

The Visibility Problem in Walkthrough Applications

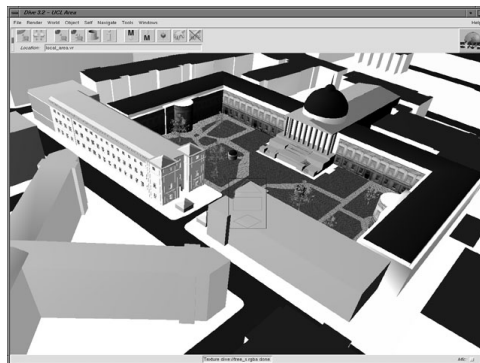
Daniel Cohen-Or and Shuly Lev-Yehudi

Computer Science Department
Tel-Aviv University

Virtual Reality Applications

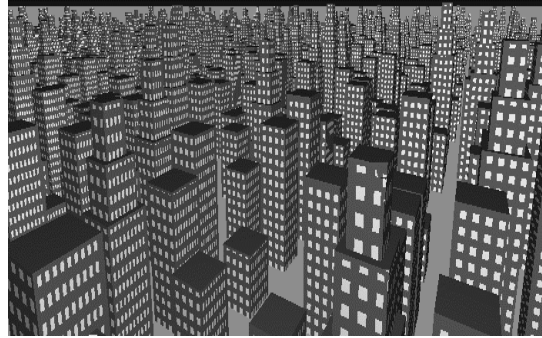
- ◆ The user "walks" interactively in a virtual polygonal environment.
Examples: model of a city, museum, mall, architectural design

The goal: to render an updated image for each view point and for each view direction in interactive frame rate

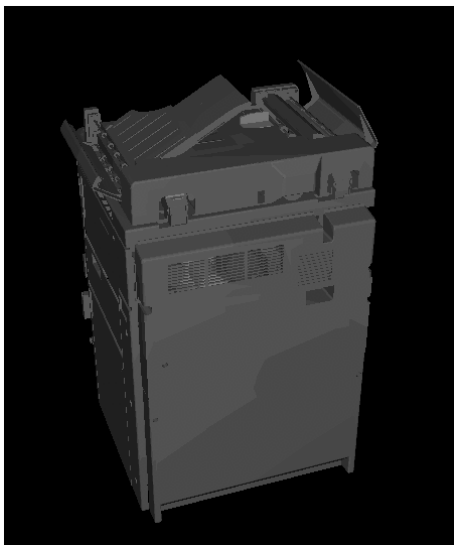


The Model

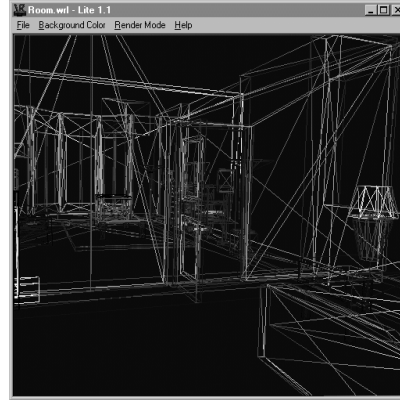
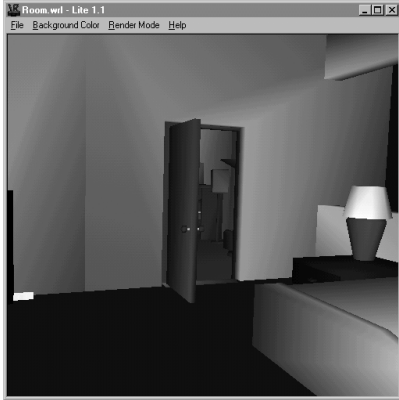
- ◆ Composed of 3D geometric objects - Lots of simple parts
- ◆ Large and complex - hundreds thousands or even millions of polygons



