Visibility

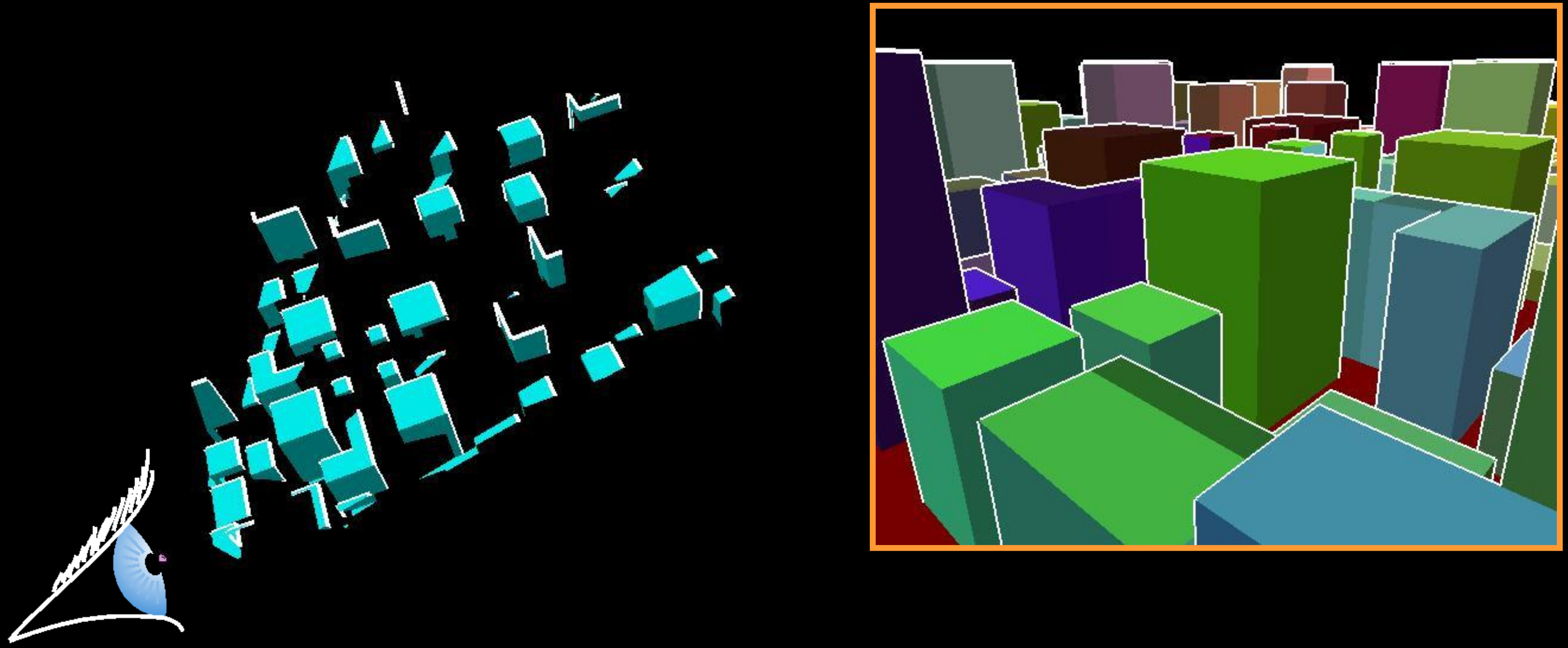
Tom Funkhouser

COS 526, Fall 2016

Slides mostly by
Frédo Durand

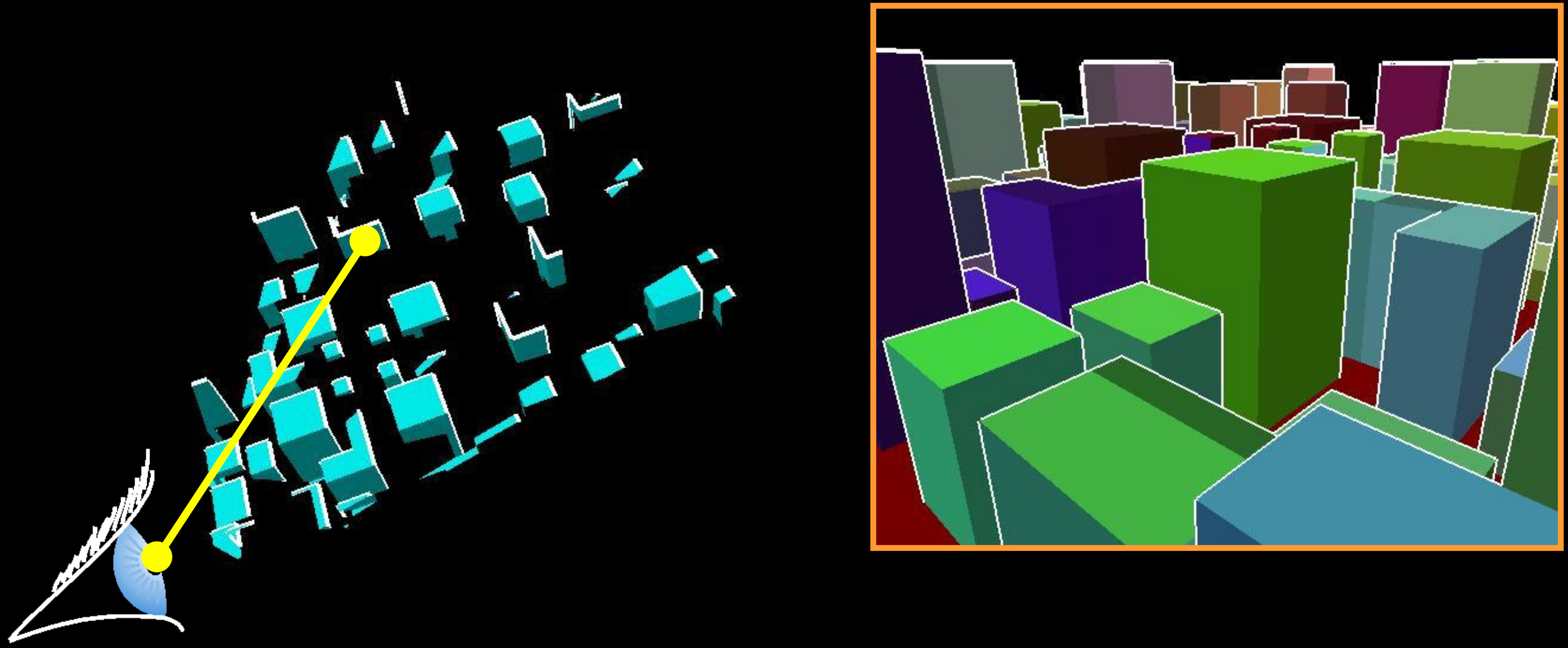
Visibility

Compute which part of scene can be seen



Visibility

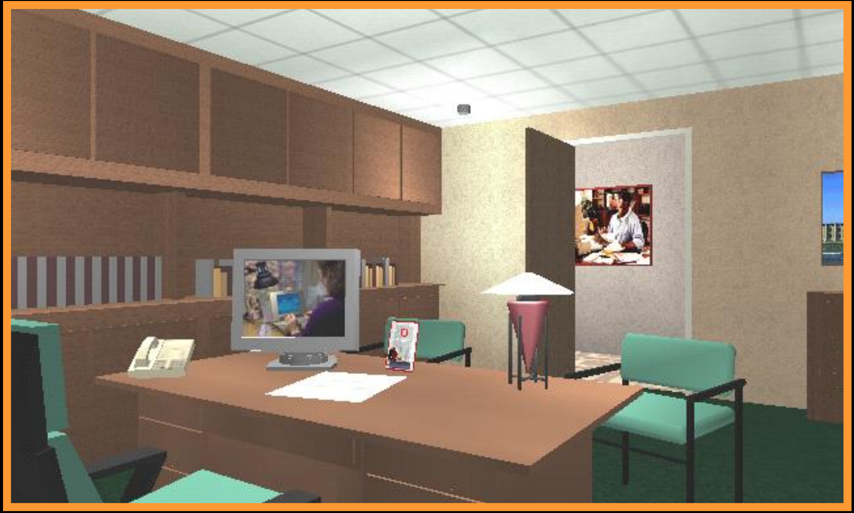
Compute which part of scene can be seen
(i.e., line segment from source to point in scene)



Visibility Applications

Computer graphics

- ➢ Hidden surface removal
- Shadow computation
- Global illumination
- Occlusion culling



Visibility Applications

Computer graphics

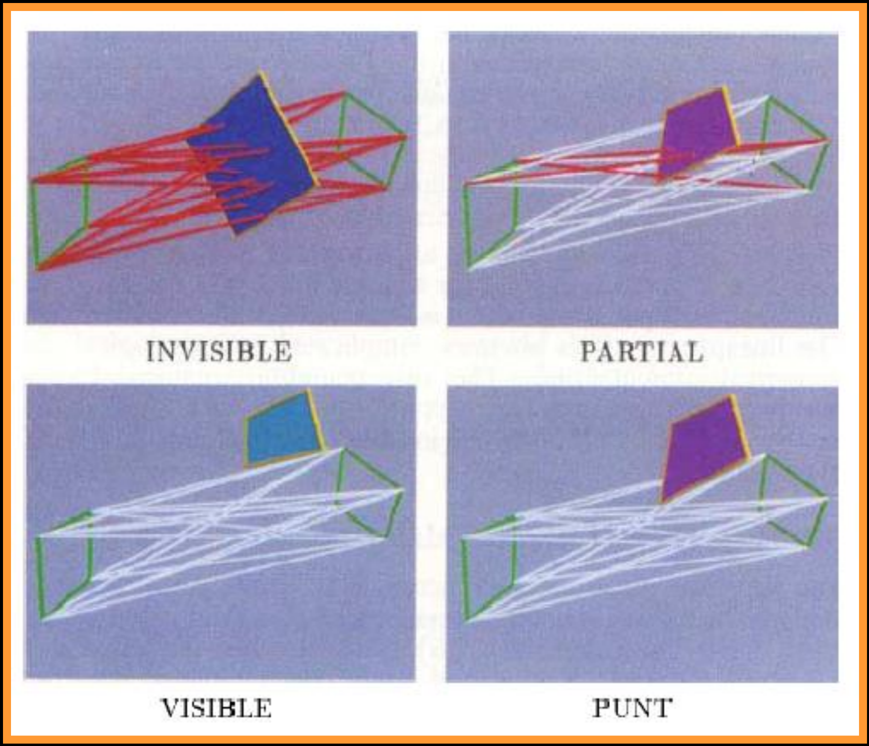
- Hidden surface removal
- Shadow computation
- Global illumination
- Occlusion culling

Drettakis

Visibility Applications

Computer graphics

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



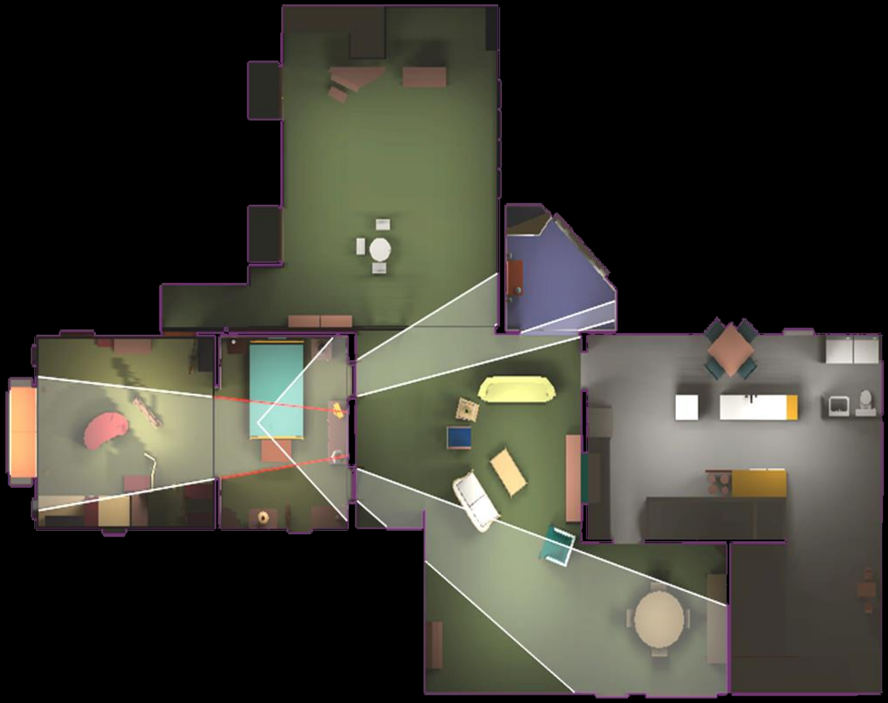
$$L(x',x'') = L_e(x',x'') + \int_S f_r(x,x',x'')L(x,x')\boxed{V(x,x')}G(x,x')dA$$

