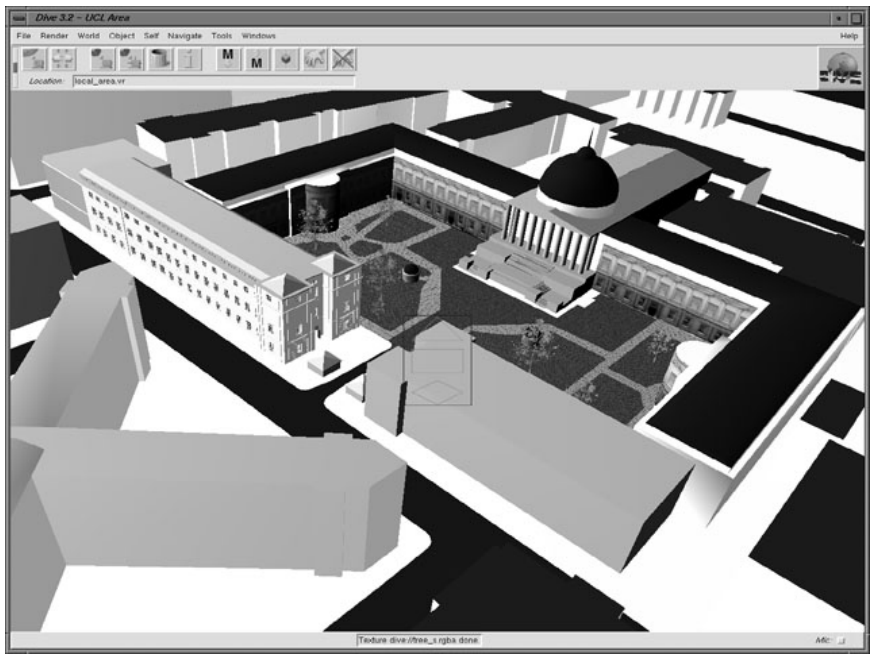
Visibility Streaming

for Network-based Remote Walkthroughs

Daniel Cohen-Or

Computer Science Department

Tel-Aviv University

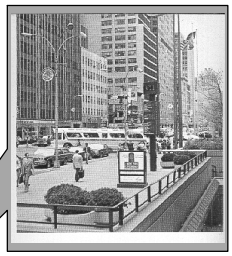


Remote network-based Walkthrough

Server

Huge 3D model

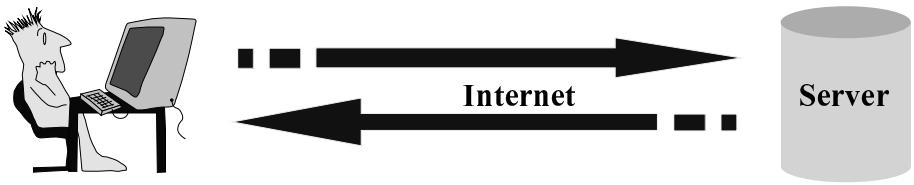
Client



Single frame of walkthrough

Remote Network-based Walkthrough

Network:
- low bandwidth
- inevitable network latency



Client
- low end thin computer
- interactive rate demand
- platform independent

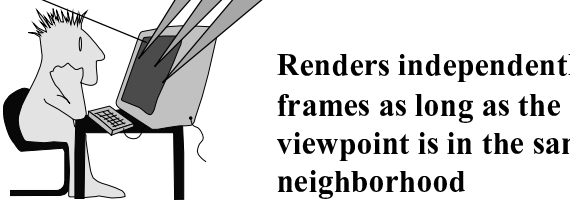
Server
- huge model database
- multi-client demands

The Visibility of a view cell

Server

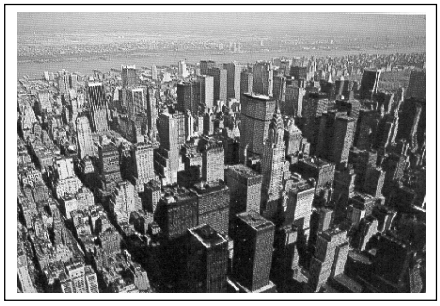
Sends the client a set of primitives valid for an entire neighborhood