Copying Machine



Indoor scene



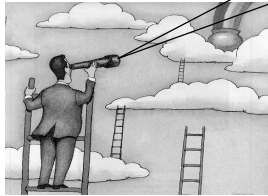
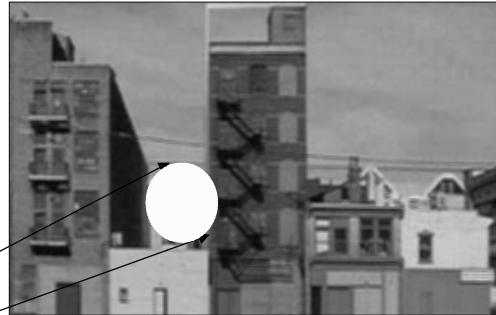
The Visibility Problem

- ◆ Selecting the (exact?) set of polygons from the model which are visible from a given viewpoint



The Visibility Problem

- ◆ Average number of polygons, visible from a viewpoint, is **much** smaller than the model size



The Visibility Problem

A small change of the viewpoint might causes large changes in the visibility

Culling

Avoid processing polygons which contribute nothing to the rendered image

A primitive can be culled by:

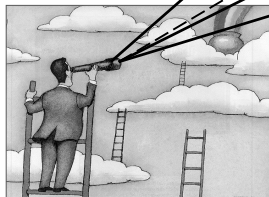
View
Frustum
Culling

Back Face
Culling

Occlusion
Culling

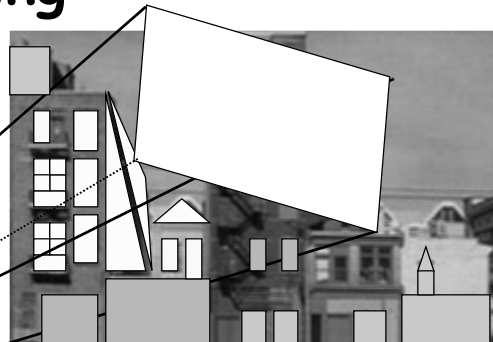
View Frustum Culling

Pass through scene primitives entirely inside frustum



Modify remaining primitives so as to pass through only the portion inside view frustum

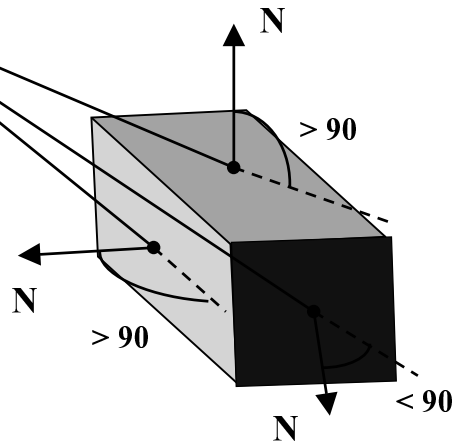
Remove primitives entirely outside the field of view



Backface Culling

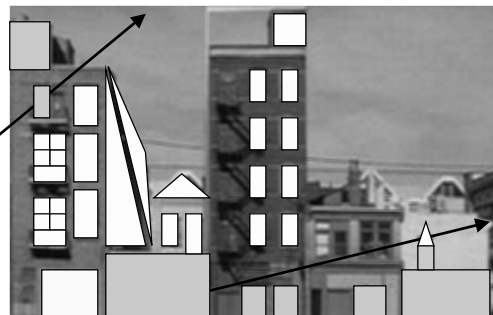
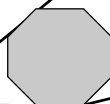
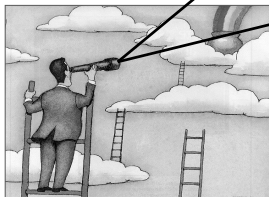


