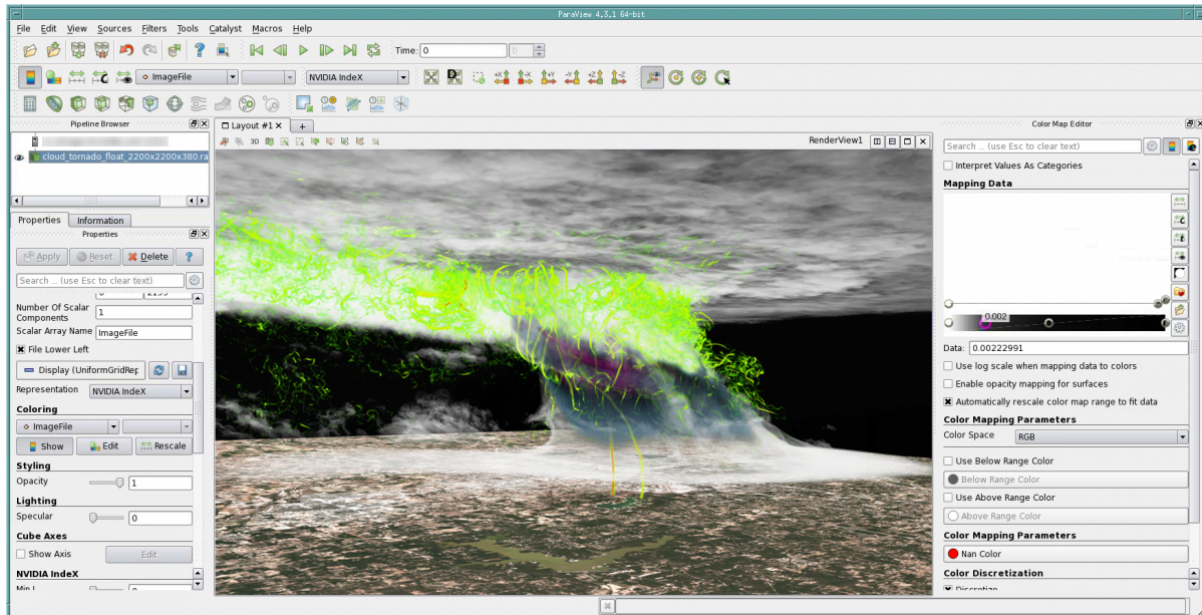


NVIDIA IndeX for ParaView Plugin

User's Guide

28 December 2017

Document version 2.1



Copyright Information

© 2017 NVIDIA Corporation. All rights reserved.

This document is protected under copyright law. The contents of this document may not be translated, copied or duplicated in any form, in whole or in part, without the express written permission of NVIDIA Corporation.

The information contained in this document is subject to change without notice. NVIDIA Corporation and its employees shall not be responsible for incidental or consequential damages resulting from the use of this material or liable for technical or editorial omissions made herein.

NVIDIA and the NVIDIA logo are registered trademarks of NVIDIA Corporation. imatter, IndeX, Iray, mental images, mental ray, and RealityServer are trademarks and/or registered trademarks of NVIDIA Corporation. Other product names mentioned in this document may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Document build number unknown

Contents

1	Introduction	1
2	Licensing	1
3	Installation	2
3.1	Linux	2
3.2	Windows	2
3.3	Compiling the plugin	3
3.4	Loading the plugin	4
4	Getting started	7
4.1	Client-only mode	7
4.2	Client-server mode on a single GPU	7
4.2.1	Client-server mode on multiple GPUs	9
4.2.2	Networking parameters for NVIDIA IndeX	10
4.2.2.1	Licensing	11
4.2.2.2	Networking options	12
5	Features	13
5.1	Structured and Unstructured grids	13
5.2	Datatypes	14
5.3	Transfer function and colormap changes	15
5.4	Region of interest changes	17
5.5	High-quality	19
5.6	Slice rendering	23
5.7	Slice rendering	23
5.8	NVIDIA IndeX visual elements	25
5.8.1	Iso-surface preset	26
5.8.2	Depth enhancement preset	26
5.8.3	Edge enhancement preset	27
5.9	Mixing ParaView primitives	27
6	Frequently Asked Questions	28
7	Useful Links	29

1 Introduction

The NVIDIA® IndeX™ for ParaView® Plugin enables large-scale and high-quality volume data visualization capabilities of the NVIDIA IndeX library inside Kitware's ParaView.

The purpose of this document is to assist an end user who is new to the plugin and wants to explore the features of NVIDIA IndeX library supported in this version of the plugin. The following sections will explain the installation procedure, various features the plugin supports, followed by a section of frequently asked questions and useful links for further reference.

If you haven't downloaded the plugin yet, you can do so from this URL:

<http://www.nvidia.com/object/index-paraview-plugin.html>

2 Licensing

The NVIDIA IndeX for ParaView plugin comes with a free license that enables exploiting the capabilities of a single GPU. If you aim to use plugin on a cluster of multiple hosts and/or with multiple NVIDIA GPUs, then please contact us for appropriate licensing via email to indexintegration@nvidia.com. Users will be notified via email whenever a new version of the plugin is released.

3 Installation

NVIDIA IndeX for ParaView plugin supports both Linux and Windows x86-64 platforms. Follow the install instructions specific to your platform.

3.1 Linux

- Run the plugin installer as shown below and follow the onscreen instructions. The installer will unpack the plugin contents into the current directory from where it was executed.

```
chmod +x <paraview-plugin-installer>.run
```

```
./<paraview-plugin-installer>.run
```

- Once installed, please set your LD_LIBRARY_PATH to the folder containing NVIDIA IndeX libraries:

```
export LD_LIBRARY_PATH=<plugin-directory>/lib/
```

Alternatively, you can also copy all the files from <plugin-directory>/lib/ into your ParaView library directory. The library directory will be located in

```
<your-paraview-installation>/lib/paraview-<version>/
```

when using official ParaView releases, or in

```
/usr/lib/paraview
```

when using distribution-specific packages. To verify that you have selected the right directory, please make sure that it contains the file libvtkPVCommon*.so.

3.2 Windows

- On Windows run the plugin installer by double clicking the executable (.exe) and follow the onscreen instructions.
- Once installed, please set your PATH environment variable to the folder containing NVIDIA IndeX libraries:

```
PATH=<plugin-directory>\lib\
```

Alternatively, you can also copy all the files from <plugin-directory>/lib/ into <your-paraview-installation>/lib directory.

Please note that installing a new plugin will override your older plugin versions.

3.3 Compiling the plugin

Please follow these steps to compile the plugin matching your build environment.

- You can download the ParaView source code from [ParaView website](#)¹
- Copy the source code folder `<plugin-directory>/src/nvindex_paraview_plugin` to `<your-paraview-source-tree>/Plugins/` directory.
- Run `cmake` on your ParaView source tree and the plugin sources will be added to the ParaView plugins list to be compiled. Make sure you have the `cmake` option set, enabled by default in the plugin source code.

```
mkdir paraview_bin
mkdir paraview_install
cd paraview_bin
```

```
ccmake ../<paraview-source-root>
```

```
PARAVIEW_BUILD_PLUGIN_pvNVIDIAIndex=ON
```

- Run `make` or `make install` from your ParaView source tree and the plugin will be compiled together with ParaView.

```
make <paraview-binary-root>
```

For additional information about compiling ParaView please refer to the instructions on [ParaView website](#)²

¹<https://www.paraview.org/download/>

²https://www.paraview.org/Wiki/ParaView:Build_And_Install

3.4 Loading the plugin

If the plugin is self compiled, plugin binary is located under `<your-paraview-installation>/lib/paraview-<version>/` and will automatically show up in the list of plugins in the **Manage Plugins** menu.

Please follow the steps below to load the plugin in ParaView.

- Start ParaView client and navigate to [Tools ► Manage Plugins] option from the menu bar.

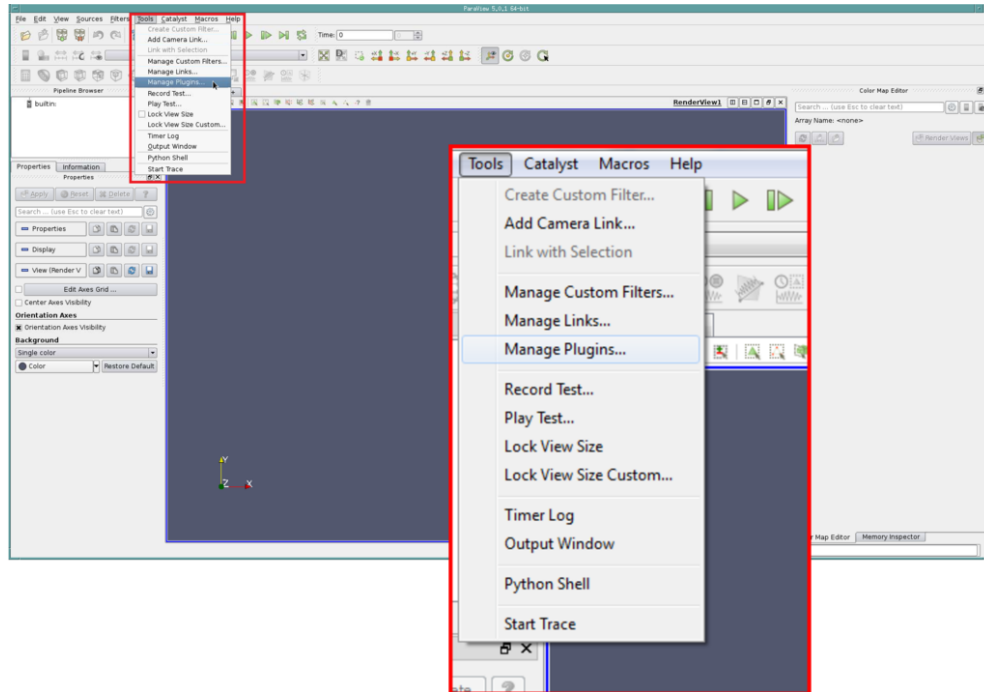


Fig. 3.1 – [Tools ► Manage Plugins] from ParaView menu

- Click on [Load new] option and select libpvNVIDIAIndex. on Linux or pvNVIDIAIndex. on Windows by navigating into the plugin install directory.

