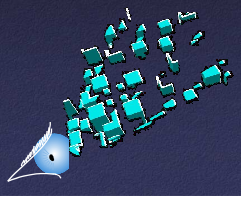
Visibility Computation

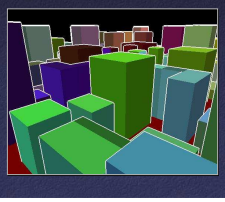
COS 526, Fall 2006

Acknowledgments: Frédo Durand, Tom Funkhouser

Visibility

- Compute part of scene visible from
  - Point
  - Surface
  - Volume





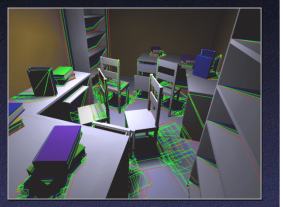
Visibility Applications

- Computer graphics
  - Hidden surface removal
  - Shadow computation
  - Occlusion culling
  - Global illumination
  - Image-based modeling
  - Good viewpoint selection
- Computational Geometry
  - Visibility graphs
  - Art galleries

Funkhouser

Visibility Applications

- Computer graphics
  - Hidden surface removal
  - Shadow computation
  - Occlusion culling
  - Global illumination
  - Image-based modeling
  - Good viewpoint selection
- Computational Geometry
  - Visibility graphs
  - Art galleries



Drettakis

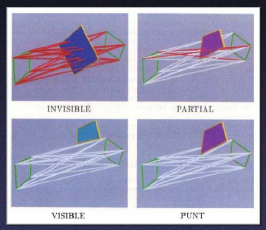
Visibility Applications

- Computer graphics
  - Hidden surface removal
  - Shadow computation
  - Occlusion culling
  - Global illumination
  - Image-based modeling
  - Good viewpoint selection
- Computational Geometry
  - Visibility graphs
  - Art galleries

Luebke

Visibility Applications

- Computer graphics
  - Hidden surface removal
  - Shadow computation
  - Occlusion culling
  - Global illumination
  - Image-based modeling
  - Good viewpoint selection
- Computational Geometry
  - Visibility graphs
  - Art galleries



$$L(x',x'') = L_e(x',x'') + \int_S f_r(x,x',x'')L(x,x')V(x,x')G(x,x')dA$$

Teller

Visibility Applications

- Computer graphics
  - Hidden surface removal
  - Shadow computation
  - Occlusion culling
  - Global illumination
  - Image-based modeling
  - Good viewpoint selection
- Computational Geometry
  - Visibility graphs
  - Art galleries

Visibility Applications

- Computer graphics
  - Hidden surface removal
  - Shadow computation
  - Occlusion culling
  - Global illumination
  - Image-based modeling
  - Good viewpoint selection
- Computational Geometry
  - Visibility graphs
  - Art galleries

Visibility Applications

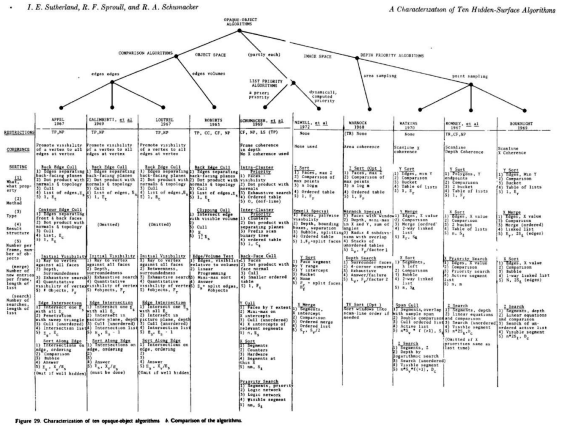
- Computer vision
  - Object recognition
  - 3D scene reconstruction
  - Next best view planning
- Robotics
  - Motion planning
  - Visibility-based pursuit-evasion
  - Self-localization

Visibility Approaches

- Conceptually, focus on visibility everywhere
  - For all points p and q, is p visible from q?
  - Classification of *rays* (or, more precisely, segments)
- Want practical data structures, algorithms
- Strategies:
  - Compute locally (one viewpoint at a time)
  - Discretize
  - Focus on combinatorial structure:
    regions of p and q may have the same visibility

