

GPU-Based Volume Rendering of Noisy Multi-Spectral Astronomical Data

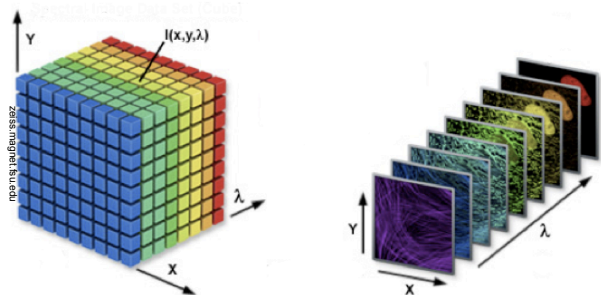
A. Hassan, C.J.Fluke, D.G.Barnes

Swinburne Centre for Astrophysics & Supercomputing, Melbourne, Victoria, Australia, 3122

Introduction

Next-generation astronomy research facilities will generate new challenges for data storage, access, analysis and system monitoring, bringing astronomy into the Petascale Data Era. But even today, astronomical knowledge is not growing at the same rate as the data. Scientific visualization is a fundamental, enabling technology for knowledge discovery. Despite recent progress, many current astronomy visualization approaches will be seriously challenged by, or are completely incompatible with, the Petascale Data Era. With an emphasis on developing new approaches compatible with data from the Square Kilometer Array and its Pathfinders, the goal of this work is to advance the field of astrophysical visualization in preparation for the Petascale Era.

Multi-Spectral Astronomical Data

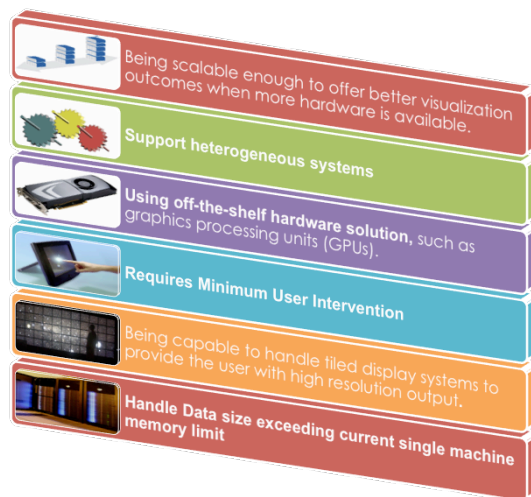


Multispectral data can be considered as a 4D data volume where three dimensions are associated to position allocation (two dimensions for the spatial position in sky coordinates and one dimension for the wavelength) and one dimension for the flux density. The data cube can be considered as a stack of images where each image presents a sky portion over a small wavelength range ($\Delta\lambda$).

Challenges and Design Objectives

Our main goal is to enable astronomers to visualize large spectral data cubes of the size that will be generated from Australia SKA Pathfinder (ASKAP) (at least 1TB of data). To achieve this target we designed a framework that utilizes the latest available hardware technologies combined with the latest software infrastructure.

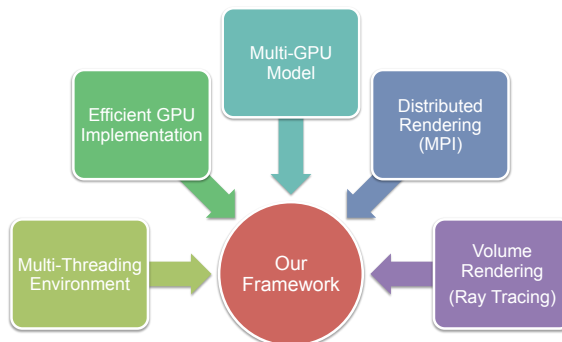
Our design objectives were:



Based on that, the following design decisions were taken:

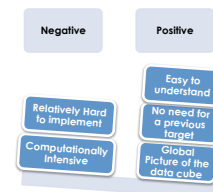
1. Use ray-casting volume rendering as our visualization technique.
2. Build a mixed framework based on a distributed GPU architecture to visualize such cubes.

