Comparison of CPU-based and GPU-based Approaches for Scalar Field Visualization(Slicing)

Kathan Mistry
SR No.: 23-1-22804
M.Tech, CSA

Introduction

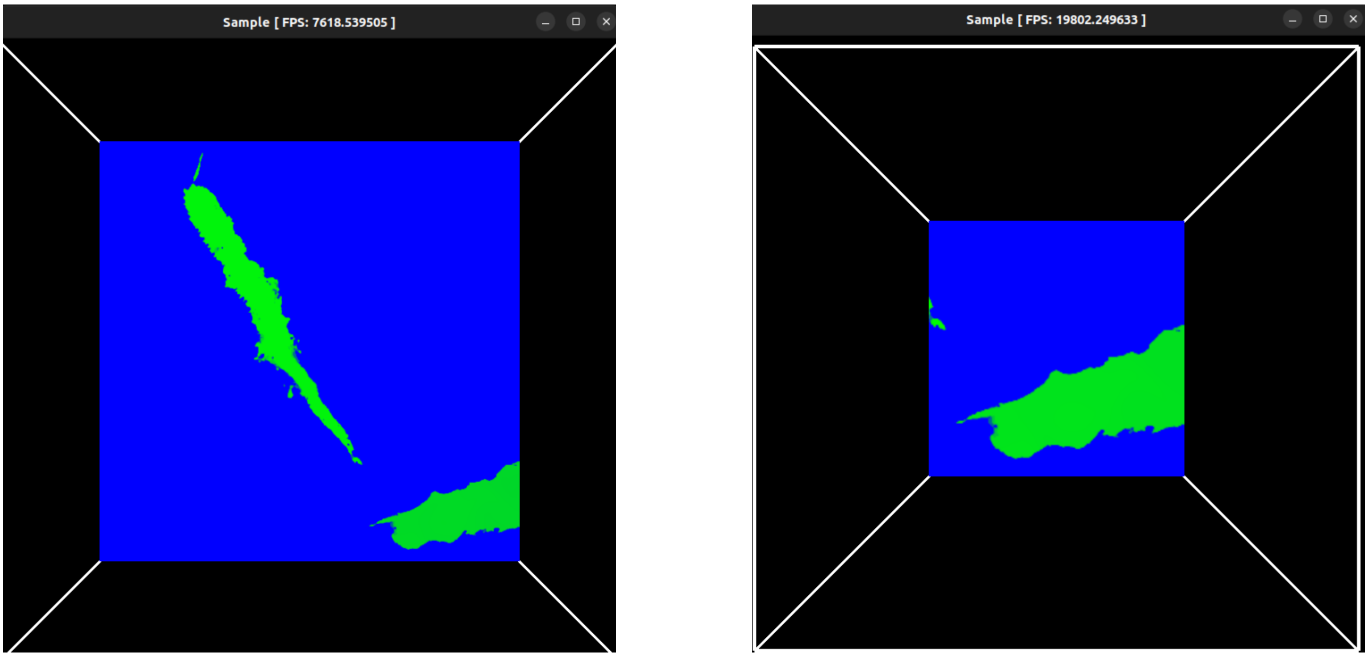
In this report, I evaluate two approaches for scalar field visualization in OpenGL: a CPU-based method and a GPU-based method, focusing on an axis-parallel slicing plane to represent a scalar field. I used a 3D bounding box to define the domain and provide context for the visualization. The GPU-based approach leverages 3D textures and a shader-based 1D colormap, while the CPU-based approach renders the data without utilizing GPU shaders for computation.

My comparison is based on performance, image quality, and the ability to handle complex data. The dataset used includes RedSea and RedSeaSmall dataset containing scalar fields of temperature and salinity etc.