Teller

Visibility Applications

Computer graphics

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



Luebke

Visibility Applications

Computational Geometry

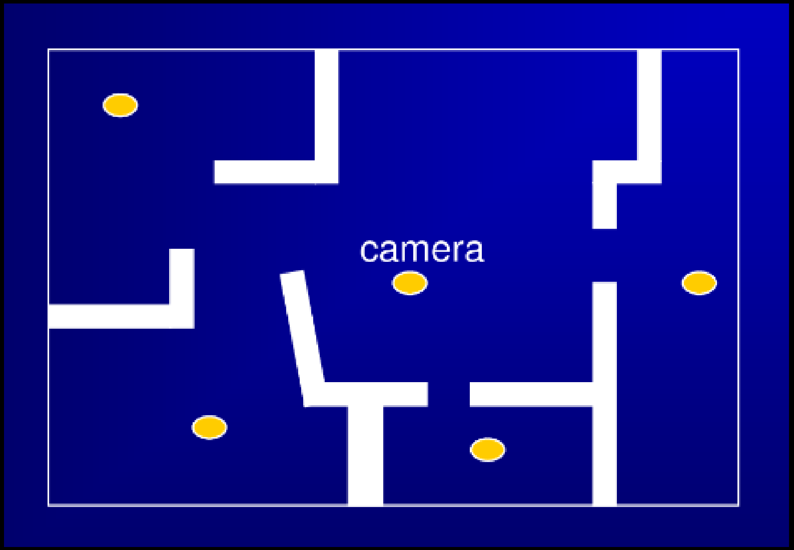
- Art galleries

Computer vision

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

Robotics

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



Outline

Hidden surface removal

- Visibility from viewpoint

Shadow map

- Visibility from point light source

Aspect graph

- Visibility from any point in space

Visibility Skeleton

- Visibility between scene elements

Outline

Hidden surface removal ←

- Visibility from viewpoint

Shadow map

- Visibility from point light source

Aspect graph

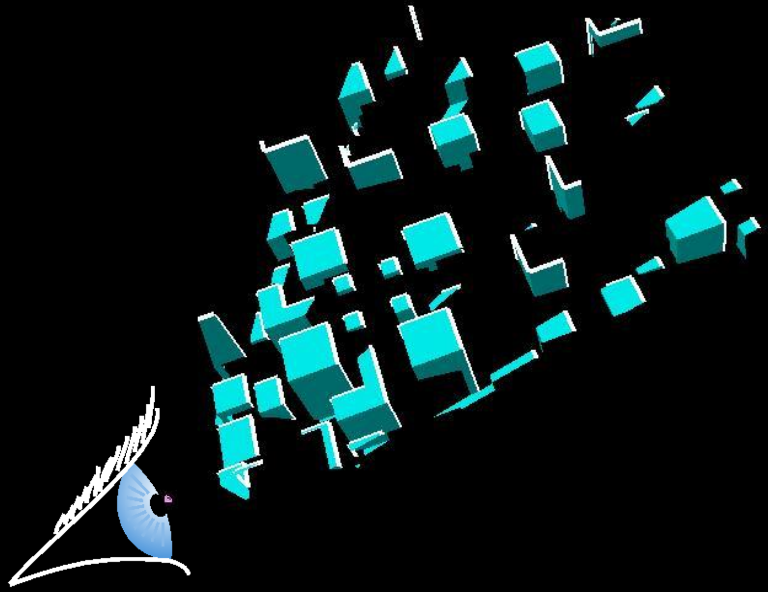
- Visibility from any point in space

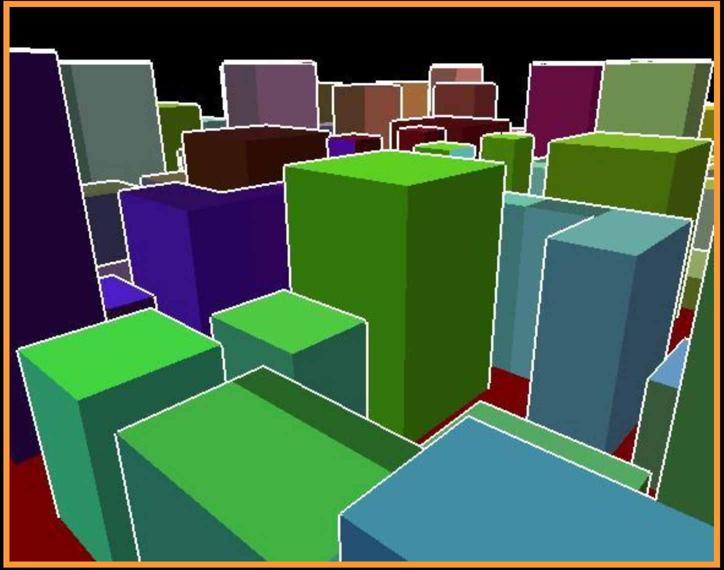
Visibility Skeleton

- Visibility between scene elements

Hidden Surface Removal

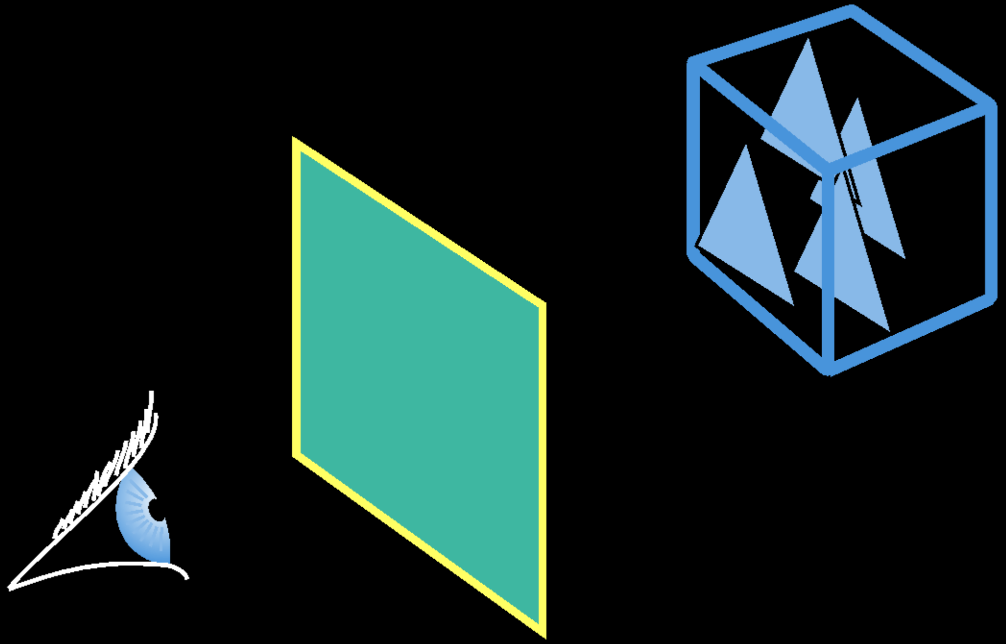
Compute which part of every primitive can be seen from a point





Hidden Surface Removal

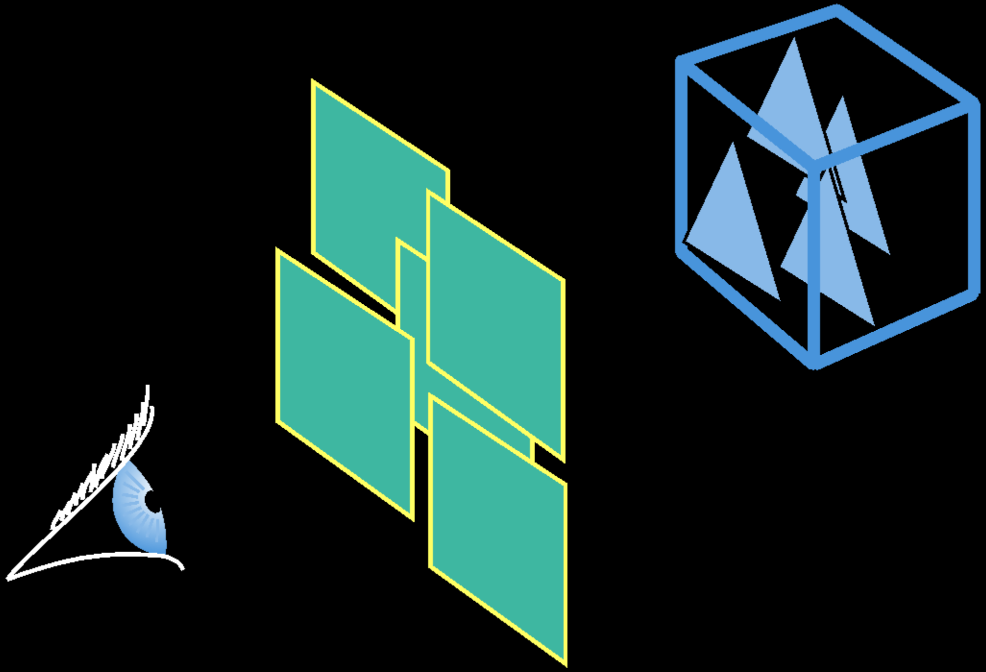
Occlusion by a single occluder



Fredo Durand

Hidden Surface Removal Problem

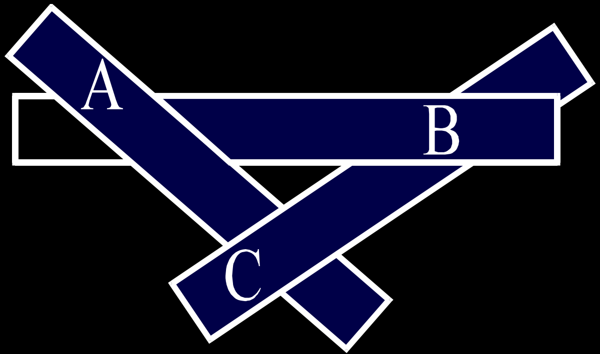
Cumulative occlusion by multiple occluders