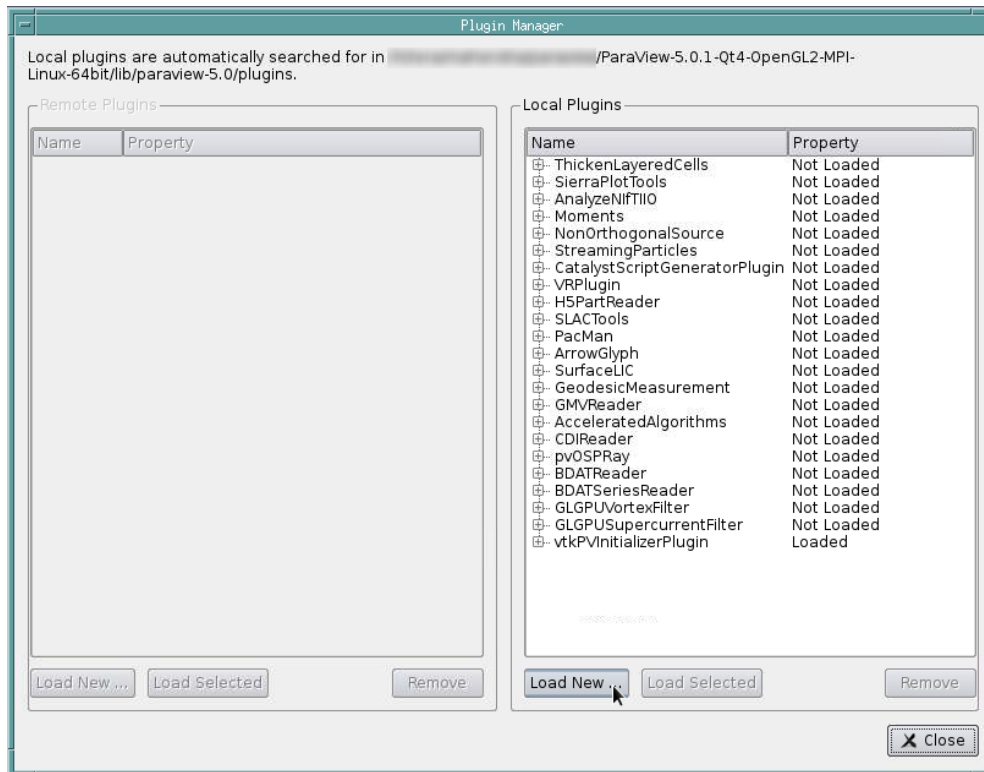


Fig. 3.2 – Click Load New option and locate the plugin

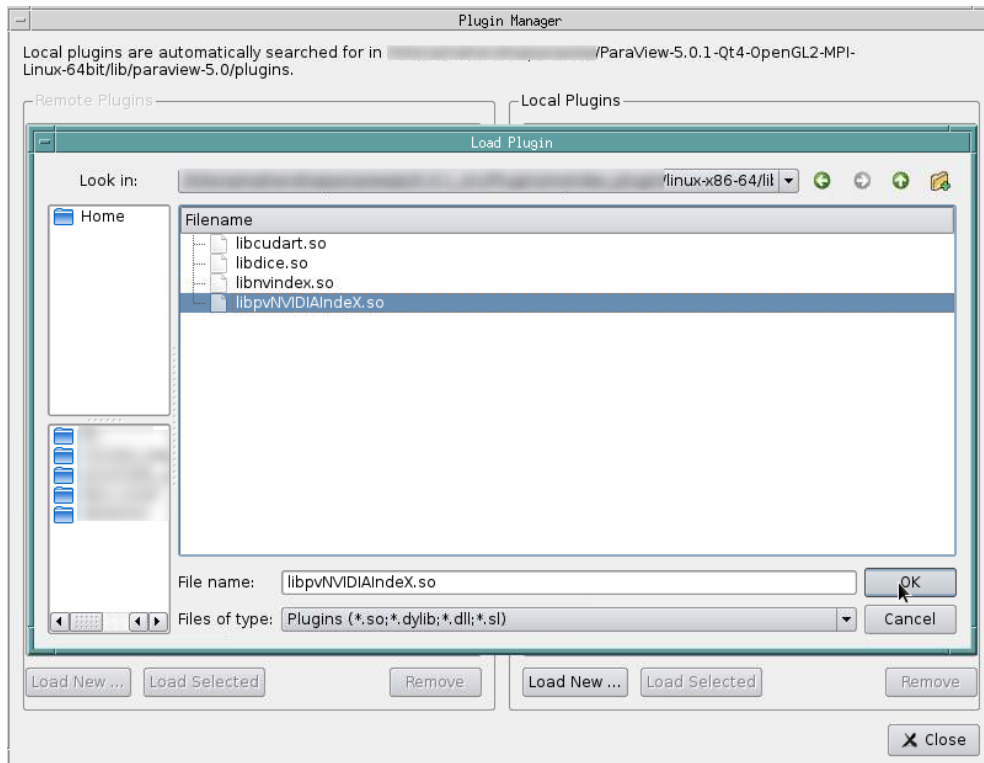


Fig. 3.3 – Load NVIDIA Index for ParaView plugin

- Once the plugin is loaded, plugin name shows up in the Manage Plugins dialog box with status changed as *loaded*. Make sure there no errors in the terminal or on ParaView's console. Also, when using the plugin in client-server mode, be sure to load the plugin in both client and server side of the Manage Plugins window.

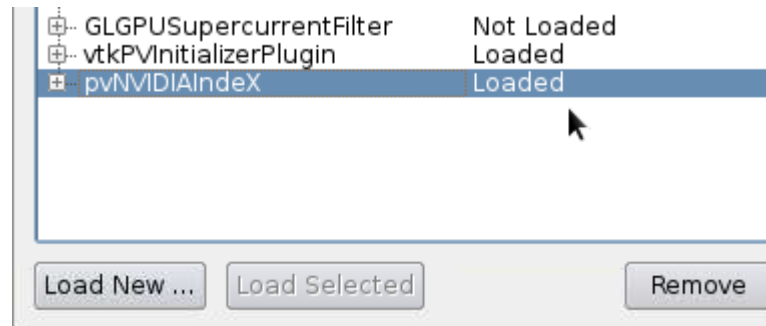


Fig. 3.4 – Status is shown as *Loaded* when no errors

4 Getting started

This section will describe instructions on how to use the NVIDIA IndeX for ParaView plugin in both client-only and client-server mode.

4.1 Client-only mode

To run the plugin in client-only mode simply launch the ParaView client and load the plugin as described in section 2.

4.2 Client-server mode on a single GPU

To run the plugin in client-server mode, start pvserver with mpirun as shown below.

```
mpirun -bynode -np 1 pvserver -display :0.0 --use-offscreen-rendering
```

Once pvserver process is launched, run the ParaView client and connect to the server where pvservers are running by using [File ► Connect ► Add Servers] option in ParaView's menubar. Typically the server address is printed out in the console where mpirun was executed, once the client-server is connected, console will update the status with "Client connected" message. Make sure to load the plugin on both client and server side as described in section 2.

To verify that the plugin is installed correctly and loaded successfully in ParaView, please create a *Wavelet* source by clicking the menu option [Sources ► Wavelet] in ParaView client.

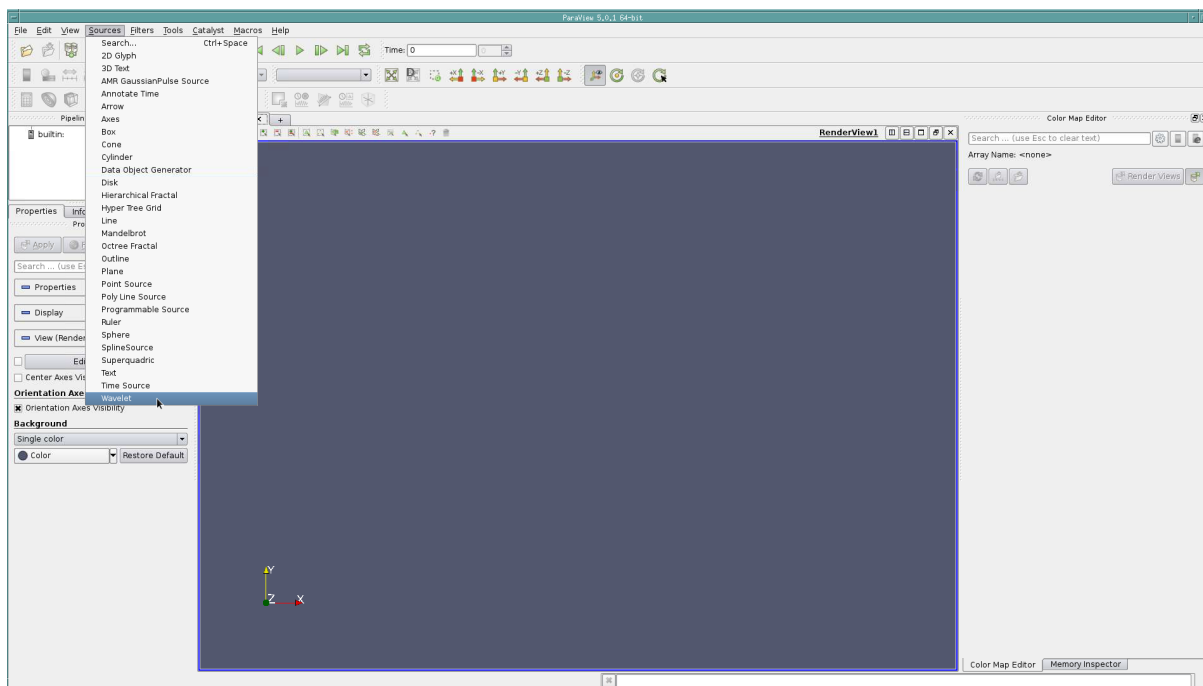


Fig. 4.1 – Create a Wavelet source

Once you click apply an *Outline* representation will be shown in the viewport. Select *RTData* as the scalar array from the dropdown box instead of *Solid Color* and *NVIDIA IndeX* instead of *Outline* as the representation as shown below.