Client

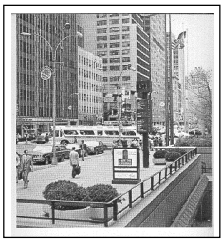


Renders independently all frames as long as the viewpoint is in the same neighborhood

The Visibility of Out-door scenes



Huge 3D model



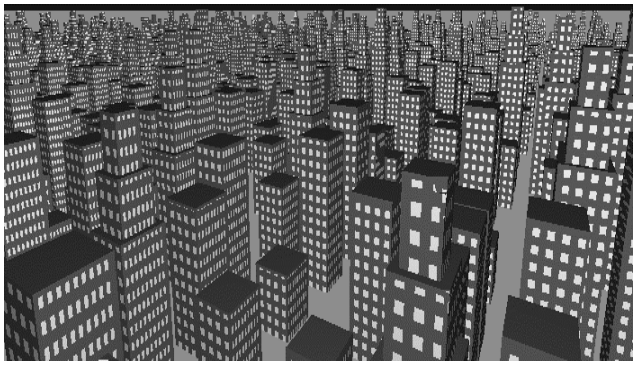
Single frame of walkthrough

Vast majority of the geometry in dense models is occluded from any single view point

The epsilon-visibility query

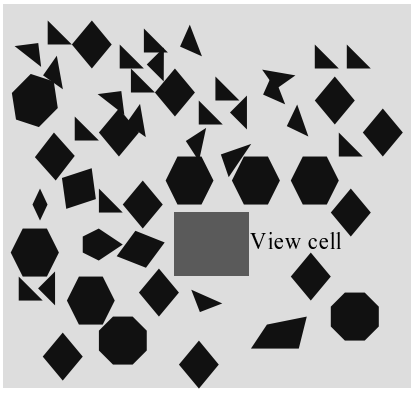
- Given a viewpoint the answer to an epsilon-visibility query is the set of all polygons visible from that viewpoint or from an epsilon-neighborhood of that point.

Out-doors scenes are much harder than In-doors scenes



No cells-and-portals

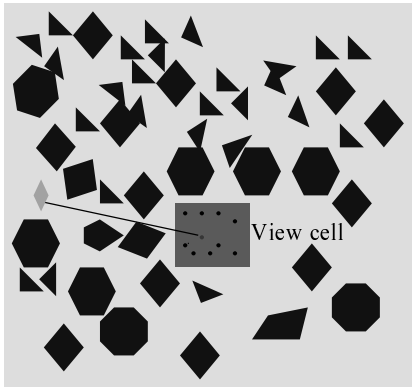
Conservative Viewspace Partitioning



The visibility set is valid for any view point within the view cell

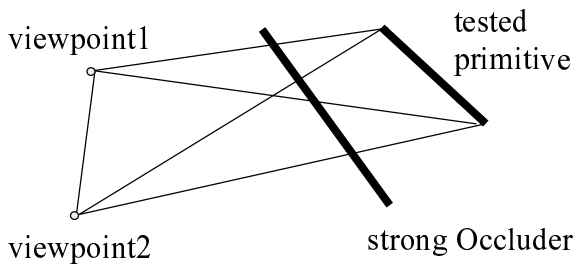
Conservative Viewspace Partitioning

Resolving visibility of an object from some random view points does not guarantee a correct interpolation of the visibility for the entire view cell

