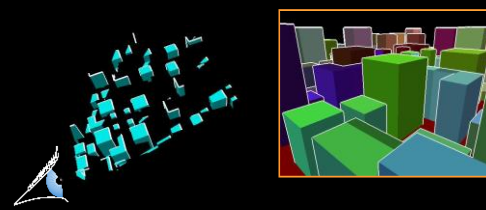
# Visibility

Tom Funkhouser
COS 526, 2012

Slides mostly by
Frédo Durand

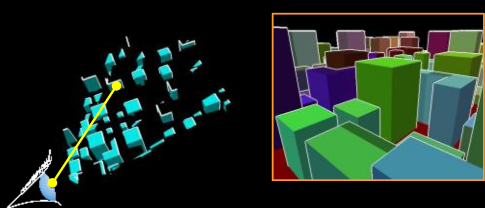
## Visibility

Compute which part of scene can be seen



## Visibility

Compute which part of scene can be seen
(i.e., line segment from source to point in scene)

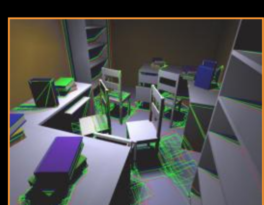


## Visibility Applications

Computer graphics
- Hidden surface removal
- Shadow computation
- Global illumination
- Occlusion culling

## Visibility Applications

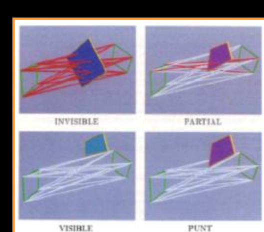
Computer graphics
- Hidden surface removal
- Shadow computation
- Global illumination
- Occlusion culling



*Drettakis*

## Visibility Applications

Computer graphics
- Hidden surface removal
- Shadow computation
- Global illumination
- Occlusion culling




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

*Teller*

## Visibility Applications

Computer graphics
- Hidden surface removal
- Shadow computation
- Global illumination
- Occlusion culling

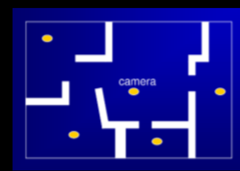
*Luebke*

## Visibility Applications

Computational Geometry
- Art galleries

Computer vision
- Object recognition
- 3D scene reconstruction
- Next best view planning

Robotics
- Motion planning
- Visibility-based pursuit-evasion
- Self-localization




## Visibility Problems

Source type
- Point, line, surface, region

Result
- Exact description of visible region for each primitive, or just tell whether each primitive is visible or not

Approximation
- Approximate, exact, conservative

Timing
- Off-line, interactive
- Amortization

## Outline

Hidden surface removal
- Visibility from viewpoint

Visibility Skeleton
- Visibility between scene elements

Aspect graph
- Visibility from any point in space

## Outline

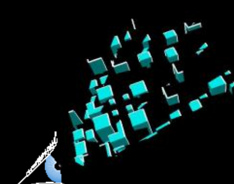
Hidden surface removal ←
- Visibility from viewpoint

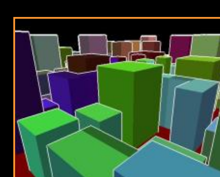
Visibility Skeleton
- Visibility between scene elements

Aspect graph
- Visibility from any point in space

## Hidden Surface Removal

Compute which part of every primitive can be seen from a point





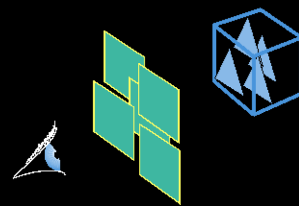
## Hidden Surface Removal

Occlusion by a single occluder

*Fredo Durand*

## Hidden Surface Removal Problem

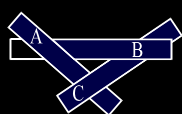
Cumulative occlusion by multiple occluders



*Fredo Durand*

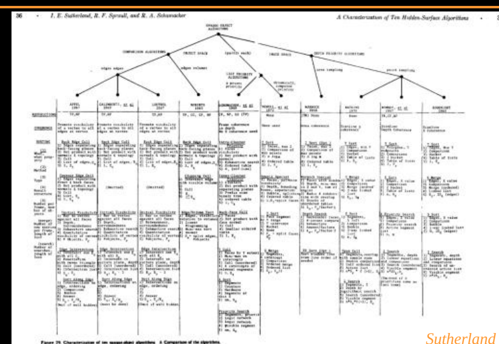
## Hidden Surface Removal Problem

Sorting according to a distance is not enough




*Fredo Durand*

## Hidden Surface Removal Methods



*Sutherland*

## Hidden Surface Removal Methods

Image-space
- Z-buffer
- Scan-line
- Warnock subdivision

Object-space
- Depth-sort
- Weiler-Atherton
- BSP

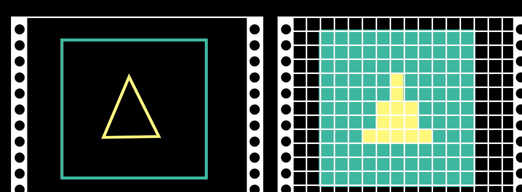
Line-space
- Ray casting

## Image-space

Computation performed in the plane of the image

E.g. is triangle inside rectangle?