Our Framework

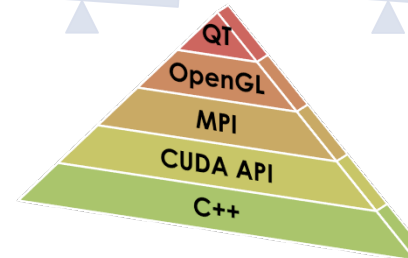
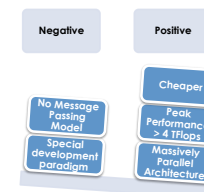


Design Decisions and Implementation tools

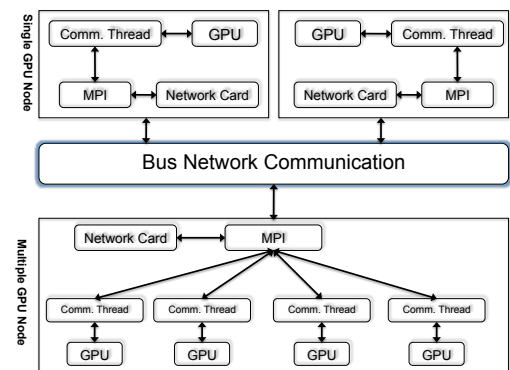
Why Volume Rendering?



Why GPU?

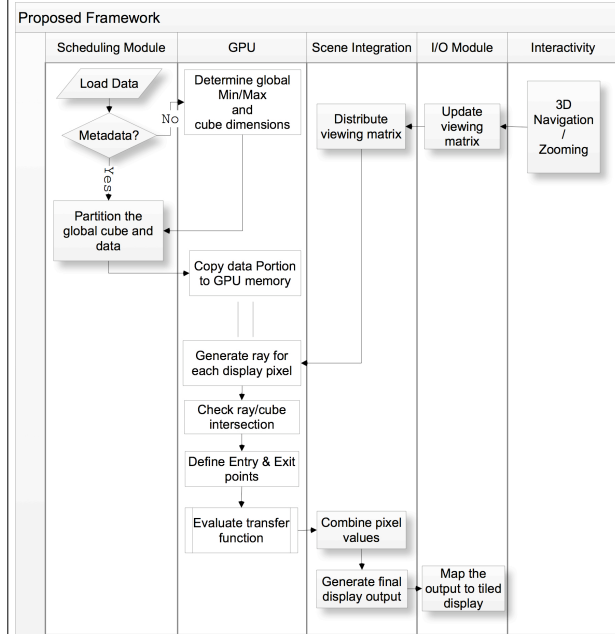


Distributed GPU Framework



Our distributed GPU framework combines the processing power of multiple GPU nodes to speed-up and enhance the spectral line cube visualization process. This framework uses Message Passing Interface (MPI), Multi-threading, and CUDA to allow many GPUs to work on the same problem.

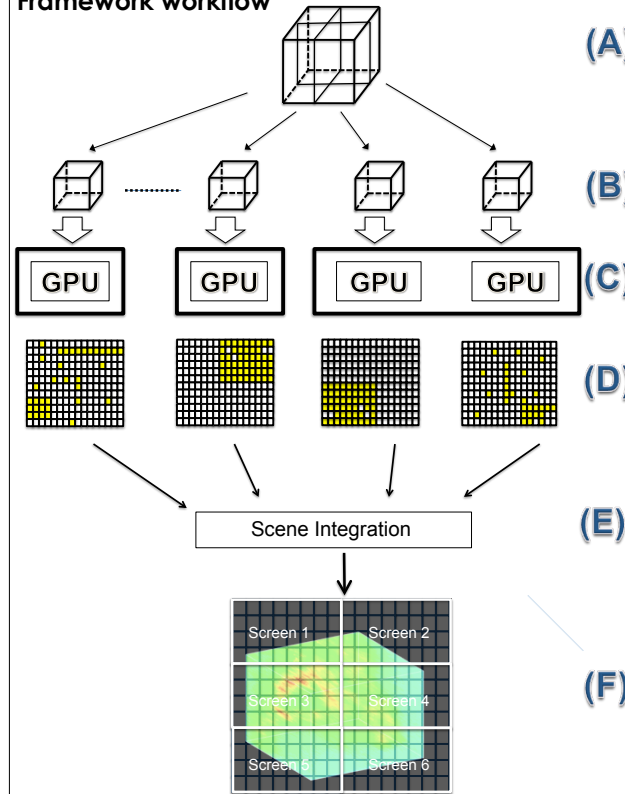
Main Framework Modules



This figure shows a cross-functional diagram containing the main components of our proposed framework. These components are partitioned based on their main functionality into:

