CS380: Computer Graphics
Clipping and Culling

Sung-Eui Yoon
(윤성의)

Course URL:
http://sgvr.kaist.ac.kr/~sungeui/CG/

KAIST

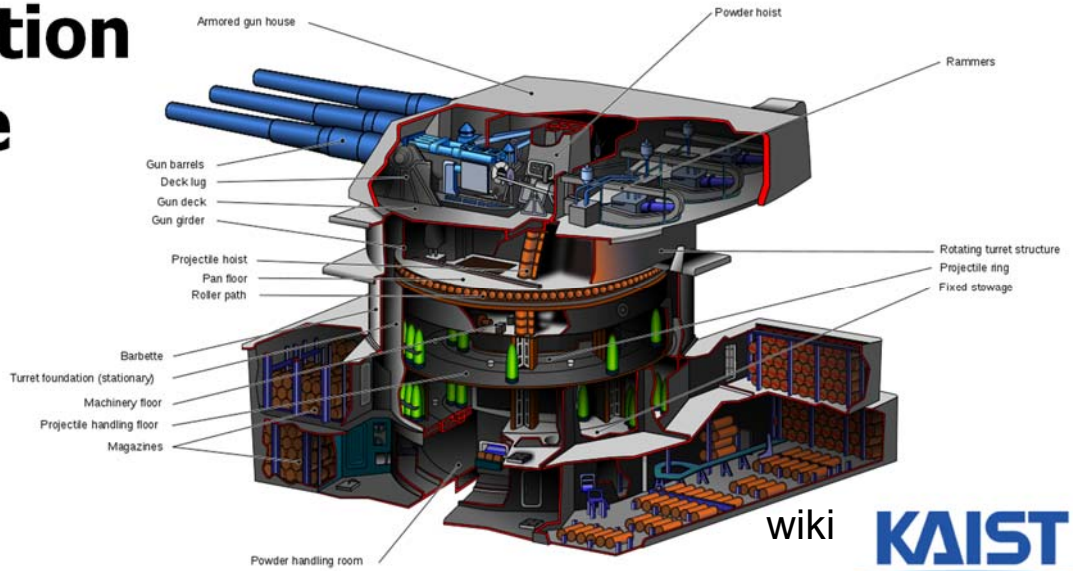
Class Objectives

- Understand clipping and culling
- Understand view-frustum, back-face culling, and hierarchical culling methods
- Know various possibilities to perform culling and clipping in the rendering pipeline

- Related chapter:
  - Ch. 6: Clipping and Culling

Culling and Clipping

- **Culling**
  - **Throws away entire objects and primitives that cannot possibly be visible**
  - **An important rendering optimization (esp. for large models)**
- **Clipping**
  - **"Clips off" the visible portion of a primitive**
  - **Simplifies rasterization**
  - **Also, used to create "cut-away" views**



