Occlusion Tiling : *Procedural Generation of Occluding Tiles*

Dorian Gomez Pierre Poulin Mathias Paulin

DIRO - Université de Montréal, Canada
IRIT - Université de Toulouse III, France

About Virtual Worlds

Our purposes

- Create huge worlds.
- Reduce working time for artists.
- Reduce rendering time and memory consumption.

Our purposes

- Create huge worlds.
- Reduce working time for artists.
- Reduce rendering time and memory consumption.

Automatic and controlable content creation

Our purposes

- Create huge worlds.
- Reduce working time for artists.
- Reduce rendering time and memory consumption.

Automatic and controlable content creation
Hardware compliant data

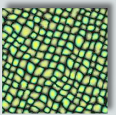
Summary

Procedural Modeling : Previous Work

- ***In a general sense : Automatic Data Generation***

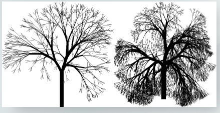
Procedural Modeling : Previous Work

- *In a general sense : Automatic Data Generation*
- ***In a computer graphics sense : Automatic Data Generation for geometries, texture,...***



Wei '00

Prusinkiewicz '90

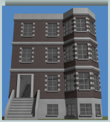


Maréchal '08

Deussen '98

Peytavie '09

Leblanc '11

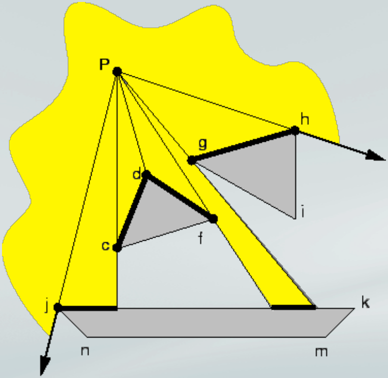


Peekaboo !

Reminder

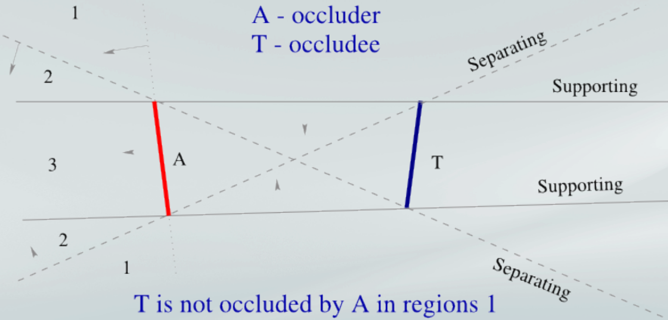
- **What ?** Reduce the number of primitives to increase framerate.
- **How ?** By not displaying hidden geometries.

From Point Visibility



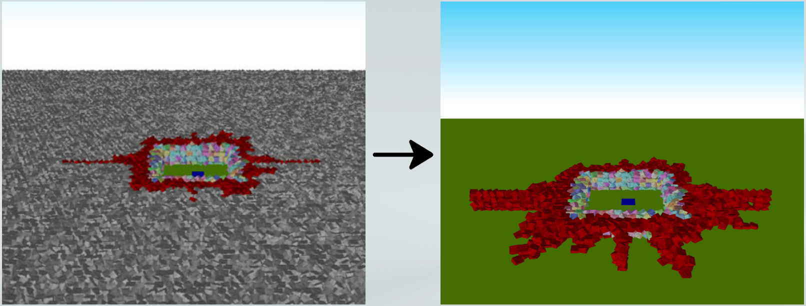
Blockers boundaries determine the discontinuities of the visibility function.

From Region Visibility : Vocabulary



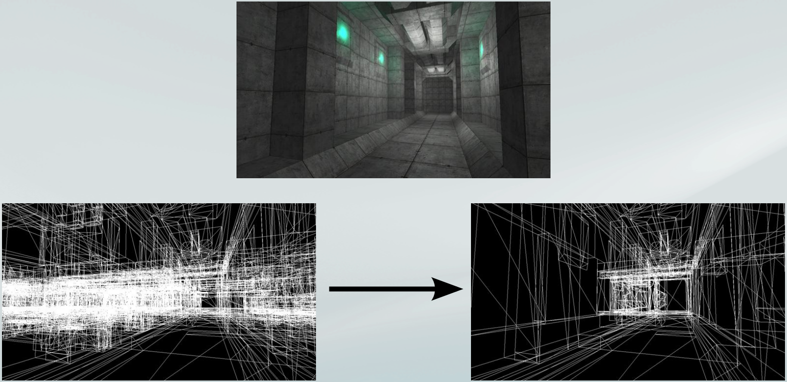
T is not occluded by A in regions 1
T is partially occluded by A in regions 2
T is fully occluded by in region 3

PVS (*Potentially Visible Sets*)



Visible objects from a view cell : potentially visible parts of the entire scene.