Fredo Durand

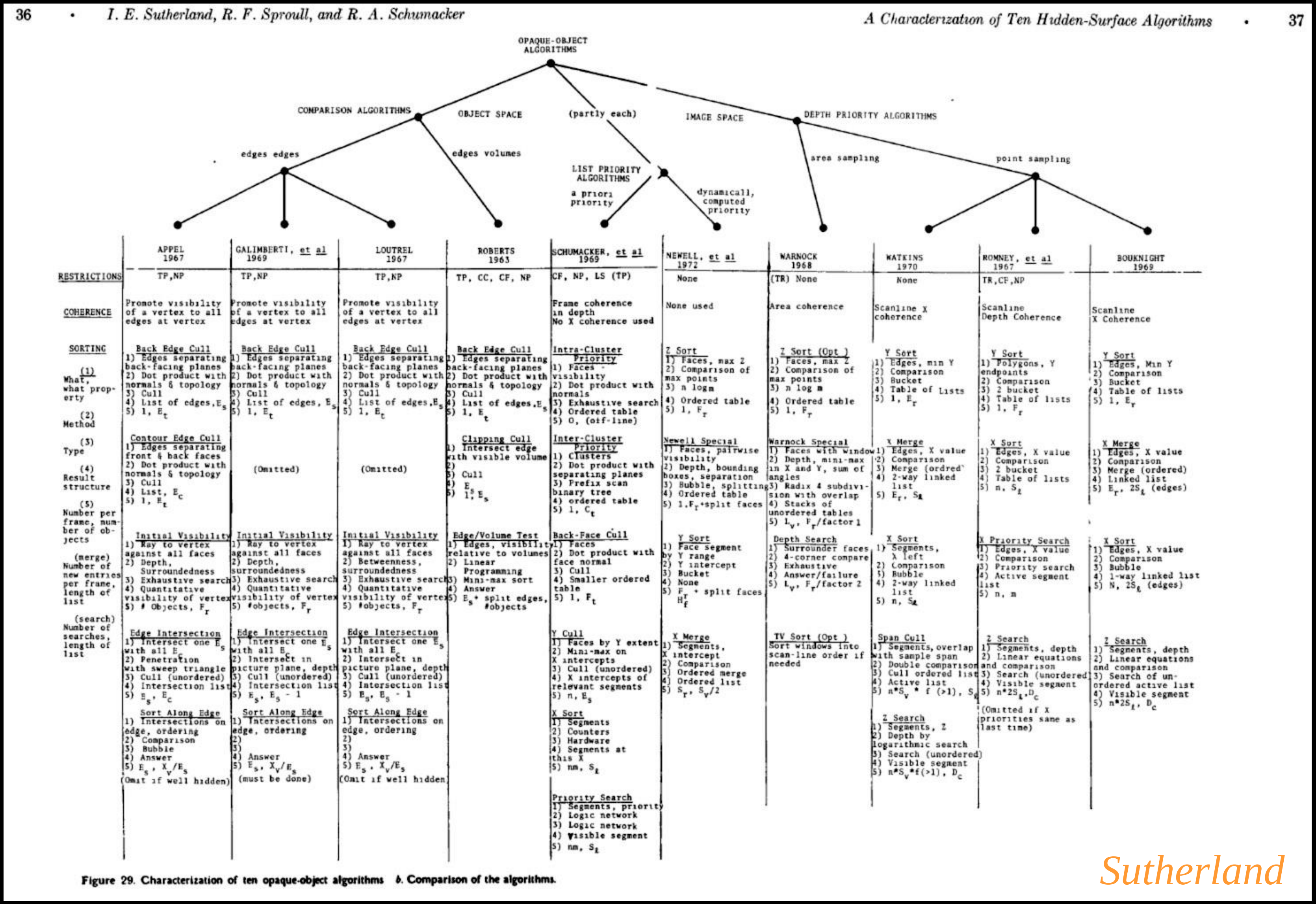
Hidden Surface Removal Problem

Sorting according to a distance is not enough



Fredo Durand

Hidden Surface Removal Methods



36 • *I. E. Sutherland, R. F. Sproull, and R. A. Schumacker*

A Characterization of Ten Hidden-Surface Algorithms • 37

	APPEL 1967	GALIMBERTI, et al 1969	LOUTREL 1967	ROBERTS 1963	SCHUMACKER, et al 1969	NEWELL, et al 1972	WARNOCK 1968	WATKINS 1970	ROMNEY, et al 1967	BOUKNIGHT 1969
RESTRICTIONS	TP,NP	TP,NP	TP,NP	TP, CC, CF, NP	CF, NP, LS (TP)	None	(TR) None	None	TR,CF,NP	
COHERENCE	Promote visibility of a vertex to all edges at vertex	Promote visibility of a vertex to all edges at vertex	Promote visibility of a vertex to all edges at vertex		Frame coherence in depth No X coherence used	None used	Area coherence	Scanline X coherence	Scanline Depth Coherence	Scanline X Coherence
SORTING (1) What, what property (2) Method	Back Edge Cull 1) Edges separating back-facing planes 2) Dot product with normals & topology 3) Cull 4) List of edges, E_s 5) 1, E_t	Back Edge Cull 1) Edges separating back-facing planes 2) Dot product with normals & topology 3) Cull 4) List of edges, E_s 5) 1, E_t	Back Edge Cull 1) Edges separating back-facing planes 2) Dot product with normals & topology 3) Cull 4) List of edges, E_s 5) 1, E_t	Back Edge Cull 1) Edges separating back-facing planes 2) Dot product with normals & topology 3) Cull 4) List of edges, E_s 5) 1, E_t	Intra-Cluster Priority 1) Faces - visibility 2) Dot product with normals 3) Exhaustive search 4) Ordered table 5) 0, (off-line)	Z Sort 1) Faces, max Z 2) Comparison of max points 3) n log m 4) Ordered table 5) 1, F_r	Z Sort (Opt.) 1) Faces, max Z 2) Comparison of max points 3) n log m 4) Ordered table 5) 1, F_r	Y Sort 1) Edges, min Y 2) Comparison 3) Bucket 4) Table of Lists 5) 1, E_r	Y Sort 1) Polygons, Y endpoints 2) Comparison 3) 2 bucket 4) Table of lists 5) 1, F_r	Y Sort 1) Edges, Min Y 2) Comparison 3) Bucket 4) Table of lists 5) 1, E_r
(3) Type (4) Result structure (5) Number per frame, number of objects	Contour Edge Cull 1) Edges separating front & back faces 2) Dot product with normals & topology 3) Cull 4) List, E_c 5) 1, E_t	(Omitted)	(Omitted)	Clipping Cull 1) Intersect edge with visible volume 2) 3) Cull 4) E_s 5) 1, E_s	Inter-Cluster Priority 1) Clusters 2) Dot product with separating planes 3) Prefix scan binary tree 4) ordered table 5) 1, C_t	Newell Special 1) Faces, pairwise visibility 2) Depth, bounding boxes, separation 3) Bubble, splitting 4) Ordered table 5) 1, F_r+split faces	Warnock Special 1) Faces with window 2) Depth, mini-max in X and Y, sum of angles 3) Radix 4 subdivision with overlap 4) Stacks of unordered tables 5) L_v, F_r/factor 1	X Merge 1) Edges, X value 2) Comparison 3) Merge (ordred` 4) 2-way linked list 5) E_r, S_ℓ	X Sort 1) Edges, X value 2) Comparison 3) 2 bucket 4) Table of lists 5) n, S_ℓ	X Merge 1) Edges, X value 2) Comparison 3) Merge (ordered) 4) Linked list 5) E_r, 2S_ℓ (edges)
(merge) Number of new entries per frame, length of list	Initial Visibility 1) Ray to vertex against all faces 2) Depth, Surroundedness 3) Exhaustive search 4) Quantitative visibility of vertex 5) # Objects, F_r	Initial Visibility 1) Ray to vertex against all faces 2) Depth, surroundedness 3) Exhaustive search 4) Quantitative visibility of vertex 5) #objects, F_r	Initial Visibility 1) Ray to vertex against all faces 2) Betweenness, surroundedness 3) Exhaustive search 4) Quantitative visibility of vertex 5) #objects, F_r	Edge/Volume Test 1) Edges, visibility relative to volumes 2) Linear Programming 3) Mini-max sort 4) Answer 5) E_s+ split edges, #objects	Back-Face Cull 1) Faces 2) Dot product with face normal 3) Cull 4) Smaller ordered table 5) 1, F_t	Y Sort 1) Face segment by Y range 2) Y intercept 3) Bucket 4) None 5) F_r + split faces H_f^r	Depth Search 1) Surrounder faces 2) 4-corner compare 3) Exhaustive 4) Answer/failure 5) L_v, F_r/factor 2	X Sort 1) Segments, X left 2) Comparison 3) Bubble 4) 2-way linked list 5) n, S_ℓ	X Priority Search 1) Edges, X value 2) Comparison 3) Priority search 4) Active segment list 5) n, m	X Sort 1) Edges, X value 2) Comparison 3) Bubble 4) 1-way linked list 5) N, 2S_ℓ (edges)
(search) Number of searches, length of list	Edge Intersection 1) Intersect one E_s with all E_c 2) Penetration with sweep triangle 3) Cull (unordered) 4) Intersection list 5) E_s, E_c	Edge Intersection 1) Intersect one E_s with all E_c 2) Intersect in picture plane, depth 3) Cull (unordered) 4) Intersection list 5) E_s, E_s - 1	Edge Intersection 1) Intersect one E_s with all E_c 2) Intersect in picture plane, depth 3) Cull (unordered) 4) Intersection list 5) E_s, E_s - 1		Y Cull 1) Faces by Y extent 2) Mini-max on X intercepts 3) Cull (unordered) 4) X intercepts of relevant segments 5) n, E_s	X Merge 1) Segments, X intercept 2) Comparison 3) Ordered merge 4) Ordered list 5) S_r, $S_v/2$	TV Sort (Opt.) Sort windows into scan-line order if needed	Span Cull 1) Segments, overlap with sample span 2) Double comparison 3) Cull ordered list 4) Active list 5) n*S_v * f (>1), S_ℓ	Z Search 1) Segments, depth 2) Linear equations and comparison 3) Search (unordered) 4) Visible segment 5) n*2S_ℓ,D_c (Omitted if X priorities same as last time)	Z Search 1) Segments, depth 2) Linear equations and comparison 3) Search of unordered active list 4) Visible segment 5) n*2S_ℓ, D_c
	Sort Along Edge 1) Intersections on edge, ordering 2) Comparison 3) Bubble 4) Answer 5) E_s, X_v/E_s (Omit if well hidden)	Sort Along Edge 1) Intersections on edge, ordering 2) 3) 4) Answer 5) E_s, X_v/E_s (must be done)	Sort Along Edge 1) Intersections on edge, ordering 2) 3) 4) Answer 5) E_s, X_v/E_s (Omit if well hidden)		X Sort 1) Segments 2) Counters 3) Hardware 4) Segments at this X 5) nm, S_ℓ			Z Search 1) Segments, Z 2) Depth by logarithmic search 3) Search (unordered) 4) Visible segment 5) n*S_v*f(>1), D_c		
					Priority Search 1) Segments, priority 2) Logic network 3) Logic network 4) Visible segment 5) nm, S_ℓ					

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

Sutherland

Hidden Surface Removal Methods

Image-space

- Z-buffer
- Scan-line
- Warnock subdivision

Object-space

- Depth-sort
- Weiler-Atherton
- BSP

Line-space

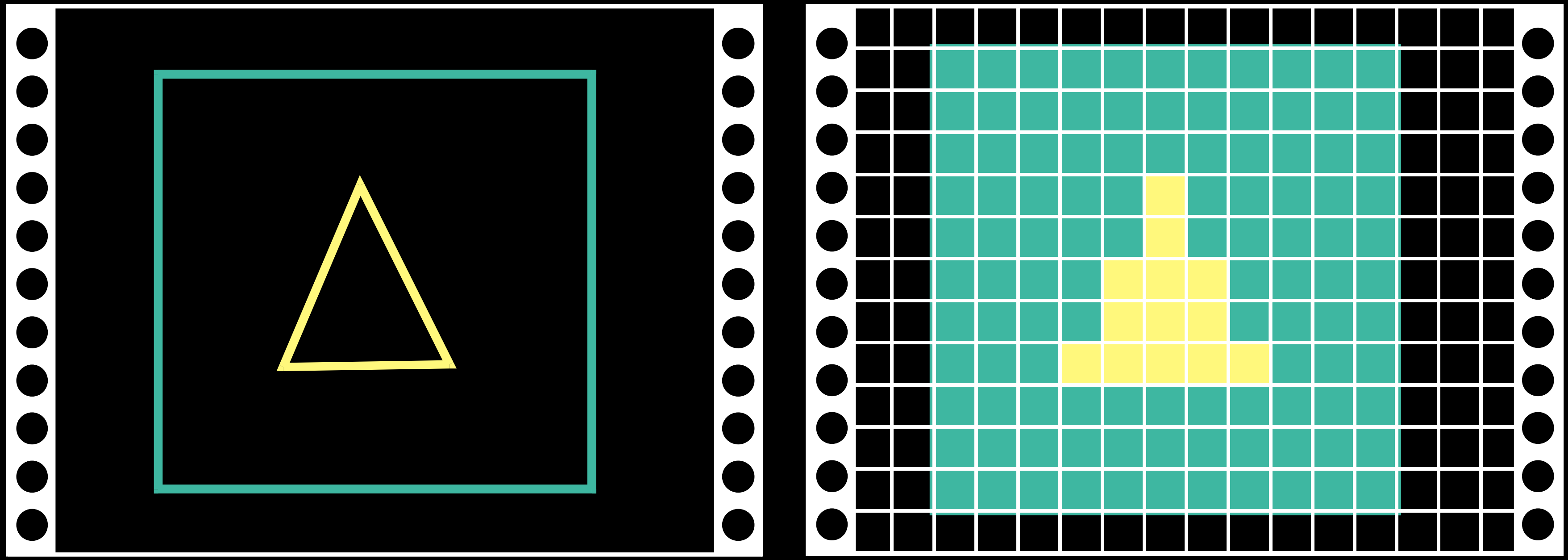
- Ray casting

Image-space

Computation performed in the plane of the image

E.g. is triangle inside rectangle?

