

Topology based transfer function for volumic rendering

1 Topic

Volumic rendering is a visualization technique enabling to see through a whole 3D data set. In order to explore these data, the user is required to define a transfer function, i.e. the mapping of the rendering color and transparency with respect to the data values. Generating such a transfer function can be a nightmare to users as they might have no knowledge of the structure of the studied data set. This structure can be described by distinct segmentation methods, for instance thanks to topological analysis, which can be used to define topological features in the data set.

This internship aims at designing and implementing techniques to automate the generation of these transfer functions according to the topology of the data set. The user experience dimension of the problem will have to be taken into account.

This research work can be divided into 3 major steps:

1. State-of-the-art study regarding automatic transfer function design w/wo considering topology, see for instance [2], or visualisation tools such as ImageVis3D, Voreen etc,
2. Design of the technique and implementation,
3. Test on industrial data sets, such as micro CT scans from core, velocity models, or combustion simulation results.

Paraview [1] will be the rendering tool.

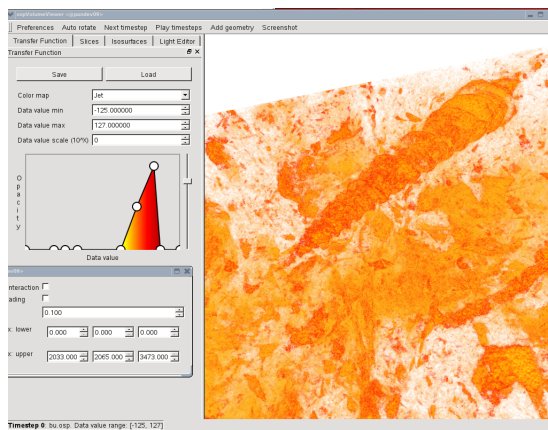


Figure 1: Transfer function used to extract micro fossil from a CTscan.

2 Appreciated skills

We are looking for highly curious and motivated student (master or engineer school), with good c++ coding skills, strong mathematical background and interest into scientific visualization research area.

3 Contact

This is a 6 months internship, to be planned in 2019.

Contact : melanie.plainchault@total.com

References

- [1] Utkarsh Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware, Inc., USA, 2015.
- [2] Gunther H. Weber, Scott E. Dillard, Hamish A. Carr, Valerio Pascucci, and Bernd Hamann. Topology-controlled volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13:330–341, 2007.