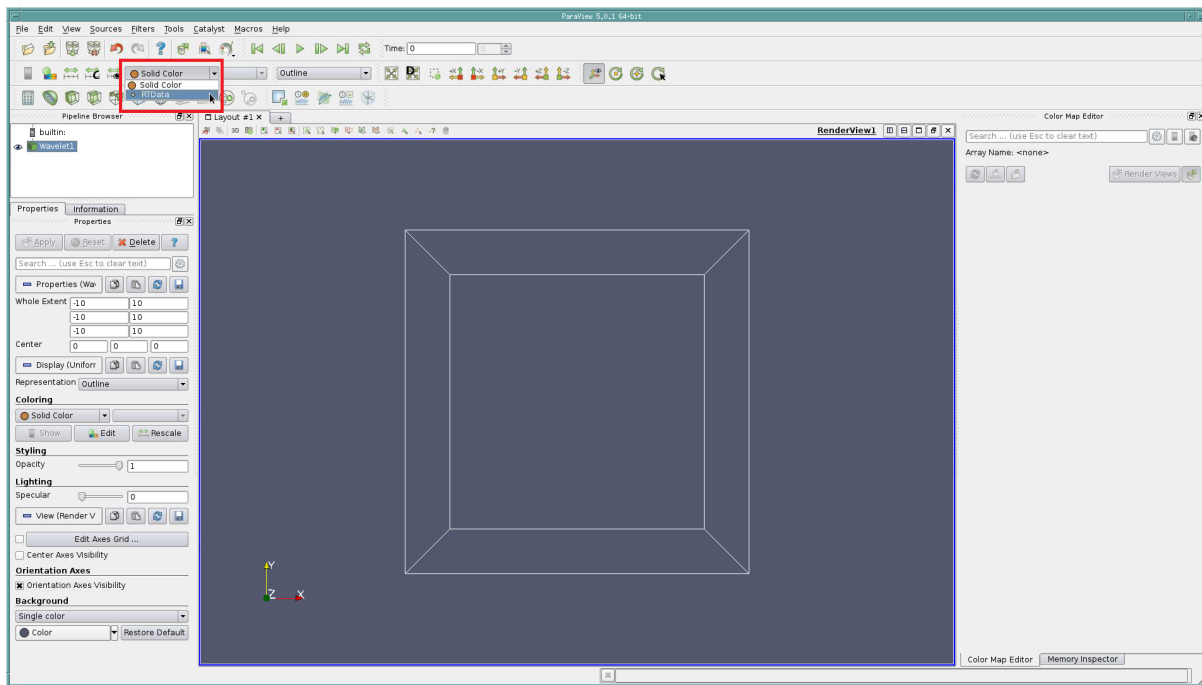


Fig. 4.2 – Outline is default representation

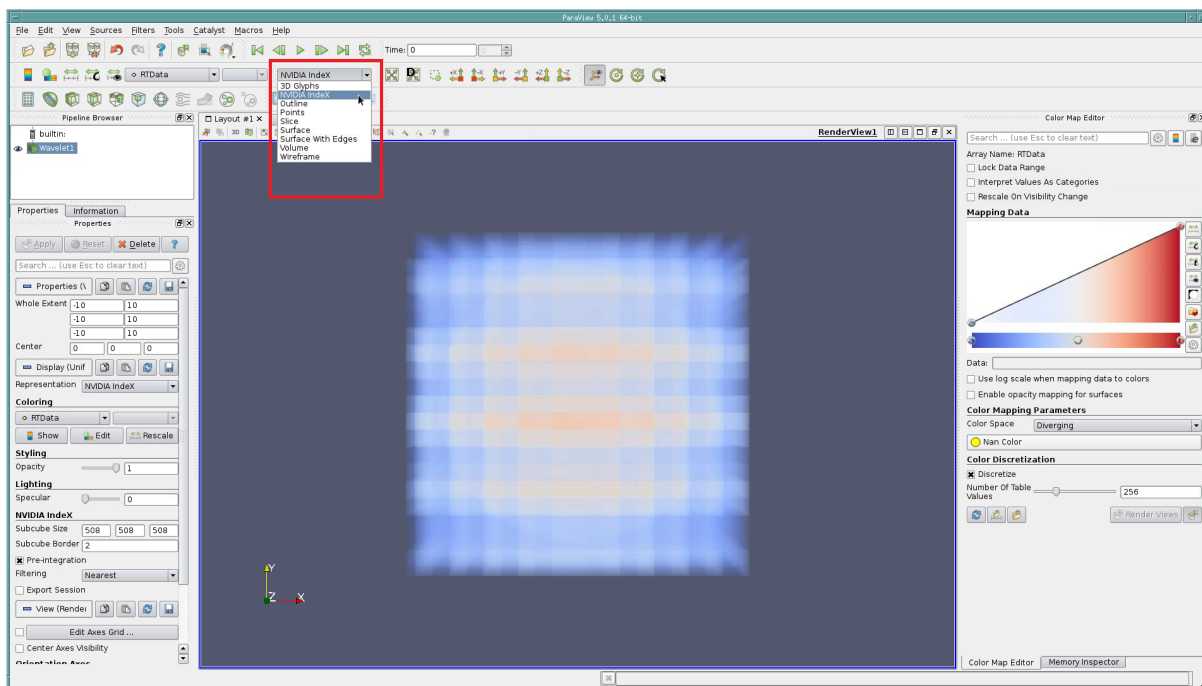


Fig. 4.3 – Wavelet source rendered by NVIDIA Index

4.2.1 Client-server mode on multiple GPUs

If you have acquired a valid license for the cluster version of the plugin, you can run the plugin in client-server mode on multiple GPUs. When using multiple GPUs on the same machine it is required to start one MPI process per GPU. For example, on a machine with 4 GPU's:

```
mpirun -bynode -np 1 pvserver -display :0.0 --use-offscreen-rendering \  
      : -np 1 pvserver -display :0.1 --use-offscreen-rendering \  
      : -np 1 pvserver -display :0.2 --use-offscreen-rendering \  
      : -np 1 pvserver -display :0.3 --use-offscreen-rendering \
```

When using multiple GPUs on multiple machines in a cluster environment MPI's hostfile functionality can be used. For example, to utilize two machines with two GPUs each:

```
mpirun --hostfile myhosts --bynode -np 2 pvserver -display :0.0 --use-offscreen-rendering \  
      : -np 2 pvserver -display :0.1 --use-offscreen-rendering
```

Where *myhosts* is a text file with list of host names where MPI will spawn pvserver instances.

Before connecting pvservers with ParaView client, please disable IceT compositing. This can be done from ParaView client's settings menu [Edit ► Settings ► Render view] and restart ParaView client.

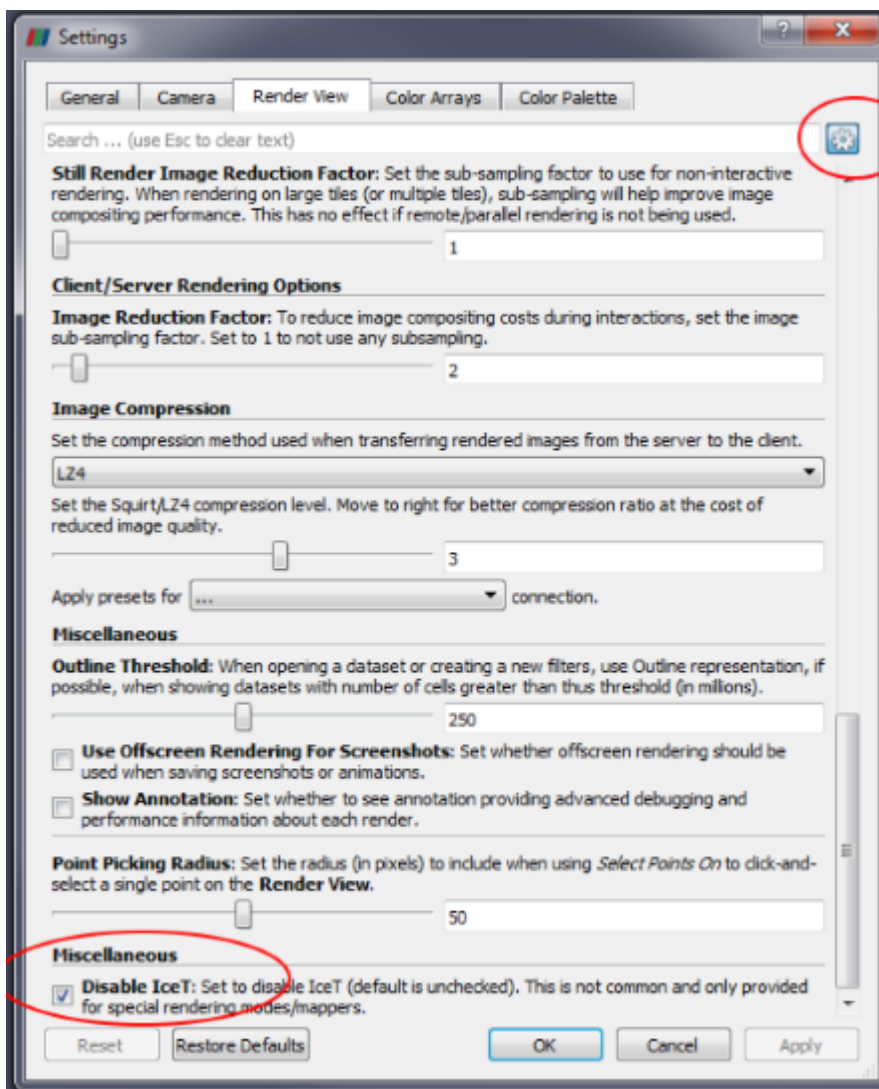


Fig. 4.4 – Disable IceT from Settings menu

Refer to [this page](http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server)³ for more details about pvserver and running ParaView in client-server mode.

4.2.2 Networking parameters for NVIDIA Index

NVIDIA Index is a distributed renderer capable of utilizing multiple GPU's on a cluster, to configure networking features of NVIDIA Index an optional configuration file `nvindex_config.xml` is provided with the plugin. Copy this file under the default config directory of ParaView.

```
cp <plugin-directory>/nvindex_config.xml ~/.config/ParaView/
```

Each networking option is enclosed within a pair of XML tags and the file is enclosed within `<index_config>...</index_config>` tags.

³http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server

4.2.2.1 Licensing

When you obtain a license for the cluster version of the plugin you will receive license.lic file with the NVINDEX_VENDOR_KEY and NVINDEX_SECRET_KEY that can be set as environment variables on all the hosts where NVIDIA IndeX is run.

```
export NVINDEX_VENDOR_KEY="input_vendor_key_here"  
export NVINDEX_SECRET_KEY="input_secret_key_here"
```

Alternatively, copy paste those keys in the <license></license> section of the config file as show below.

```
<license>  
    <vendor_key>input_vendor_key_here</vendor_key>  
    <secret_key>input_secret_key_here</secret_key>  
</license>
```

4.2.2.2 Networking options

All the networking configuration options for the plugin are specified under `<network></network>` section of the config file.

Tag `<cluster_mode>` defines the networking mode of NVIDIA IndeX with modes **UDP** and **TCP** supported, with UDP being the preferred mode.

```
<cluster_mode>your-networking-mode</cluster_mode>
```

Tag `<cluster_interface_address>` defines the Network Interface Card(NIC) that is used for communication between the nodes. On Linux, *ifconfig* command gives the NIC address as *inet addr*. If not set, any address is valid. The string may end with `:` and a port number to select which port to listen to for UDP and TCP. If no port is set and unicast only mode is set, port 10000 will be used.

```
<cluster_interface_address>  
    172.161.123.0/24:10001  
</cluster_interface_address>
```

```
172.161.123.0/24:10001
```

Tag `<multicast_address>` defines the multicast address for the nodes to communicate. This is valid only when `<cluster_mode>` is set to UDP.

```
<multicast_address>224.1.3.2</multicast_address>
```

Tag `<discovery_address>` defines the discovery address used for TCP `<cluster_mode>`.

```
<discovery_address>224.1.3.3:5555</discovery_address>
```

Tag `<use_rdma>` can be used to switch **on** or **off** RDMA mode for networking.

```
<use_rdma>no</use_rdma>
```

5 Features

This section will provide a walk-through on individual plugin features.

5.1 Structured and Unstructured grids

NVIDIA IndeX for ParaView plugin enables volume rendering of both structured and unstructured grid types.

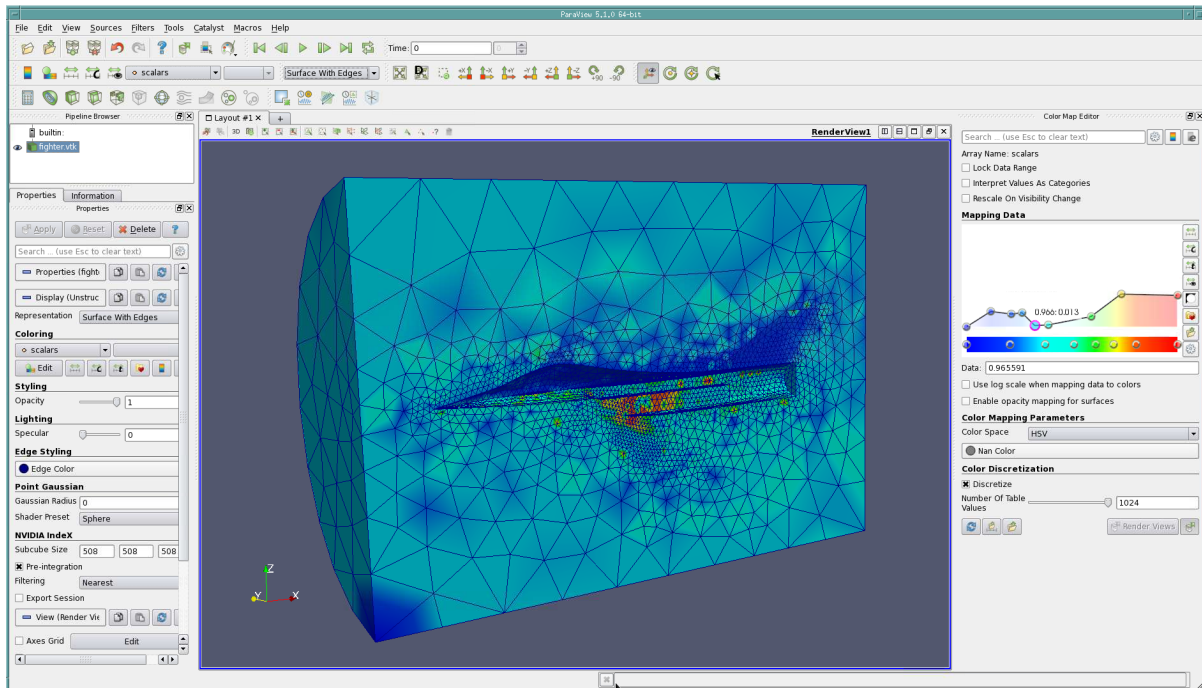


Fig. 5.1 – Grid rendered as a surface

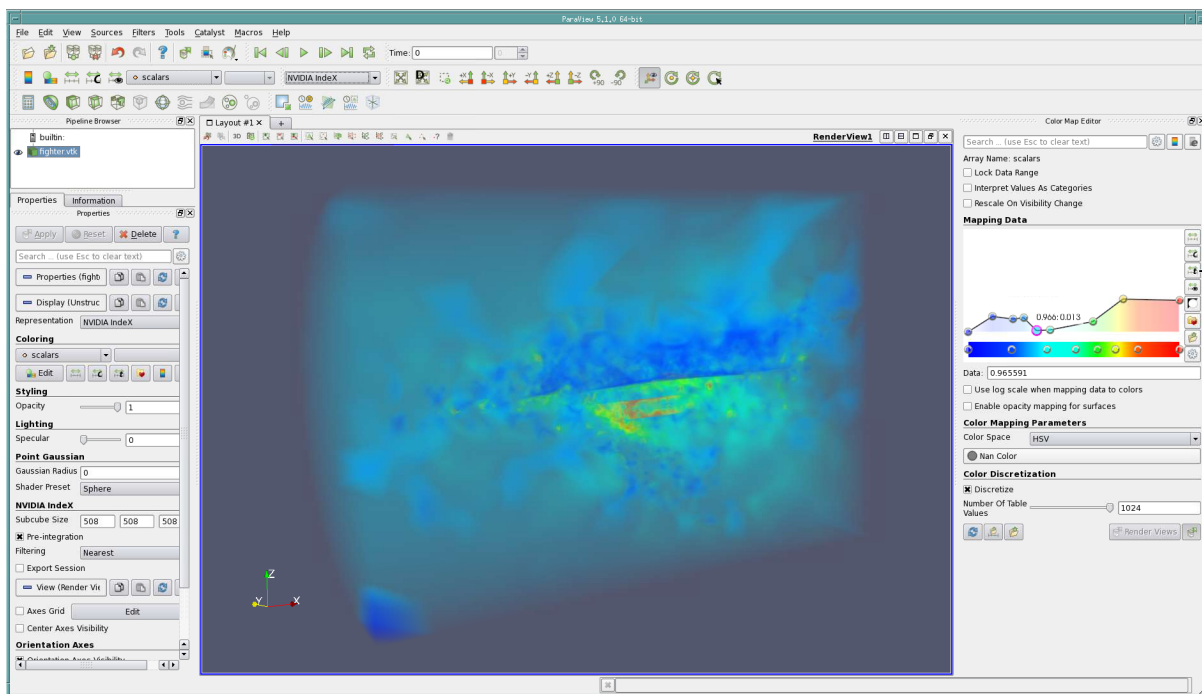


Fig. 5.2 – Grid rendered as a volume using NVIDIA IndeX

5.2 Datatypes

NVIDIA IndexX supports different datatype formats such as *unsigned char*, *unsigned short* and *floating point*. Make sure appropriate byte endianness is chosen when loading unsigned short and floating point datatypes otherwise your visualization might look like artifacts or even look completely random.

