A SXGA 3D Display Processor with Reduced Rendering Data and Enhanced Precision

Seok-Hoon Kim
MVLSI Lab., KAIST
Contents

● Background
● Motivation
  ● 3D Graphics + 3D Display
● Previous Works
  ● Conventional 3D Image Synthesis Process
  ● Real-time 3D Display Processor
● Conclusion
Background: Principle of 3D Display

A separate image to each eye

Perception of depth

Stereoscopy
Background: Target 3D Display

- **Lenticular display**
  - Convey different image to each eye
  - Feel depth-perception
  - Widely used

- **Multiview lenticular display**
  - Wide view range
  - Multiplexing

- **Slanted lenticular display**
  - Improve resolution characteristic

9-view Slanted Lenticular Display
Motivation: 3D Graphics

Ray Tracing
Vertex texture
Shadow Mapping
Motivation:

- 3D Graphics
  - 3D graphics hardware
    - Photorealistic rendering effects
  - 2D display
    - Lack of depth-perception
    - Not provide 3D effects

No True Realism
Motivation: 3D Display

- Depth-perception of 3D display → True Realism

- 3D image processing is too complicated.
  - Display static or pre-processed data
  - Difficult for supporting interactive applications
Motivation

: 3D Graphics + 3D Display

3D graphics + 3D display supports interactive applications & true realism
Previous Works: Conventional Method

- **View interpolation**
  - Interpolate 7 view images between left and right view image

- **Multiplexing**
  - Allocate sub-pixels of 9 view images to lenticular LCD
  - View image (427x342), 3D image (1280x1024)
Previous Works
: Conventional Method – View Interpolation

Shift a pixel of left image to left by disparity

\[ \text{disparity} = \text{const.} \times \text{view} \times \text{depth} \]

\( x^* = x - \text{disparity} \)

\( y^* = y \)
Previous Works

: Conventional Method – View Interpolation

- **Hole**
  - Not filled pixels in intermediate view images
  - Visible from only one view
  - Remove holes bringing pixels from **right image**

interpolation

left image

intermediate view image

Hole

KAIST
Previous Works
: Conventional Method – View Interpolation

disparity

left view image line

intermediate view line

right view image line

before filling holes

after filling holes
Previous Works
: Conventional Method – Multiplexing

- Allocate sub-pixels to appropriate positions in LCD
- View numbers are calculated at sub-pixel level

After view interpolation

R: \( \text{view}_{\text{LCD}} = (\alpha \cdot (x_{\text{LCD}} \cdot 3 + 0) + \beta \cdot y_{\text{LCD}} + \gamma) \mod 9 \)

G: \( \text{view}_{\text{LCD}} = (\alpha \cdot (x_{\text{LCD}} \cdot 3 + 1) + \beta \cdot y_{\text{LCD}} + \gamma) \mod 9 \)

B: \( \text{view}_{\text{LCD}} = (\alpha \cdot (x_{\text{LCD}} \cdot 3 + 2) + \beta \cdot y_{\text{LCD}} + \gamma) \mod 9 \)
Previous Works
: Conventional Method – Problems

- Frequent external memory accesses
  - Difficult to support interactive applications
- Large memory for storing intermediate images
Previous Works
: ISSCC2007 – Merged Architecture(1)

3D Graphic Pipeline

Geometry Engine (GE) ➔ Rasterization Engine (RE)

3D Image Synthesis Pipeline

- Both frame buffers store the same data (color, position, and depth)

→ Merged Architecture
Previous Works
ISSCC 2007 – Merged Architecture(2)

- Disparity is calculated using a linear equation

\[
d_{\text{eye}} = \frac{f}{z_{\text{eye}}} \cdot b \cdot \frac{\text{view}}{8}
\]

\[
d_{\text{eye}} : h_{\text{eye}} = d_{\text{screen}} : h_{\text{screen}}
\]

\[
z_{\text{screen}} = \frac{F}{F - N} \cdot \left(1 - \frac{N}{z_{\text{eye}}}\right)
\]

\[
d_{\text{eye}} = \frac{h_{\text{eye}}}{h_{\text{screen}}} \cdot d_{\text{screen}}
\]

\[
\frac{N}{z_{\text{eye}}} = \frac{f}{z_{\text{eye}}} = 1 - \left(\frac{F - N}{F}\right) \cdot z_{\text{screen}}
\]

\[
\frac{h_{\text{eye}}}{h_{\text{screen}}} \cdot d_{\text{screen}} = \left\{1 - \left(\frac{F - N}{F}\right) \cdot z_{\text{screen}}\right\} \cdot b \cdot \frac{\text{view}}{8}
\]

\[
d_{\text{screen}} = \left\{1 - \left(\frac{F - N}{F}\right) \cdot z_{\text{screen}}\right\} \cdot b \cdot \frac{\text{view}}{8} \cdot \frac{h_{\text{eye}}}{h_{\text{screen}}}
\]

\[
d_{\text{screen}} = (\alpha - \beta \cdot z_{\text{screen}}) \cdot \text{view}
\]
Previous Works
: ISSCC 2007 – Real-time Synthesis(1)