Usually discretized in pixels

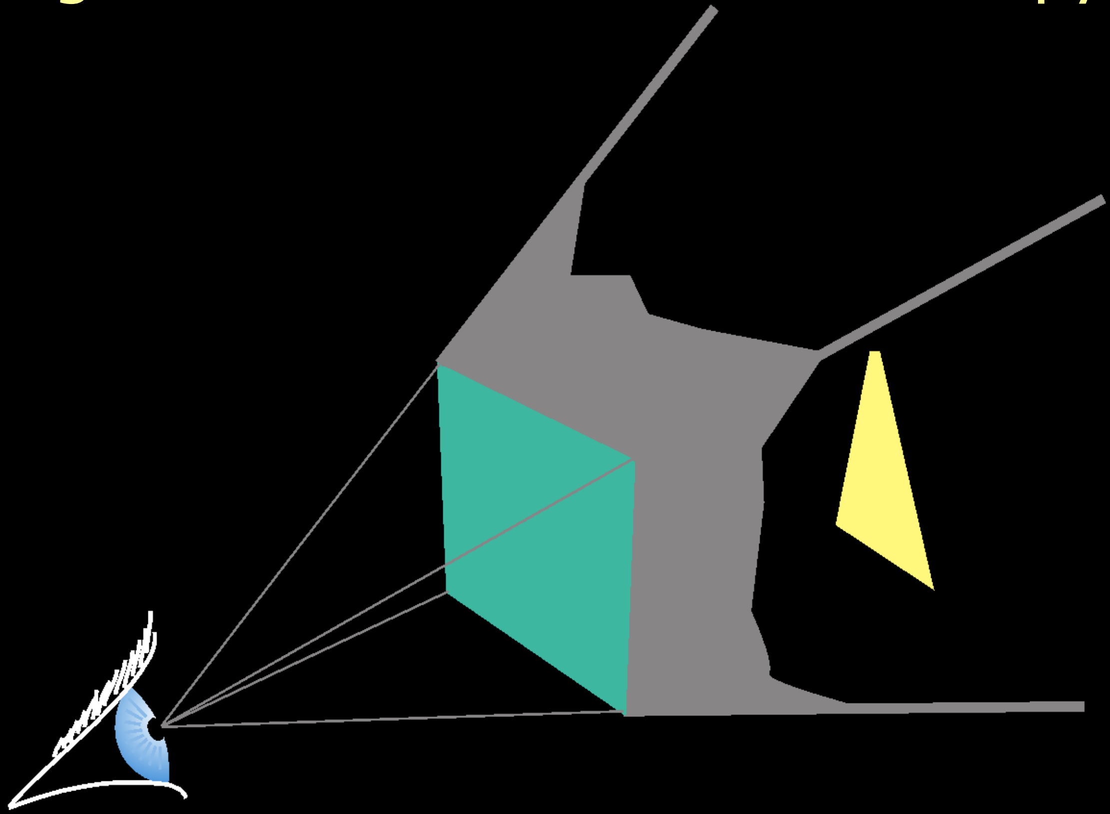


Fredo Durand

Object-space

3D space where the scene is defined

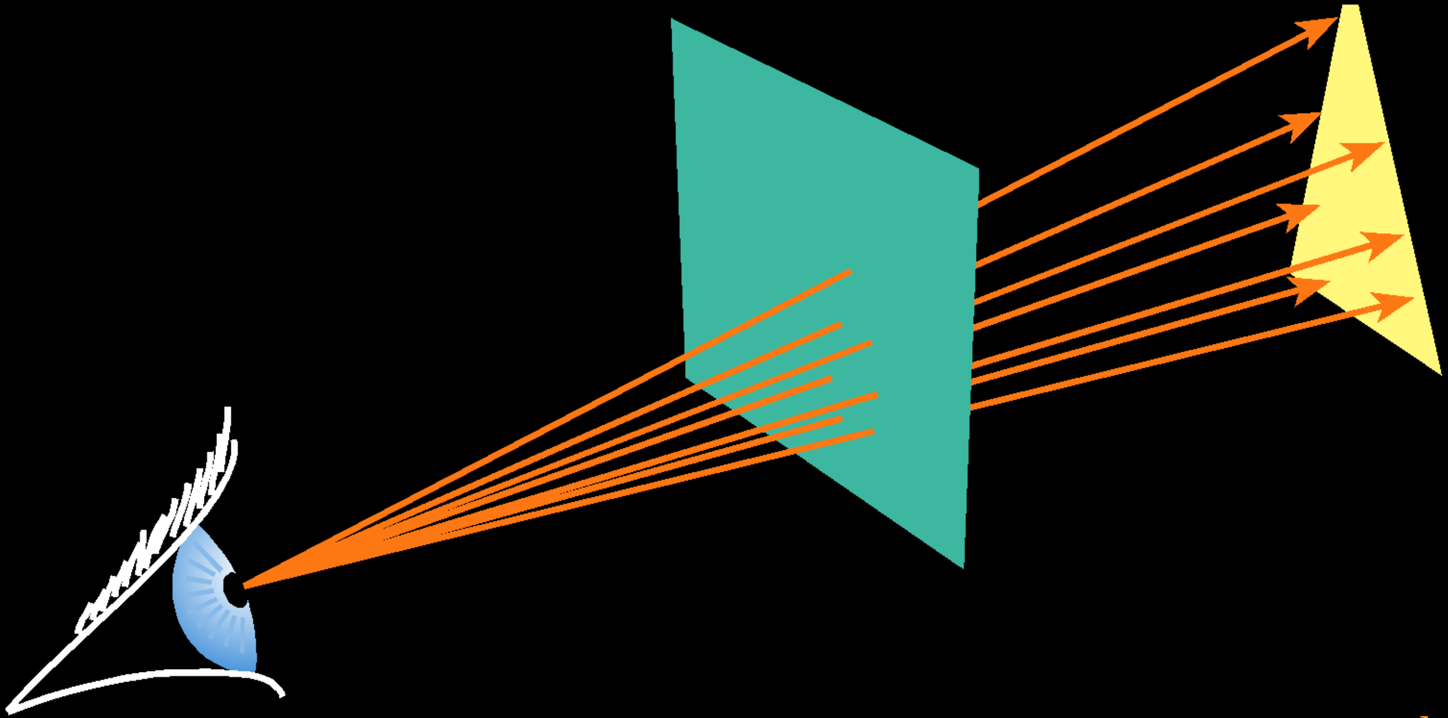
E.g., triangle is occluded if it is inside the pyramid



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

Shadow map ←

- Visibility from point light source

Aspect graph

- Visibility from any point in space

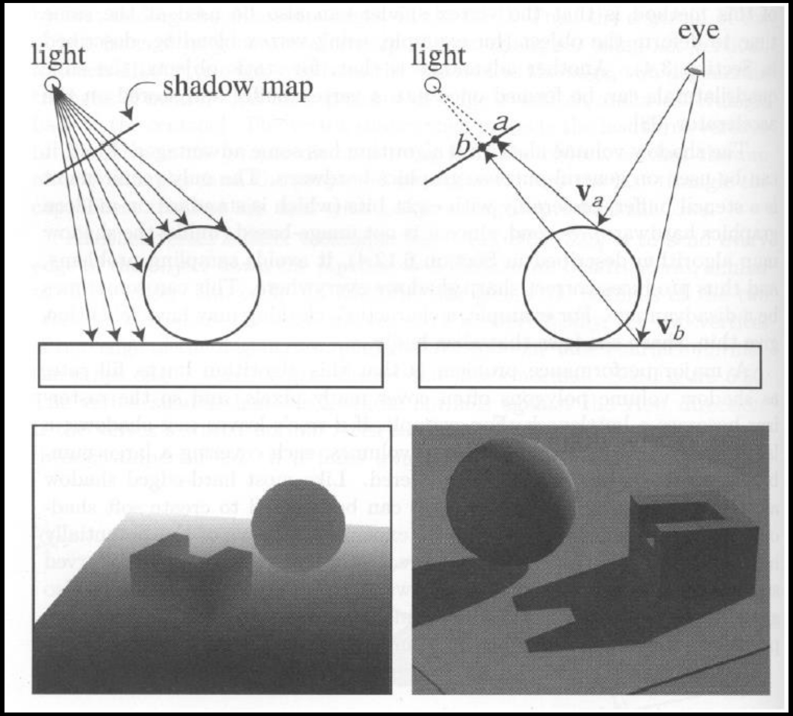
Visibility Skeleton

- Visibility between scene elements

Shadow Maps

Precompute image of depths from light

- Store image of distances from light
- Lookup depth of surface point in image when shade



[Foley et al.]

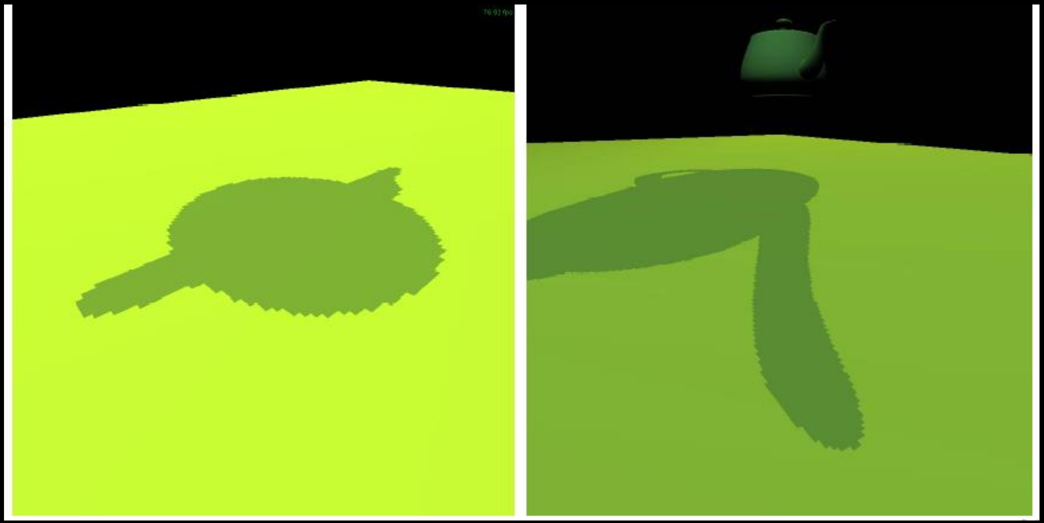
Shadow Maps

Suitable for hardware pipeline

- Projection into light coordinate system is 4x4 matrix
- Shadow map stored in texture

Problems

- Field of view
- Aliasing



Aliasing

[Teller & Durand, MIT]

Outline

Hidden surface removal

- Visibility from viewpoint

Shadow map

- Visibility from point light source

Aspect graph ←

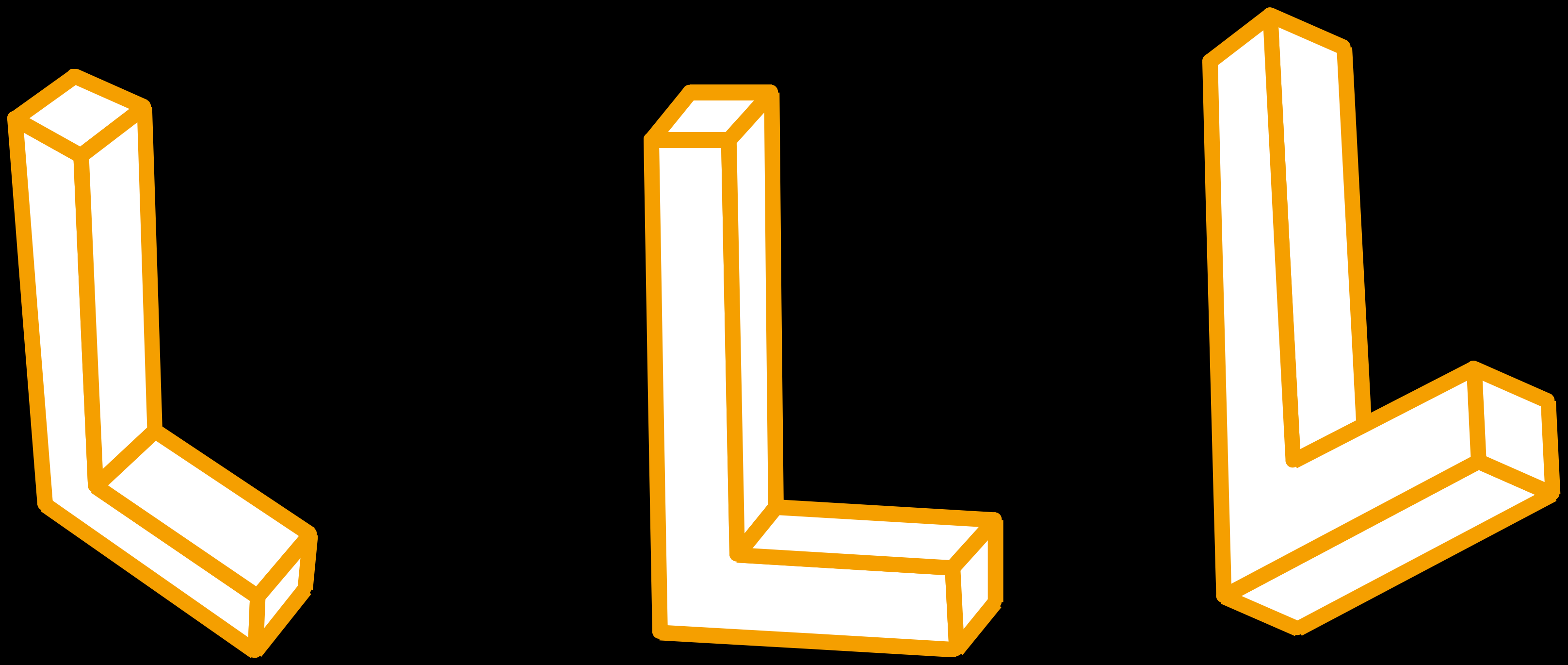
- Visibility from any point in space

Visibility Skeleton

- Visibility between scene elements

Aspect Graph

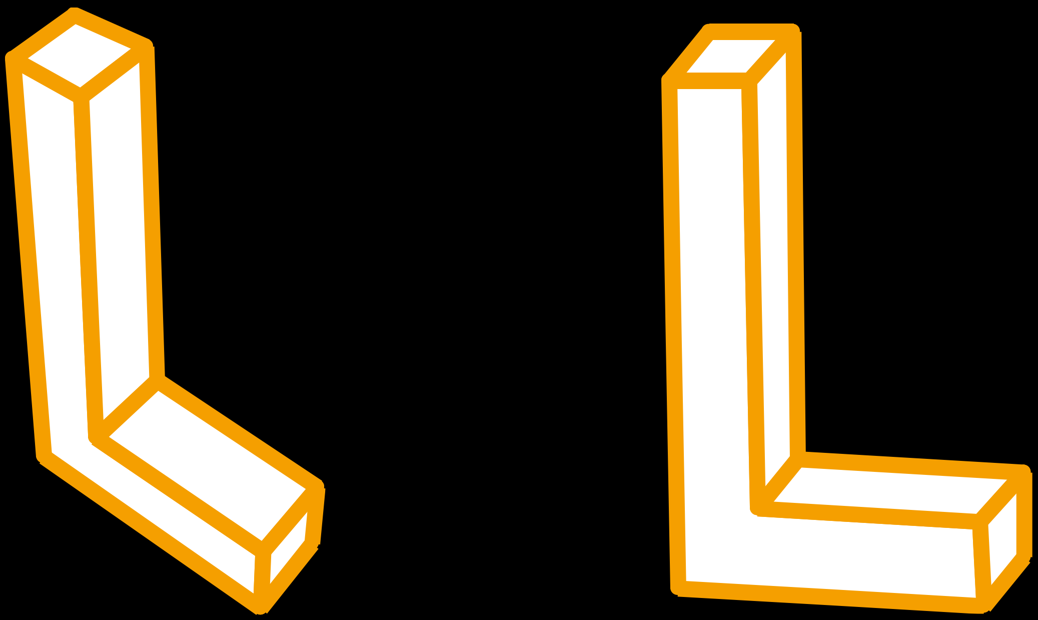
There are many possible views of any 3D object



