A Layered Depth-of-Field Method for Solving Partial Occlusion

David C. Schedl  Michael Wimmer
schedl@cg.tuwien.ac.at  wimmer@cg.tuwien.ac.at

Institute of Computer Graphics and Algorithms
Vienna University of Technology

WSCG – 26th June 2012
Real life partial occlusion

f=18 mm, N=4
Intro/Overview

- depth-of-field approximation
  - post processing
  - partial occlusion in realtime
Thin lens

\[ \text{sensor} \]

\[ d_{\text{coc}} \]

\[ q \quad p \quad q' \quad p' \]

\[ z_p \quad z_{\text{focus}} \]
Previous work

- Potmesil and Chakravarty, 1981
  - CoC - equation
  - first post-processing method
  - blur according to CoCs
  - still a reference
  - artifacts
Artifacts

- color bleeding:

- depth discontinuity:
Partial Occlusion

- pinhole vs. finite aperture
Partial Occlusion

- pinhole:
- finite aperture:
Previous work – solve partial occlusion

- **non-realtime:**
  - ray-tracing (Cook et al., 1984)
  - Accumulation B. (Haeberli and Akeley, 1990)

- **layered methods:**
  - Kraus and Strengert, 2007
    - occluded scene content only interpolated
  - Lee et al., 2010
    - image composition via ray traversal
    - simulate more lens effects
    - more complex and slower than ours
Our Method

David C. Schedl
Our Method
Rendering & Depth Peeling

Images

$I_0$

$I_1$

$I_2$
Matting – functions

weight

depth

David C. Schedl
Matting – functions

weight

depth

David C. Schedl
Matting – layers

Input

$I_0$

$I_1$

$I_2$
Matting – layers

Input

$I_0$ → $I_1$ → $I_2$ → $L_7$
Matting – layers

Input

\( I_0 \)

\( I_1 \)

\( I_2 \)

Layers

\( L_6 \)

\( L_7 \)

\( L_8 \)

\( L_K \)
Matting – layers

Input

$I_0$

$I_1$

$I_2$

Layers

$L_6$

$L_7$

$L_8$

$L_K$

David C. Schedl
Blurring

- uniformly blur layers
- Gaussian filter

$H_0 \times L_0 = L'_0$

$H_1 \times L_1 = L'_1$
Blurring

\[ L'_{k_{\text{focus}}} = L_{k_{\text{focus}}}' = L_0 * H_0 \]

\[ L_1' = L_1 * H_1 \]
Blurring

\[ L'_{k_{\text{focus}}} = L_{k_{\text{focus}}} \]

\[ H_8 \]

\[ H_9 \]

\[ L'_8 \]

\[ L'_9 \]
Compose

- alpha-blend back to front
Optimization

- reduce filter width
- recursive Gaussians

\[ \sigma_a \ast \sigma_b = \sqrt{\sigma_a^2 + \sigma_b^2} \]
\[ \hat{H}_1 \]

\[ \hat{L}_1 \]

\[ L_2 \]
Optimization - front

\[ \hat{H}_2 \]

\[ \hat{L}_2 \]
Optimization - Compositing

\[ I' \]

\[ \hat{L}_i L_{k_{\text{focus}}} \]
Results - Homunculus

f=100mm, N =1.4, focus=18 500 mm, 17 layers, 3x DP
Results - Dragons

f=100mm, N =1.4, focus=3 000 mm, 22 layers, 3x DP
## Results – Benchmarks

- Intel Core i7 920, GeForce GTX 480
- OpenGL and GLSL
- 1024 x 1024px

<table>
<thead>
<tr>
<th>Model</th>
<th>Ours Optimized</th>
<th>Ours Non-Rec.</th>
<th>Lee et. al. 2010 256 Rays</th>
<th>Accum. B. 256 Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homunculus (74k tri.)</td>
<td>102 ms</td>
<td>1.4x</td>
<td>13.2x</td>
<td>47x</td>
</tr>
<tr>
<td>Dragons (610k tri.)</td>
<td>98 ms</td>
<td>1.3x</td>
<td>14.7x</td>
<td>42x</td>
</tr>
</tbody>
</table>
Conclusion

- layered DoF method
- partial occlusion solved
- comparison to:
  - Accumulation Buffer
  - Lee et al., 2010
- optimized by recursive Gaussians
- efficient composition with alpha blending
Outlook

- screen-spaced antialiasing
- avoid empty layers: clustering
- inaccurate but faster blurring methods
- combine with eye-tracker
Thank you!

slides will be available at:
http://www.cg.tuwien.ac.at/research/publications/2012/schedl-2012-dof/