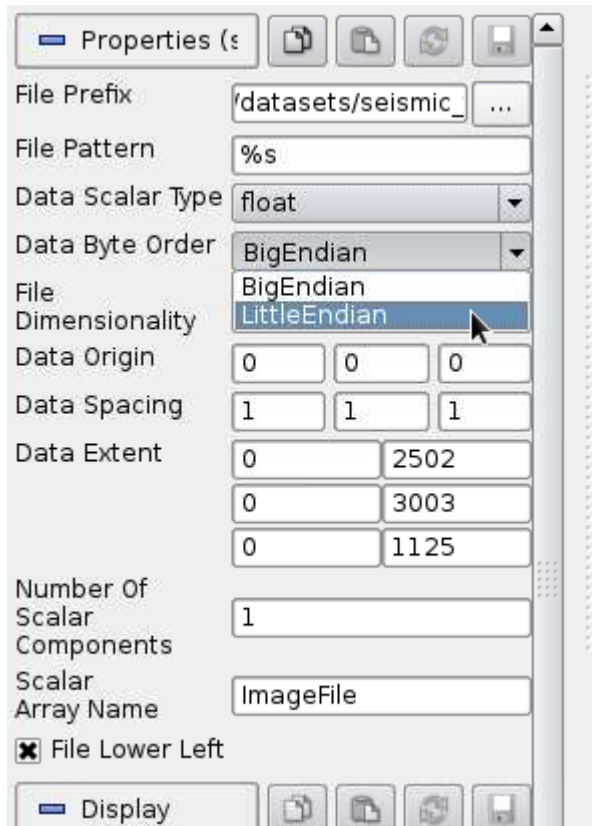


Fig. 5.3 – Choose endianness of the dataset

5.3 Transfer function and colormap changes

Transfer function changes can be done using ParaView's colormap editor. If the colormap editor is not open you can do so by using the menu option.

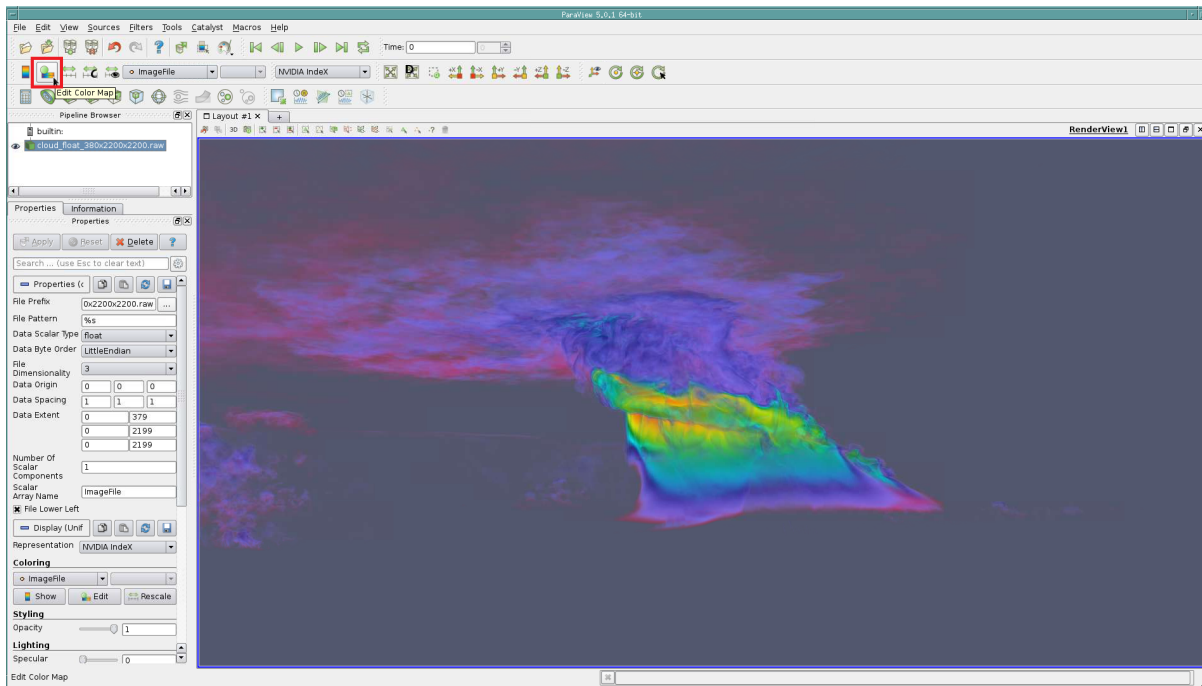


Fig. 5.4 – Click on the icon to open up Colormap Editor in ParaView

Using the colormap editor user interface you can visualize parts of the dataset that is interesting for you. This can be achieved by changing the colortable, by adjusting the transparency, or by setting custom domain range values to isolate parts of the dataset that is uninteresting.