Methodology

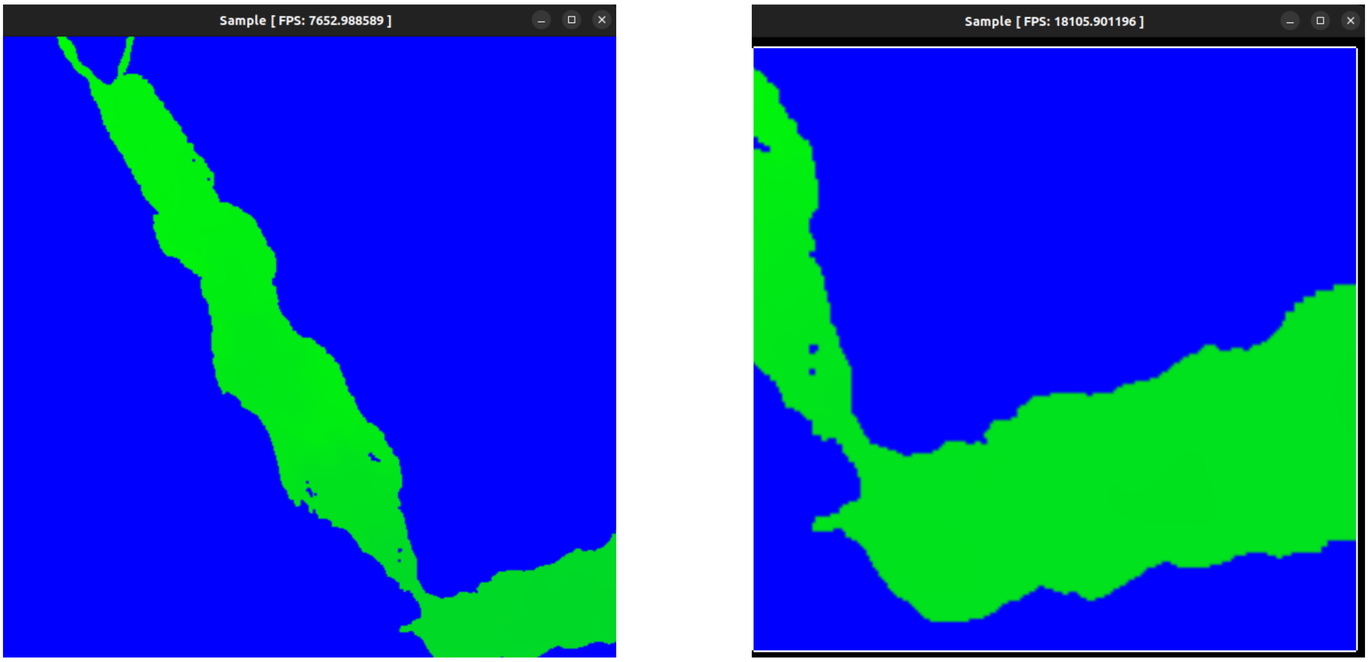
1. **Performance Measurement**:
   I measured frames per second (FPS) for both CPU and GPU approaches.
2. **Image Quality Comparison**:
   I visually compared the precision of each approach in terms of clarity, color gradients, and the ability to display fine details within the scalar field. Special attention was given to any artifacts or discrepancies between the two methods.
3. **Observations of Rendering Artifacts and Smoothness**:
   I checked for artifacts or visual irregularities that may arise due to interpolation limitations in the CPU-based approach and compared them against the GPU's native handling of textures and colormaps.