Visibility Strategy Examples

- From-region, conservative
  - Cells and portals
- Local (one point): hidden surface removal
  - Discretized in image, object space, ray space
- One object, any viewpoint, combinatorial structure
  - Aspect graphs
- Occlusion due to one object from one viewpoint
  - Umbra and penumbra volumes
- Global visibility
  - Visibility graphs (polyhedral scene, discretized at vertices)
  - Visibility skeleton (combinatorial structure of ray space)



I. E. Sutherland, R. F. Sproull, and R. A. Schumacker — A Characterization of Ten Hidden-Surface Algorithms

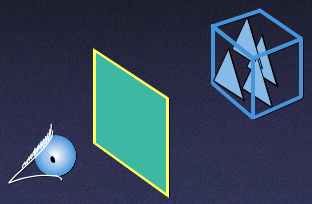
Figure 29. Characterization of ten opaque-object algorithms. A Comparison of the algorithms.

Hidden Surface Removal Methods

- Image-space
  - Z-buffer
  - Scan-line
  - Warnock subdivision
- Object-space
  - Depth-sort
  - Weiler-Atherton
  - BSP
- Line-space
  - Ray casting

Hidden Surface Removal

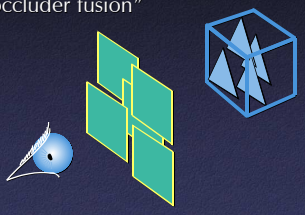
- Occlusion by a single occluder



Durand

Hidden Surface Removal Problem

- Cumulative occlusion by multiple occluders: "occluder fusion"



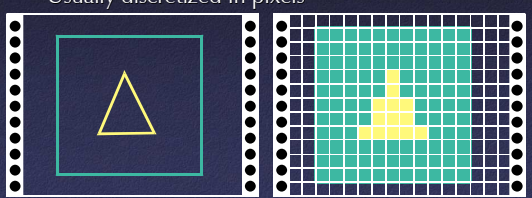
Durand

Hidden Surface Removal Problem

- Sorting w.r.t. to distance (Painter's algorithm) is not enough

Image-Space

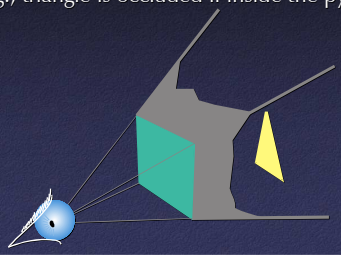
- Computation performed in image plane
- E.g. is triangle inside rectangle?
- Usually discretized in pixels



Durand

Object-Space

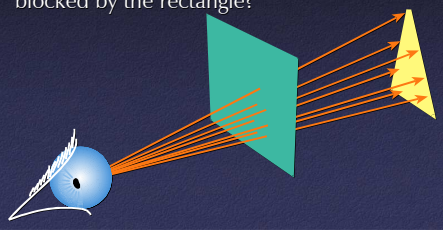
- 3D space where the scene is defined
- E.g., triangle is occluded if inside the pyramid



Durand

Line-Space

- Visibility expressed in terms of rays
- E.g. are all rays between the eye and the triangle blocked by the rectangle?



Durand

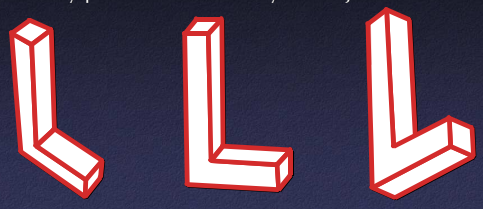
Typical Advantages and Drawbacks

- Image-space
  - \+ Robust, easier to code, occluder fusion, can use hardware
  - – Limited to one viewpoint, aliasing, needs hardware
- Object-space
  - \+ Precision, can handle from-region visibility
  - – Often robustness problems, occluder fusion is harder
- Line space
  - \+ Natural space, simple atomic operation (ray-casting)
  - – 4D, often requires approximation, or too complex

Durand

Visibility of Faces of an Object

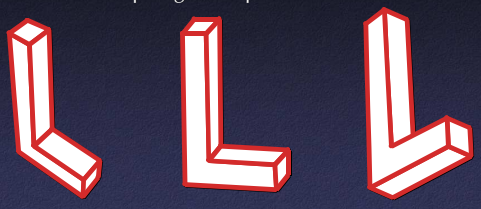
- Many possible views of any 3D object



Durand

Visibility of Faces of an Object

- Focus on topological ("qualitative") differences



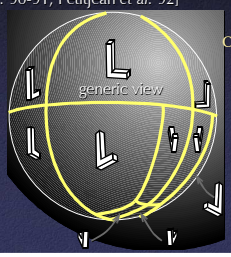
Qualitatively equivalent (same aspect)