wiki

KAIST

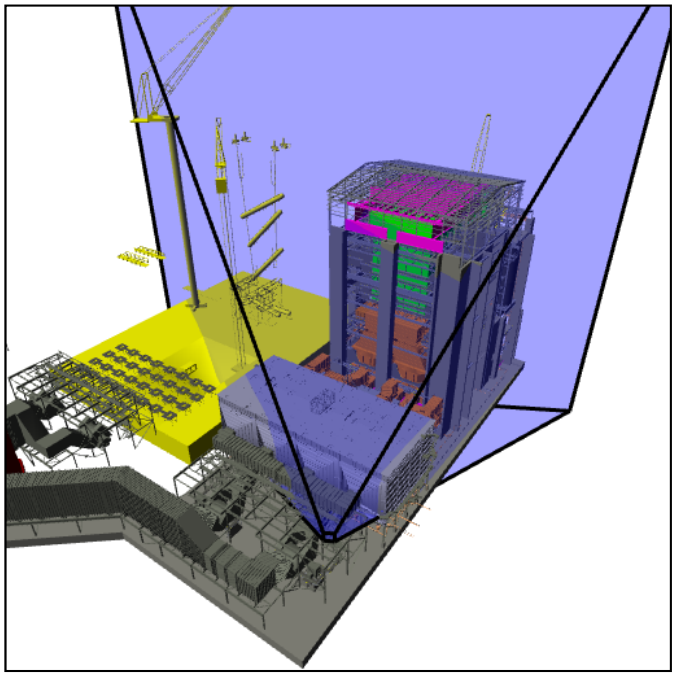
Culling Example

Power plant model

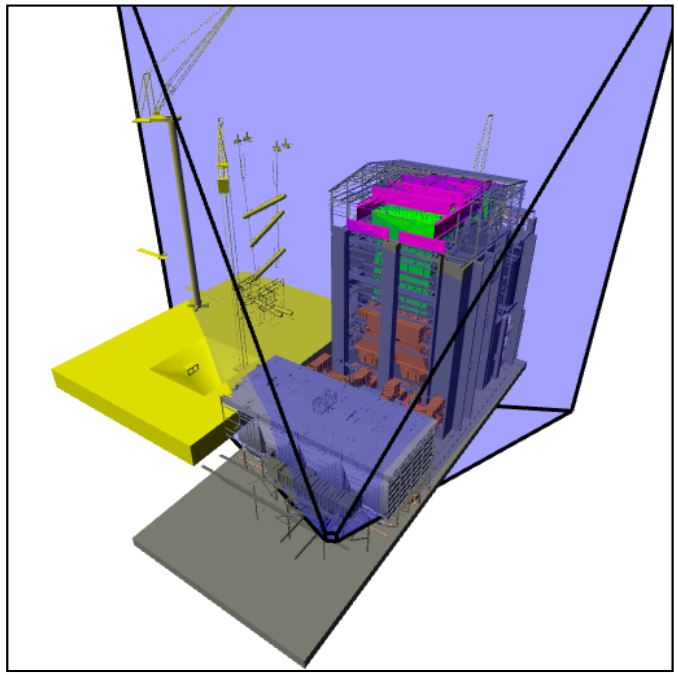
(12 million triangles)

KAIST

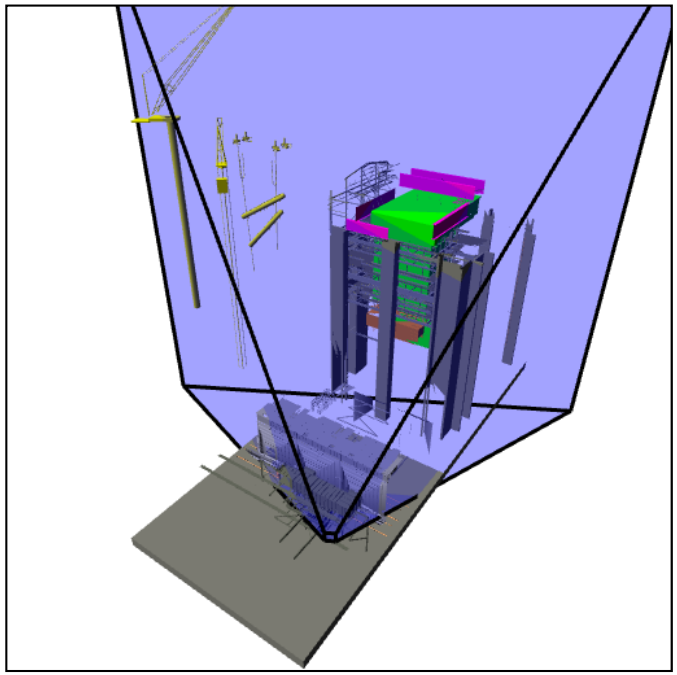
Culling Example



Full model
12 Mtris



View frustum culling
10 Mtris



Occulsion culling
1 Mtris

KAIST

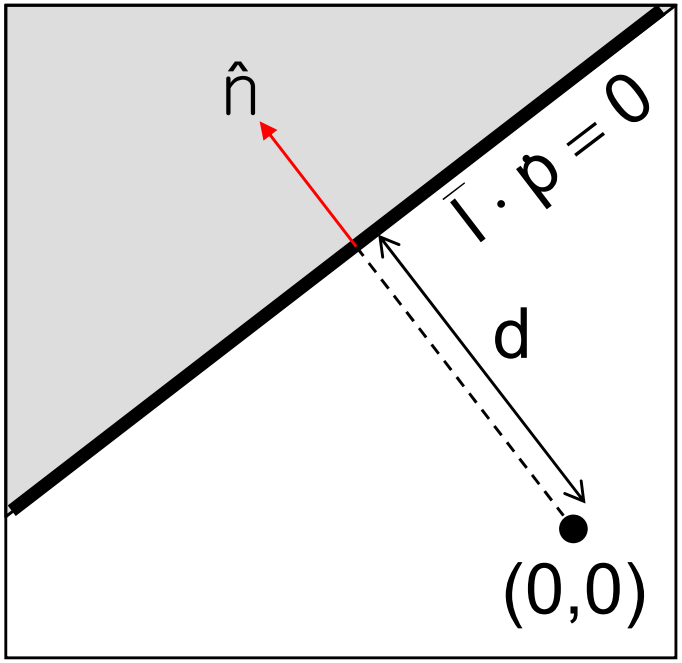
Lines and Planes

- **Implicit equation for line (plane):**

$$n_x x + n_y y - d = 0$$

$$\begin{bmatrix} n_x & n_y & -d \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$\Rightarrow \quad \bar{l} \cdot \dot{p} = 0$$