Fredo Durand

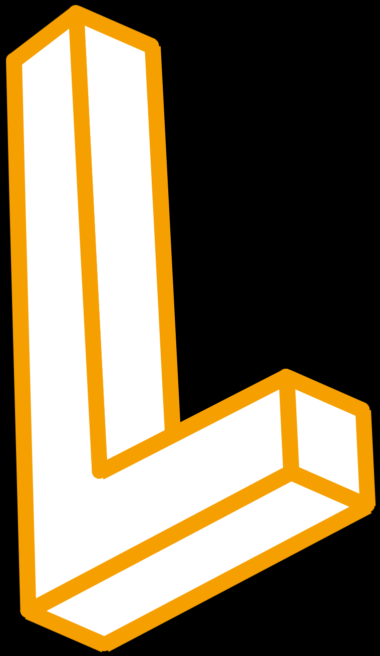
Aspect Graph

Some produce topologically equivalent visibility solution



Qualitatively equivalent

(same aspect)



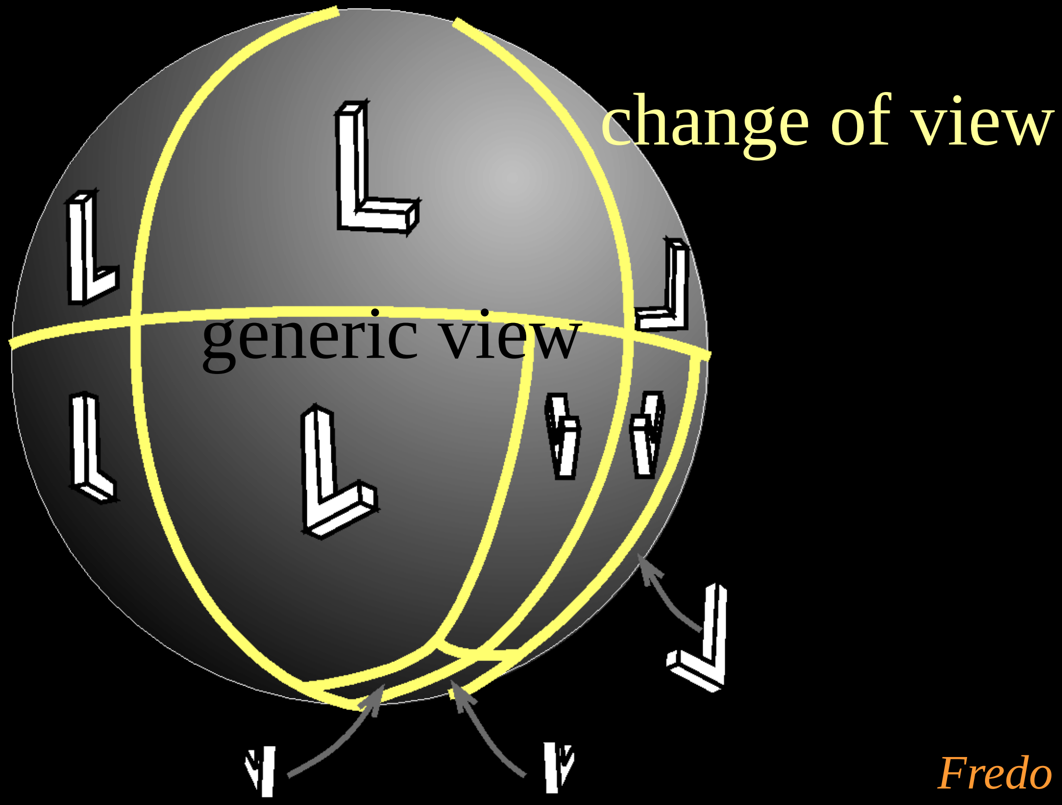
Qualitatively different
(different aspect)

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!

Outline

Hidden surface removal

- Visibility from viewpoint

Shadow map

- Visibility from point light source

Aspect graph

- Visibility from any point in space

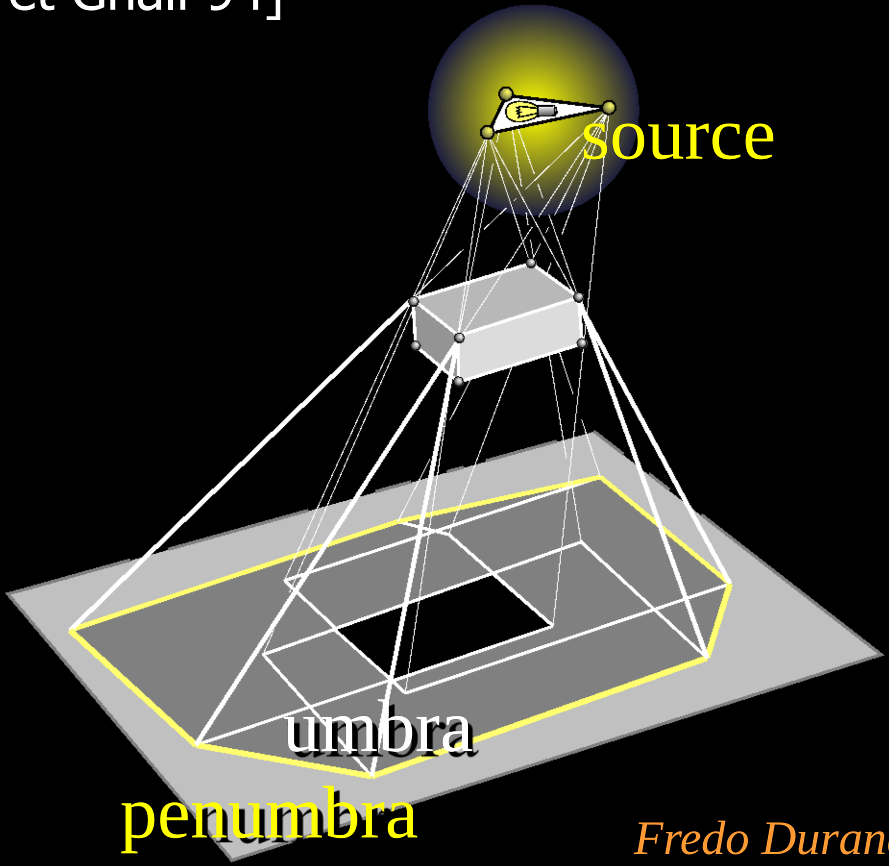
Visibility Skeleton ←

- Visibility between scene elements

Visibility from Polygon

Umbra and Penumbra

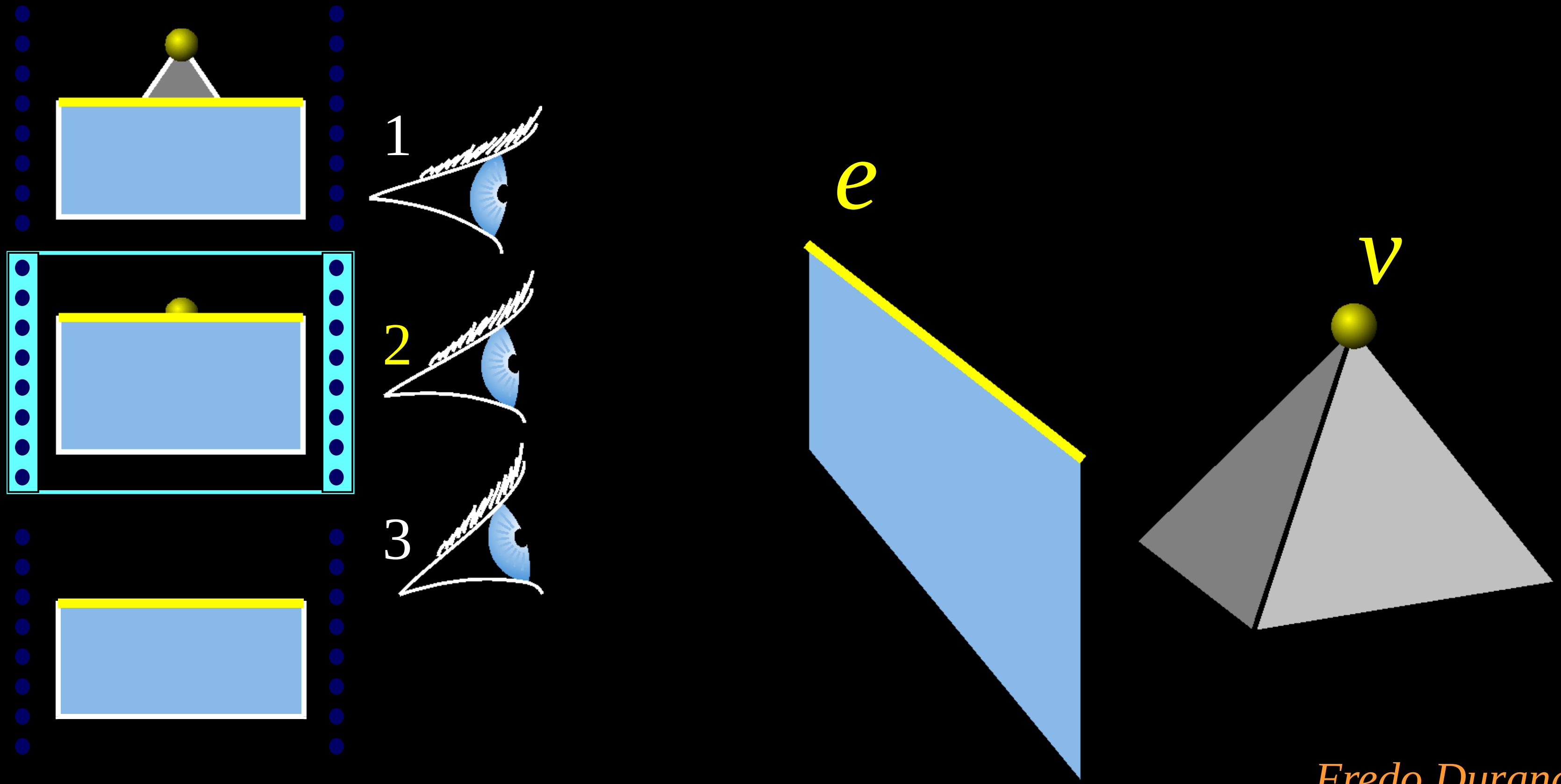
- [Nishita et Nakamae 85, Heckbert 92, Teller 92, Lischinski *et al.* 93, Drettakis et Fiume 94, Stewart et Ghali 94]



