POTENTIALLY VISIBLE HIDDEN-VOLUME RENDERING FOR MULTI-VIEW WARPING

ACM TRANS. GRAPHICS, 42(4), 86

JANGHUN KIM AND SUNGKIL LEE*
SUNGKYUNKWAN UNIVERSITY, SOUTH KOREA
Novel-View Warping (in Real-Time Rendering)

- Single-image warping can introduce disocclusions (or holes), which are not rendered for the known view.
Multi-Fragment Rendering (MFR)

• A-buffer, $k$-buffer, and Depth Peeling (DP)
  • typically used for transparency or global illumination
  • can handle disocclusions by including hidden fragments in warping.

* A-buffer [Carpenter 1984; Yang et al. 2010], $k$-buffer [Bavoil et al. 2007], and Depth Peeling (DP) [Everitt 2001; Mammen 1989]
Redundancy of MFR

• Many of the fragments
  • are **invisible** from any novel views and
  • do **not contribute** to the final outcome.
Challenges

• Reduction of redundancy in MFR and warping

• However,
  • In general, disocclusions are revealed after warping.
  • Also, multi-view warping even requires to be iterated.
Our Goal

• Early test of the visibilities for fragment culling in MFR
  • Capture fragments for the known views, but pre-test their visibilities against novel views.
  • in particular for Depth Peeling (DP)
Previous Solutions

- Potentially Visible Set (PVS)
  - Offline visibility culling [Teller and Séquin 1991; Cohen-Or et al. 2003]
- Umbra fragment culling in Depth-of-Field (DOF) rendering
  - Based on pixel-as-geometry occluders [Lee et al. 2010]
Our Contributions

• Potentially Visible Hidden Volume (PVHV)
  • **Definition and modeling of PVHVs** for MFR
  • PVHVs are 3D volumes that are hidden at the known source view but visible at novel views.

• Effective Depth Peeling (EDP) Algorithm
  • PVHV-based **on-the-fly** real-time fragment culling
Benefit of Our Solution

- Multi-view warping with EDP
  - produce **fewer fragments/layers** for the same quality
  - higher rendering **performance**
  - **higher memory efficiency** (packing from sparser fragments)
Potentially Visible Hidden Volumes (PVHVs)
Definition of PVHV:

- $O_s$: Hidden volume from $s$
Definition of PVHV:

- $V_n$ : Visible volume from $n$
Definition of PVHV:

- PVHV: $H_s(n) = O_s \cap V_n$
PVHVs for Linear Views

• The simplest shape of a PVHV

• Key elements for finding PVHV
  • *inner blocker fragment* $f$
  • *two edge rays* passing through $f$
Local Circle of Confusion (LCOC)

- PVHV is characterized by LCOC (---)
- Similar to COC in DOF rendering
- LCOC Radius from triangle similarity:

\[ R(p, f) = \left( \frac{p_z - f_z}{f_z} \right) E \]

\( p \): Incoming fragment to test
\( f \): Inner blocker fragment
\( E \): distance from \( s \) to \( n \)
Other Types of PVHV

• It is possible to extend PVHVs to point and areal view types.
Effective Depth Peeling (EDP)
Depth Peeling

• Standard Depth Peeling (DP) [Everitt 2001]
  • captures all hidden fragments

• Our Effective Depth Peeling (EDP)
  • DP + PVHV-driven fragment culling
Problem for Efficient Implementation

• Problem:
  • PVHVs require to find edges.
  • **Finding edges explicitly** needs to be avoided for efficiency.
  • We just want to test edge exists (rather than where the edges are).
EDP: Backward-Search Algorithm

- Actually, we need $f_z$ for LCOC.
  - This needs precise edge detection.

- Assuming blocker is almost flat
  - blocker depth: $q_z \approx f_z$

- Search bound for finding edges
  - LCOC projection ( ) onto blocker
  - **When an edge exists**, fragment $p$ is in PVHV; $p$ has be kept during DP.
Experimental Analysis
Test Configurations

- Experimental configurations
  - NVIDIA GTX 3090, Full-HD (1920×1080), OpenGL 4.6
  - Three camera-animated scenes (1.25M—242M faces, 2.2—36K objects)
Fragments in the First Hidden Layer

- Resulting fragments are already sparse
  - Most fragments behind large occluders are discarded well

Baseline DP [Everitt 2001]  Umbra DP [Lee et al. 2010]  Effective DP [DP+PVHV; ours]

Fragments: 0.994  Fragments: 0.915  Fragments: 0.392
Performance in Multi-View DOF Warping

• Speed-ups with respect to Standard DP (1024 views)
  • Ruins scene: 3.2-3.4×
  • Safari scene: 2.3-2.9×
  • Satellites scene: 2.4×
Memory Efficiency

• Packed EDP (PEDP)
  • GPU-based linked list, storing only sparse fragments, can greatly reduce the memory consumption.

<table>
<thead>
<tr>
<th>Scene</th>
<th>E (mm)</th>
<th>Memory (MB)</th>
<th>Memory Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PEDP</td>
<td>EDP</td>
<td>UDP</td>
</tr>
<tr>
<td>RU</td>
<td>50</td>
<td>17.7</td>
<td>316.4 (10)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>26.5</td>
<td>379.7 (12)</td>
</tr>
<tr>
<td>SF</td>
<td>50</td>
<td>16.4</td>
<td>221.5 (07)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>20.5</td>
<td>316.4 (10)</td>
</tr>
<tr>
<td>ST</td>
<td>50</td>
<td>9.1</td>
<td>94.9 (03)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>20.6</td>
<td>443.0 (14)</td>
</tr>
</tbody>
</table>
Thank you for attention!