cull away polygons whose front sides face away from the viewer



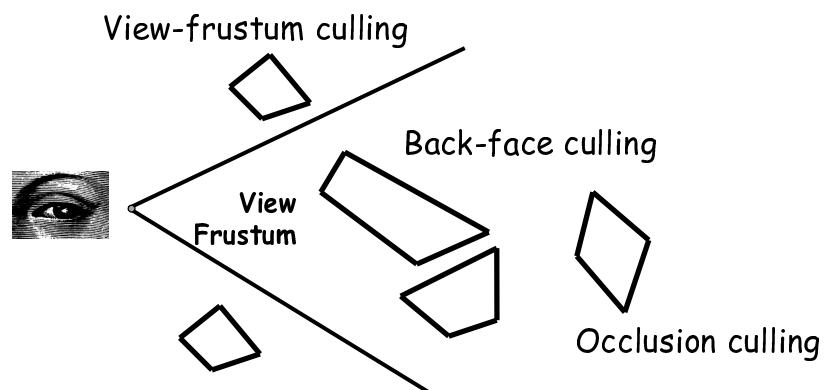
Occlusion Culling

- ◆ Cull the polygons occluded by other objects in the scene



In typical urban scene most of the objects are hidden

Visibility Culling



Conservative Visibility

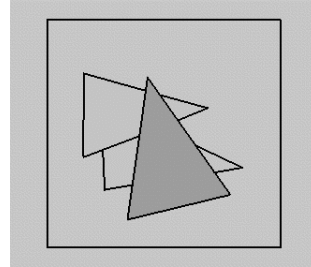
May classify invisible object as visible
but may never classify visible object as
invisible

Conservative Set

Includes at least all the visible objects
plus maybe some additional invisible
objects

Hidden Surface Removal

- ◆ For most interesting scenes and viewpoints, some polygons will overlap
- ◆ Somehow, we must determine which portion of each polygon is visible (or closer) to eye



- Algorithms:
- Painter (Depth-Sort)
- Z-Buffer
- Ray Casting

Image-Space vs. Object-Space

◆ Image Space

- ◆ for each pixel in the image
- ◆ {
 - ◆ determine the object closest to the viewer that is pierced by the projector through the pixel
 - ◆ draw the pixel in the appropriate color
- ◆ }

Ray Casting, Z-Buffer

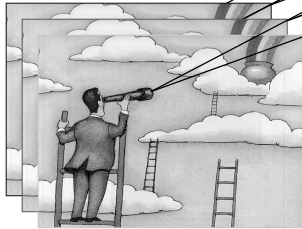
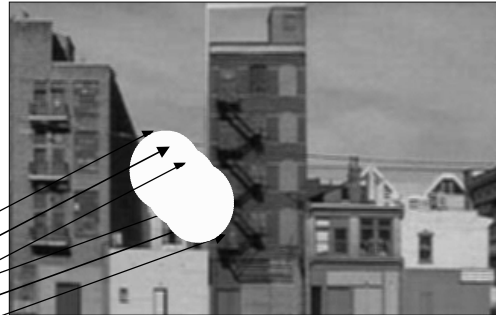
◆ Object Space

- ◆ for each object in the scene
- ◆ {
 - ◆ determine the parts of the object whose view is occluded by the other parts of it or other object
 - ◆ draw those parts in the appropriate color
- ◆ }

Painter

Global Visibility

Compute the set of all polygons visible from every possible viewpoint from a region (view-cell)

