Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Representing and rendering large noisy radio data cubes

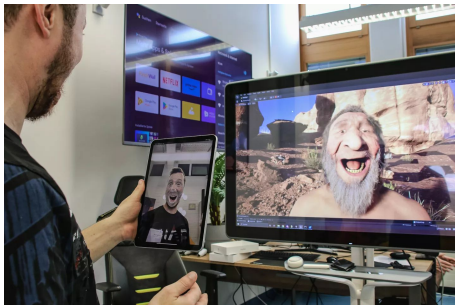
David Bernhardt, Tom David Eibich, Anton Kindsvater, André Hinkenjann

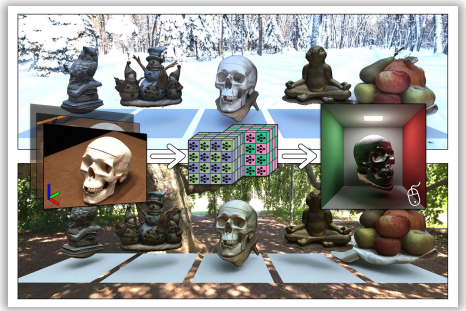
Institute of Visual Computing

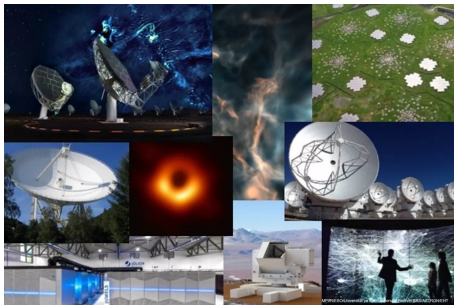
Bonn-Rhein-Sieg University of Applied Sciences

Institute of Visual Computing @ H-BRS

- 40-50 people working on **visualisation, HCI, computer vision, interactive systems**
- From **basic** to **applied** research







B3D project

Vision for a new type of working environment

Three dimensional interactive visual analysis and collaboration on large HI data cubes

- Explore datasets with multiple users
- In person & remote
- Apply common tools and algorithms to selected regions
- Intuitive user interface
- Stereoscopic 3D view

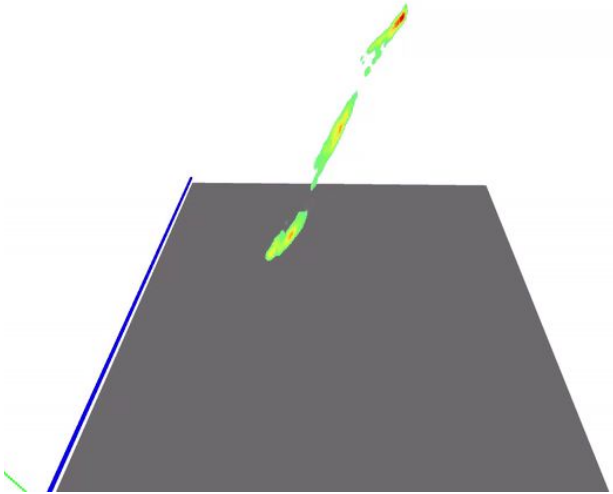
image generated by Leonardo.ai

Volume renderer for HI Data Cubes

- Distributed volume renderer for large datasets
  - Large data visualisation on multiple GPU
  - XR requirements not met
  - Hybrid Image-/Data-Parallel Rendering Using Island Parallelism
    [Zellman22]

- Volume renderer using domain knowledge
  - Identified (**filtered**) sources for empty space skipping
  - Maximum intensity projection
  - Transfer function (maps values to color and transparency) can be altered real-time
  - Renders whatever **fits in GPU memory** (eg. 48GB)

Unfiltered Test-Dataset (10GB) rendered with “Hybrid Image-/Data-Parallel Rendering Using Island Parallelism"



Filtered NGC4656. Dataset rendered with our volume renderer. Red/Green axis: spatial coordinates. Blue axis: frequencies