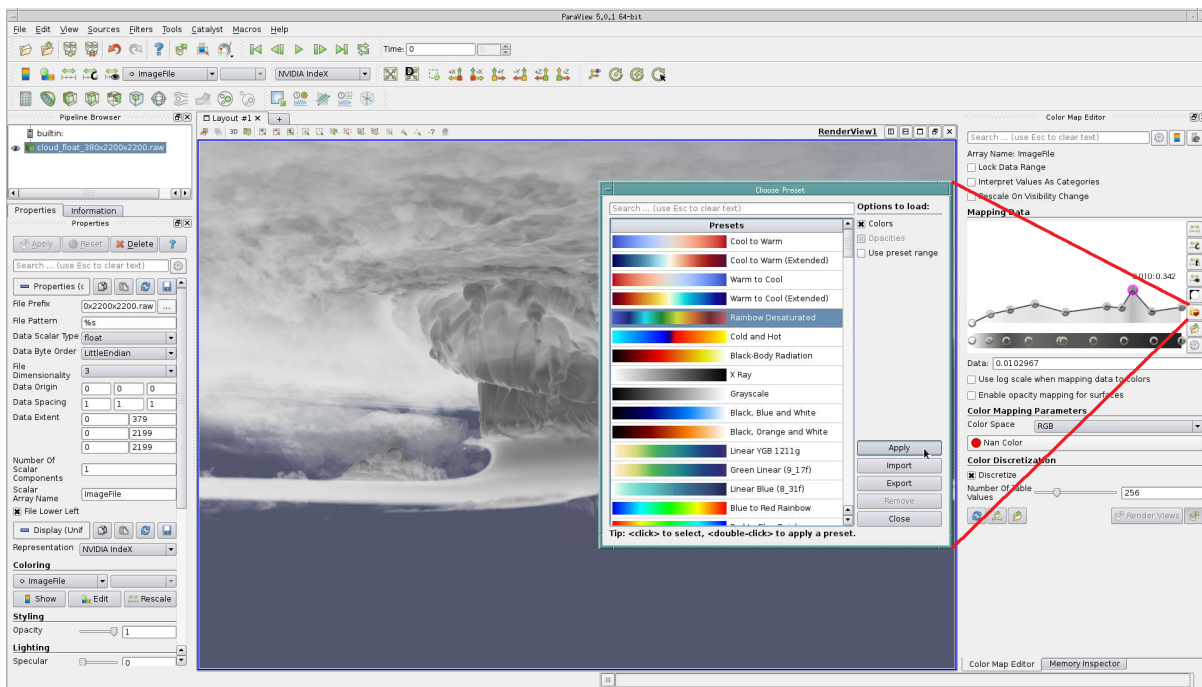


Fig. 5.5 – Choose a suitable colortable and click Apply

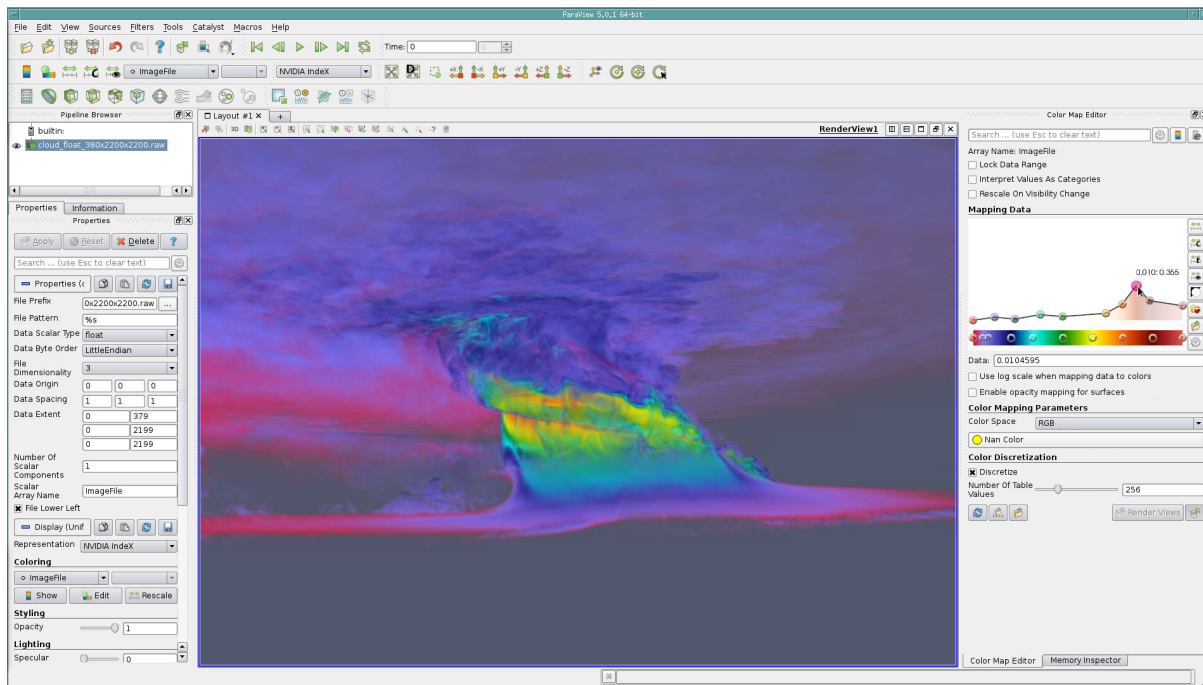


Fig. 5.6 – Colortable matching your dataset domain

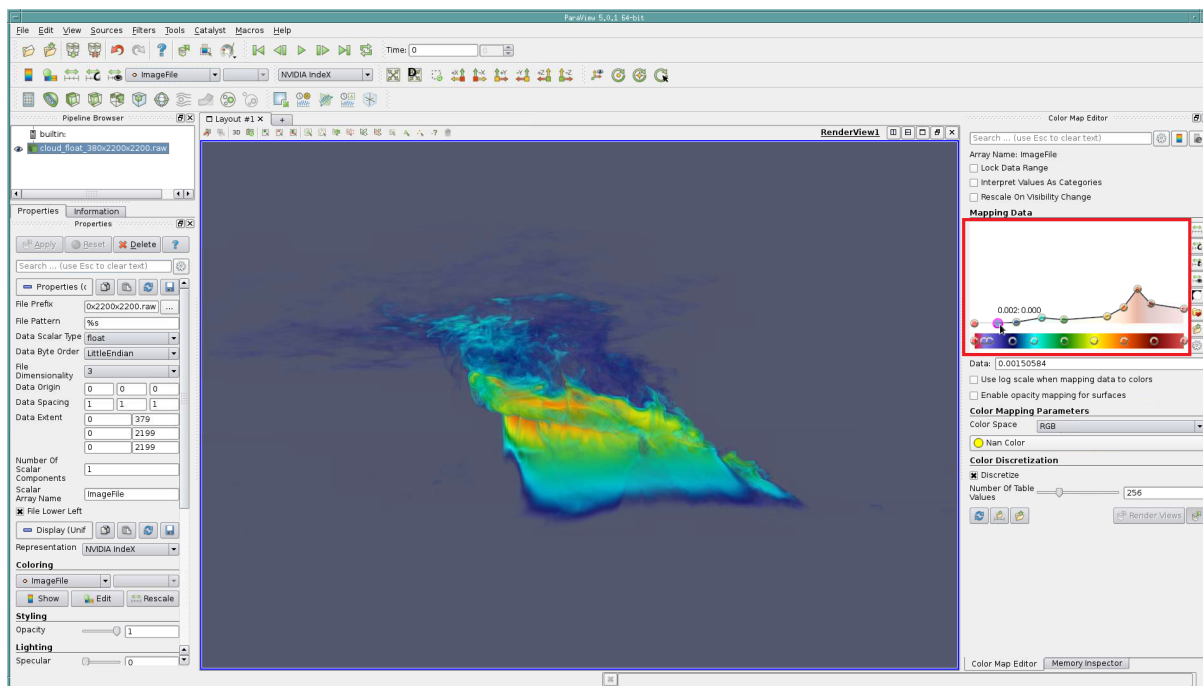


Fig. 5.7 – Changing the opacity values

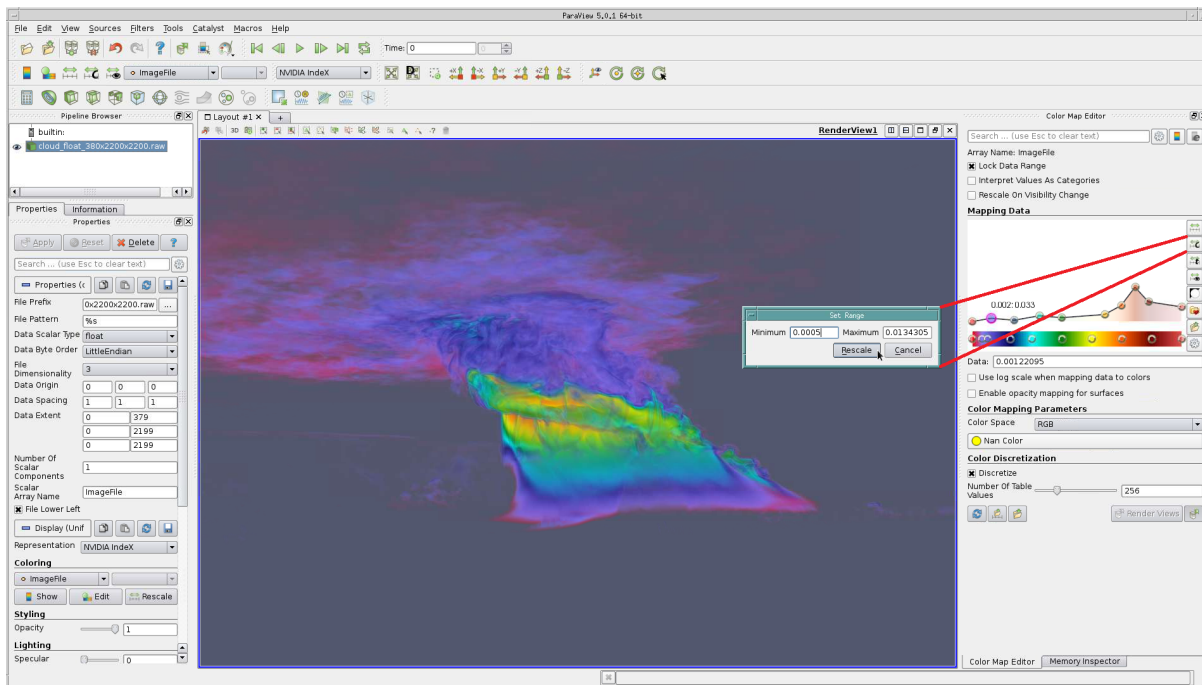


Fig. 5.8 – Custom data domain range

5.4 Region of interest changes

Users can select a custom region of interest to visualize specific sections of the dataset using the sliders in the properties panel. This is tagged as an experimental feature since changing region of interest is restricted to axis aligned directions.

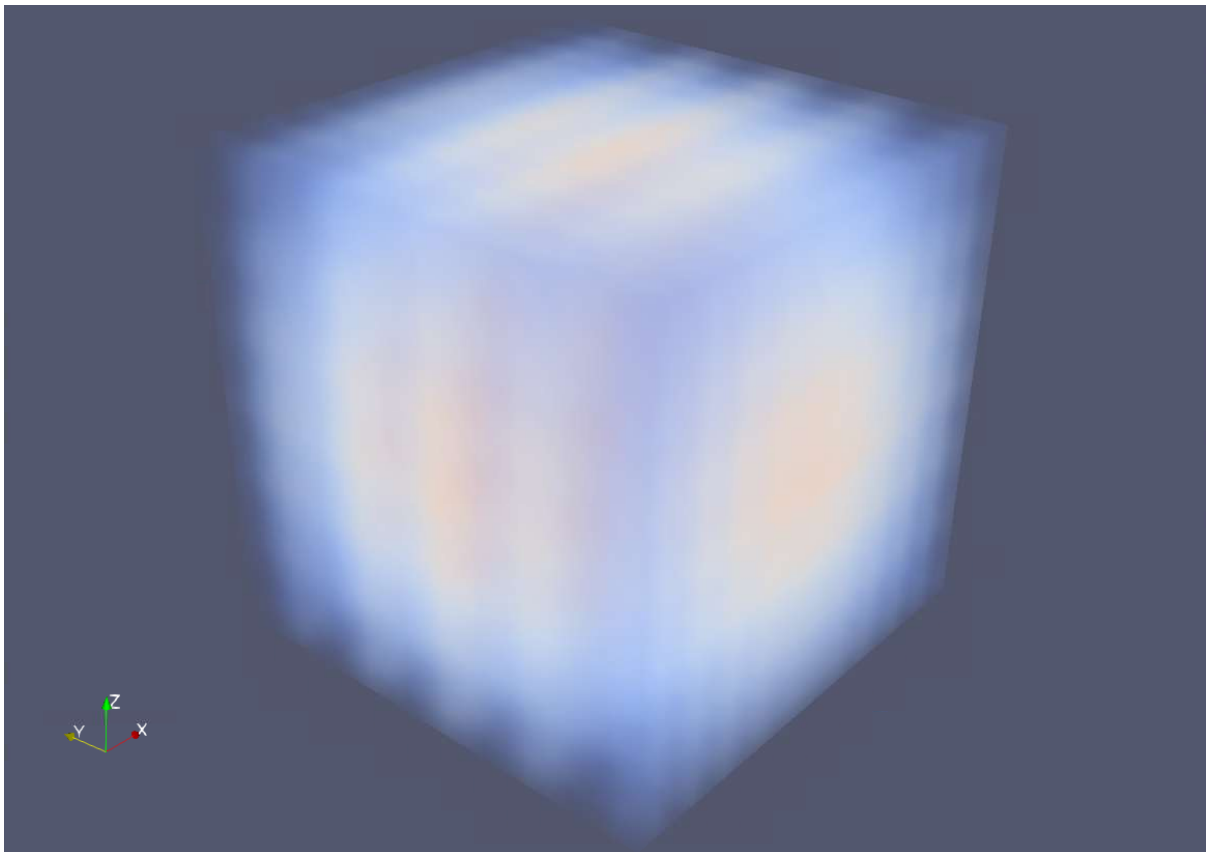


Fig. 5.9 – Wavelet volume

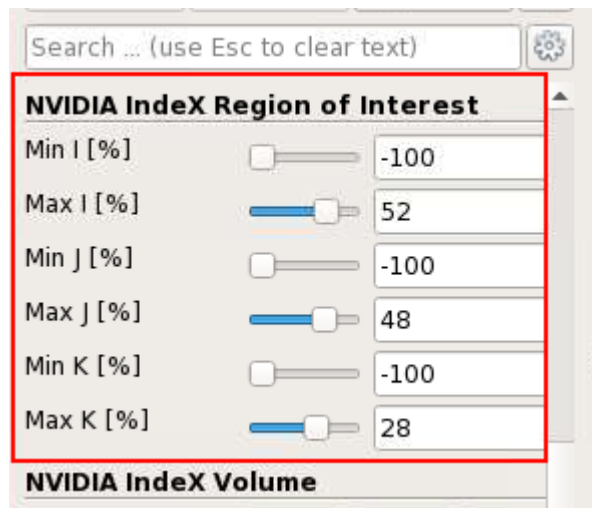


Fig. 5.10 – Changing region of interest

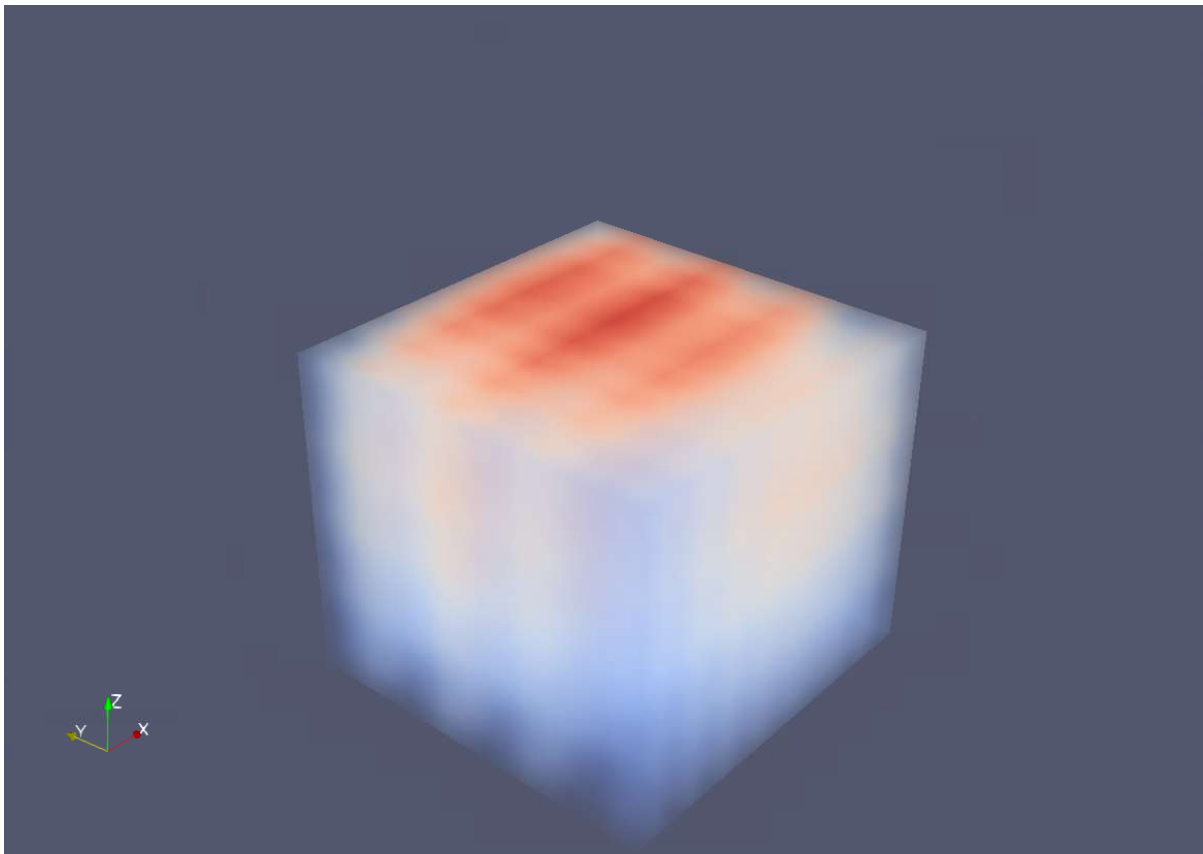


Fig. 5.11 – Wavelet volume with a custom region of interest

5.5 High-quality

High quality rendering can be achieved by using pre-integration and filtering techniques of NVIDIA IndeX exposed in the plugin, there is a known performance-quality trade off when using some of these filtering techniques.

