A Log-Rectilinear Transformation for Foveated 360-Degree Video Streaming

David Li†, Ruofei Du‡, Adharsh Babu†, Camelia Brumar†, Amitabh Varshney†

† University of Maryland, College Park  ‡ Google
A Log-Rectilinear Transformation for Foveated 360° Video Streaming

David Li†, Ruofei Du†, Adharsh Babu†, Camelia Brumar†, Amitabh Varshney†
† University of Maryland, College Park  ‡ Google
Outline

• Background on 360° Video Streaming and Log-Polar Foveation
• Summed Area Tables and Log-Rectilinear Transformation
• Foveated Streaming Pipeline
• Qualitative and Quantitative Results
Current Landscape

- 360 Cameras and VR headsets are increasing in resolution.
- Video streaming is quickly increasing in popularity.
Current Landscape

- Commercial VR headsets are getting eye-tracking capabilities.
360° Videos

- 360 cameras capture the scene in every direction with a full 360 degree spherical field of regard.
- These videos are typically stored in the equirectangular projection parameterized by spherical coordinates $(\theta, \varphi)$.
360° Videos

- When viewed in a VR headset, 360° videos cover the entire field-of-view for more immersive experiences.
- However, transmitting the full field-of-regard either has worse perceived quality or requires far more bandwidth than for conventional videos.
Viewport Dependent 360° Video Streaming

• Existing work in 360° streaming focuses on viewport dependent streaming by using tiling to transmit only visible regions based on the user’s head rotation.
Foveated Rendering

- Foveated rendering renders the fovea region of the viewport at a high-resolution and the peripheral region at a lower resolution.
- Kernel Foveated Rendering (Meng et al., PACMCGIT 2018) uses a log-polar transformation to render foveated images in real-time.

![Log-polar Transformation, Image from (Meng et al., 2018)](image)
Log-Polar Foveated Streaming

- Applying log-polar subsampling to videos results in flickering and aliasing artifacts in the foveated video.
Research Questions

- Can foveation techniques from rendering be used to optimize 360 video streaming?
- How can we reduce foveation artifacts by leveraging the full original video frame?
Log-Polar Foveated Streaming

- Artifacts are caused by subsampling of the original video frame.
Log-Polar Foveated Streaming

- Artifacts are caused by subsampling of the original video frame.
Log-Polar Foveated Streaming

- Subsampled pixels should represent an average over an entire region of the original video frame.
- Computationally, this would take $O(\text{region size})$ time to compute for each sample.
Summed-Area Tables

- One way to compute averages quickly is using summed-area tables, also known as integral images.
- Sampling a summed area table only takes $O(1)$ time.

$$\text{Sum}(D) = a - b - c + d$$
Log-Rectilinear Transformation

- Apply exponential drop off along x-axis and y-axis independently.
- Rectangular regions allow the use of summed area tables for subsampling.
- A one-to-one mapping near the focus region preserves the resolution of the original frame.
Log-Polar vs Log-Rectilinear

log-polar buffer

log-rectilinear buffer
Foveated Streaming Pipeline

Video Streaming Server

- Decoding 360° Video
  - FFmpeg
  - GPU-driven Summed-Area Table Generation
  - OpenCL
  - Computing the Log-Rectilinear Buffer
  - OpenCL
  - Encoding the Log-Rectilinear Video Stream
  - FFmpeg

Client

- Video Streaming Request
  - socket
  - Updating the Foveal Position
  - FFmpeg
  - Decoding the Log-Rectilinear Video Stream
  - OpenCL
  - Transforming into a Full-resolution Video Frame
Qualitative Results

• Shown with gaze at the center of the viewport
Quantitative Results

We perform quantitative evaluations comparing the log-rectilinear transformation and the log-polar transformation in 360° video streaming.

- Performance overhead of summed-area tables.
- Full-frame quality.
- Bandwidth usage.

<table>
<thead>
<tr>
<th>Sampling Method</th>
<th>Decoding (ms)</th>
<th>Processing (ms)</th>
<th>Sampling (ms)</th>
<th>Encoding (ms)</th>
<th>Total (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log-Polar</td>
<td>6.14</td>
<td>1.91</td>
<td>0.55</td>
<td>2.86</td>
<td>11.46</td>
</tr>
<tr>
<td>Log-Rectilinear</td>
<td>6.13</td>
<td>1.91</td>
<td>0.53</td>
<td>2.85</td>
<td>11.43</td>
</tr>
<tr>
<td>SAT Log-Rectilinear</td>
<td>6.14</td>
<td>3.00</td>
<td>0.46</td>
<td>2.84</td>
<td>12.44</td>
</tr>
</tbody>
</table>
Quantitative Results

• Pairing the log-rectilinear transformation with summed area table filtering yields lower flickering while also reducing bandwidth usage and returning high weighted-to-spherical signal to noise ratio (WS-PSNR) results.
Quantitative Results

• Pairing the log-rectilinear transformation with summed area table filtering yields lower flickering while also reducing bandwidth usage and returning high weighted-to-spherical signal to noise ratio (WS-PSNR) results.
Conclusion

- We present a log-rectilinear transformation which utilizes foveation, summed-area tables, and standard video codecs for foveated 360° video streaming.

![Foveation](image1.png) ![Summed-Area Tables](image2.png) ![Standard Video Codecs](image3.png)

Foveated 360° Video Streaming