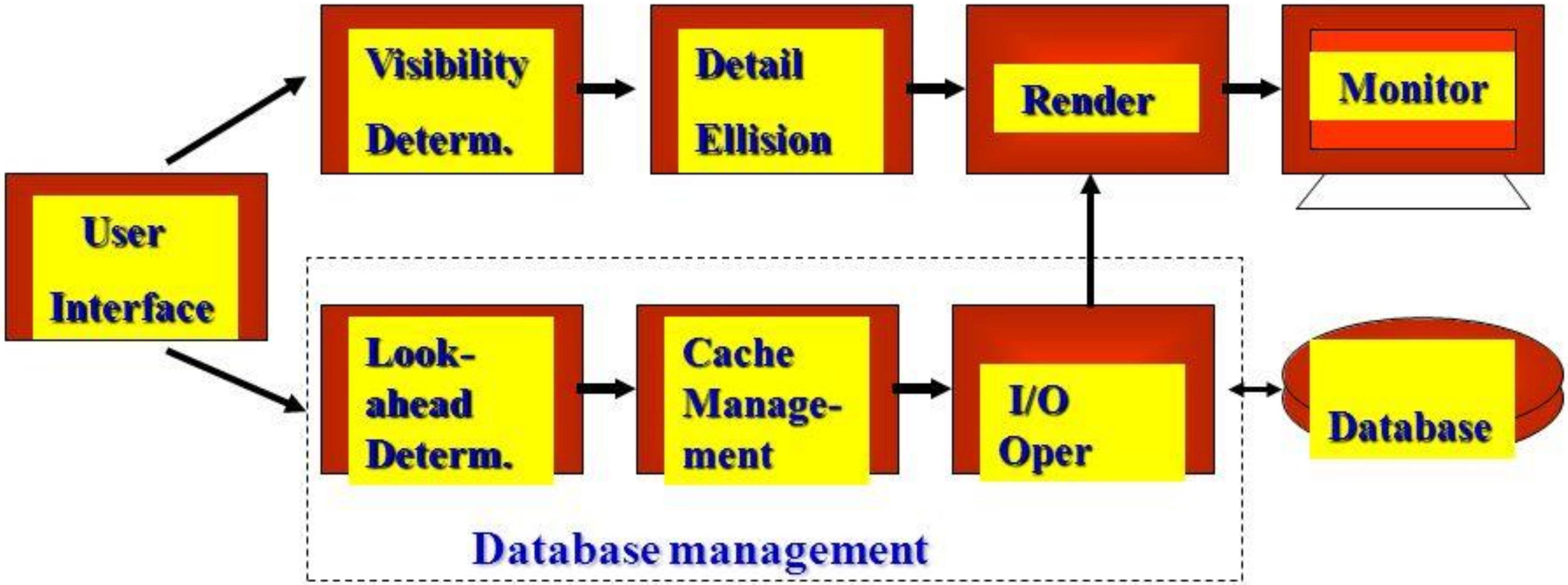


From (Funkhauser, 1993)

Combined Cell, LOD and database methods

✓It is possible to add database management techniques to prevent page faults and improve fps uniformity during walk-through;