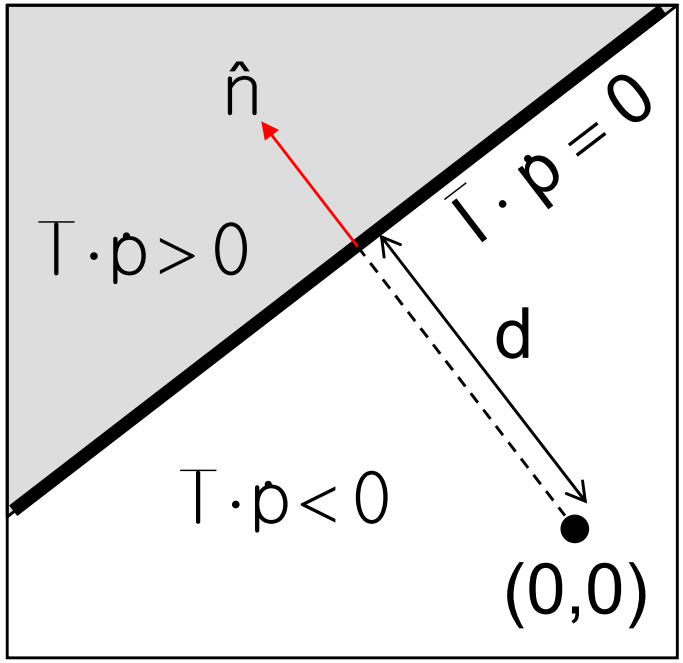


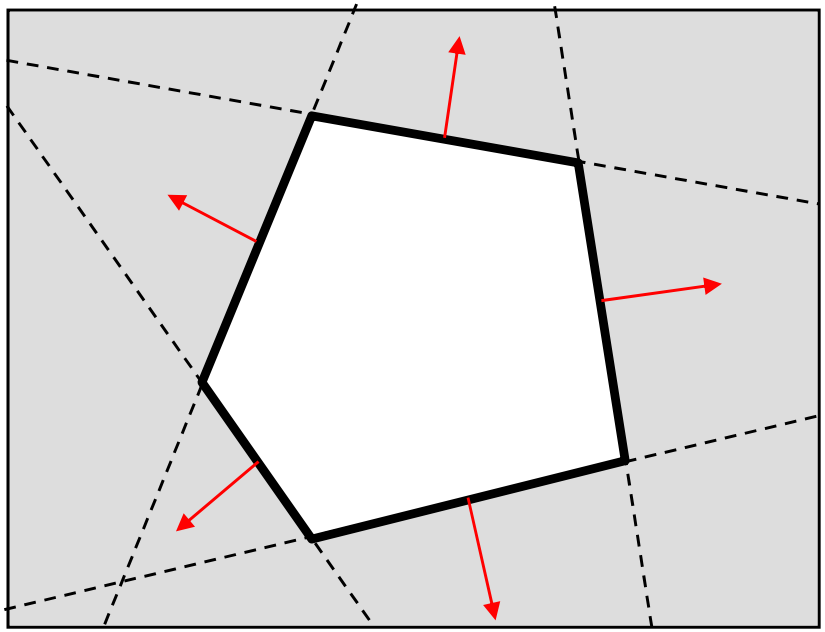
- **If $\vec{n}$ is normalized then d gives the distance of the line (plane) from the origin along $\vec{n}$**

KAIST

Lines and Planes

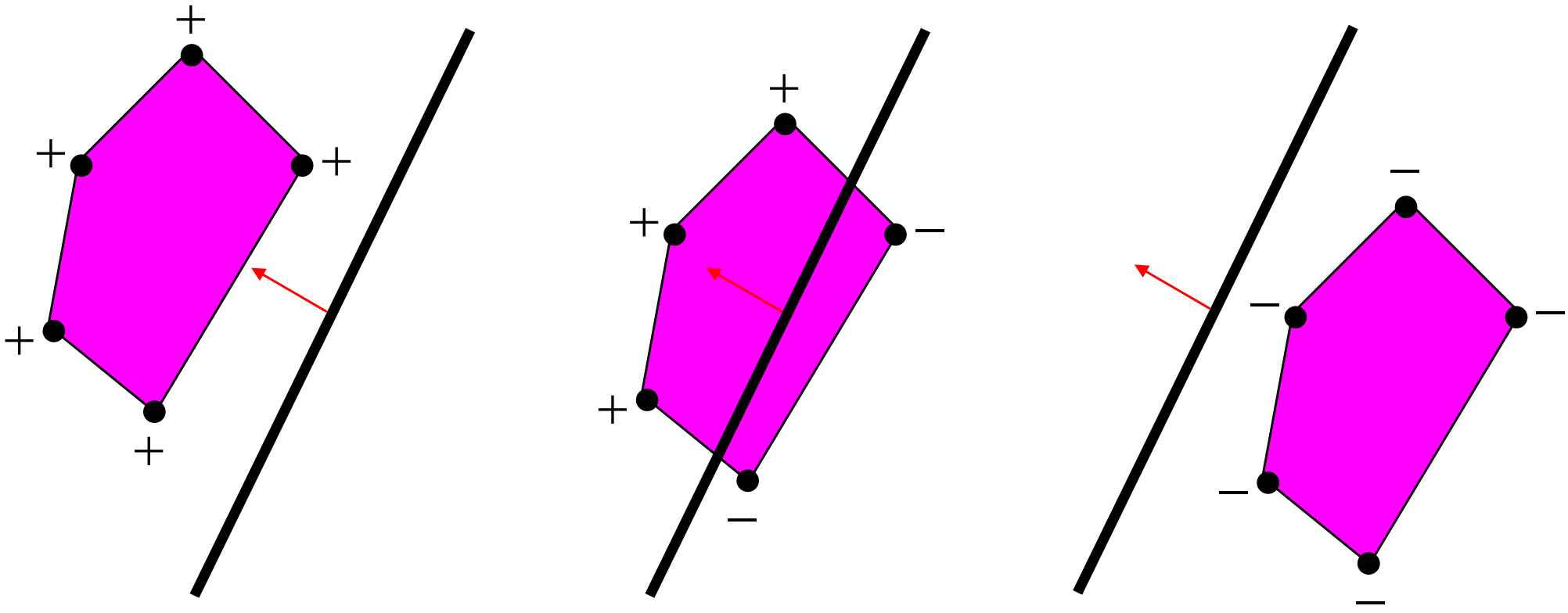
- **Lines (planes) partition 2D (3D) space:**
  - **Positive and negative *half-spaces***
- **The intersection of negative half-spaces defines a convex region**