Screenshots



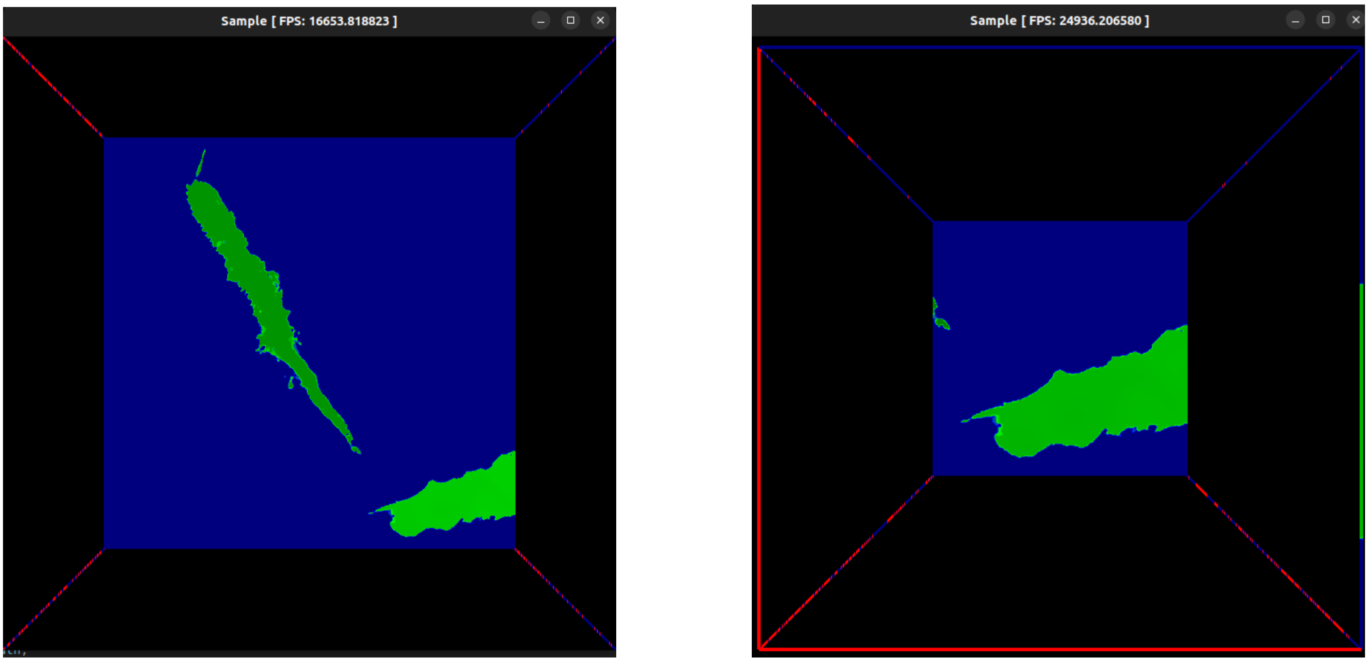
((a)) CPU-based output for ‘redsea‘ ((b)) CPU-based output for ‘redseasmall‘

Figure 1: CPU-based outputs for ‘redsea‘ and ‘redseasmall‘ datasets



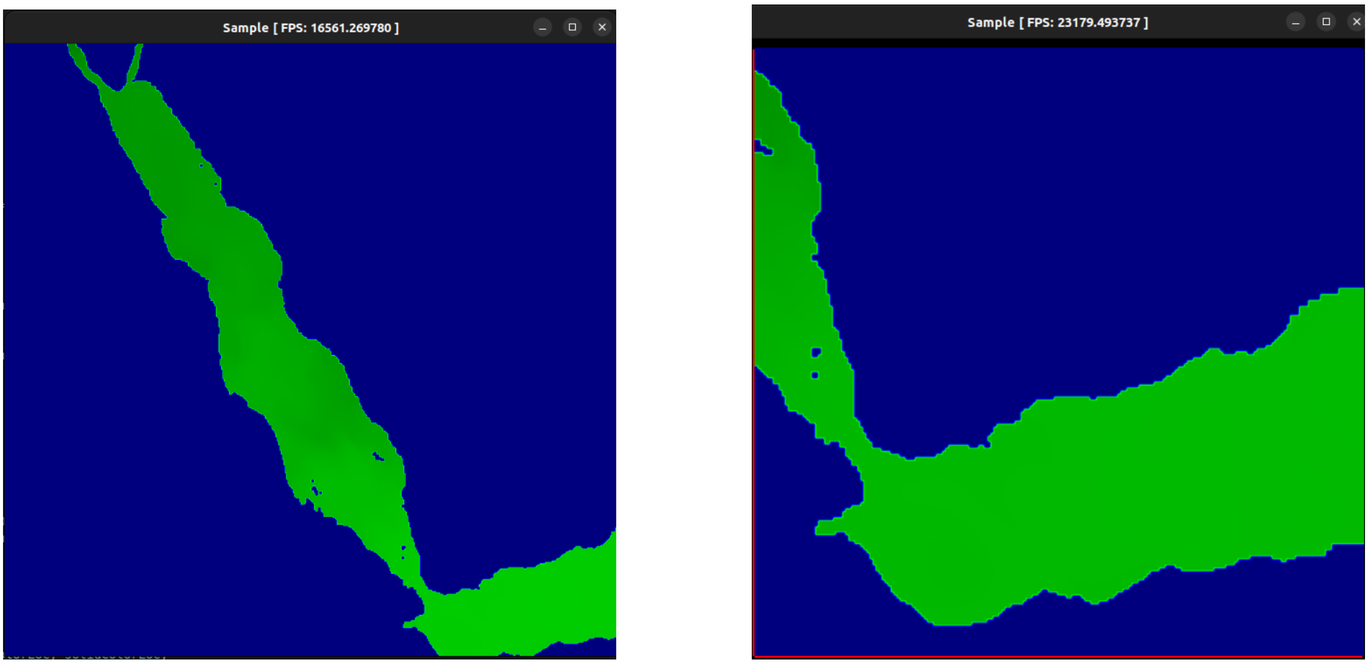
((a)) CPU-based output for ‘redsea‘ ((b)) CPU-based output for ‘redseasmall‘