Usually discretized in pixels

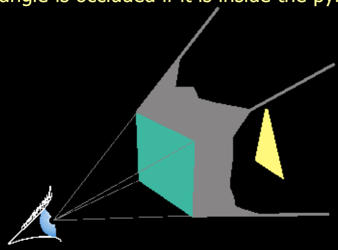


*Fredo Durand*

## Object-space

3D space where the scene is defined

E.g., triangle is occluded if it is inside the pyramid

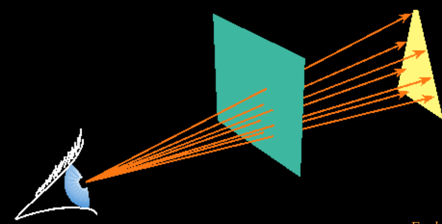


*Fredo Durand*

## Line space

Visibility expressed in terms of rays

E.g. are all rays between the eye and the triangle blocked by the rectangle?



*Fredo 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

*Fredo Durand*

## Outline

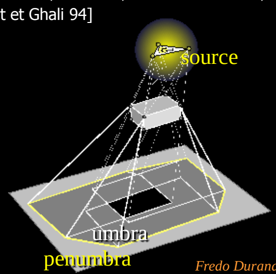
Hidden surface removal
- Visibility from viewpoint

Visibility Skeleton ←
- Visibility between scene elements

Aspect graph
- Visibility from any point in space

## Visibility from Polygon

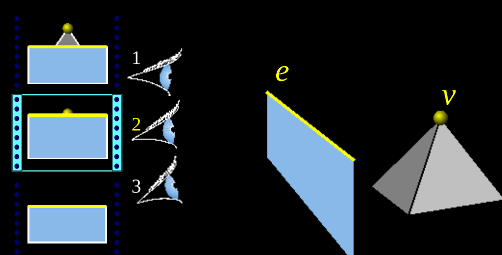
Umbra and Penumbra
- [Nishita et Nakamae 85, Heckbert 92, Teller 92, Lischinski *et al.* 93, Drettakis et Fiume 94, Stewart et Ghali 94]




*Fredo Durand*

## Visual event

Appearance-disappearance of objects (qualitative change of a view)



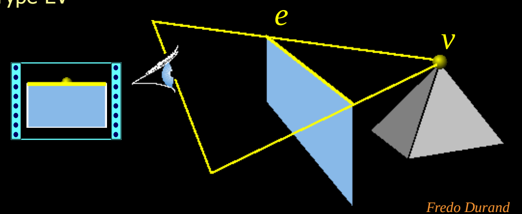

*Fredo Durand*

## Visual event

Appearance-disappearance of objects
(qualitative change of a view)

« Wedge » defined by a vertex and an edge

Type EV

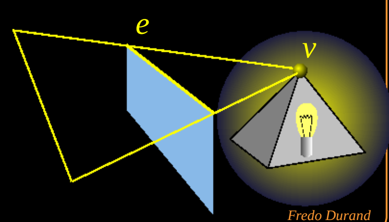



*Fredo Durand*

## Visual event

Appearance-disappearance of objects

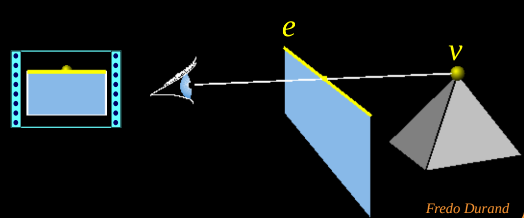
Limits of umbra




*Fredo Durand*

## Critical line

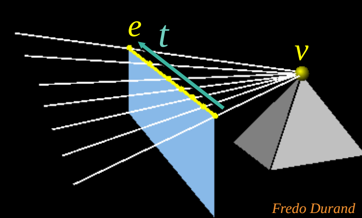
Line going through *e* and *v*




*Fredo Durand*

## Critical lines

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

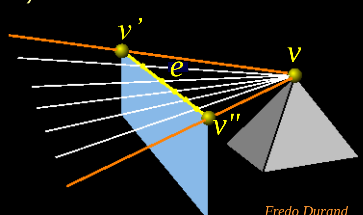



*Fredo Durand*

## Extremal stabbing line

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

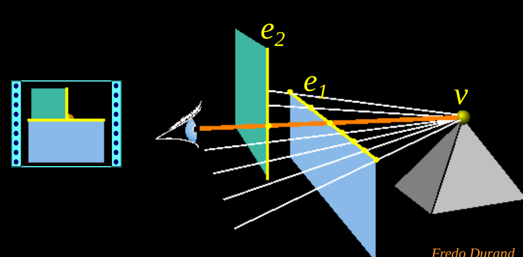
Extremity: extremal stabbing line (VV)
(0 degree of freedom)