KAIST

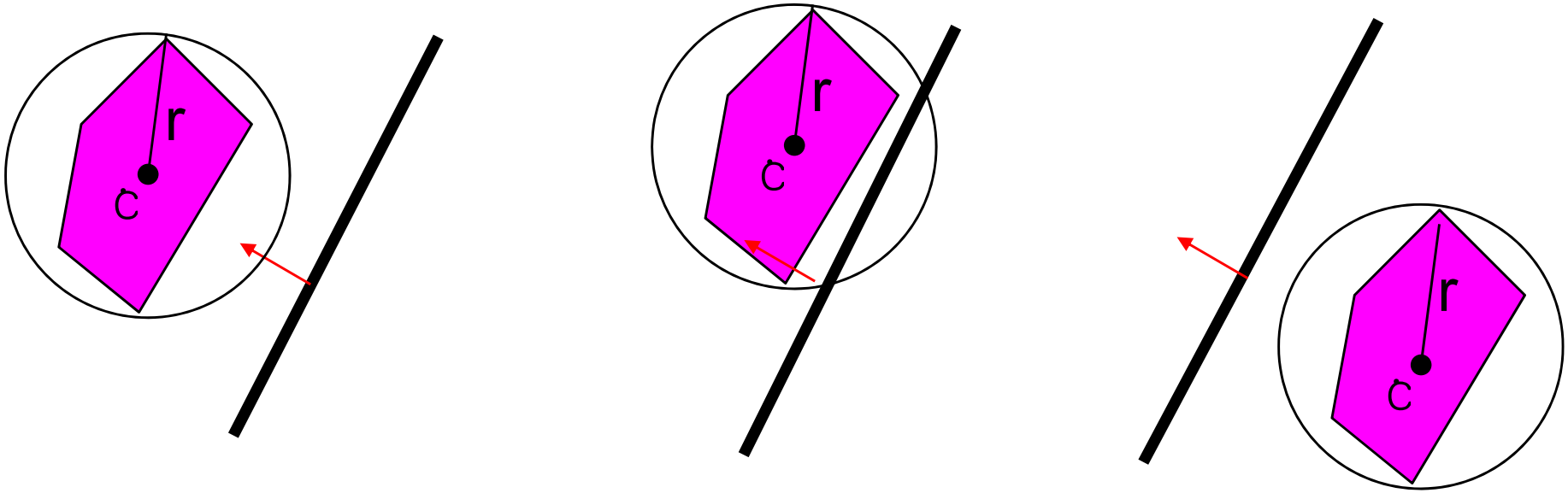
Testing Objects for Containment



Outside Straddling Inside

KAIST

Conservative Testing



Outside

$\bar{l} \cdot \dot{c} > r$

Indeterminate

$-r < \bar{l} \cdot \dot{c} < r$

Inside

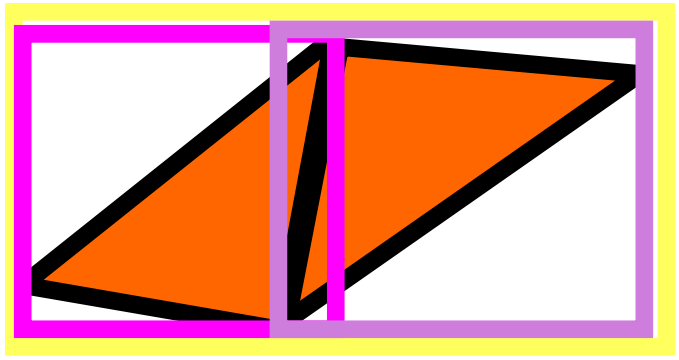
$\bar{l} \cdot \dot{c} < -r$

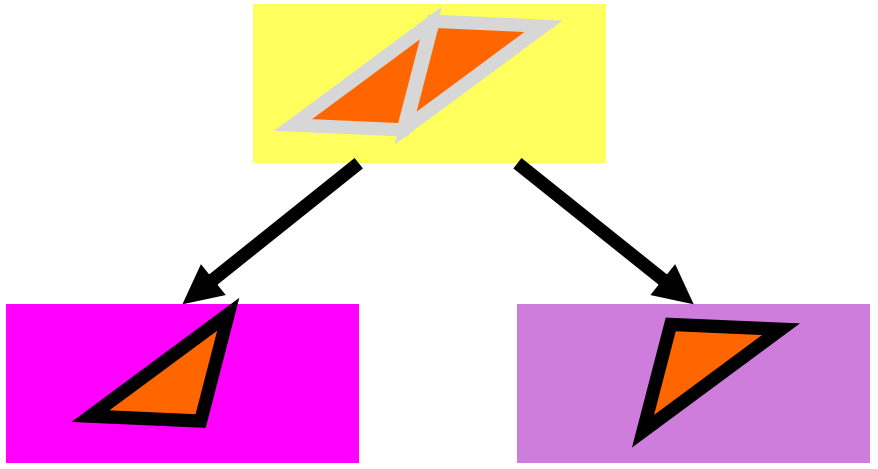
- **Use cheap, conservative bounds for trivial cases**
- **Can use more accurate, more expensive tests for ambiguous cases if needed**