PVS (*Potentially Visible Sets*)



Potentially **V**isible **S**ets reduce display costs.

PVS (*Potentially Visible Sets*)

Huge **display savings** but also huge **memory costs**.

Ze solution

Goals : Feng Shui ´-_-`

Elaborate a procedural occlusion tiling technique.

Goals : Feng Shui ´-_-`

Elaborate a **procedural** occlusion tiling technique.

Goals : Feng Shui ´-_-`

Elaborate a procedural **occlusion** tiling technique.

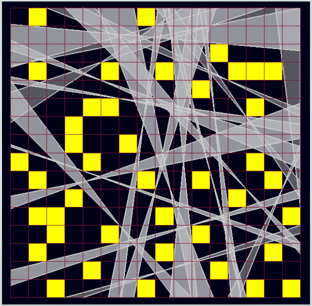
Goals : Feng Shui ´-_-`

Elaborate a procedural occlusion **tiling** technique.

Idea

- *A priori* hidden surface removal.
- Object procedural modeling VS object procedural positioning.
- Replacing **creation**, **display**, and **memory** costs by **construction precomputing** costs.

Occlusion Tiling : Definition

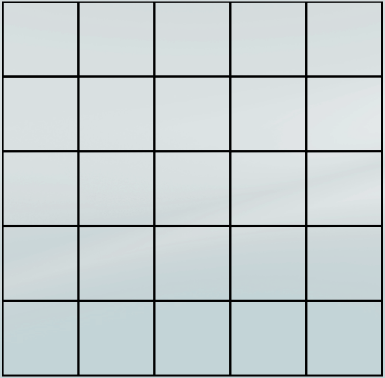


2D visibility through a tile :

- How to ensure any line of sight is blocked ?
- How to determine visible regions ?

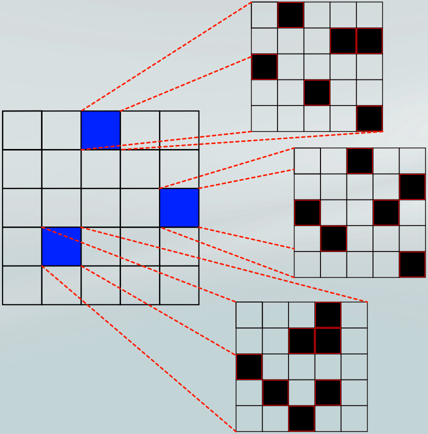
Occlusion Tiling

- **Square tiling**



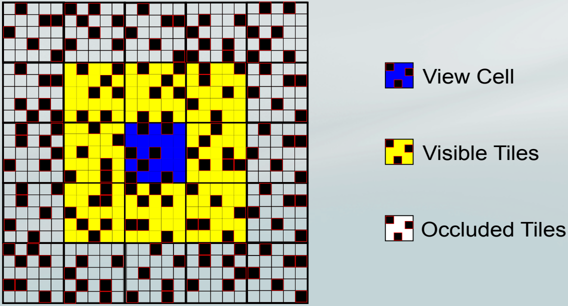
Occlusion Tiling

- Square tiling
- **1 tile = 1 set of blockers**



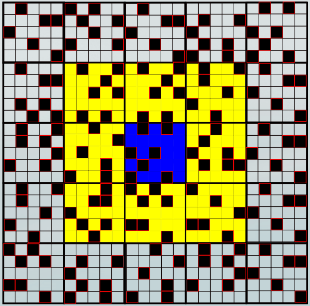
Occlusion Tiling

- Square tiling
- 1 tile = 1 set of blockers
- **1 tile = 1 independent view cell**