Fredo Durand

Visual event

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



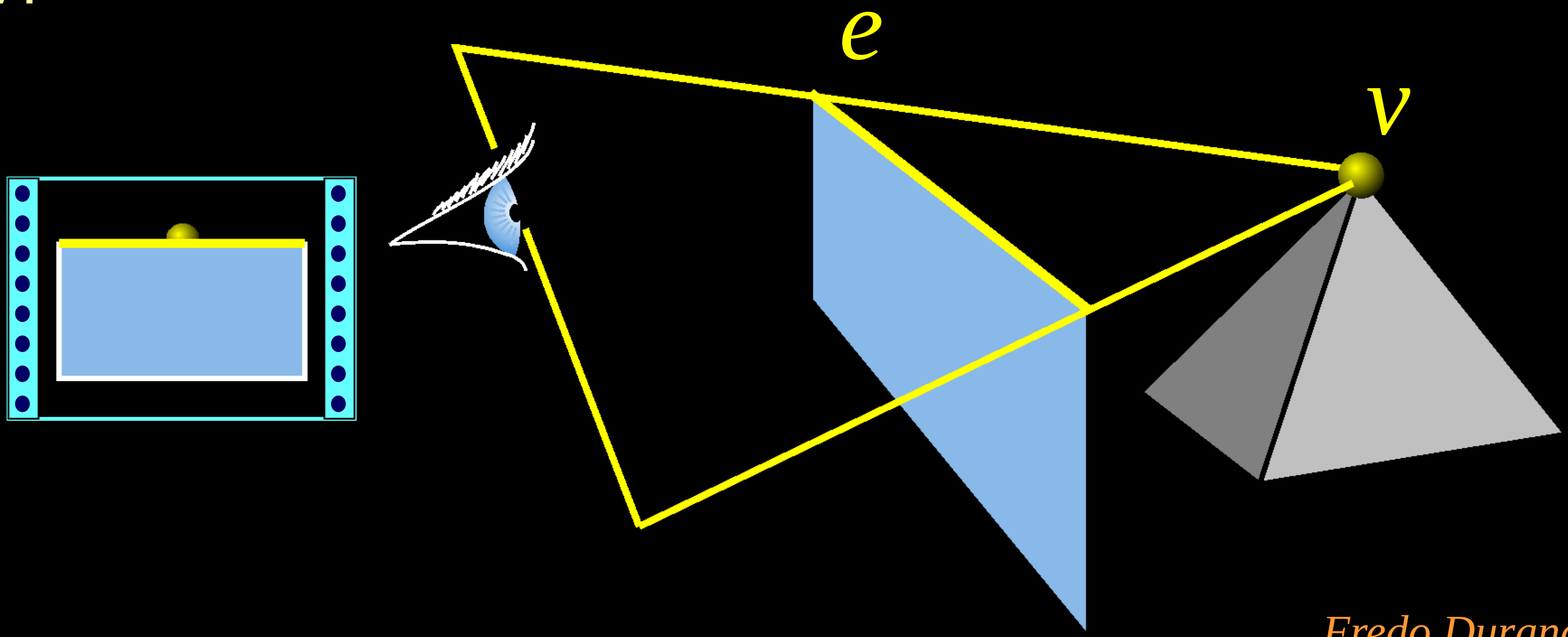
Fredo Durand

Visual event

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

« Wedge » defined by a vertex and an edge

Type EV

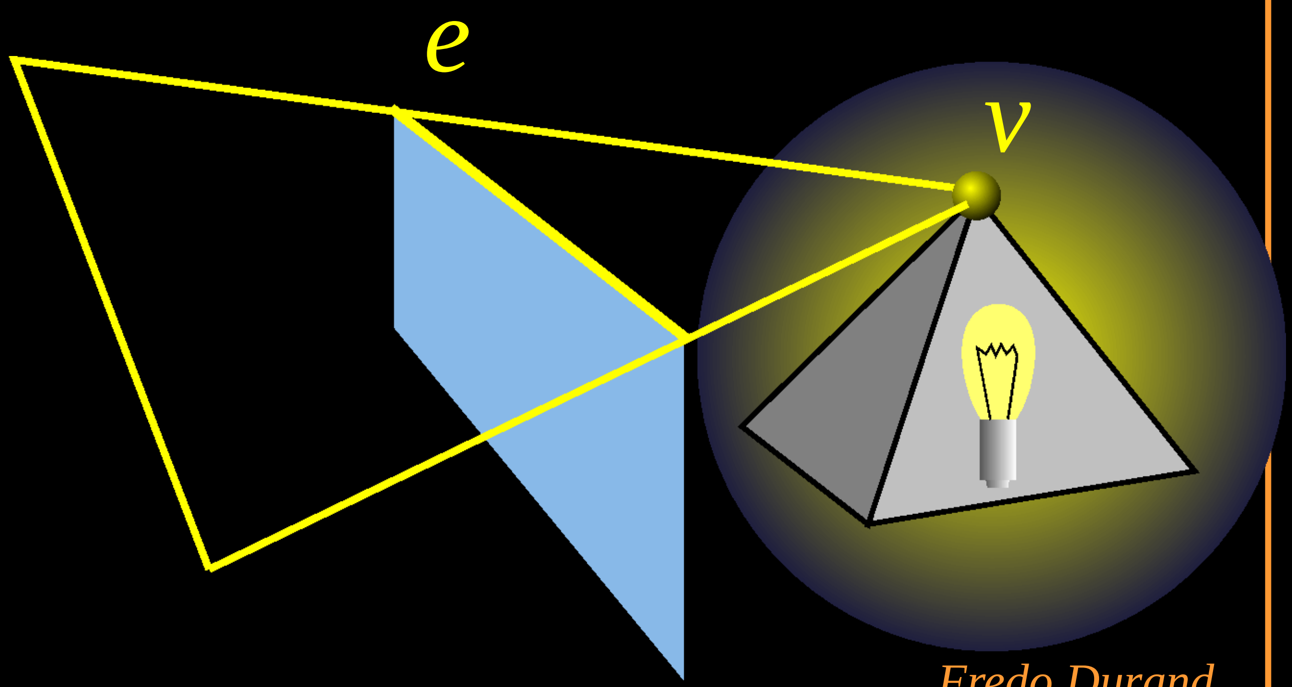


Fredo Durand

Visual event

Appearance-disappearance of objects

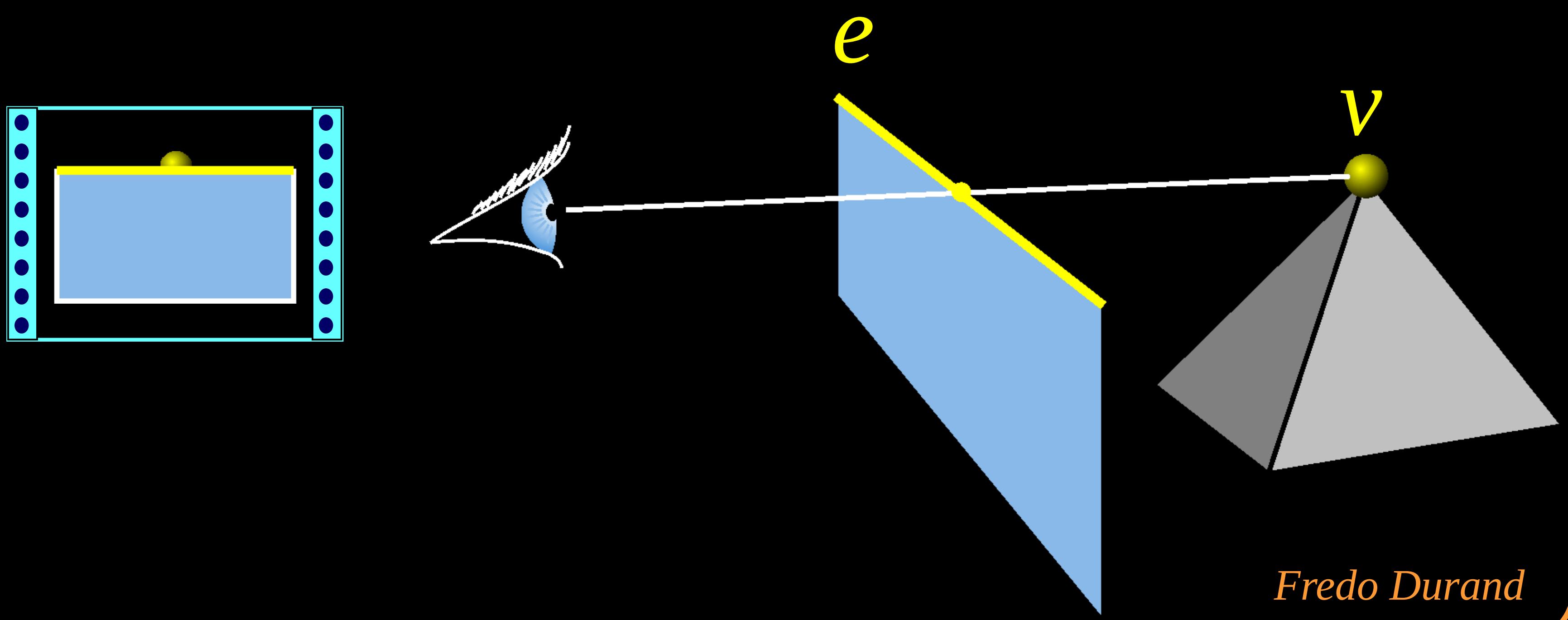
Limits of umbra



Fredo Durand

Critical line

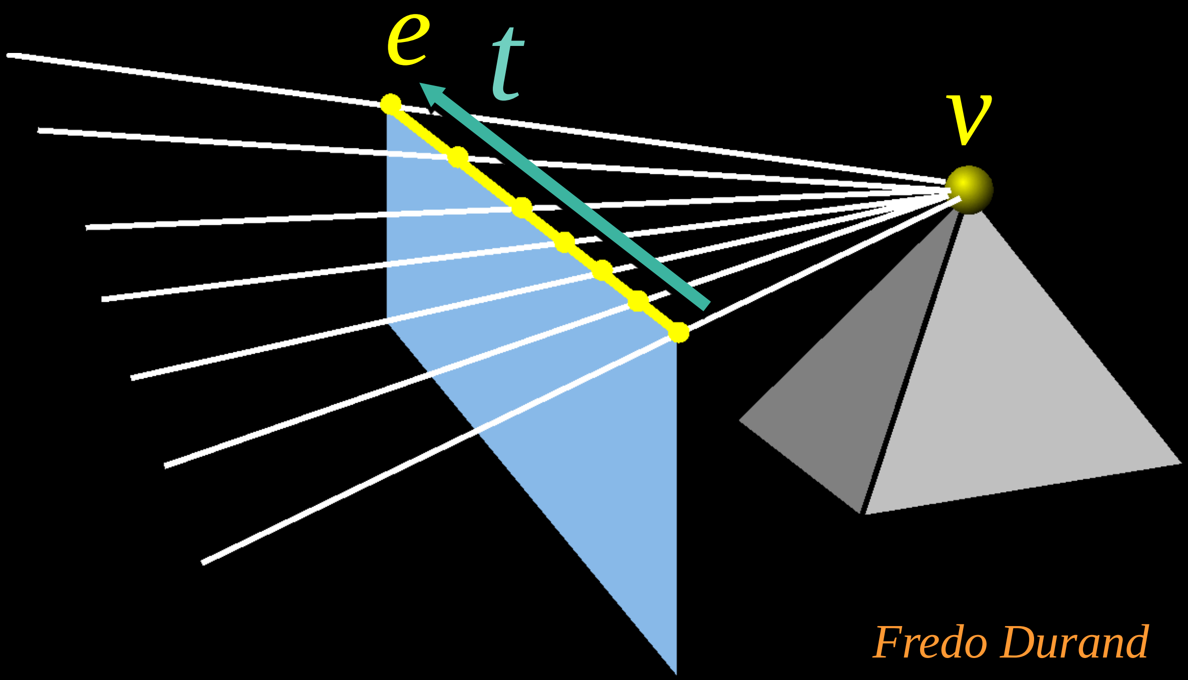
Line going through *e* and *v*



Fredo Durand

Critical lines

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

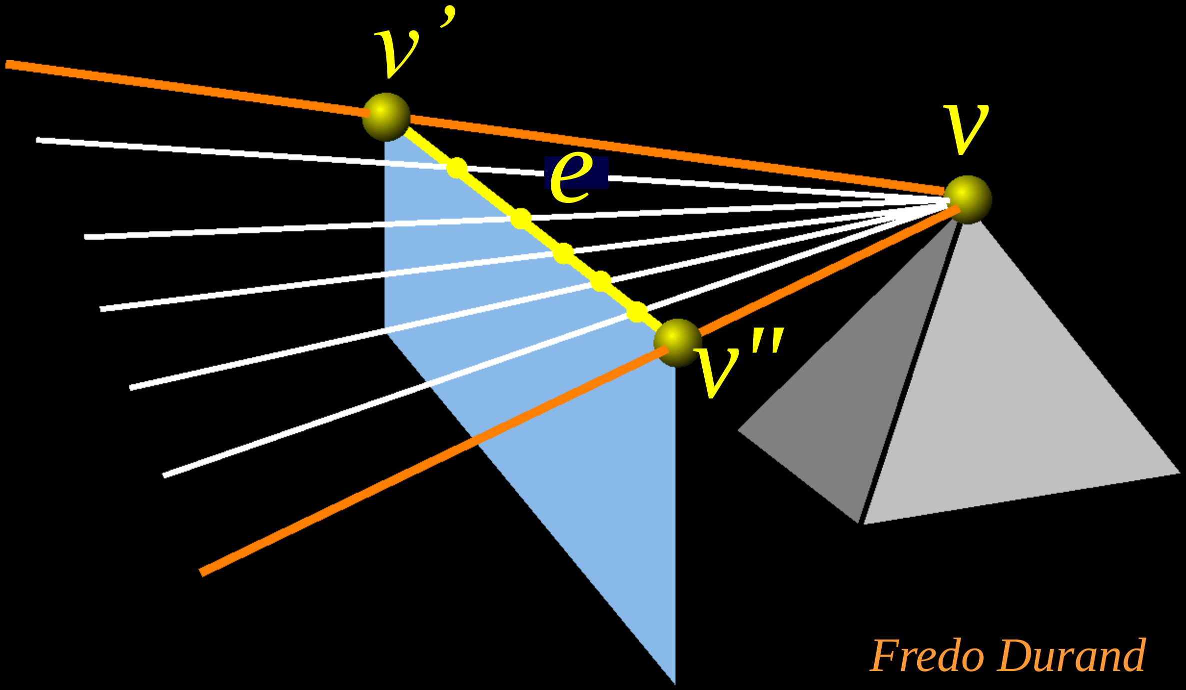


Fredo Durand

Extremal stabbing line

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

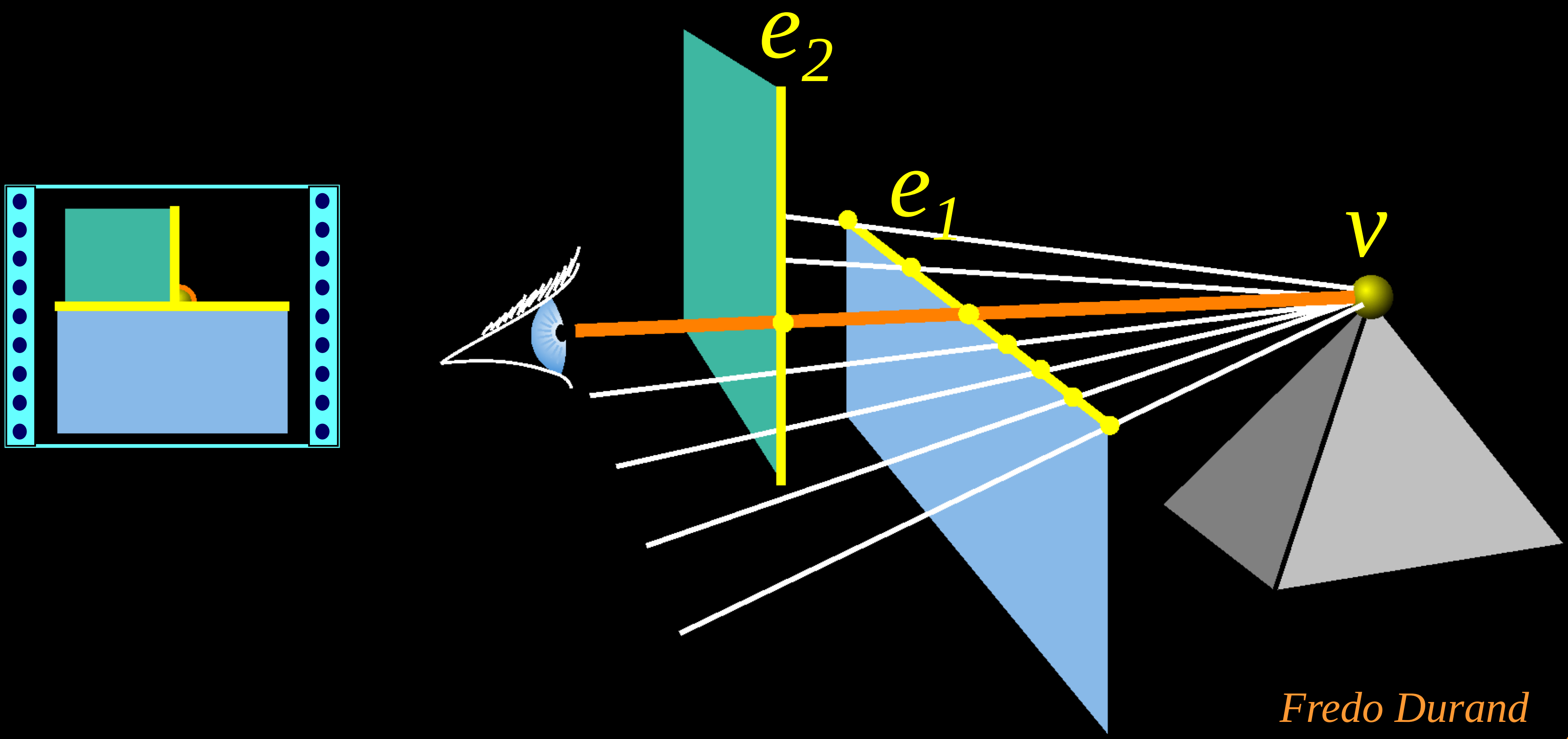
Extremity: extremal stabbing line (VV)
(0 degree of freedom)



Fredo Durand

Extremal stabbing line

Type *VEE* (0 degree of freedom)