KAIST

Hierarchical Culling

- **Bounding volume hierarchies (BVHs)**
  - **Accelerate culling by rejecting/accepting entire sub-trees at a time**
  - **Uses axis-aligned bounding boxes**
  - **Also known as object partitioning hierarchies**

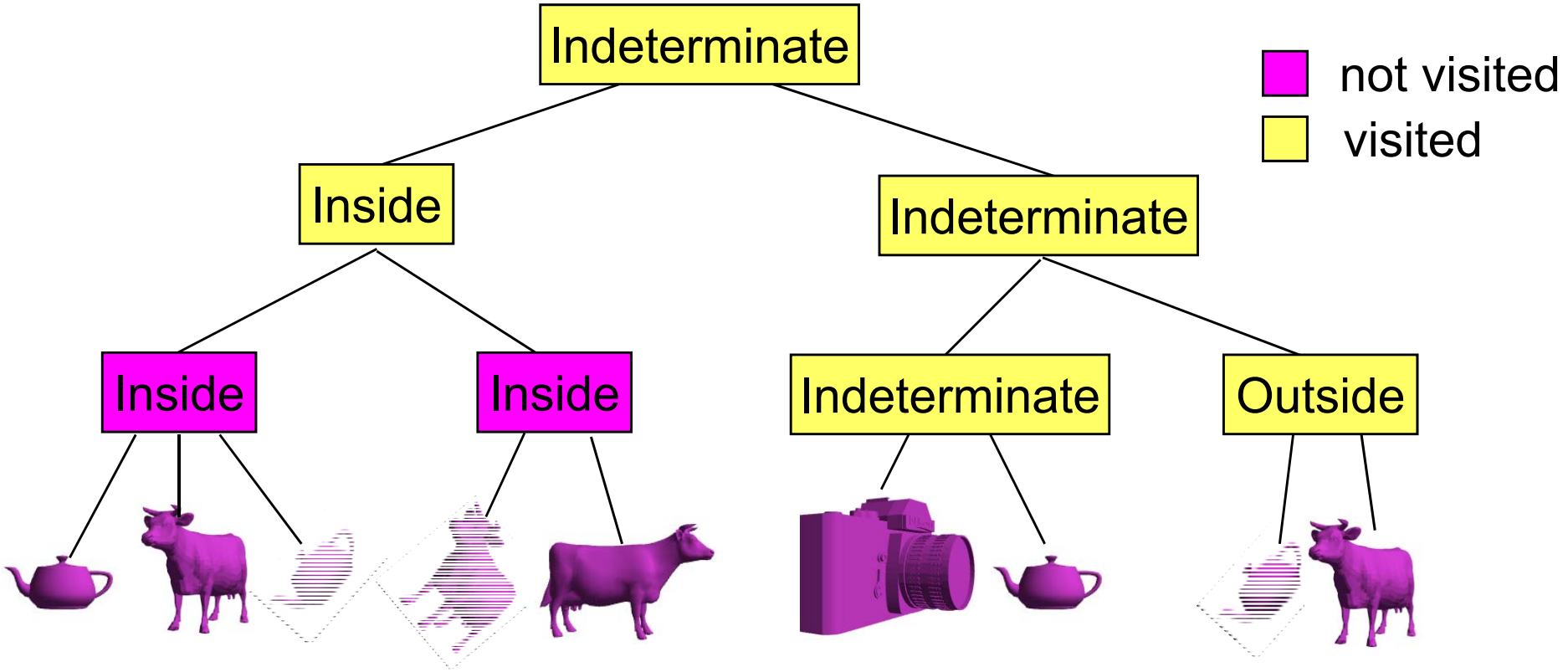




A BVH

Hierarchical Culling w/ BVH

- **Simple traversal algorithm:**
  while(node is indeterminate) recurse on children