Figure 2: CPU-based outputs for ‘redsea‘ and ‘redseasmall‘ datasets with offset=1



((a)) GPU-based output for ‘redsea‘ ((b)) GPU-based output for ‘redseasmall‘

Figure 3: GPU-based outputs for ‘redsea‘ and ‘redseasmall‘ datasets



((a)) GPU-based output for ‘redsea‘ ((b)) GPU-based output for ‘redseasmall‘

Figure 4: GPU-based outputs for ‘redsea‘ and ‘redseasmall‘ datasets with offset=1

Results

1. Performance Analysis

- **Frame Rate**:
The GPU-based approach achieved significantly higher FPS, maintaining around 16,000 FPS at all resolutions, while the CPU-based approach averaged about 12,000 FPS. With lower resolutions, the CPU approach improved but never surpassed GPU performance.

- **Scalability**:
The GPU-based method scaled well, handling larger datasets and additional slicing planes without noticeable performance degradation. The CPU approach struggled with complex datasets, with a marked increase in render time as data complexity rose.

2. Image Quality Comparison

- **Detail and Precision**:
The GPU-based visualization produced smoother and more precise color transitions, accurately representing the scalar field's temperature and salinity variations. The CPU-based method showed some pixelation and lacked the subtle gradient details visible in the GPU rendering.

- **Artifacts and Rendering Issues**:
While the GPU rendering appeared artifact-free, the CPU-based visualization exhibited minor aliasing issues along the slicing plane edges, especially at higher resolutions. These artifacts were more pronounced in regions with high scalar value gradients.

- **Colormap Rendering**:
Both approaches successfully implemented the 1D texture colormap, but the GPU-based method provided smoother gradient transitions due to better interpolation capabilities within shaders. The CPU implementation, while functional, had more abrupt color changes, reducing overall visual quality.

Observations and Conclusion

The GPU-based approach significantly outperformed the CPU-based implementation in both speed and image quality. Specifically:

- **Speed and Responsiveness**: The GPU's dedicated resources for handling large textures and parallel computations made it ideal for high-resolution, real-time visualizations. It maintained stable FPS and reduced frame render times, supporting a more interactive user experience.
- **Image Quality and Artifacts**: The GPU's use of shaders for color interpolation resulted in higher image fidelity and smoother transitions, which were critical for detailed field visualization. The CPU struggled to replicate this accuracy, particularly in high-gradient areas, and displayed noticeable artifacts in some cases.

Recommendation

For interactive or real-time applications, I highly recommend the GPU-based approach due to its superior performance and image quality. The CPU-based method could be considered in scenarios where GPU resources are unavailable or need to be allocated elsewhere, although this comes with a sacrifice in both performance and visual clarity.

In summary, the GPU-based approach provides a clear advantage for complex scalar field visualizations, especially when scalability and image quality are priorities.