Qualitatively different (different aspect)

Durand

Aspect Graph

- Characterization of the set of possible views of an object
  - – [Koenderink and Van Doorn 79, Plantinga and Dyer 90, Gigus *et al.* 90-91, Petitjean *et al.* 92]



changes of view

Durand

Aspect Graph Size

- For a polygonal scene with n edges

	orthographic	perspective
convex	$O(n^2)$	$O(n^3)$
non-convex	$O(n^6)$	$O(n^9)$

- More typically: < $O(n^4)$ and $O(n^6)$

Durand

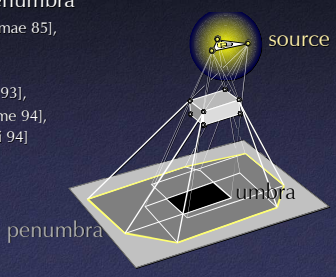
Global Visibility

- Soft shadows
  - For each point, visible portion of the source
- Mutually visible objects
  - Sampling of the objects?
- Set of visible objects from a volume
- Qualitative visibility
  - Changes in visibility
    (disappearance of objects, limits of shadows, etc.)

Durand

Visibility from Polygon

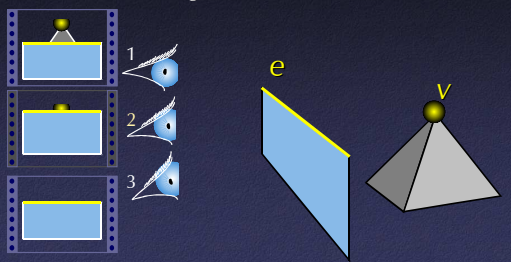
- Umbra and Penumbra
  - [Nishita et Nakamae 85], [Heckbert 92], [Teller 92], [Lischinski *et al.* 93], [Drettakis & Fiume 94], [Stewart et Ghali 94]



Durand

Visual Events

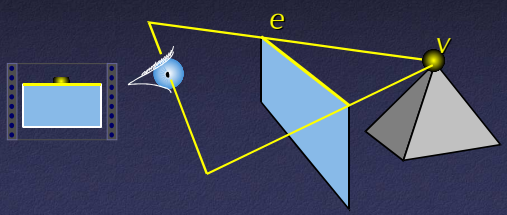
- Qualitative changes in view



Durand

EV Events

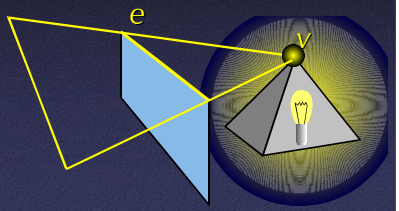
- "EV" event: object begins to occlude vertex
- "Wedge" defined by vertex and edge



Durand

EV Events

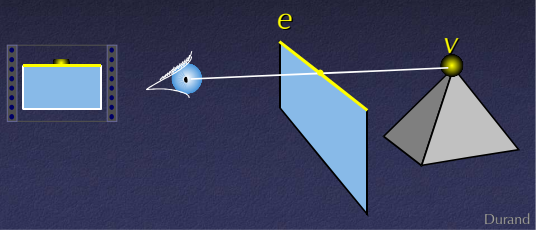
- Limits of umbra



Durand

Critical Lines

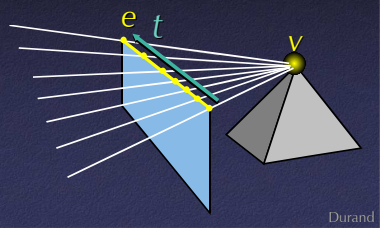
- Line going through e and v



Durand

Critical Lines

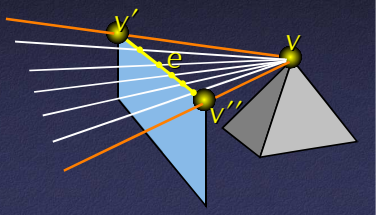
- 1D set of lines going through e and v (1 degree of freedom)



Durand

Extremal Stabbing Lines

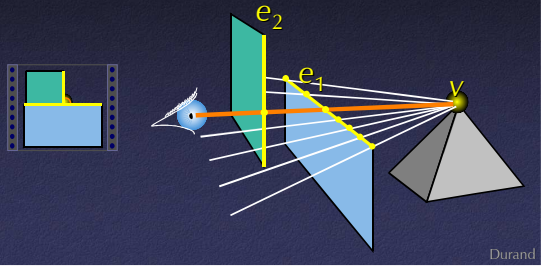
- Extremity: extremal stabbing line (VV event) (0 degree of freedom)