KAIST

Test-Of-Time 2006 Award

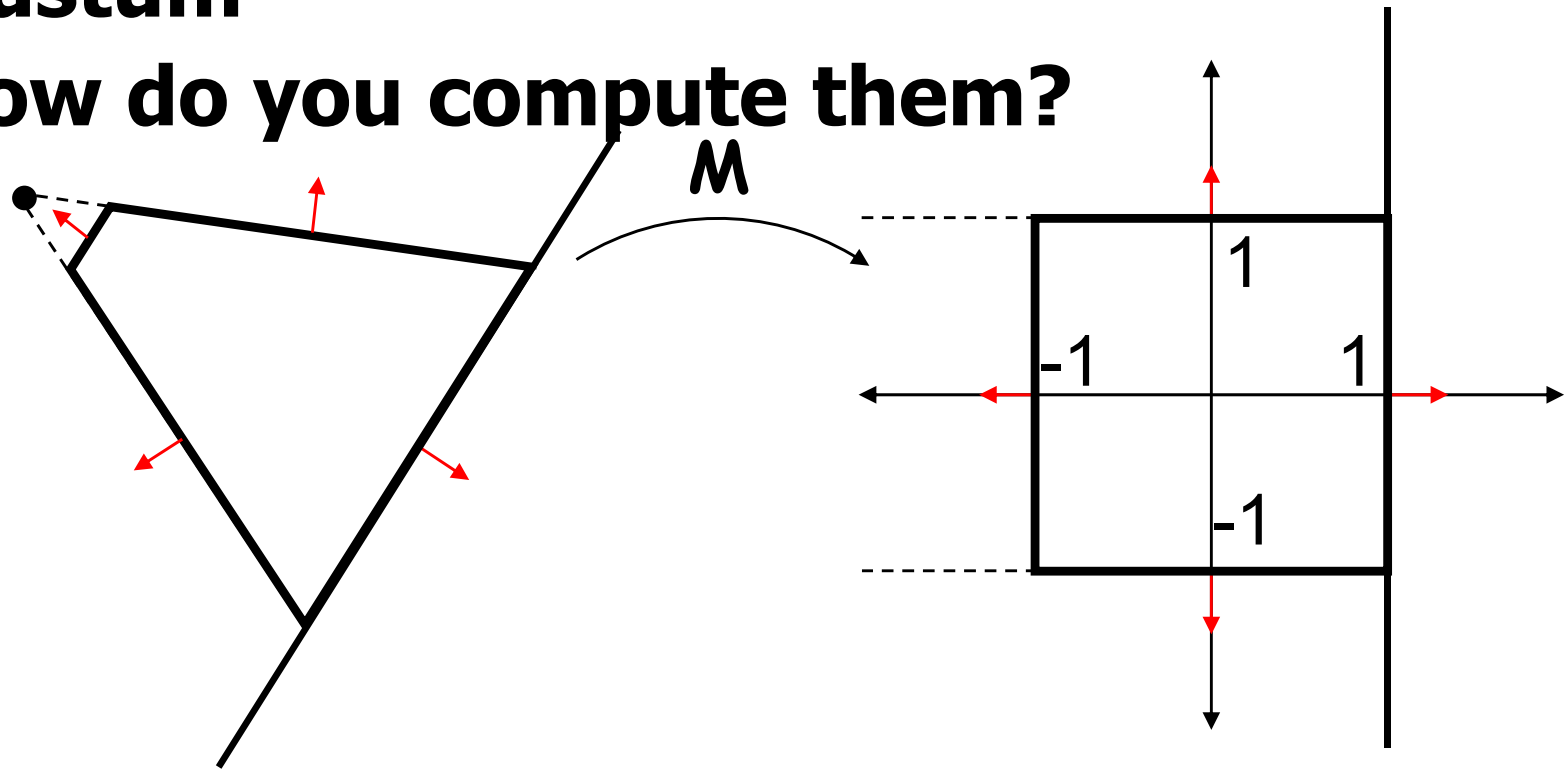

RT-DEFORM: Interactive Ray Tracing of Dynamic Scenes using BVHs

Christian Lauterbach, Sung-eui Yoon, David Tuft, Dinesh Manocha

IEEE Interactive Ray Tracing, 2006

View Frustum Culling

- Test objects against planes defining view frustum
- How do you compute them?