These filtering options can be found in ParaView's **Properties** panel typically on the left hand side of the ParaView client user interface when the plugin is loaded.

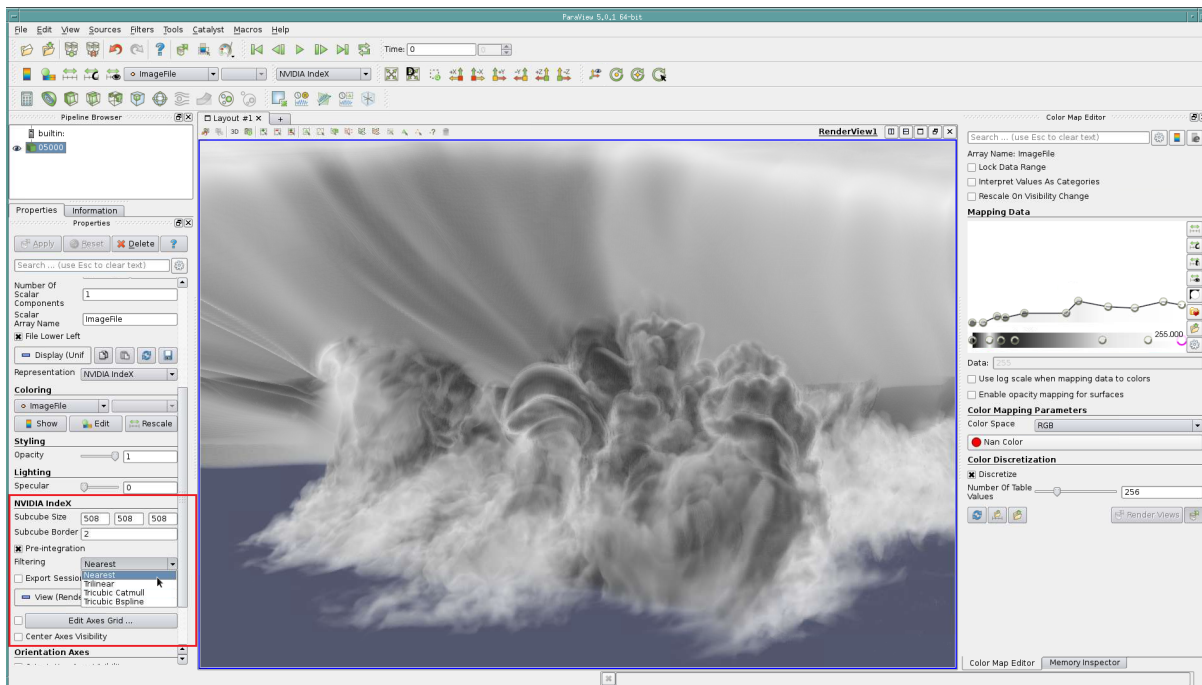


Fig. 5.12 – Properties panel in ParaView

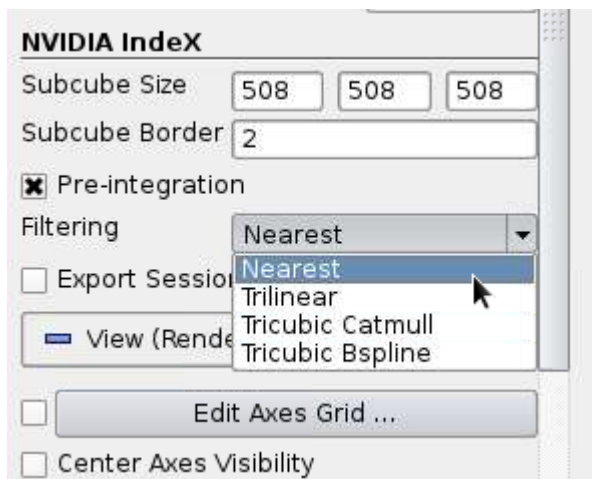


Fig. 5.13 – Filtering options in NVIDIA IndexX properties panel

There is no one filtering option optimal for all the datasets, each filtering option achieves different levels of quality with different datasets and transfer function combinations with nearest neighbor interpolation being the most basic one. Some example images comparing different filtering options are shown below.