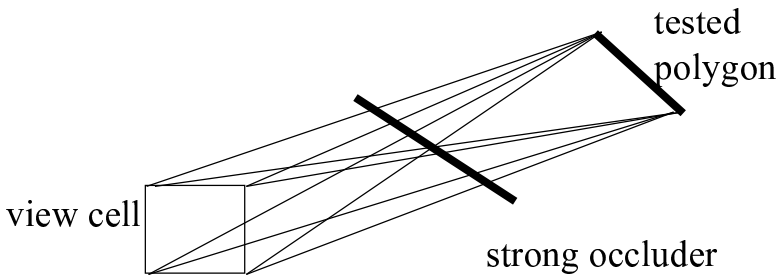


Instead of visibility interpolation occlusion interpolation may be used

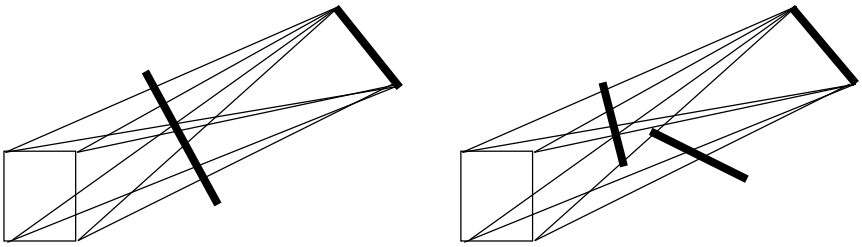
Strong Occlusion



View Cell Occlusion Interpolation



Strong and weak Occluders



Strong occluder

Weak occluders

The method effectiveness

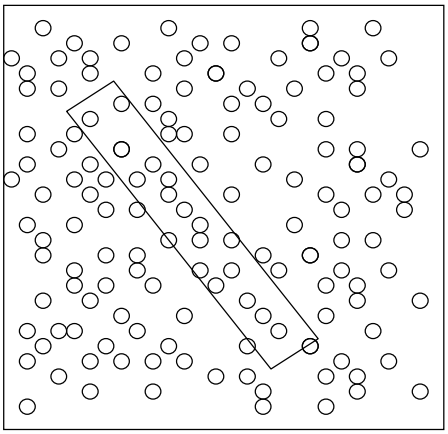
- the number of weakly occluded objects in each of our conservative visibility sets is relatively small, and
- we can construct the potentially visible sets efficiently (far more efficiently than it would have taken to compute the visibility set precisely).

The mathematical model

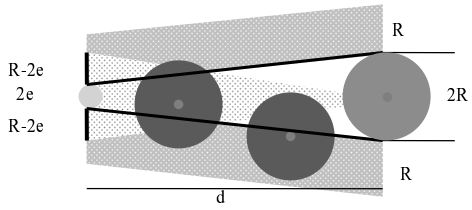
- Most occluded objects are strongly occluded
- Most objects are hidden
- An efficient view cell size

The mathematical model

Uniform distribution of spheres



Strong Occluder Analysis



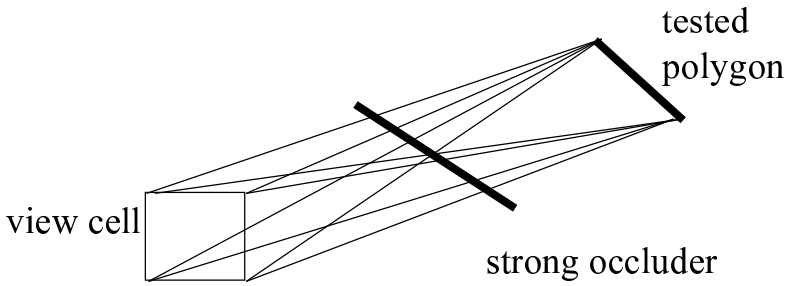
Strong Occluder Analysis

- The majority of occluded objects are strongly occluded
- The vast majority of distant objects are strongly occluded
- View cells must be smaller than the occluding objects.