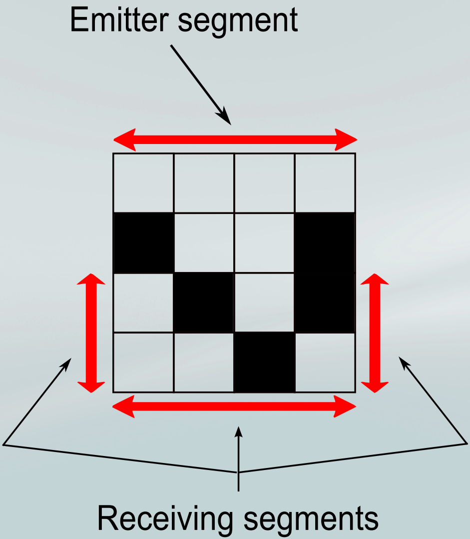
Occlusion Tiling

- Square tiling
- 1 tile = 1 set of blockers
- 1 tile = 1 independent view cell

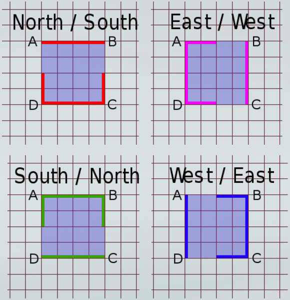


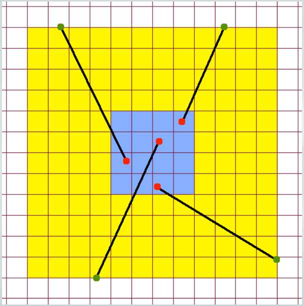
Goal : Ensure that all lines of sight coming from the view cell are blocked beyond the first ring of tiling neighborhood.

$\bar{u}$-occlusion Scheme



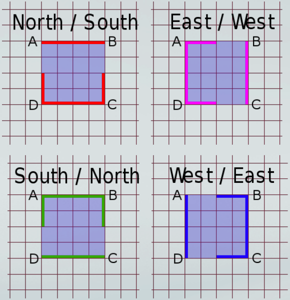
$\bar{u}$-occlusion Scheme

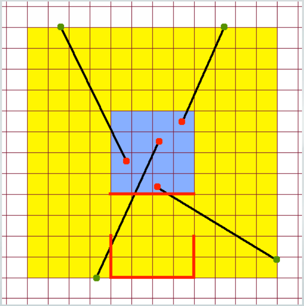




4 visibility constraints to ensure on each tile.

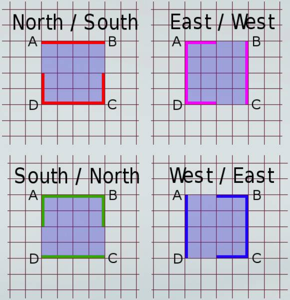
$\bar{u}$-occlusion Scheme

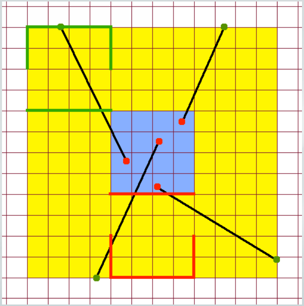




4 visibility constraints to ensure on each tile.

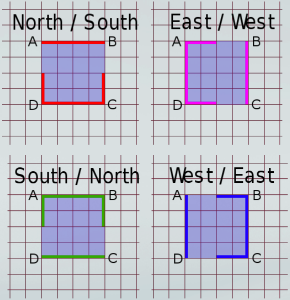
$\bar{u}$-occlusion Scheme

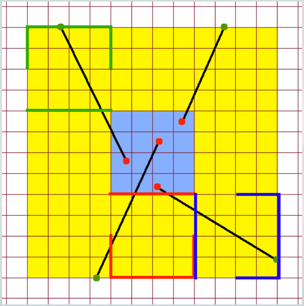




4 visibility constraints to ensure on each tile.

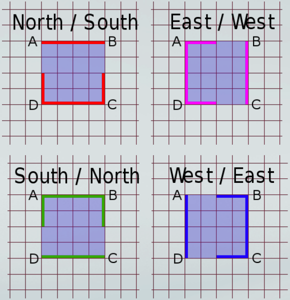
$\bar{u}$-occlusion Scheme

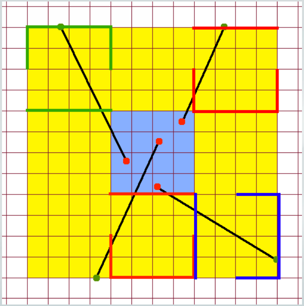




4 visibility constraints to ensure on each tile.

$\bar{u}$-occlusion Scheme



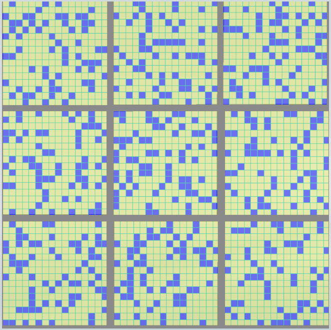


4 visibility constraints to ensure on each tile.