*Fredo Durand*

## Extremal stabbing line

Type *VEE* (0 degree of freedom)

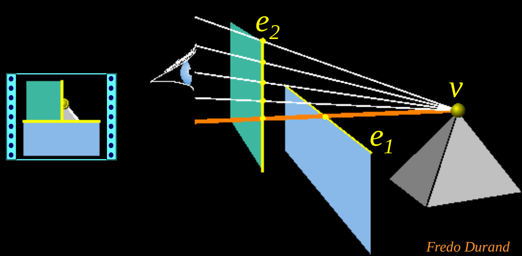



*Fredo Durand*

## Adjacent critical line set

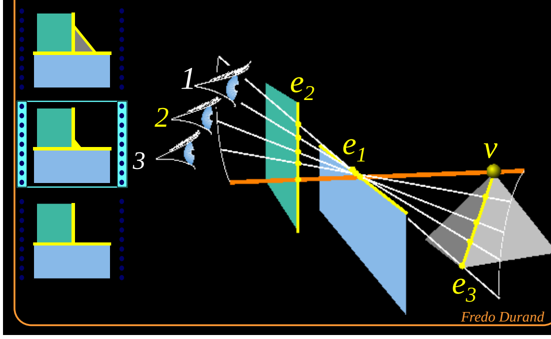
Generated by the second edge

Same extremity $ve_1e_2$



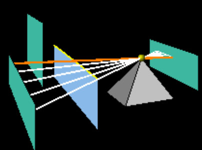

*Fredo Durand*

## Triple-edge event

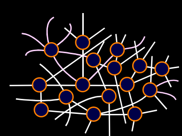



*Fredo Durand*

## Visibility skeleton



Scene



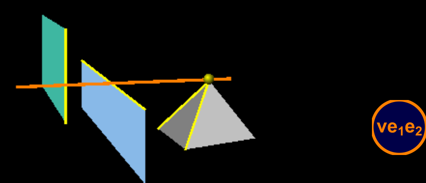
Graph in line space

Encodes adjacencies of extremal stabbing lines and critical line sets

*Fredo Durand*

## Visibility Skeleton

Extremal stabbing line = Node

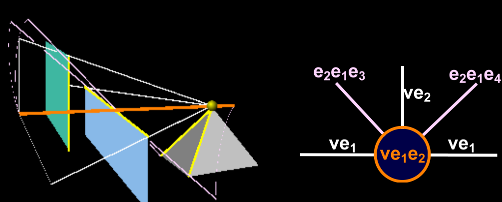



*Fredo Durand*

## Visibility Skeleton

Extremal stabbing line = Node

Critical line set = Arc




*Fredo 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

*Fredo Durand*

## Radiosity with Visibility Skeleton

Exact computation of form-factors
- point-polygon

Discontinuity meshing
- scene subdivision along shadow boundaries
- also for indirect lighting

Refinement criterion
- perceptual metric
- error estimation



## Radiosity with Visibility Skeleton

492 polygons : 10 minutes 23 seconds



## Radiosity with Visibility Skeleton



## Radiosity with Visibility Skeleton



## Outline

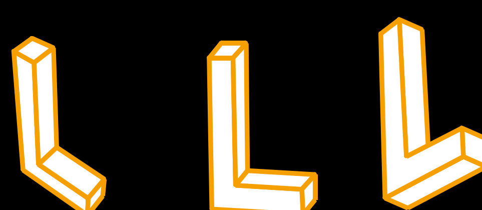
Hidden surface removal
- Visibility from viewpoint

Visibility Skeleton
- Visibility between scene elements

Aspect graph ←
- Visibility from any point in space

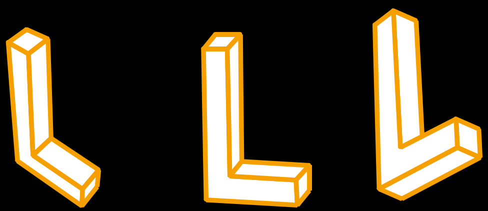
## Aspect Graph

There are many possible views of any 3D object



## Aspect Graph

Some produce topologically equivalent visibility solution



Qualitatively equivalent

(same aspect)

Qualitatively different

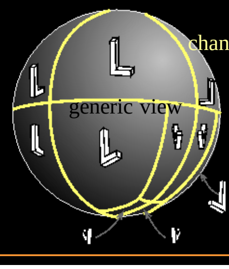
(different aspect)

*Fredo 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]




*Fredo Durand*

## Aspect Graph

For a polygonal scene with *n* edges

- $O(n^3)$ visual events
- $O(n^6)$ for orthographic views
- $O(n^9)$ for perspective views

More reasonable estimate may be

- $O(n^4)$ and $O(n^6)$

Not practical to compute and store!

*Fredo Durand*

## Summary

Object-space visibility

- Help understand the nature of visibility
- Offer insights about which algorithms will work well
- Generally difficult to code an make robust

Image-space visibility

- Usually only for visibility from a point
- Can be implemented with graphics hardware
- Usual benefits/problems of image-precision computation