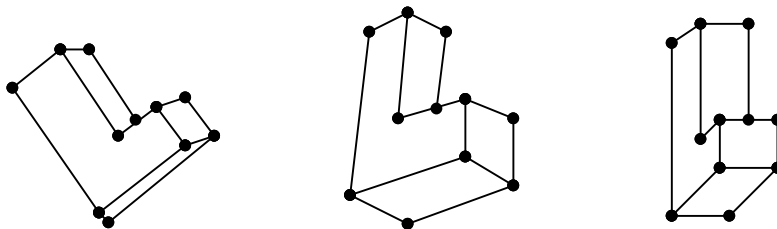


The whole environment is partitioned into view-cells

The Aspect Graph

◆ ISG – Image Structure graph

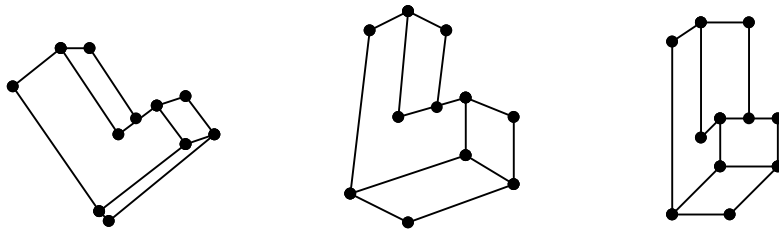
The planner graph, defined by the outlines of an image, created by projection of a polyhedral object, in a certain view direction



The Aspect Graph (cont.)

◆ Aspect

Two different view directions of an object have the same aspect iff the corresponding ISG graphs are isomorphic



The Aspect Graph (Cont.)

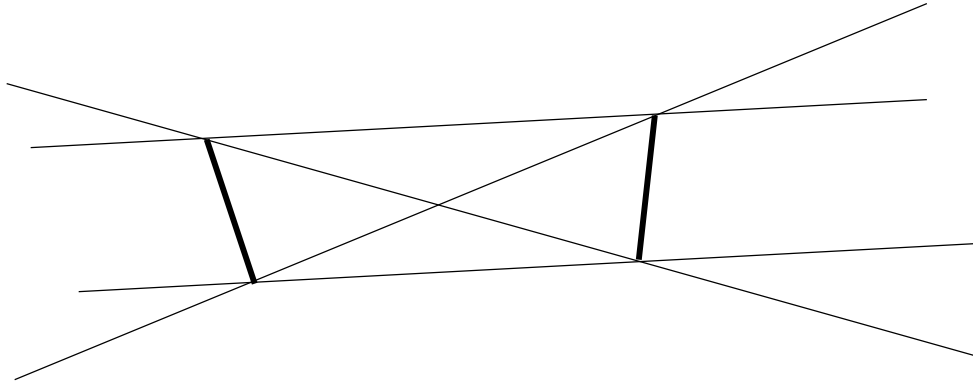
◆ VSP – Visibility Space Partition

- ◆ Partitioning the viewspace into maximal connected regions in which the viewpoints have the same view or aspect
- ◆ All viewpoints within a region has one representative ISG

◆ Visual Event

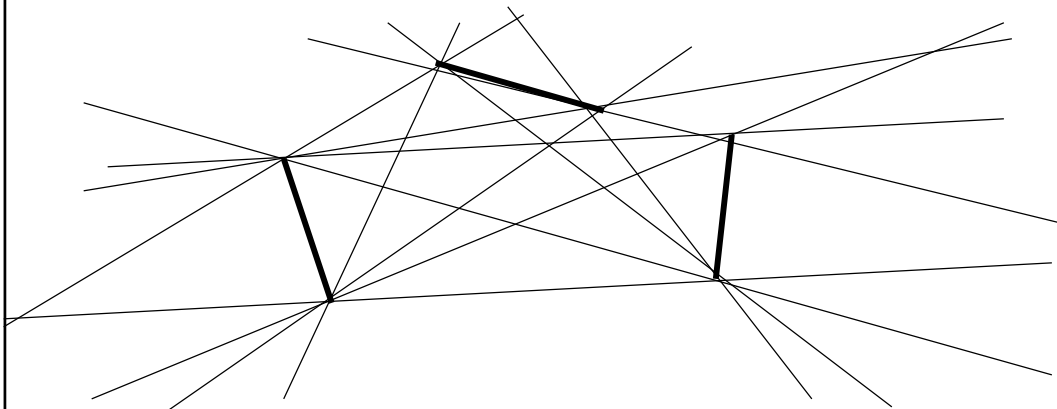
A boundary of a VSP region called a VE for it marks a change in visibility

Aspect graph (cont.)