Durand

Extremal Stabbing Lines

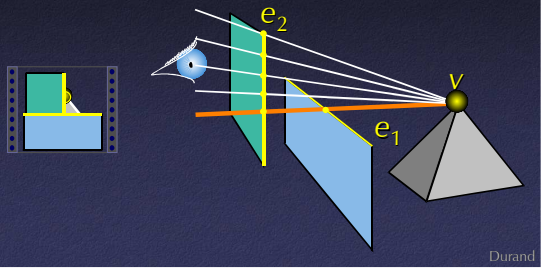
- Type *VEE* (0 degree of freedom)



Durand

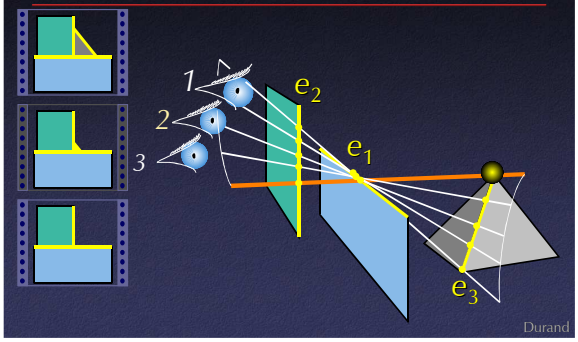
Adjacent Critical Line Set

- Generated by e_2; same extremity ve_1e_2



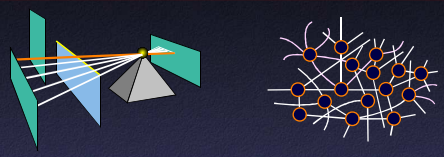
Durand

Triple-Edge Events



Durand

Visibility Skeleton



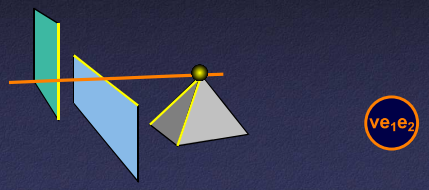
Scene | Graph in line space

- Encodes adjacencies of extremal stabbing lines and critical line sets

Durand

Visibility Skeleton

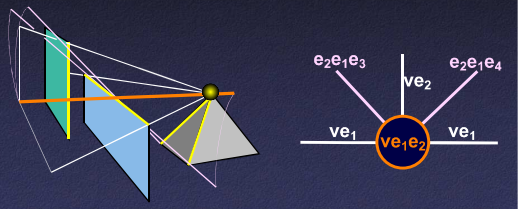
- Extremal stabbing line = Node



Durand

Visibility Skeleton

- Extremal stabbing line = Node
- Critical line set = Arc



Durand

Visibility Skeleton

- Idea:
  - Graph representation of visual events
- Complexity
  - Memory: $O(n^4)$ in theory, n^2 observed
  - Time: $O(n^5)$ in theory, $n^{2.4}$ observed
- Results
  - Scenes up to 1500 polygons
    - 1.2 million nodes
    - 32 minutes for computation

Durand

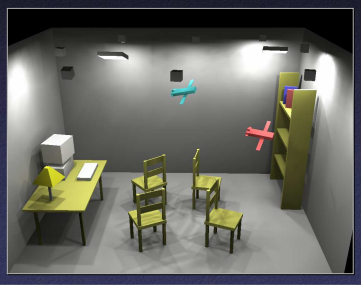
Radiosity with Visibility Skeleton

- Exact computation of form-factors
  - "How much of each polygon visible from a point"
- Discontinuity meshing
  - Wcene subdivision along shadow boundaries
  - Also for indirect lighting

Durand

Radiosity with Visibility Skeleton

- 492 polygons : 10 minutes 23 seconds



Durand

Radiosity with Visibility Skeleton

With skeleton
10 minutes 23 seconds

[Gibson 96]
1 hour 57 minutes

Durand

Summary

- Object-space visibility
  - Help understand the nature of visibility
  - Offer insights about which algorithms will work well
  - Often large time and/or space requirements
- Image-space visibility
  - Usually only for visibility from a point
  - Can be implemented with graphics hardware
  - Usual benefits/problems of image-precision computation