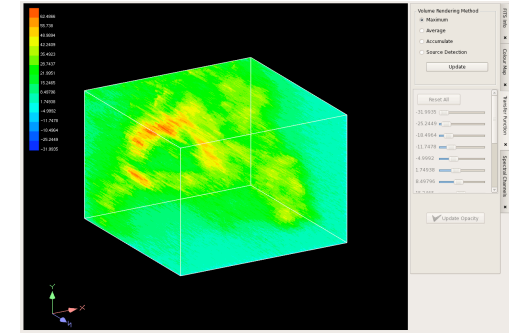
- Scheduling module:** responsible for managing and controlling the overall process.
- GPU module:** includes different execution kernels and utilizes the parallelization paradigm of the GPUs.
- Scene integration module:** responsible for generating the final display output by combining the output of the contributing GPUs.
- I/O and User interaction modules:** support the user's interactivity and change the output according to the user's input.

Framework workflow

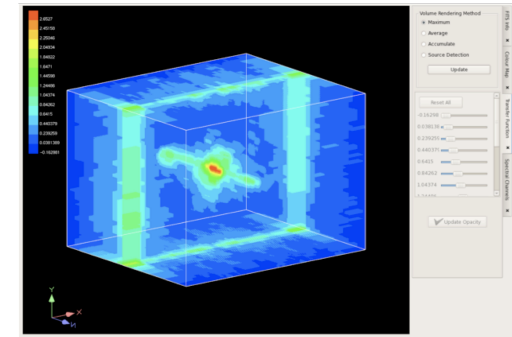


- The spectral line cube will be partitioned into a set of smaller sub-cubes.
- The scheduler module will assign these sub-cubes to the processing Nodes/GPUs.
- Each GPU will apply ray-tracing volume rendering to map each output pixel into a color (RGBA) based on a pre-selected transfer function.
- The process of volume rendering will produce (N) images with the same resolution as the output screen(s).
- The scene integration module will reapply the same transfer function to these images to combine them into the final output.
- The final output will be directed to the output display(s).

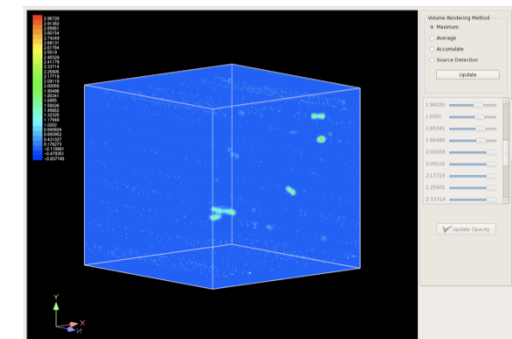
Sample Output



An expanding HI shell from the Canadian Galactic Plane Survey observed with the Dominion Radio Astrophysical Observatory interferometer. X and Y axes are Galactic longitude and latitude (714 x 520 pixels), and the Z axis is velocity with respect to local standard of rest (30 channels). Note that axis labels have yet to be implemented. Data courtesy Jayanne English (U.Manitoba).



115.27 GHz 12CO(J=1-0) observation of EP Aqr obtained with the IRAM interferometer (Winters et al. 2007, A&A, 475, 559; VizieR On-line Data Catalog: J/A+A/475/559). The spectral cube comprises 26 velocity channels (Z axis) of 256x256 sky pixels.



Neutral hydrogen (21cm) observation of part of the Ursa Major galaxy cluster, made with the Lovell radio-telescope at Jodrell Bank, Manchester. The image is 157x148 sky pixels by 148 velocity channels. Data courtesy Virginia Kilborn (Swinburne).



Conclusion:

The focus of this work is to improve multi-spectral data visualization to cope with the vast amount of data to be produced by ASKAP and similar facilities. By employing GPUs combined with distributed processing, we are aiming to implement a scalable system capable to interactively visualize "greater than memory" astrophysical datasets.

Web-site: <http://astronomy.swin.edu.au>

More information: E-mail ahassan@swin.edu.au or cfluke@swin.edu.au