Visualisation and representation methods

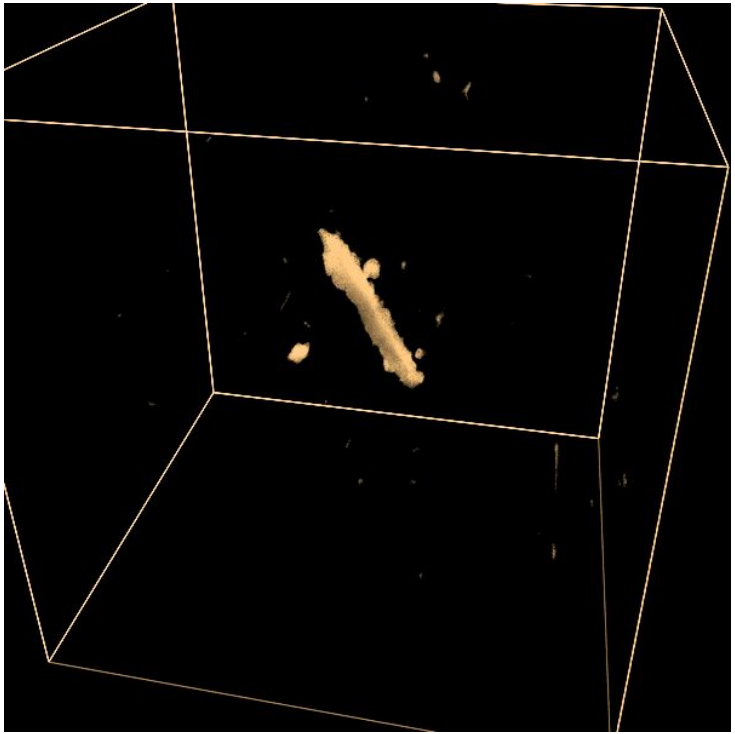
- Some teams want to explore the unfiltered data cubes: Current cubes already too big
- SKA cubes up to 100.000^3 voxels
  - way big for “traditional” rendering on a local machine
- Possible Solutions
  - Distributed remote rendering (high latency)
  - Hybrid local rendering
  - Compression by neural representation
  - Neural Rendering

Image: SKAO

Visualizing noisy data with NN

- Trials: Local Neural Rendering
  - Apply source finding (filtering) on data cube
  - Extract NanoVDB from identified sources
  - Use NVIDIA Instant Neural Graphics Primitives code to quickly train and render NN (MLP)
  - Example:
    - Original **2 GByte** FITS
      ➔ **8 MByte** NanoVDB
    - Traning time: order of seconds

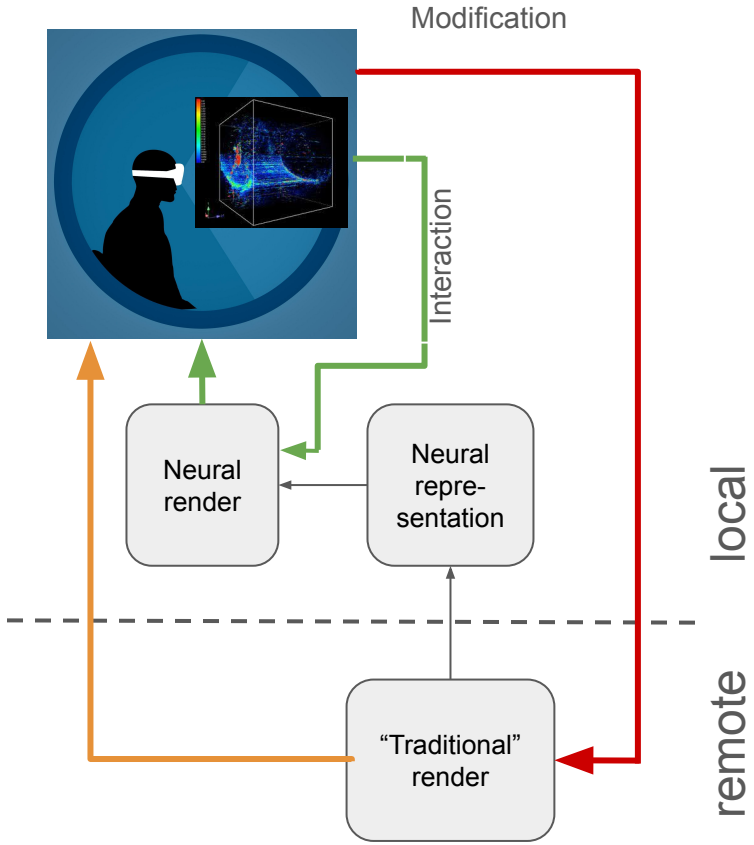
- Trials with non-filtered, noisy data did not show satisfying results
  - MLP tried to learn noise (which is dominant) and spent its capacity on it
  - Open questions: Which neural representation fits problem best? Can denoising be done like in image synthesis? Stream (updates on) neural representations? How to handle dynamic cubes / “scenes”?



Filtered NGC45652 Dataset learned & rendered with Instant-NGP

Visualizing noisy data with NN

- Open: Hybrid/local Neural Rendering
  - Remote Renderer generates high resolution images
  - Continuously train model locally with remotely generated images while user explores data set
  - Use local neural rendering to get (lower quality) images
  - Manipulating data or mapping requires re-training
    - Training has to be very fast (eg. Instant Neural Graphics Primitves)



Thank you very much!

Andre.Hinkenjann@h-brs.de

http://ivc.h-brs.de