Visibility Map for Global Illumination in Point Clouds

R. Goradia\textsuperscript{1} A. Kanakanti\textsuperscript{1} S. Chandran\textsuperscript{1} A. Datta\textsuperscript{2}

\textsuperscript{1}Indian Institute of Technology, Bombay
Mumbai, India
{rhushabh, kanil, sharat}@cse.iitb.ac.in

\textsuperscript{2}University of Western Australia
Perth, Australia
datta@csse.uwa.edu.au

2.12.2007
Overview

1. Problem Definition

2. Visibility Map
   - What is a V-map?
   - Construction of a V-map

3. Results

4. Conclusion and Future Work
Overview

1. Problem Definition

2. Visibility Map
   - What is a V-map?
   - Construction of a V-map

3. Results

4. Conclusion and Future Work
Problem Definition

Problem Statement

To compute a Visibility Map (V-map) for complex scenes represented as point-models, for the purpose of global illumination.
Application Domains
Visibility Between Point Pairs

View Independent Visibility calculation between point pairs is essential to give correct GI results as a point receives energy from other point only if it is visible.
Visibility Between Point Pairs

VISIBILITY IN POLYGONAL MODELS  VISIBILITY IN POINT MODELS
Hierarchical Visibility approach helps in achieving faster GI solution (eg. hierarchical radiosity).
## Problem Definition

<table>
<thead>
<tr>
<th>Model</th>
<th>Points (millions)</th>
<th>$N^2$ links (millions)</th>
<th>V-Map Links (millions)</th>
<th>% Decrease</th>
<th>Memory(MB) $N^2$ links</th>
<th>Memory(MB) V-Map links</th>
<th>Build V-Map Time(secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECR</td>
<td>0.1</td>
<td>1.4</td>
<td>0.27</td>
<td>79.5%</td>
<td>5.35</td>
<td>1.09</td>
<td>20.6</td>
</tr>
<tr>
<td>PCR</td>
<td>0.14</td>
<td>3.85</td>
<td>0.67</td>
<td>82.62%</td>
<td>15.43</td>
<td>2.68</td>
<td>23.8</td>
</tr>
<tr>
<td>BUN</td>
<td>0.15</td>
<td>1.53</td>
<td>0.38</td>
<td>74.64%</td>
<td>6.09</td>
<td>1.5</td>
<td>21.7</td>
</tr>
<tr>
<td>DRA</td>
<td>0.55</td>
<td>2.75</td>
<td>0.43</td>
<td>84.54%</td>
<td>11.0</td>
<td>1.7</td>
<td>23.5</td>
</tr>
<tr>
<td>BUD</td>
<td>0.67</td>
<td>1.58</td>
<td>0.39</td>
<td>74.75%</td>
<td>6.33</td>
<td>1.6</td>
<td>23.9</td>
</tr>
<tr>
<td>GAN</td>
<td>0.15</td>
<td>1.56</td>
<td>0.38</td>
<td>75.64%</td>
<td>6.2</td>
<td>1.55</td>
<td>22.0</td>
</tr>
<tr>
<td>GOD</td>
<td>0.17</td>
<td>1.62</td>
<td>0.4</td>
<td>75.31%</td>
<td>6.4</td>
<td>1.63</td>
<td>22.9</td>
</tr>
</tbody>
</table>

- ECR – Empty Cornell room
- PCR – Packed Cornell room
- BUN – Bunny in Cornell room
- DRA – Dragon in Cornell room
- BUD – Buddha in Cornell room
- GAN – Indian God Ganesha in a Cornell room
- GOD – Indian Goddess Satya in a Cornell room
Overview

1. Problem Definition

2. Visibility Map
   - What is a V-map?
   - Construction of a V-map

3. Results

4. Conclusion and Future Work
The visibility map for a tree is a collection of visibility links for every node in the tree.

The visibility link for any node $p$ is a list $L$ of nodes.