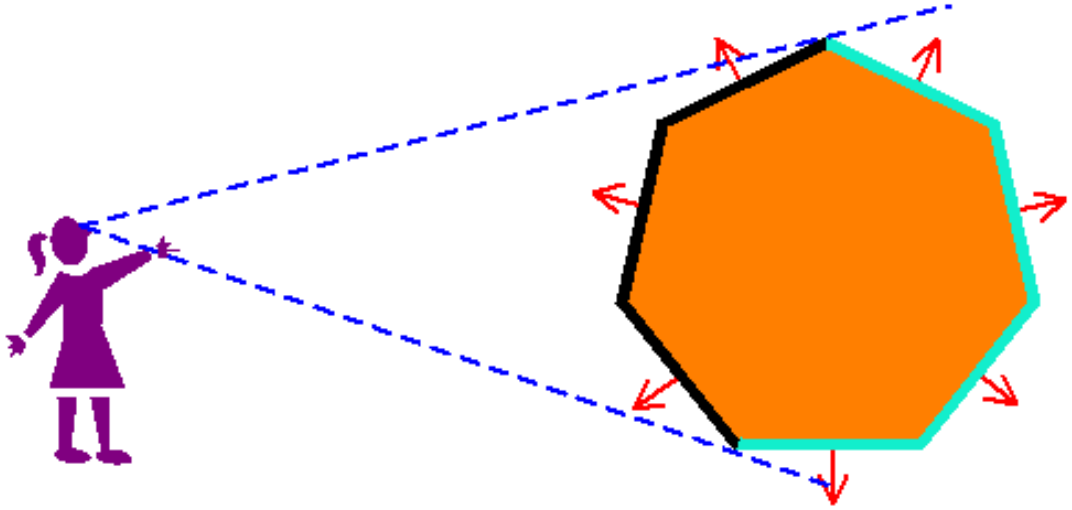
$$\bar{l} = [1 \quad 0 \quad -1]$$

- Other planes can be computed similarly

KAIST

Back-Face Culling

- Special case of occlusion - convex self-occlusion
  - For closed objects (has well-defined inside and outside) some parts of the surface must be blocked by other parts of the surface
- Specifically, the backside of the object is not visible

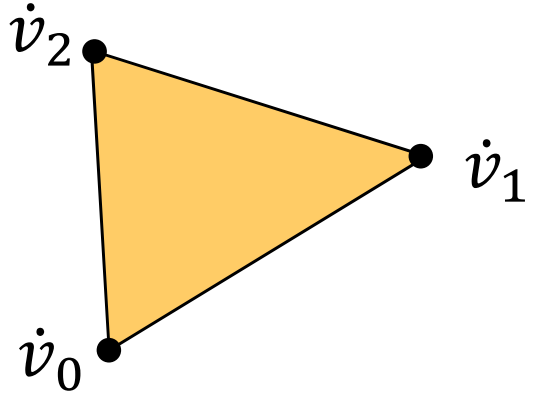


KAIST

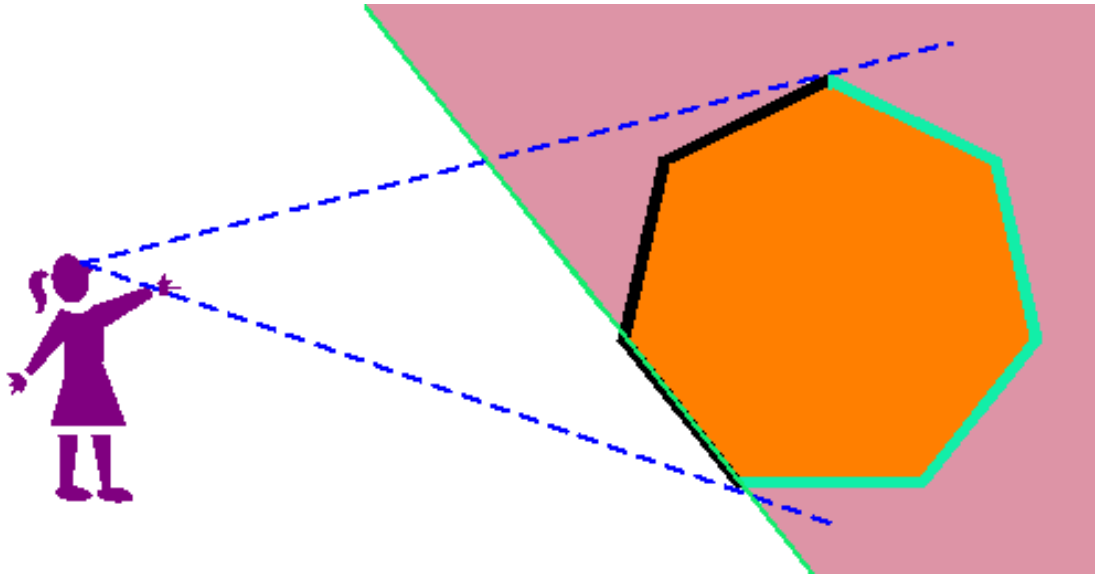
Face Plane Test

- **Compute the plane for the face:**

$$\vec{n} = (\dot{v}_1 - \dot{v}_0) \times (\dot{v}_2 - \dot{v}_0)$$
$$d = \vec{n} \cdot \dot{v}_0$$



