Using Opaque Image Blur for Real-Time Depth-of-Field Rendering

Martin Kraus

Department of Architecture and Media Technology
Aalborg University, Denmark

GRAPP 2011
Outline

• Previous Work
• Depth-of-Field Rendering
• Opaque Image Blur
• Results
• Discussion
Previous Work (I)

• Based on 3D scene information:
  • Stochastic sampling (Cook et al., 1984; etc.)
  • Splatting of image points (Potmesil and Chakravarty, 1982; etc.)
  • Also on GPUs (Lee et al. 2008 and 2010; etc.)
  • All dependent on scene complexity.
Previous Work (II)

- Based on color image and depth image:
  - Blurring according to depth (Rokita, 1993; etc.)
  - Also on GPUs (Demers, 2004; etc.)
  - Artifacts due to color bleeding between objects at different depths.
Previous Work (III)

• Based on color image and depth image:
  • Independent blurring of subimages (Barsky, 2004)
  • Also on GPUs (Kraus and Strengert, 2007)
  • This work is based on the latter.
Depth-of-Field Rendering

subimages

opaque blur

blending
Depth-of-Field Rendering

• Problem:
  Blurred subimages have decreased opacity at silhouettes; thus, pixels that don't exist in the image are disoccluded. This requires expensive extrapolation.

• Solution:
  Image blur of subimages that never decreases opacity: “opaque image blur.”
Opaque Image Blur

• How to blur the opacity channel without decreasing the opacity of any pixel?

• Easier question: how to blur a grayscale image without decreasing the intensity of any pixel?

• Answer: image glow, i.e. blur and blend with original (possibly multiple times)
Opaque Image Blur

- blur
- opacity
- rescaling of colors
- glow
Opaque Image Blur

• Three steps:
  • Apply image glow to opacity channel.
  • Apply standard blur to color channels and original opacity channel.
  • Rescale blurred colors according to the ratio between glowing and blurred opacity.
Results

ray-traced with stochastic sampling

GPU disocclusion (Kraus & Strengert) (208 ms)

this work (110 ms)
Discussion

- Our new approach for depth-of-field rendering:
  - Is almost twice (1.9x) as fast.
  - Is easier to implement. (No extrapolation.)
  - Doesn't offer same image quality.
  - Thus, it's a new compromise between performance, image quality, and ease of implementation.
Discussion

• Future applications:
  • Motion blur
  • Generalized depth-of-field effects
  • Blur of arbitrary parts of images without color bleeding
Thanks!

Questions?