Generation Principle

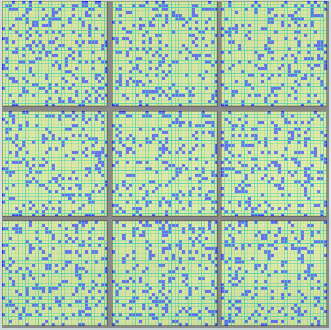
- Randomly generate a grid of blockers (uniform or according to user specification).
- Test occlusion.
- If occluding, add to the tiling set.

Results



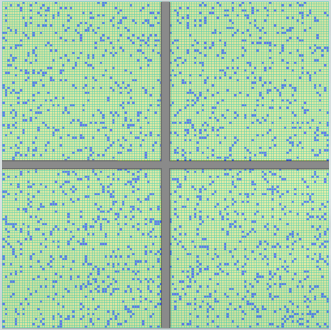
16×16 occluding tiles, occluding density : $1/5$ (20%)

Results



32×32 occluding tiles, occluding density : $1/5$ (20%)

Results



64×64 occluding tiles, occluding density : $1/7 (14,3\%)$

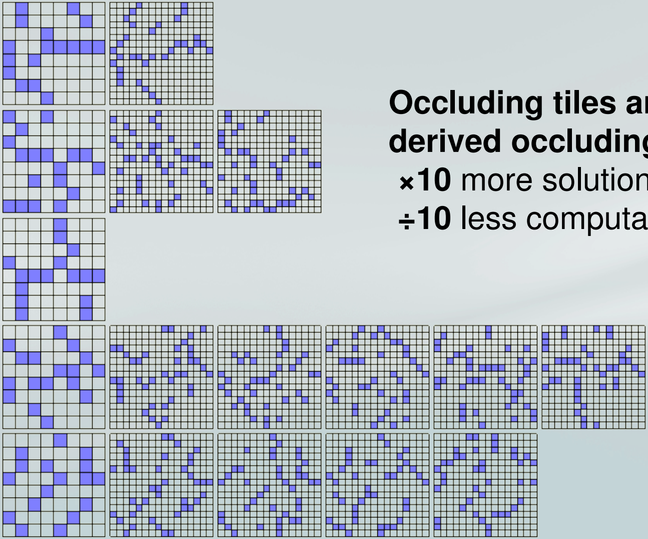
Results – 90 minutes

Dimension	% Occlusion Density	# Tested	# Positive
16×16	1/4 (25%)	260725	2960
16×16	1/5 (20%)	1064670	209
16×16	1/6 (16,6%)	2482720	5
16×16	1/7 (14,3%)	4399280	0
32×32	1/4 (25%)	801	329
32×32	1/5 (20%)	3710	112
32×32	1/6 (16,6%)	19608	9
32×32	1/7 (14,3%)	62224	0
64×64	1/4 (25%)	8	8
64×64	1/5 (20%)	10	10
64×64	1/6 (16,6%)	20	10
64×64	1/7 (14,3%)	59	4
64×64	1/8 (12,5%)	244	2
64×64	1/9 (11,1%)	909	0

Occlusion Tiling : Hierarchical Scheme

1. Generate a $n \times n$ random tile satisfying the 4 $\bar{u}$-schemes.
2. Double the resolution.
3. Mutate pixels in the $2n \times 2n$ resolution by randomly moving occluders ;
   Test new tiles for occlusion.
4. Repeat steps 2 and 3 with the resulting mutated tiles that passed the occlusion test until the desired resolution is reached.

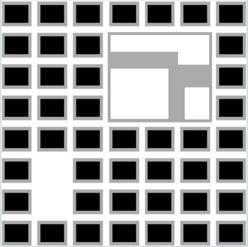
Occlusion Tiling : Hierarchical Scheme

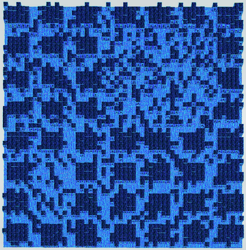


Occluding tiles and derived occluding tiles

- ×**10** more solutions
- ÷**10** less computation time

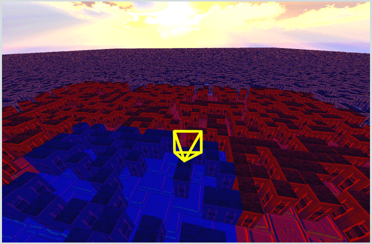
Occlusion Tiling : Customization

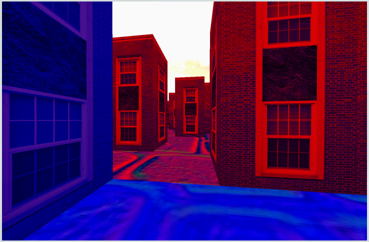




Each tile pixel is generated with a probability linearly derived from the source image.