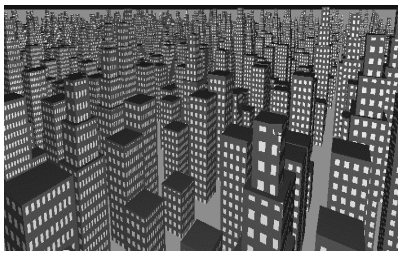
Optimizations

- **Space subdivision**
  - ray intersections by hierarchical ray traversal
- **Bounding boxes**
  - culls all polygons in an occluded box
- **Leading ray**
  - single ray traversal is sufficient to detect a strong Occluder
- **Redundant occluders**
  - an occluded object is a redundant Occluder
- **Shaft optimization**
  - rays not lying on the convex hull are redundant for conservative occlusion

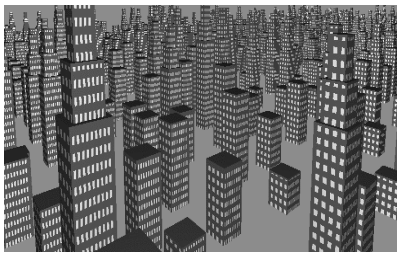
Optimizations (cont.)



Occlusion Test Models



Dense (city A)



Sparse (city B)

Total polygons:21280

Occlusion results

	Dense	Sparse
total polygons	21280	21280
strongly occluded polygons	20062 (94%)	18713 (88%)
total boxes	2128	2128
strongly occluded	1954 (91%)	1755 (82%)
first occluders	1427 (67%)	1225 (58%)

View cell ratio: 1/4

Strong occluders dominate occlusion of dense models

Optimizations results

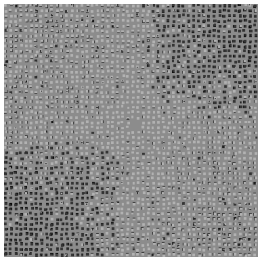
	Hierarchical model	bounding box	redundant occluders	redundant rays	full optimization
total time	30170	8900	8850	7040	7030
box occlusion	-	**7450**	**7340**	**5550**	**5540**
- ray traversal	-	3460	3530	3460	3420
- ray exclusion	-	-	-	640	630
- intersection	-	2350	2310	880	850
polygon occlusion	**29820**	**1410**	**1400**	**1410**	**1390**
- ray traversal	24070	940	970	990	850
- intersection	4000	380	290	280	400
misc. procedures	**350**	**40**	**110**	**80**	**100**
intersection tests					
- ray/box	-	135278	135089	45570	42414
- ray/polygon	517800	38648	38648	38648	38648

- times in milliseconds on SGI R4400
- model size = 21280 polygons

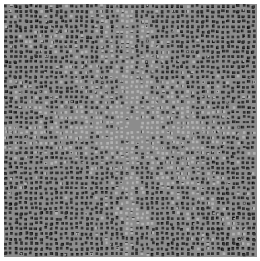
Cost effective analysis - the view cell size

- view cell size has great effect on the culling results
- smaller cells provide greater culling but are valid for a smaller area

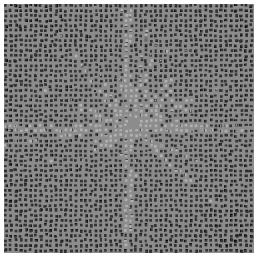
cell size	2.0x2.0	1.5x1.5	1.25x1.25	1.0x1.0	0.5x0.5	0.1x0.1
vis-set size	74%	62%	51%	34%	8%	4%
strongly occluded bounding boxes intersections	3%	7%	13%	29%	87%	94%
ray/box	132703	171463	208280	273029	222611	211715
ray/polygon	1175813	1346399	1385075	1204630	109210	48002



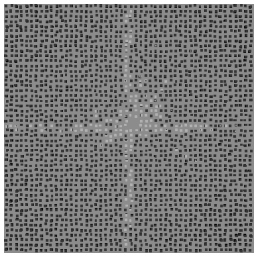
Cell size:1.0x1.0



Cell size:0.8x0.8



Cell size:0.5x0.5



Cell size:0.2x0.2

Conclusions

- Algorithm is suited for view space partitioning and occlusion culling of out-door scenes
- The vast majority of the occluded primitives has a strong occluder in dense models
- Neighborhood visibility superset may be regarded as a conservative aspect graph approximation of a given scene with much lower time and space complexity applicable for many computer graphics applications.

Open Problem

Non-convex occluders may be represented by a union of convex parts (not necessary disjoint)