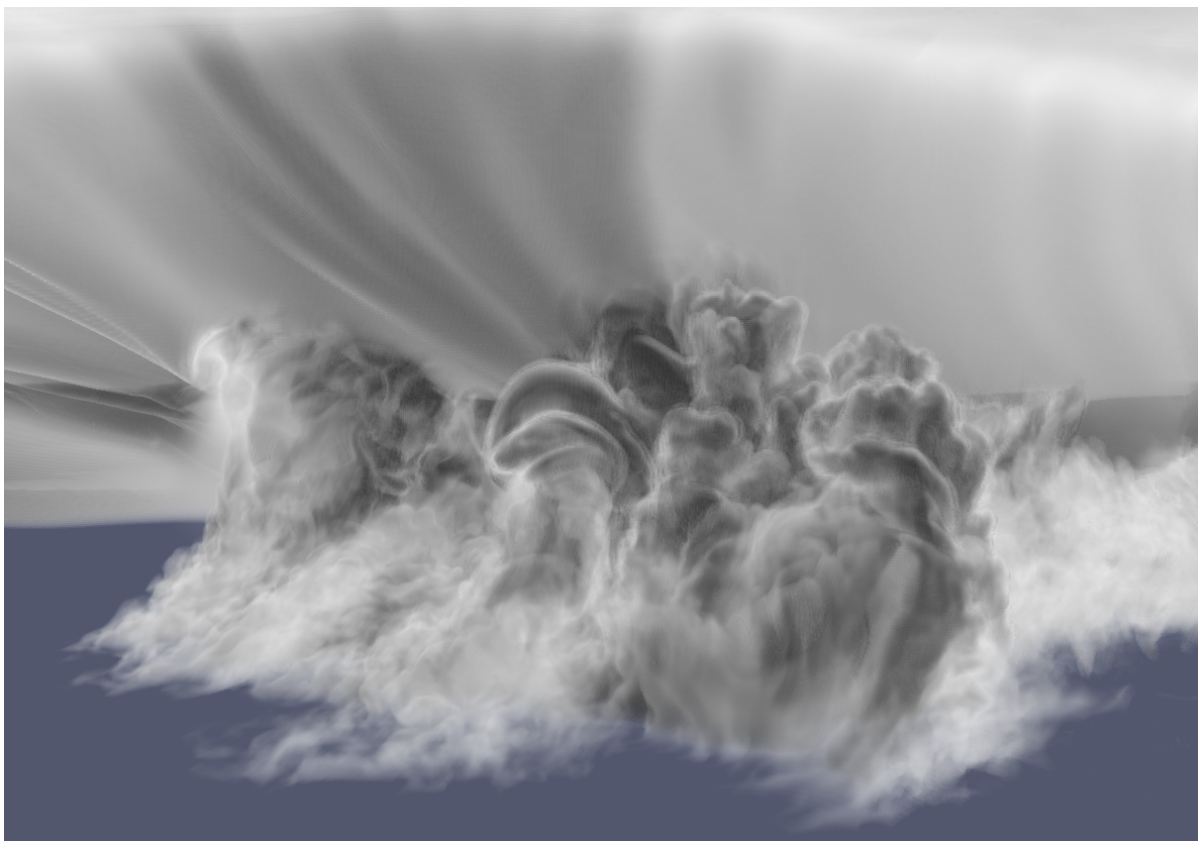


Fig. 5.14 – Base of an EF5 tornado cloud

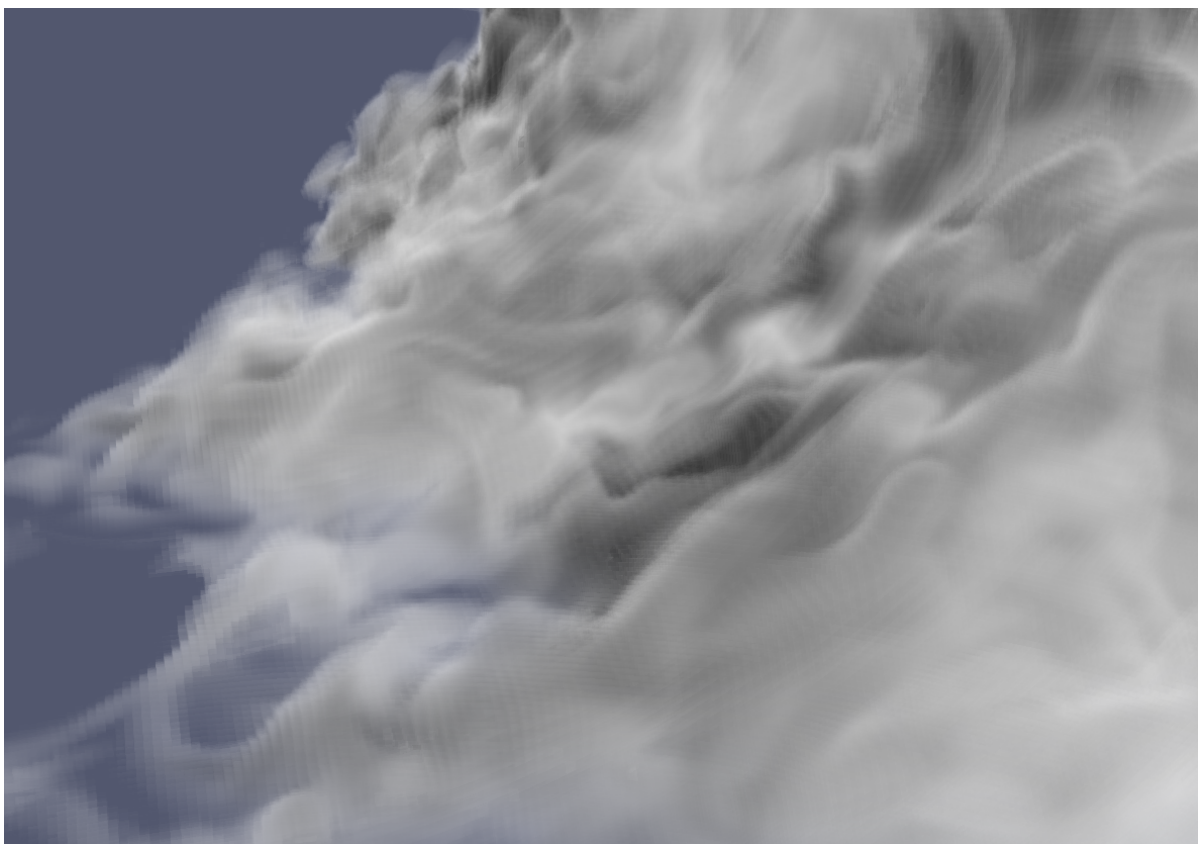


Fig. 5.15 – Nearest neighbor interpolation

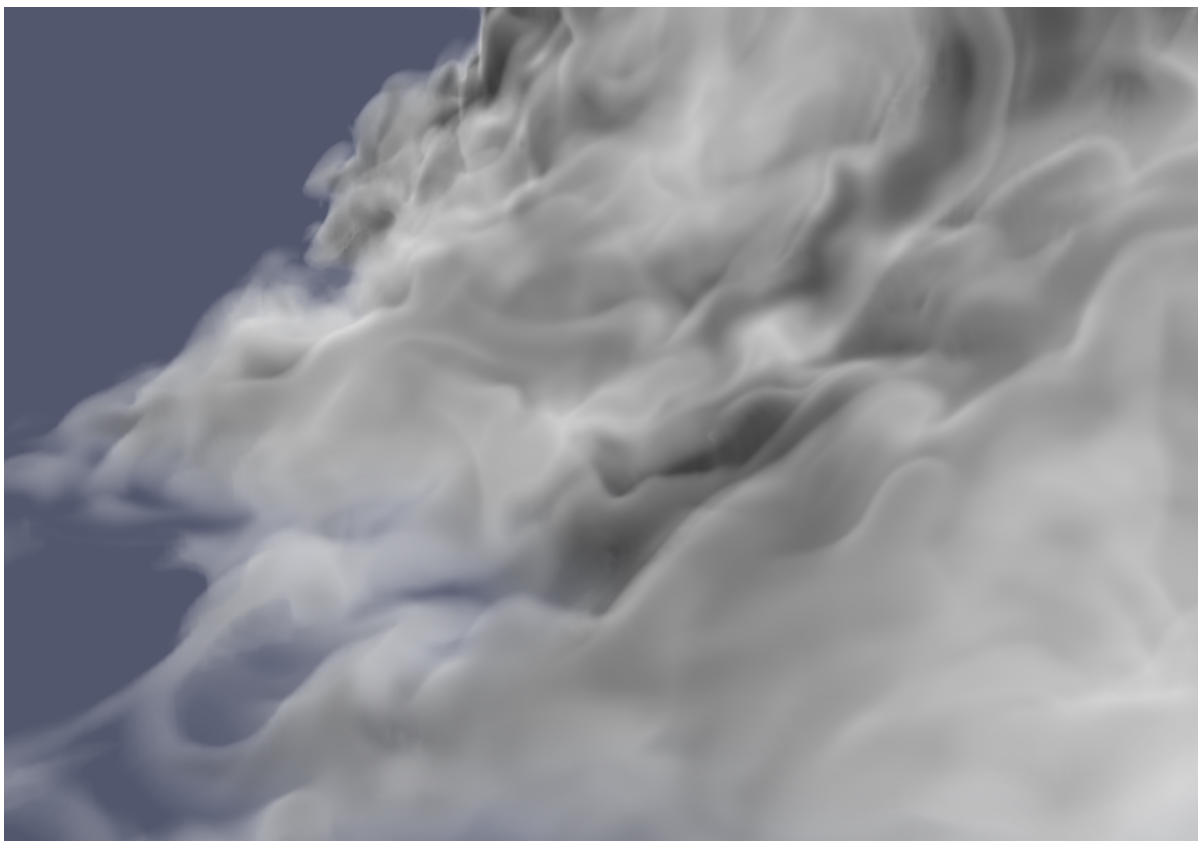


Fig. 5.16 – Trilinear interpolation

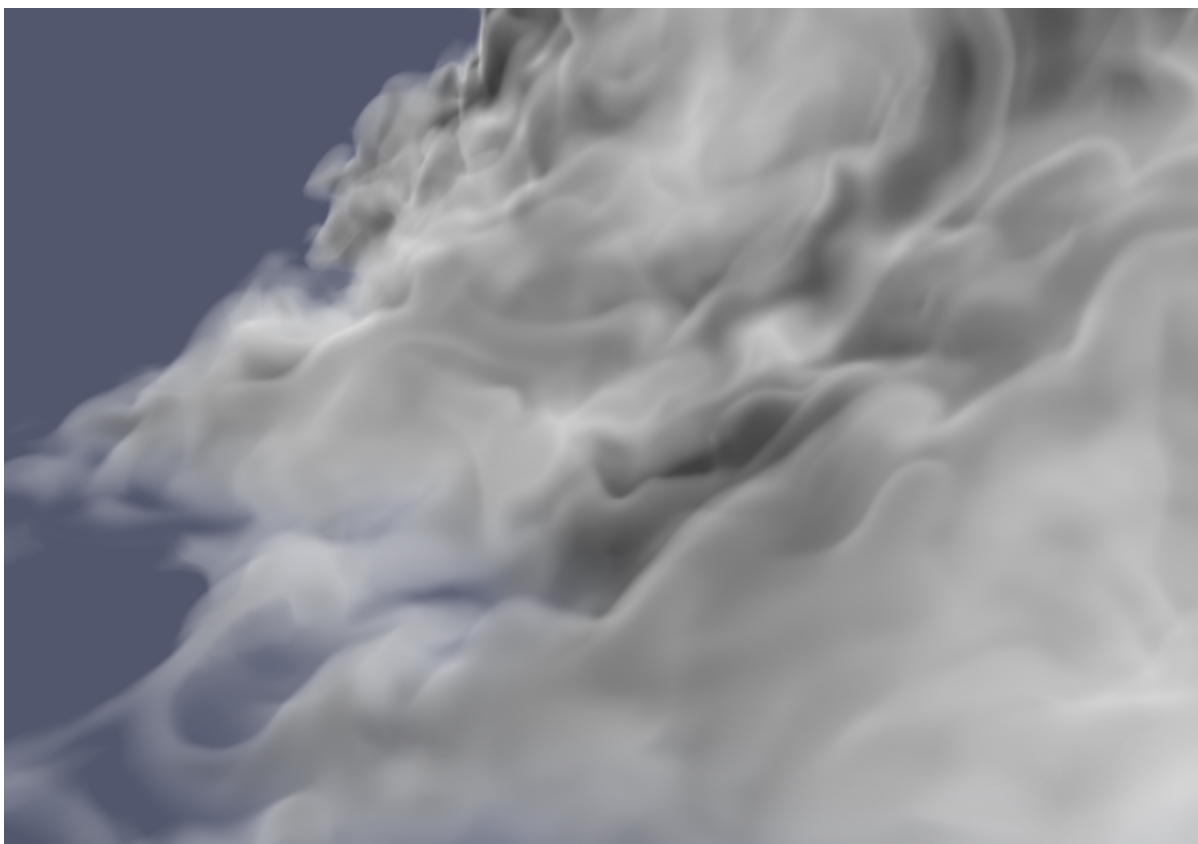


Fig. 5.17 – Tricubic Bspline interpolation

5.6 Slice rendering

Volume slices can be enabled from the GUI. Currently it is limited to a single axis aligned slice, native support with ParaView slices is coming soon.

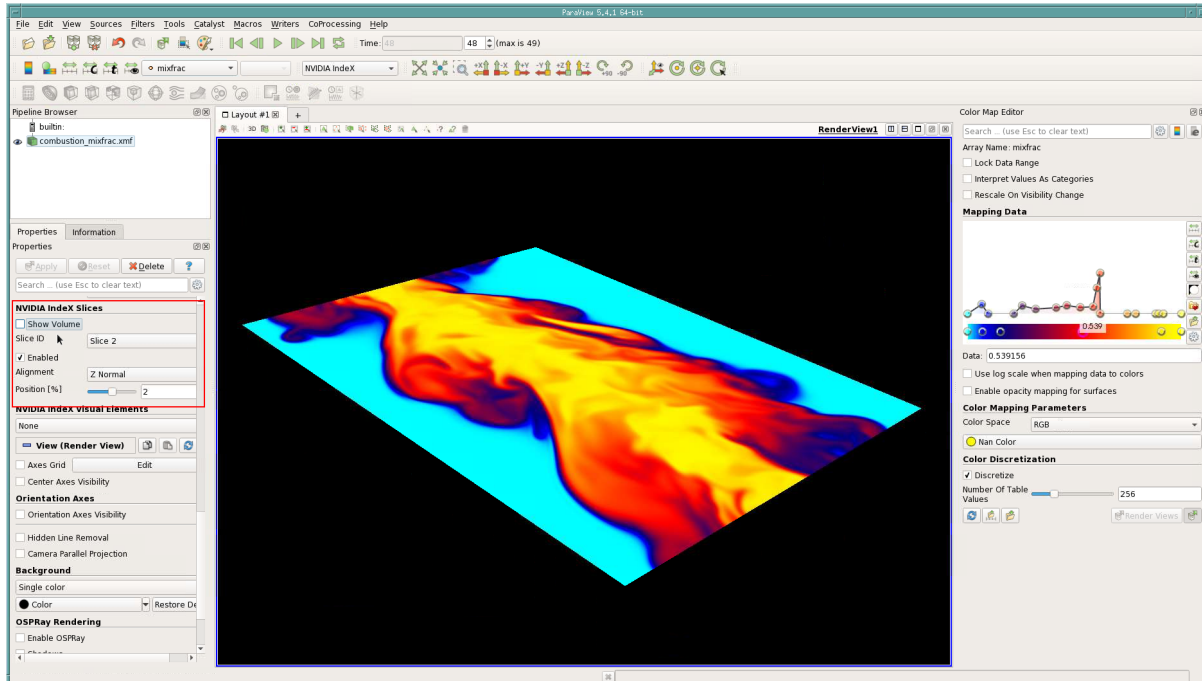


Fig. 5.18 – Properties panel in ParaView

5.7 Slice rendering

Volume slices can be enabled from the GUI. Currently it is limited to three axis aligned slices with ability to move individual slice positions, native support with ParaView slices is work in progress. This is tagged as an experimental feature since the slice rendering is not part of the ParaView slice rendering mechanism in the user interface.

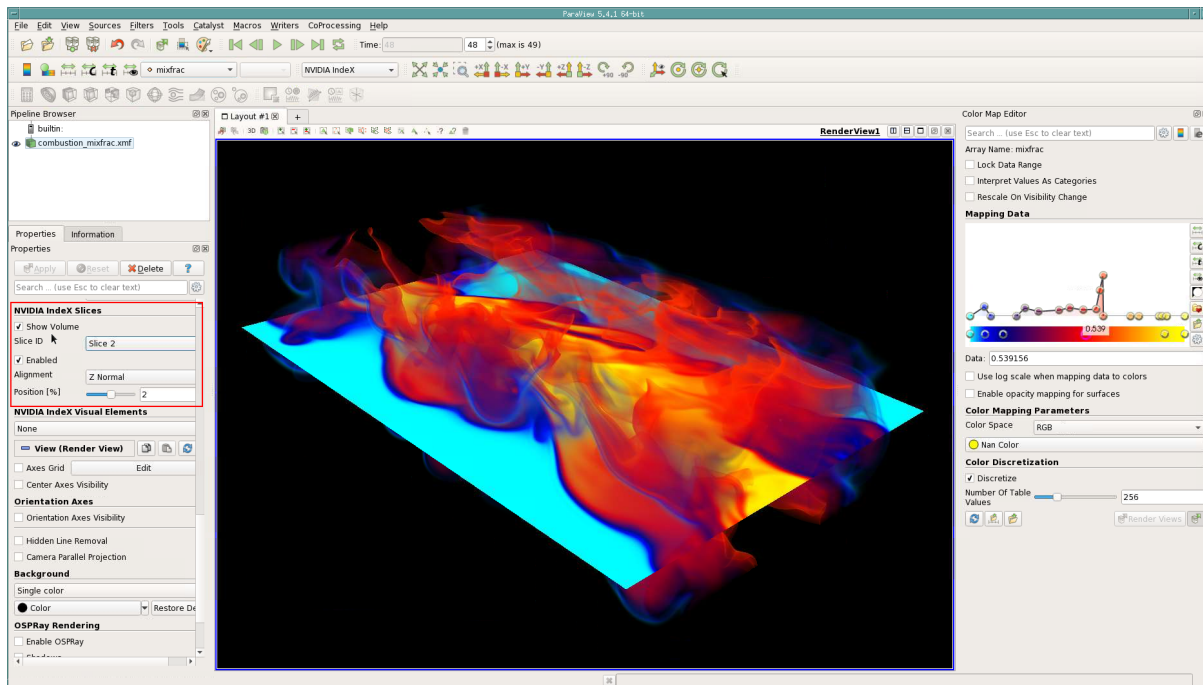


Fig. 5.19 – Slice rendering options through properties panel. Dataset is made available by Dr. Jacqueline Chen at Sandia Laboratories through US Department of Energy's SciDAC Institute for Ultrascale Visualization.