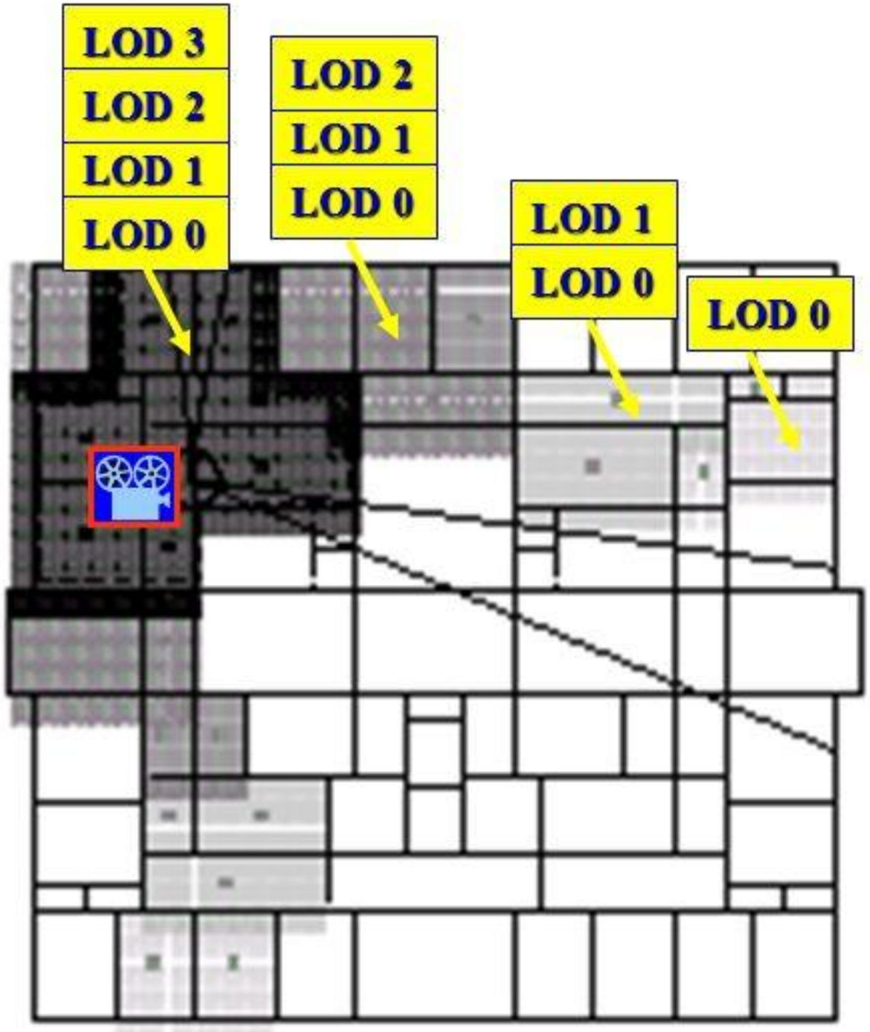
✓It is possible to estimate how far the virtual camera will rotate and translate ***over the next N frames*** and pre-fetch from the hard disk the appropriate objects.

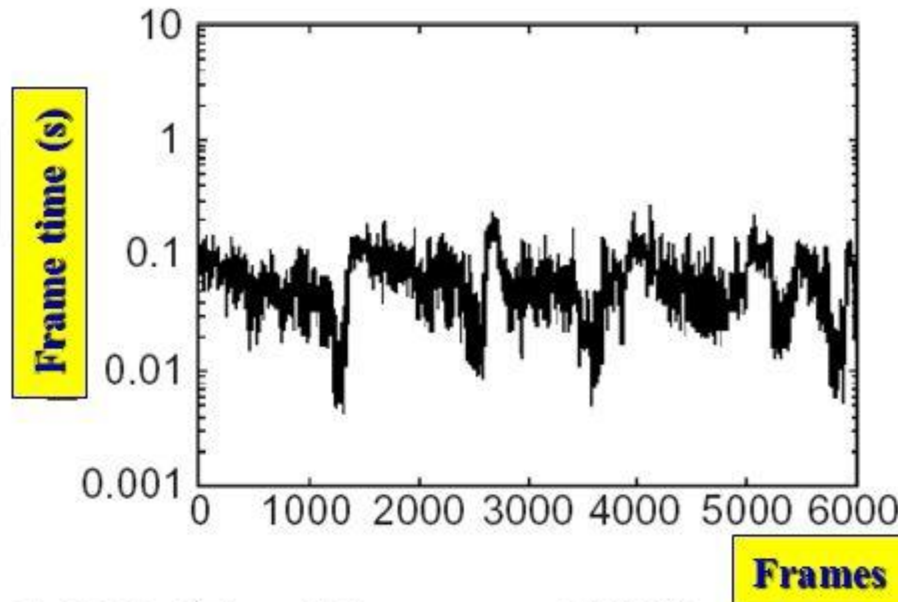


Floor plan partition (Funkhouser, 1993)

Database management

LOD 0 – lowest level of detail (loaded first)
….
LOD 3 - highest level of detail (loaded last)





Floor plan visibility and highest LOD (Funkhouser, 1990)