2 polygons - 12 aspect regions

Aspect graph (cont.)



3 polygons - "many" aspect regions

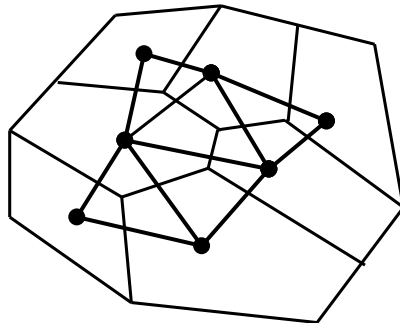
Different Aspects, but same visibility sets



The Aspect Graph (Cont.)

◆ Aspect Graph

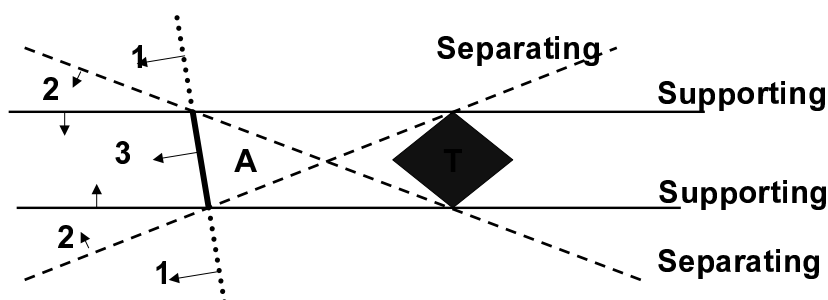
- ◆ A vertex for each region of the VSP
- ◆ An edge connecting adjacent regions



The Aspect Graph (Cont.)

- ◆ Regions of the VSP are not maximal but maximal connected regions.
Two non adjacent regions can have the same aspect
- ◆ The set of visible polygons remains constant when moving within the region of the VSP
- ◆ Different regions can have equal sets of visible polygons
- ◆ The number of regions is $\theta(n^9)$ for the case of a VSP with non-convex polyhedra and perspective projection

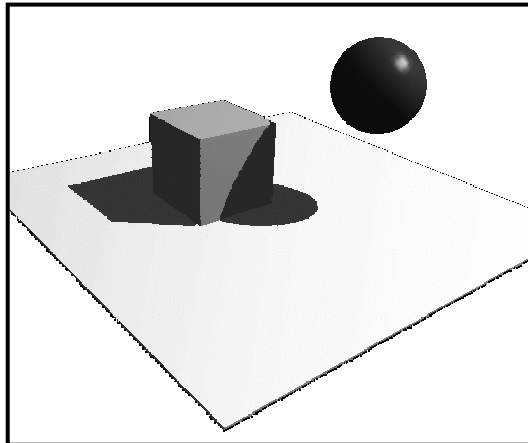
Supporting & Separating Planes



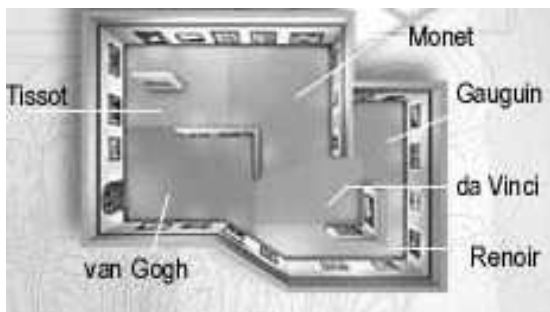
T is not occluded by A - 1
 T is partially occluded by A - 2
 T is completely occluded by A - 3

A - occluder
 T - occludee

Visibility from the light source



The Art Gallery Problem



Classification of visibility algorithms

- Exact vs. Approximated
- Conservative vs. Exact
- Precomputed vs. Online
- Point vs. Region
- Image space vs. Object space

Thanks for Listening