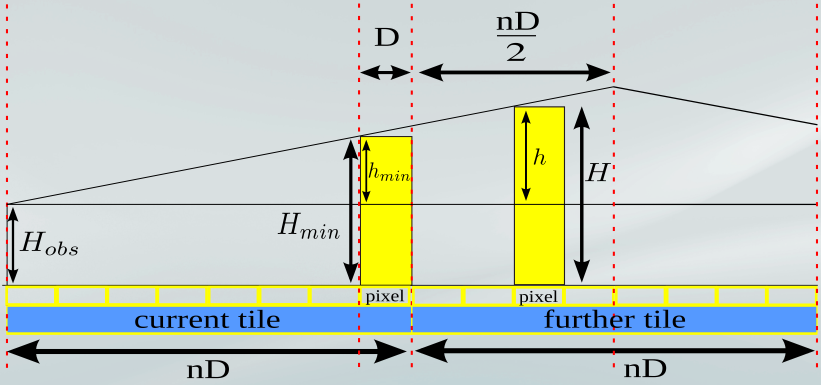
Occlusion Tiled City





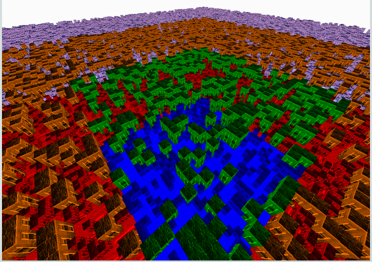
Aerial and ground views of our extruded 3D city from our 2D blockers.

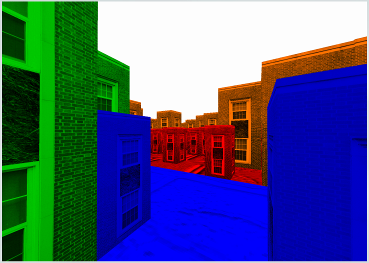
Height Variations



Height variation in building construction.

Multi-scale Occlusion Tiled City





Aerial and ground views of our extruded 3D city from our 2D blockers.

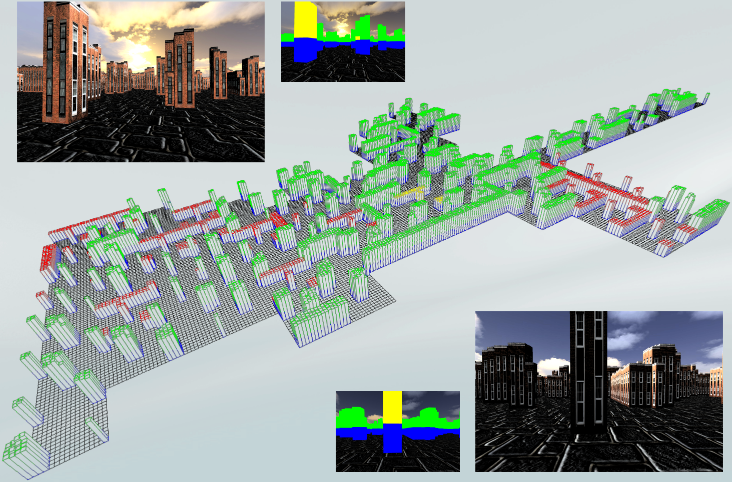
Zen Attitude

1. Procedural Modeling
2. Visibility & Occlusion Culling
3. Occlusion Tiling
4. Conclusion

Conclusion

- Increasing visual field occlusion.
- *A priori* Hidden Surface Removal.
- Replacing creation, rendering and memory costs by construction of visibility relations between tiles.

Current Work



Visibility propagation.

Future Work

- Other types of worlds.
- Other shapes of tiles.
- Non planar worlds.
- 3D visibility computations.

}<((('> Thank you ! <')))>{

email me at dorian.gomez@irit.fr
for any question concerning fish cod(e) recipe.

Annex

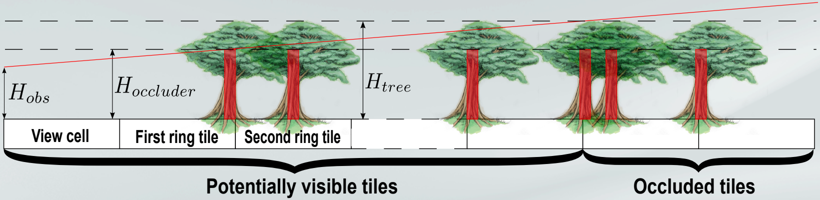


FIGURE: Occlusion Tiling : instanciation by trees.