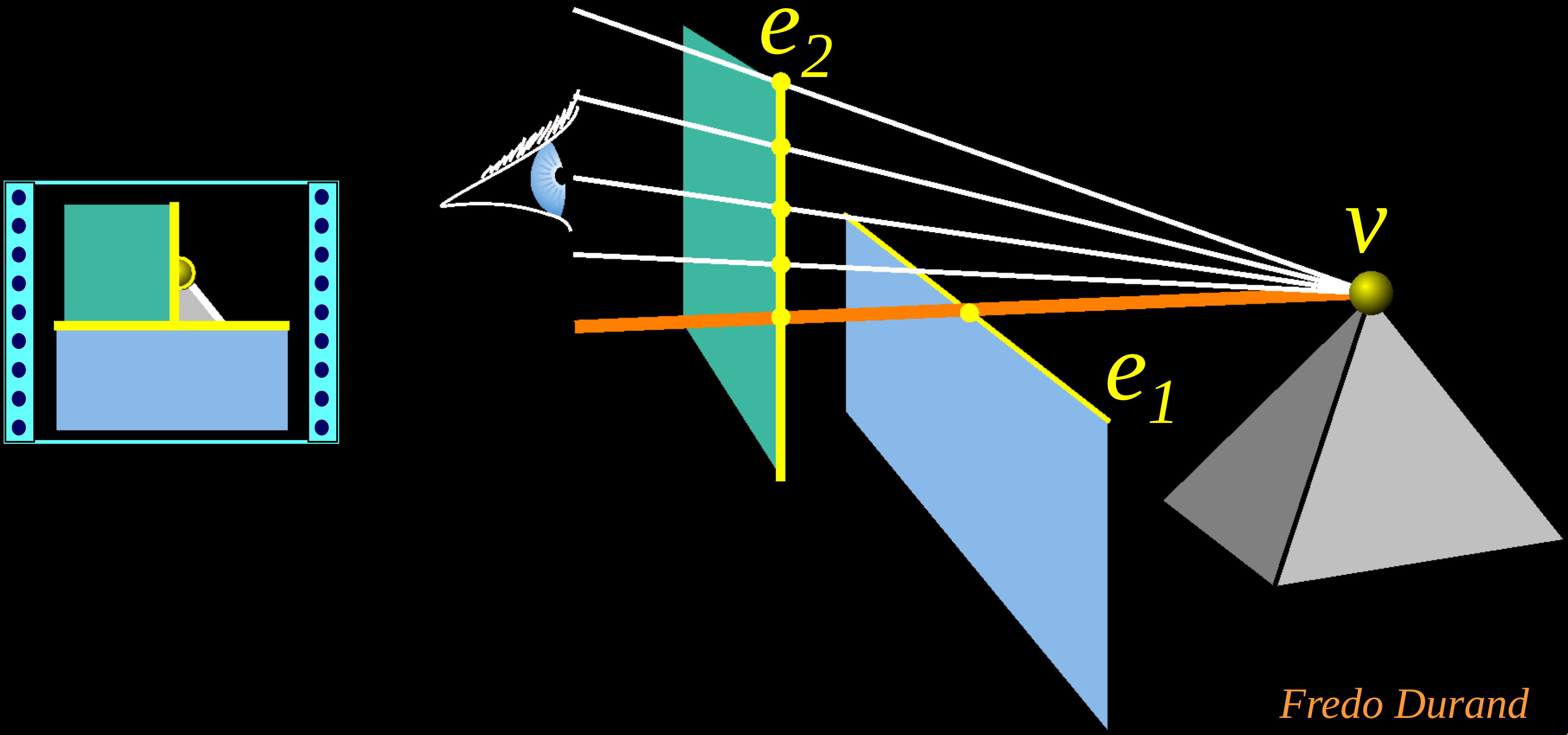


Fredo Durand

Adjacent critical line set

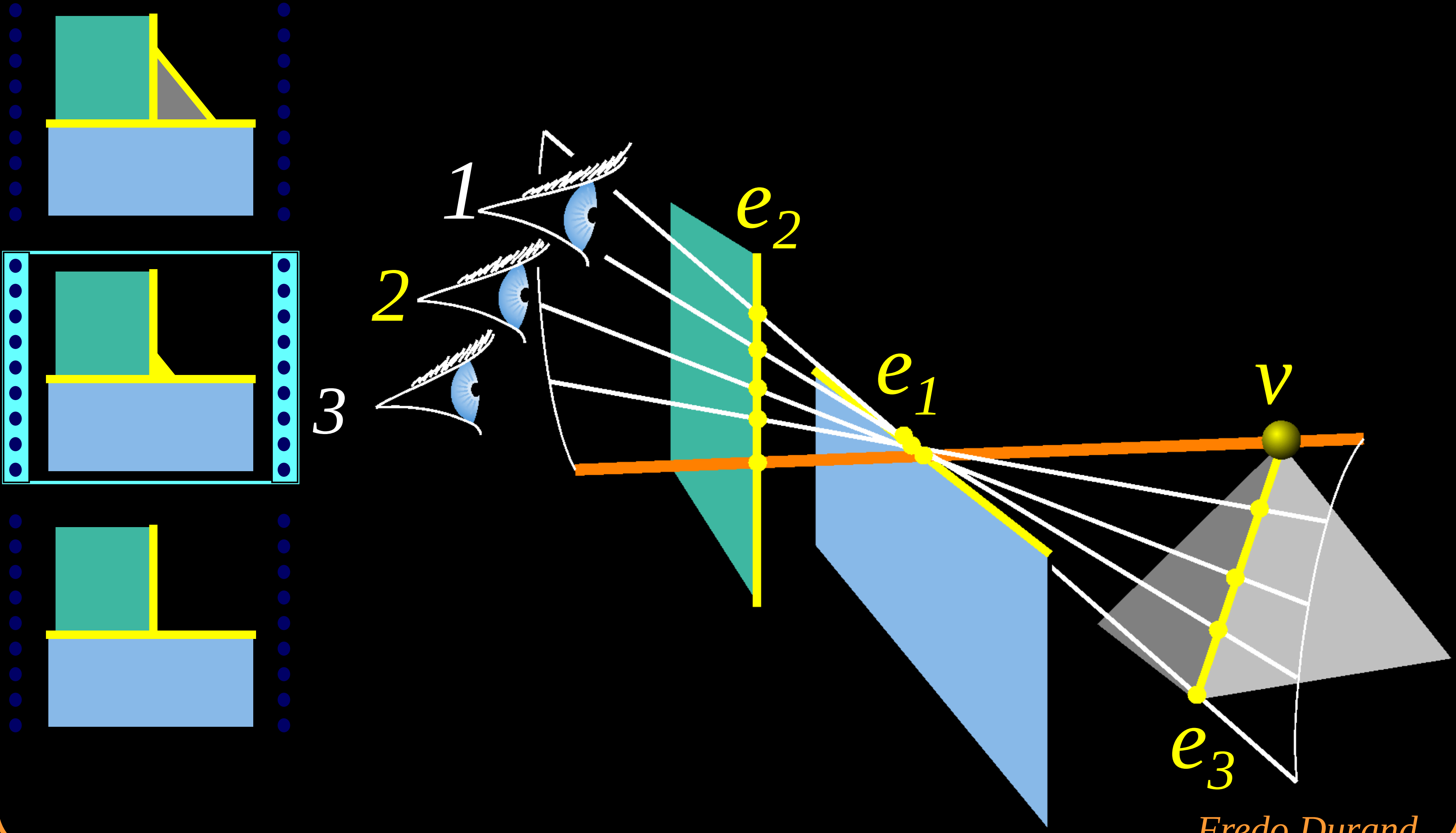
Generated by the second edge

Same extremity ve_1e_2



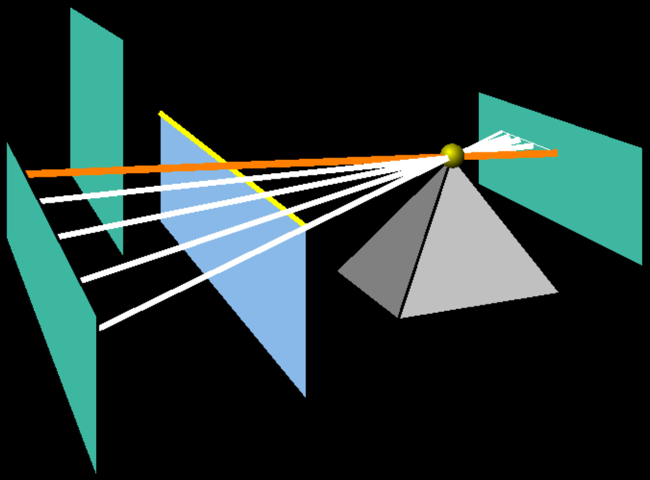
Fredo Durand

Triple-edge event

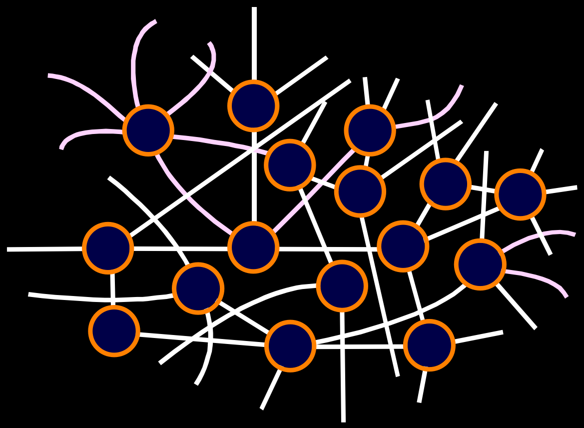


Fredo Durand

Visibility skeleton



Scene

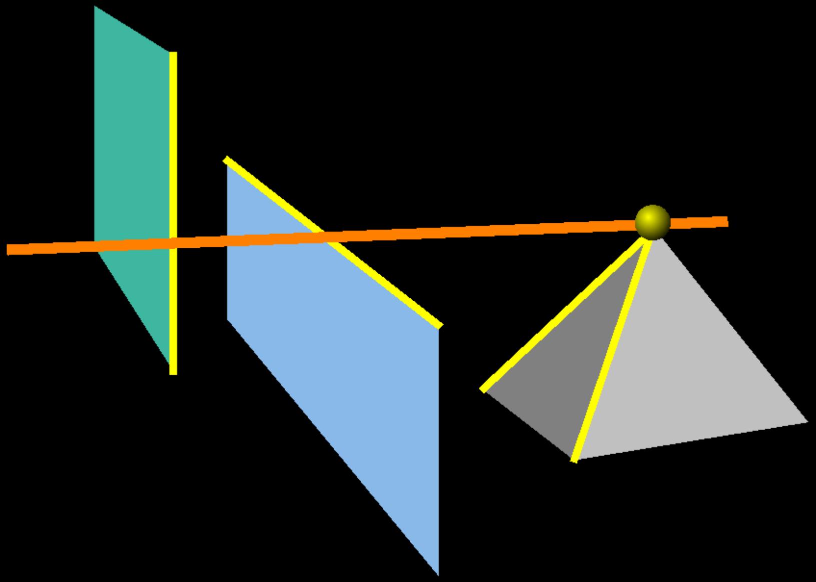


Graph in line space

Encodes adjacencies of extremal stabbing lines and critical line sets

Visibility Skeleton

Extremal stabbing line = Node



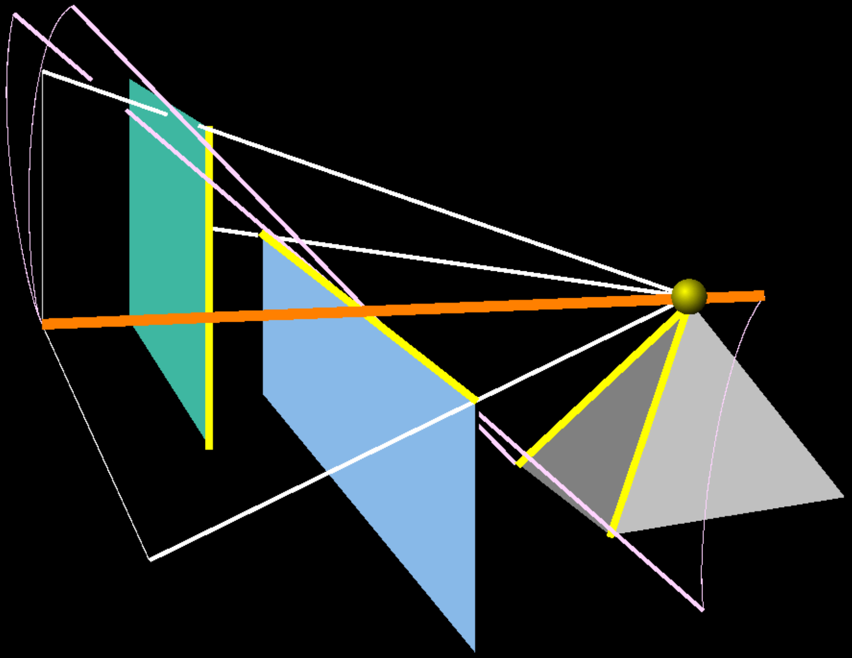
ve_1e_2

