Exploiting Temporal and Spatial Coherence in Hierarchical Visibility Algorithms

Jiří Bittner and Vlastimil Havran
Centre of Applied Cybernetics
Czech Technical University in Prague

- Introduction, hierarchical visibility.
- Hierarchy updating, conservative hierarchy updating.
- Visibility propagation.
- Results, conclusion, future work.
Hierarchical Visibility Algorithms

Speedup rendering of large scenes by visibility culling

Spatial hierarchy
- kD–tree, octree, OBB tree

Visibility test of a region:
- visible (V)
- invisible (I)
- partially visible (PV)

- for each viewpoint start at the root of the hierarchy
- recursively apply visibility test on PV regions
Hierarchical Visibility Algorithms

Image space
- Hierarchical z–buffer. Greene ’93.
- Hierarchical polygon tiling. Greene ’96.

Object space
Hierarchy Updating

- idea: avoid repeated visibility tests of **interior** hierarchy nodes
- **cut of the hierarchy** used as a starting level in next frame
- **propagate** visibility changes up the hierarchy
Conservative Hierarchy Updating

Motivation

- hierarchy updating saves almost 50% of visibility tests
- we can save more: partially visible and visible nodes marked visible with certain probability $p$

Properties

- keeps conservative behaviour of the algorithm
- not all changes from PV or V to I are captured
- higher $p$ – less visibility tests, but usually more visible nodes and more objects to render
Visibility Propagation

- **front–to–back** processing of the hierarchy nodes
- use visibility classification of already processed nodes
- combine their visibility – if *visible* or *invisible* we are done
Visibility Propagation – cont.

- use neighbour links (ropes) to locate all processed neighbours
Modified Visibility Pipeline

- Determine Visibility
- Refine Visibility
- Test Visibility
- Conservative Hierarchy Updating
- Visibility Propagation
- Culling
- View-Frustum Culling
- Estimated Computational Cost
- Refine Visibility

Legend:
- Invisible
- Partially Visible
- Visible
- Probabilistic Path

Diagram:
- Box with arrows and flowchart indicating the pipeline steps.
Results

Test path through building interior

Visibility classification
Walk through a building interior
Avg. number of visibility tests vs. relative walking speed

![Graph showing the relationship between AvgVisNodes and MovieStep for different conditions A, B, C, and D. The graph illustrates how AvgVisNodes increase with MovieStep, with each condition showing a different trend.](image-url)
Avg. frame time [ms] vs. relative walking speed

![Graph showing the relationship between Avg. frame time and relative walking speed. The graph includes lines for different conditions labeled A, B, C, and D.](image)
Results – cont.

Avg. frame time [ms] vs. probability $p$

![Graph showing AvgTime vs. VSkipProb]
Results – summary

Hierarchy updating
■ saves almost 50% of visibility tests

Conservative hierarchy updating
■ another 20% of visibility tests
■ faster visibility changes ⇒ more objects to render

Visibility propagation
■ successful on only few rather large regions
Conclusion and Future Work

Hierarchy updating

- very simple and useful
- eliminates overhead induced by hierarchy, keeps advantages

Conservative hierarchy updating

- useful for certain settings
- how to estimate optimal probability of skipping visibility test?

Visibility propagation

- not very successful
- spatial coherence already exploited well by using the hierarchy!
- determine that region is PV with high probability?
QUESTIONS?
CONSTRAINED DFS

ROOT

TERMINATION NODES

U N N 1 2 3

DFS ROOT ROPE

U 3 2 1