- **Cull if eye point in the negative half-space**



KAIST

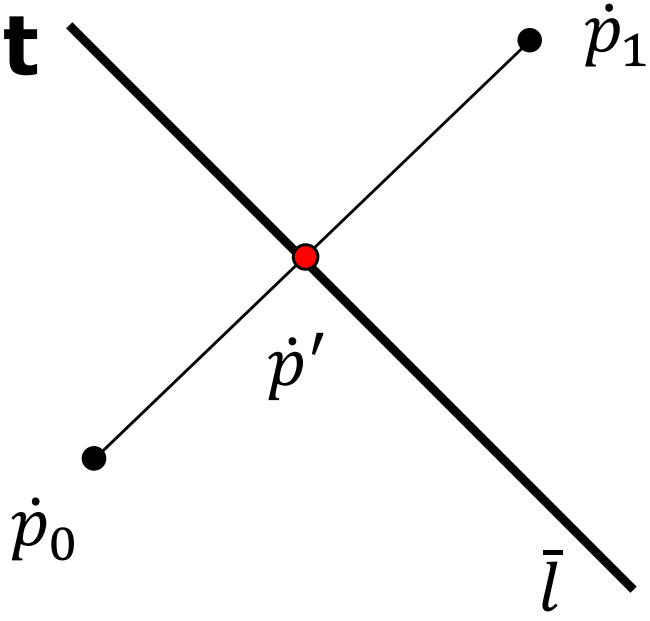
Clipping a Line Segment against a Line

- First check endpoints against the plane
  - If they are on the same side, no clipping is needed
- Interpolate to get new point

$$\dot{p}' = \dot{p}_0 + t(\dot{p}_1 - \dot{p}_0) \qquad \bar{l} \cdot \dot{p}' = 0$$

$$\bar{l} \cdot (\dot{p}_0 + t(\dot{p}_1 - \dot{p}_0)) = 0$$

$$t = \frac{-(\bar{l} \cdot \dot{p}_0)}{\bar{l} \cdot (\dot{p}_1 - \dot{p}_0)}$$



- Vertex attributes interpolated the same way

KAIST

Clipping in the Pipeline - Too much details; skipped

$(x, y, z, 1)$

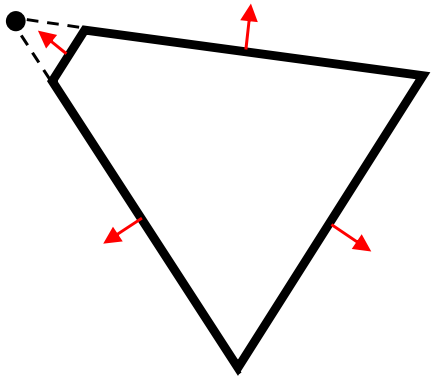
View/Projection transformations

(x', y', z', w')

Homogeneous divide

$(x'/w', y'/w', z'/w', 1)$

Option 1

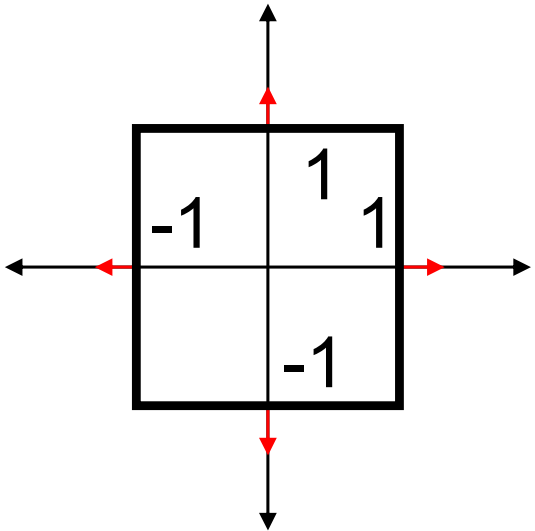


Option 2

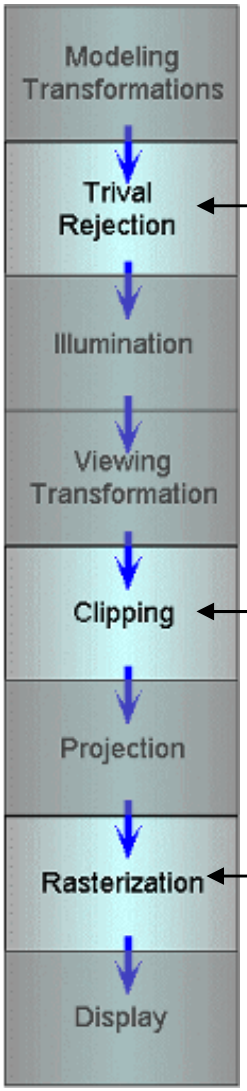
What is the best place?

- Option 2 (clip space)

Option 3 (NDC space)



Culling and Clipping in the Rendering Pipeline



View frustum culling, but performed in the application level

View frustum clipping and back-face culling can be done here

Back-face culling done in setup phase of rasterization

KAIST

Class Objectives were:

- Understand clipping and culling
- Understand view-frustum, back-face culling, and hierarchical culling methods
- Know various possibilities to perform culling and clipping in the rendering pipeline

Homework

- **Go over the next lecture slides before the class**
- **Watch 2 SIGGRAPH videos and submit your summaries before every Mon. class**
- **Submit your questions two times during the whole semester**

Next Time

- Rasterizing triangles
  - Triangulating a polygon
  - Interpolating parameters