Every point in any node in $L$ is guaranteed to be visible from every point in $p$. 
What is a Visibility Map (V-map)?

- The *visibility map* for a tree is a collection of visibility links for every node in the tree.
- The *visibility link* for any node $p$ is a list $L$ of nodes.
- Every point in any node in $L$ is guaranteed to be visible from every point in $p$.
What is a Visibility Map (V-Map)?

With respect to a node at any level,
- **Partially Visible**
- **Completely Invisible**
- **Completely Visible**

- **Green** --- Completely Invisible
- **Blue** --- Completely Visible
- **Brown** --- Partially Visible
Visibility Map Queries?

Visibility maps entertain efficient answers to the following queries.

1. Is point \( x \) visible to point \( y \)?
2. What is the visibility status of \( u \) points around \( x \) with respect to \( v \) points around \( y \)?
   - Repeat a “primitive” point-point visibility query \( uv \) times
   - V-Map gives the answer with \( O(1) \) point-point visibility queries.
3. Given a point \( x \) and a ray \( R \), determine the first object of intersection.
4. Is point \( x \) in the shadow (umbra) of a light source?

All the above queries are done with a simple traversal of the octree.
V-map Construction Algorithm

- Initialize the o-IL of every node to be its seven siblings
V-map Construction Algorithm

procedure OctreeVisibility(Node A)
    for each node B in old interaction list (o-IL) of A do
        if NodetoNodeVisibility(A,B) == VISIBLE then
            add B in new interaction list (n-IL) of A
            add A in new interaction list (n-IL) of B
        end if
        remove A from old interaction list (o-IL) of B
    end for
    for each C in children(A) do
        OctreeVisibility(C)
    end for

- V-map constructed by calling initially for the root, which sets up the relevant visibility links in n-IL
- NodetoNodeVisibility(A,B) constructs the visibility links for all descendants of A w.r.t all descendants of B (and vice-versa) at the best (i.e. highest) possible level. This ensures an optimal structure for hierarchical radiosity as well as reduces redundant computations.
V-map Construction Algorithm

- LEVEL 0
- LEVEL L−4
- LEVEL L−3
- LEVEL L−2
- LEVEL L−1
- LEVEL L

Graph showing the construction of a visibility map (V-map) with nodes labeled A and B. The algorithm involves computing visibility at different levels (L−4 to L) and connecting nodes accordingly.
V-map Construction Algorithm

[root]

LEVEL 0

LEVEL L–4

LEVEL L–3

LEVEL L–2

LEVEL L–1

LEVEL L

LOOK-UP

A

B

GRAPHITE 2007

Visibility Map

Construction of a V-map

IIT-Bombay
V-map Construction Algorithm
V-map Construction Algorithm

Complete Visibility

New Visibility Link
V-map Construction Algorithm

PARTIAL VISIBILITY

LEVEL 0
LEVEL L−4
LEVEL L−3
LEVEL L−2
LEVEL L−1
LEVEL L

NEW VISIBILITY LINK

ROOT

A

B

LEVEL L
Consider centroid and **NOT** leaf center
Leaf-Leaf Visibility Algorithm
Leaf-Leaf Visibility Algorithm

- Distance $R$ is unique for each leaf and depends on the distribution of points in the leaf ($R$ is not a user-input).
- Imposing a strict visibility condition balances the leniency introduced.
- Faster, as we exit on finding the first potential occluder.
- Dense point models help in achieving better results.

**NOTE:** We perform this visibility computation (with help of averaged normals) only for the leaves. There are no average normals defined for internal nodes of the tree.
Point Pair Visibility

Visibility between points $p$ and $q$ using the Bresenham line algorithm.
Assume $N = \Theta(n^2)$, $n$ = points in input model.

Visibility problem provides answer to $N$ pairwise queries. Hence we measure the efficiency w.r.t $N$.

Octree Visibility has the recurrence: $T(h) = 8T(h - 1) + N$ (for a Node $A$ at height $h$).