- Simultaneous view interpolation and multiplexing
  - No intermediate memory
  - Reduced memory size & external memory accesses

Some part of lenticular LCD
Previous Works
: ISSCC 2007 – Real-time Synthesis(2)

source image \((x_{\text{view}}, y_{\text{view}})\) \(\xrightarrow{ }\) \(x_{\text{LCD}} = 3x_{\text{view}}, y_{\text{LCD}} = 3y_{\text{view}}\) \(\xrightarrow{ }\) 3D image in LCD \((x_{\text{LCD}}, y_{\text{LCD}})\)

4th view image (view position = 4)

R: \(\text{view}_{\text{LCD}} = (\alpha*(x_{\text{LCD}}*3+0)+\beta*y_{\text{LCD}}+\gamma)\mod 9\)
G: \(\text{view}_{\text{LCD}} = (\alpha*(x_{\text{LCD}}*3+1)+\beta*y_{\text{LCD}}+\gamma)\mod 9\)
B: \(\text{view}_{\text{LCD}} = (\alpha*(x_{\text{LCD}}*3+2)+\beta*y_{\text{LCD}}+\gamma)\mod 9\)
Previous Works: ISSCC 2007 – Architecture

- Real-time frame rate: more than 30fps
  - Generate one pixel per cycle @ 50MHz
  - $5 \times 10^7 / 1280/1024 \approx 38$fps

- SE consists of three sub-pixel units.
  - A sub-pixel unit generates a sub-pixel per cycle

![Diagram showing Red, Green, and Blue Sub-pixel Units](image-url)
Previous Works
: ISSCC2007 – performance

- 73.3% reduced external memory accesses
- Increase synthesis rate
- Support interactive applications

External Memory Access (MBytes/Frame)

- 73.3% reduction

<table>
<thead>
<tr>
<th>Image by image processing</th>
<th>Pixel by pixel processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Image</td>
<td>View-Image</td>
</tr>
</tbody>
</table>
Previous Works
:ISSCC2007 – Chip Micrograph
Problem of RE in Previous Work

- Previous work renders the same scene twice for two different view-positions
  - Waste rendering time to generate duplicate data
  - Require additional memory to store & to load duplicate data
    - Large memory size & increasing external memory accesses

![Left view-image](image1)

![Right view-image](image2)
Proposed Idea of RE(1)

- Renders the same scene only once not twice
- Completely removes duplicate data
- Modified clipping
  - Including both left view volume & right view volume
  - Check the pixels in the expanded view volume
    - Use them for filling holes
Proposed Idea of RE(2)

- **Modified back-face culling**
  - Back-face culling operations are executed both left view-position and right view-position
  - Check the pixels culled from left view-position but not culled from right view-position
    - Use them for filling holes

![Diagram showing modified back-face culling](image)
Problem of SE in Previous Work

- **View-number error**
  - View-number evaluation equation
    - \( \text{view}_{LCD} = (A \times (x_{LCD} \times 3 + 0,1,2) + B \times y_{LCD} + C) \% 9 \)
    - A, B, C are floating number constants
  - For blending, reverse multiplexing has been used
  - Previous work rounds off the view-number to the nearest integer
  - Allocates sub-pixels without blending operations
Proposed Idea of SE(1)

- **Precision-enhanced multiplexing**
  - Based on the previous work [ISSCC2007]
  - Conserve the interactive synthesis rate
  - Rounds off the view-number to 2\textsuperscript{nd} decimal place to reduce view-number error to less than 1%
    → improve synthesis image quality
Proposed Idea of SE(2)

- Evaluated view-number = integer part + fragment part
- Integer part is used for finding matching positions
- Fragment part is used for blending
- Additional H/W
  - 5 bits more for the view-number evaluator
  - 2 flag bits per a sub-pixel for avoiding overlapping cases → only 5 bytes memory
Conclusion

● **Combine 3D graphics and 3D display**
  - Support both true-realism & interactive applications

● **Expanded clipping & back-face culling**
  - Completely remove duplicate data
  - Reduce required memory size & memory accesses
    → lower H/W cost & improved performance

● **Precision-enhanced Multiplexing**
  - Based on the previous work, conserve interactive synthesis rate
  - Reduce view-number error to less than 1%
  - Require a few of additional H/W