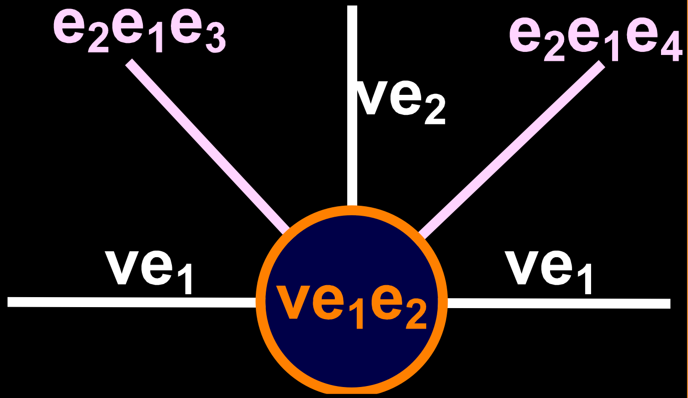
Fredo Durand

Visibility Skeleton

Extremal stabbing line = Node

Critical line set = Arc





$e_2e_1e_3$

ve_2

$e_2e_1e_4$

ve_1

ve_1e_2

ve_1

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

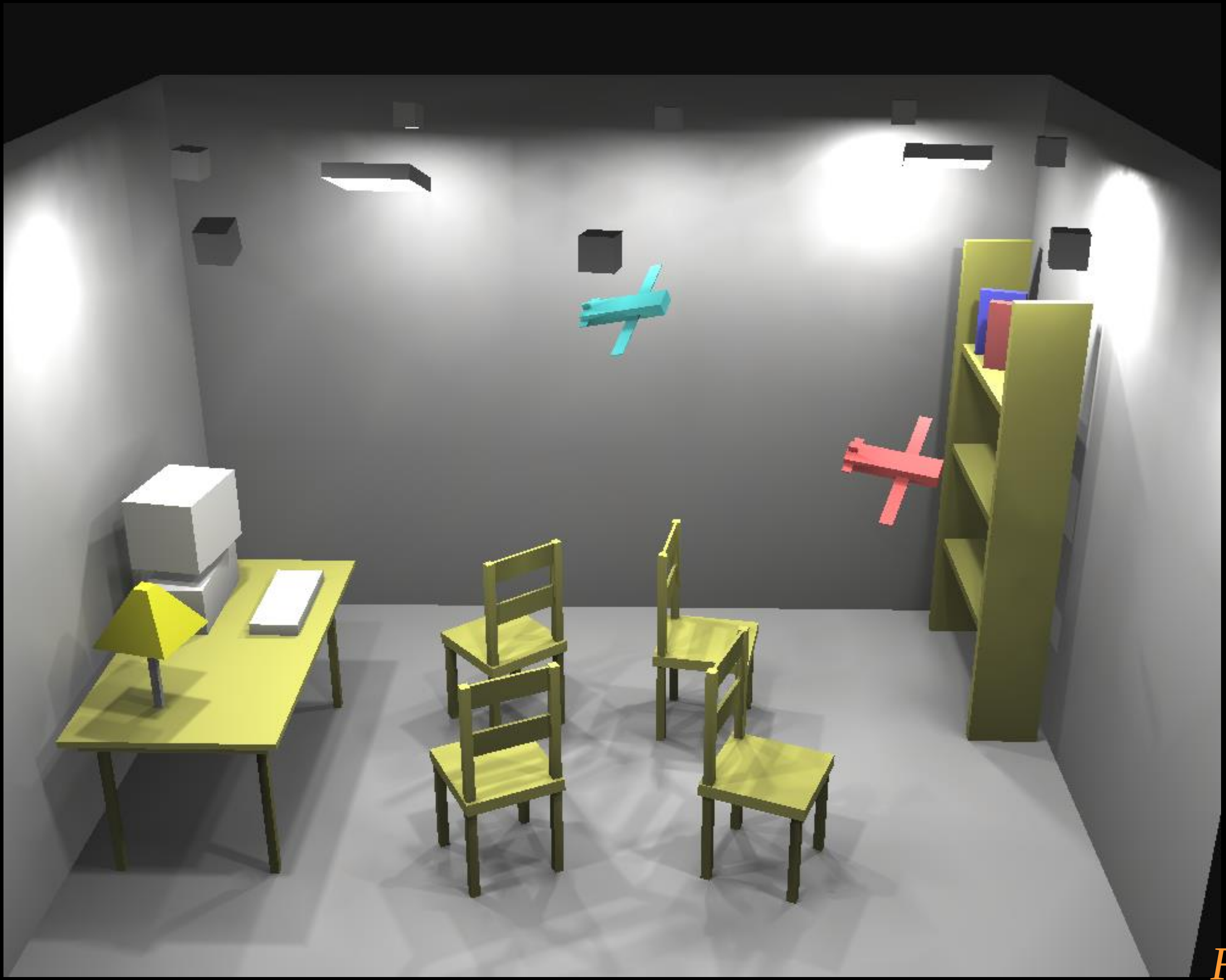
Refinement criterion

- perceptual metric
- error estimation

Fredo Durand

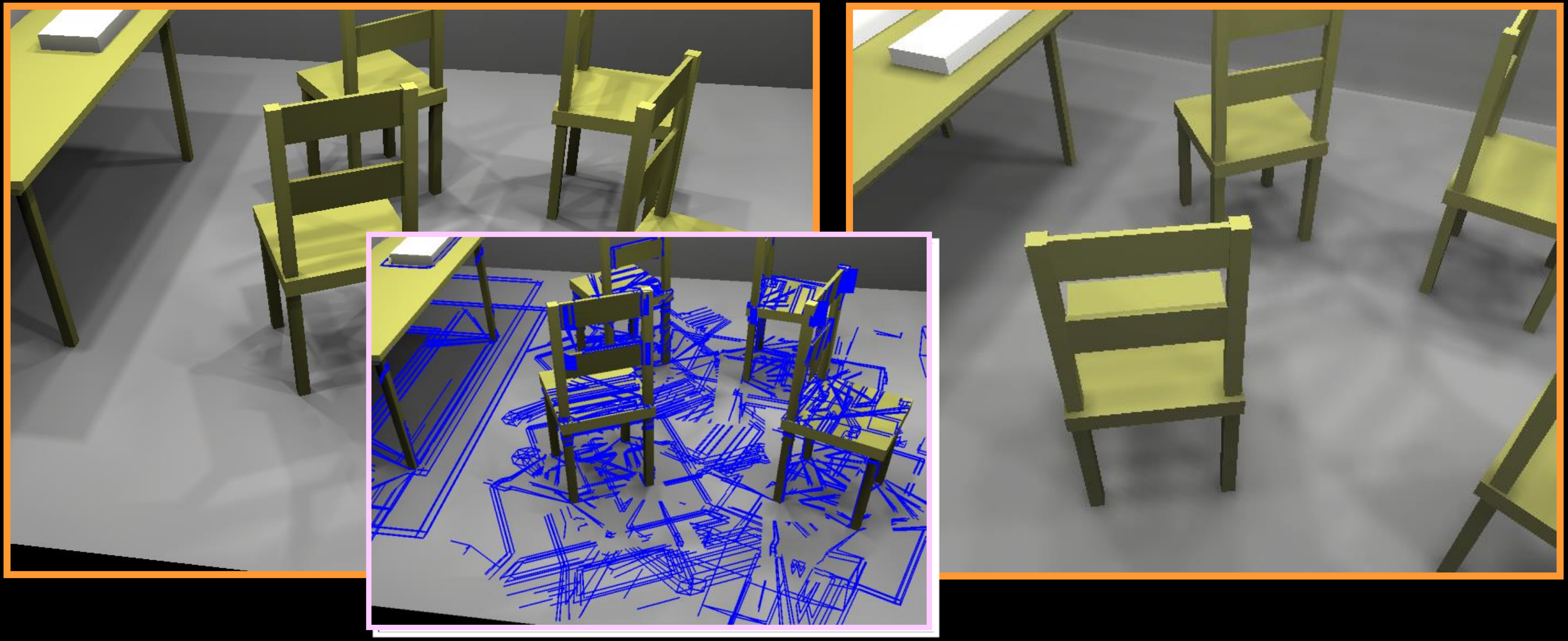
Radiosity with Visibility Skeleton

492 polygons : 10 minutes 23 seconds



Fredo Durand

Radiosity with Visibility Skeleton



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