Complexity for $\text{NodeToNodeVisibility}(A, B)$ is determined by the calls to point-pair visibility algorithm.

Assuming the latter to be $O(1)$, the recurrence relation for the former is $T(h) = 64T(h - 1) + O(1)$.

The overall algorithm consumes a small amount of memory (for storing $M$) during runtime.
Overview

1. Problem Definition

2. Visibility Map
   - What is a V-map?
   - Construction of a V-map

3. Results

4. Conclusion and Future Work
Qualitative Results: Visibility Correctness
Qualitative Results: Visibility Correctness
Qualitative Results: Visibility Correctness
Qualitative Results: Visibility Correctness
Qualitative Results: Visibility Correctness
Qualitative Results: Visibility Correctness
Qualitative Results: Global Illumination
Qualitative Results: Global Illumination
Qualitative Results: Global Illumination
Qualitative Results: Global Illumination
## Quantitative Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Points (millions)</th>
<th>$N^2$ possible links (millions)</th>
<th>V-Map Links (millions)</th>
<th>% Decrease</th>
<th>Memory (MB) $N^2$ links</th>
<th>Memory (MB) V-Map links</th>
<th>Build V-Map Time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECR</td>
<td>0.1</td>
<td>1.4</td>
<td>0.27</td>
<td>79.5%</td>
<td>5.35</td>
<td>1.09</td>
<td>20.6</td>
</tr>
<tr>
<td>PCR</td>
<td>0.14</td>
<td>3.85</td>
<td>0.67</td>
<td>82.62%</td>
<td>15.43</td>
<td>2.68</td>
<td>23.8</td>
</tr>
<tr>
<td>BUN</td>
<td>0.15</td>
<td>1.53</td>
<td>0.38</td>
<td>74.64%</td>
<td>6.09</td>
<td>1.5</td>
<td>21.7</td>
</tr>
<tr>
<td>DRA</td>
<td>0.55</td>
<td>2.75</td>
<td>0.43</td>
<td>84.54%</td>
<td>11.0</td>
<td>1.7</td>
<td>23.5</td>
</tr>
<tr>
<td>BUD</td>
<td>0.67</td>
<td>1.58</td>
<td>0.39</td>
<td>74.75%</td>
<td>6.33</td>
<td>1.6</td>
<td>23.9</td>
</tr>
<tr>
<td>GAN</td>
<td>0.15</td>
<td>1.56</td>
<td>0.38</td>
<td>75.64%</td>
<td>6.2</td>
<td>1.55</td>
<td>22.0</td>
</tr>
<tr>
<td>GOD</td>
<td>0.17</td>
<td>1.62</td>
<td>0.4</td>
<td>75.31%</td>
<td>6.4</td>
<td>1.63</td>
<td>22.9</td>
</tr>
</tbody>
</table>
Overview

1. Problem Definition

2. Visibility Map
   - What is a V-map?
   - Construction of a V-map

3. Results

4. Conclusion and Future Work
Conclusion

- The lack of surface information in point models creates difficulties in operations like generating global illumination effects and computing point-pair visibility.
- Point-to-Point Visibility is arguably one of the most difficult problems in rendering since the interaction between two primitives depends on the rest of the scene.
- One way to reduce the difficulty is to consider clustering of regions such that their mutual visibility is resolved at a group level (V-Map).
- Visibility Map data structure we propose enables efficient answer to common rendering queries.
- In this paper, we have given a novel, provably efficient, hierarchical, visibility determination scheme for point based models.
- By viewing this visibility map as a ’preprocessing’ step, photo-realistic global illumination rendering of complex point-based models have been shown.
- If analyzed properly, the visibility algorithm is embarrassingly parallel.
The digital michelangelo project: 3D scanning of large statues.

Marc Levoy and Turner Whitted.
The use of points as a display primitive.
Thank you for your time!

Questions?