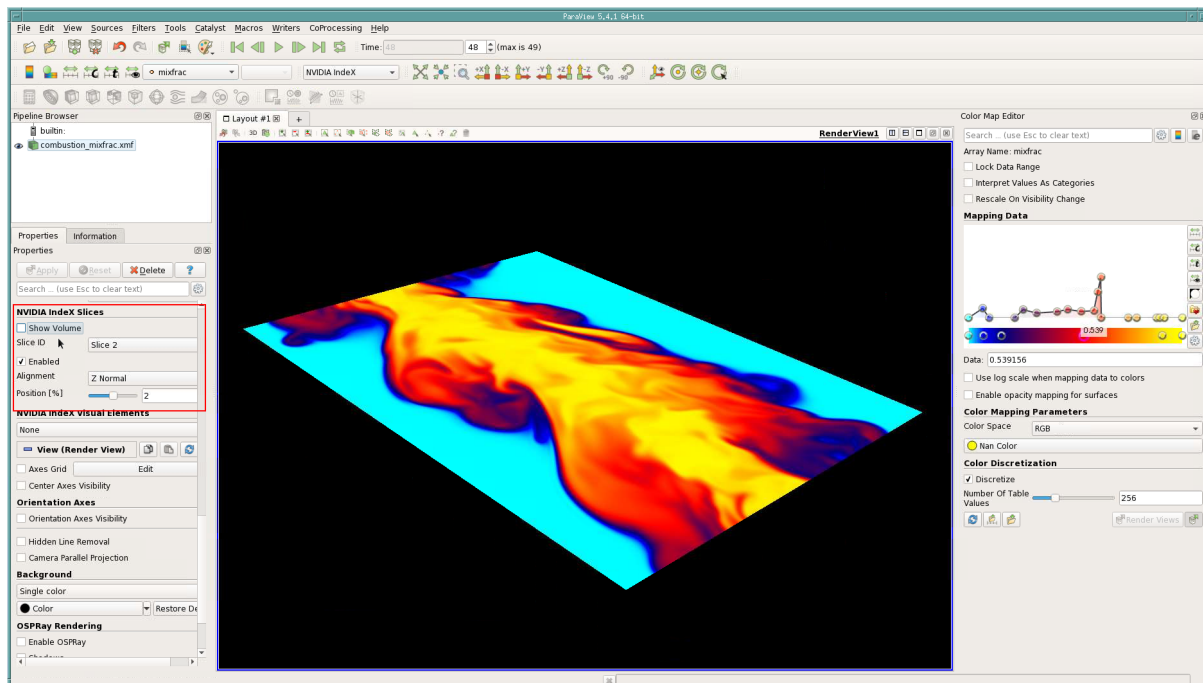


Fig. 5.20 – Slice rendering with volume disabled

5.8 NVIDIA IndeX visual elements

Visual elements feature of NVIDIA IndeX library enables you to enhance the visual cues in the dataset. In this version of the plugin there are three visual presets available, each preset has one or more parameters for finer control over that visual element.

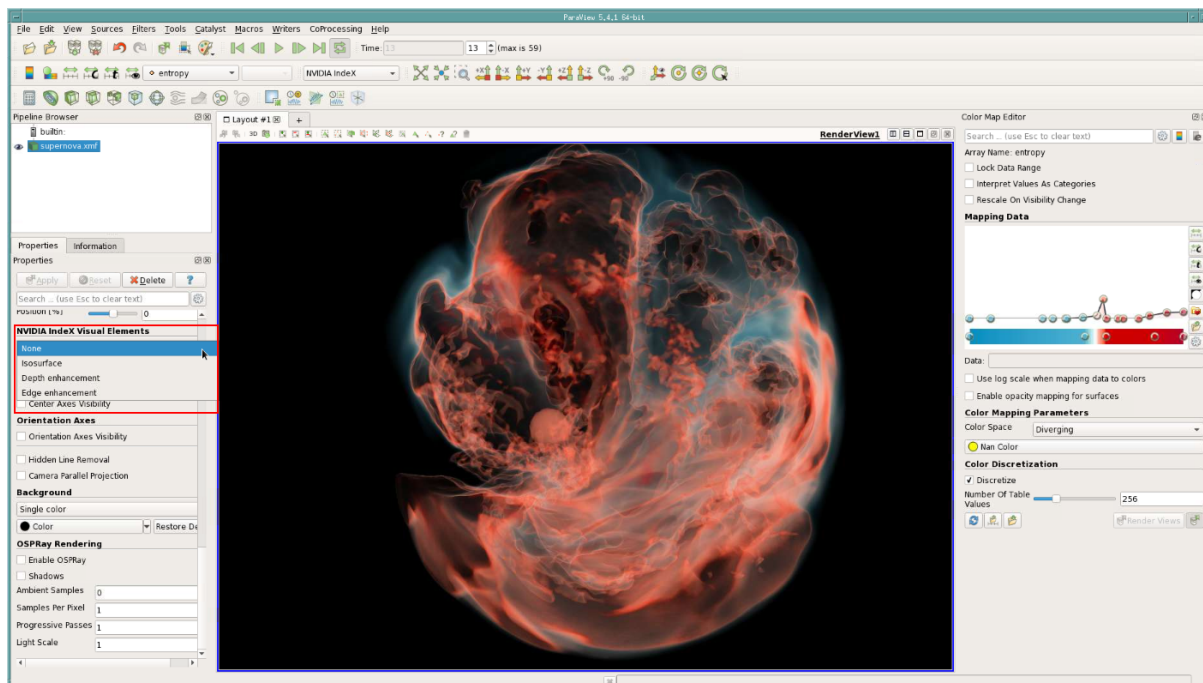


Fig. 5.21 – Supernova SASI visualized as a volume. Dataset is made available by Dr. John Blondin at the North Carolina State University through US Department of Energy’s SciDAC Institute for Ultrascale Visualization.

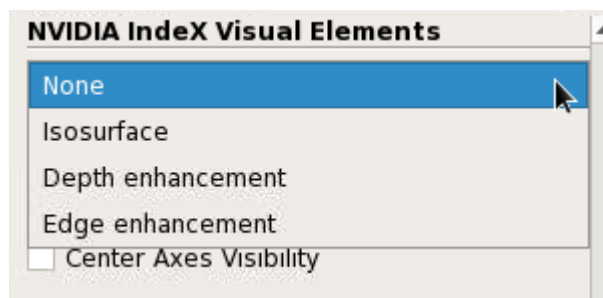


Fig. 5.22 – Visual element presets in the properties panel

5.8.1 Iso-surface preset

Iso-surface preset allows you to extract an iso-surface based on the min/max voxel range with different ways to map color values.

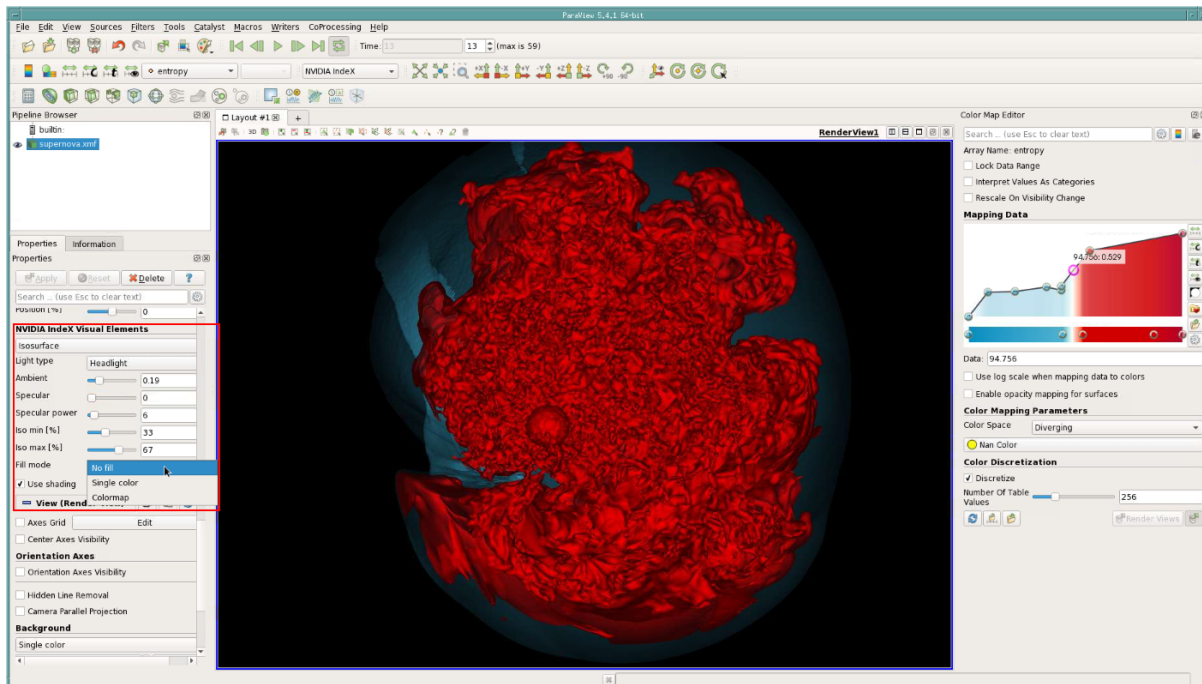


Fig. 5.23 – Supernova SASI visualized as an iso-surface

5.8.2 Depth enhancement preset

Depth enhancement preset allows you to enhance depth perception in the dataset by isolating features with high opacity values in the transfer function mapping.

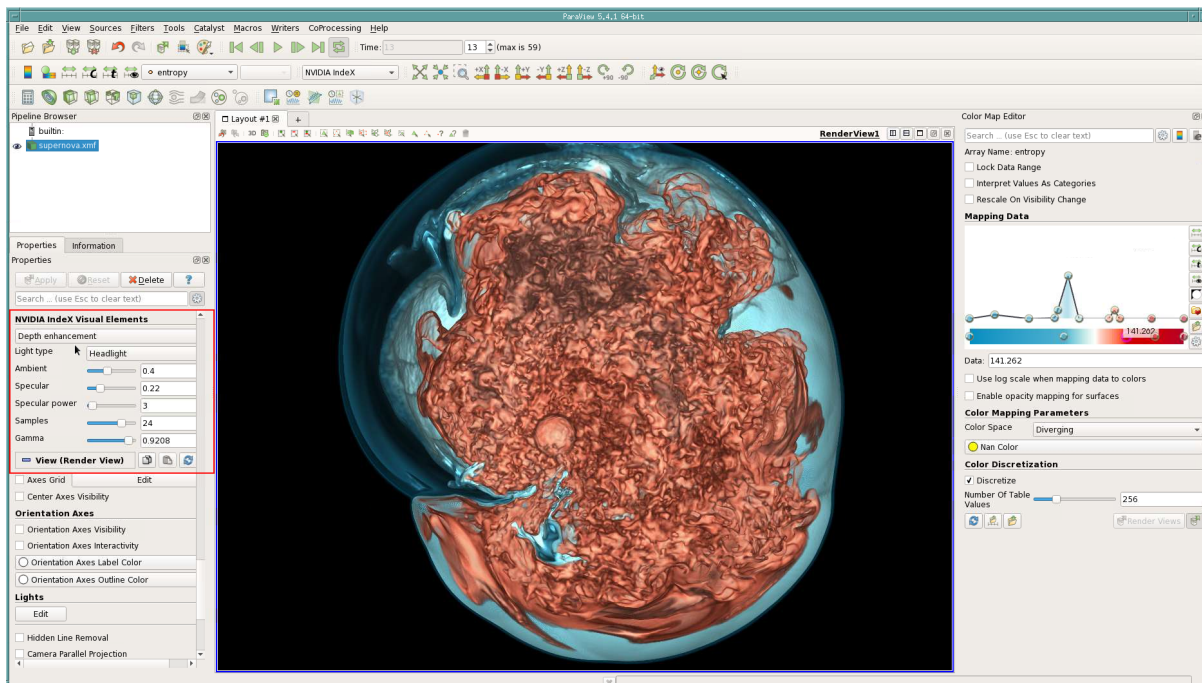


Fig. 5.24 – Supernova SASI visualized using depth enhancement preset

5.8.3 Edge enhancement preset

Edge enhancement preset allows you to enhance the edges or the "silhouettes" based on the transfer function setting.

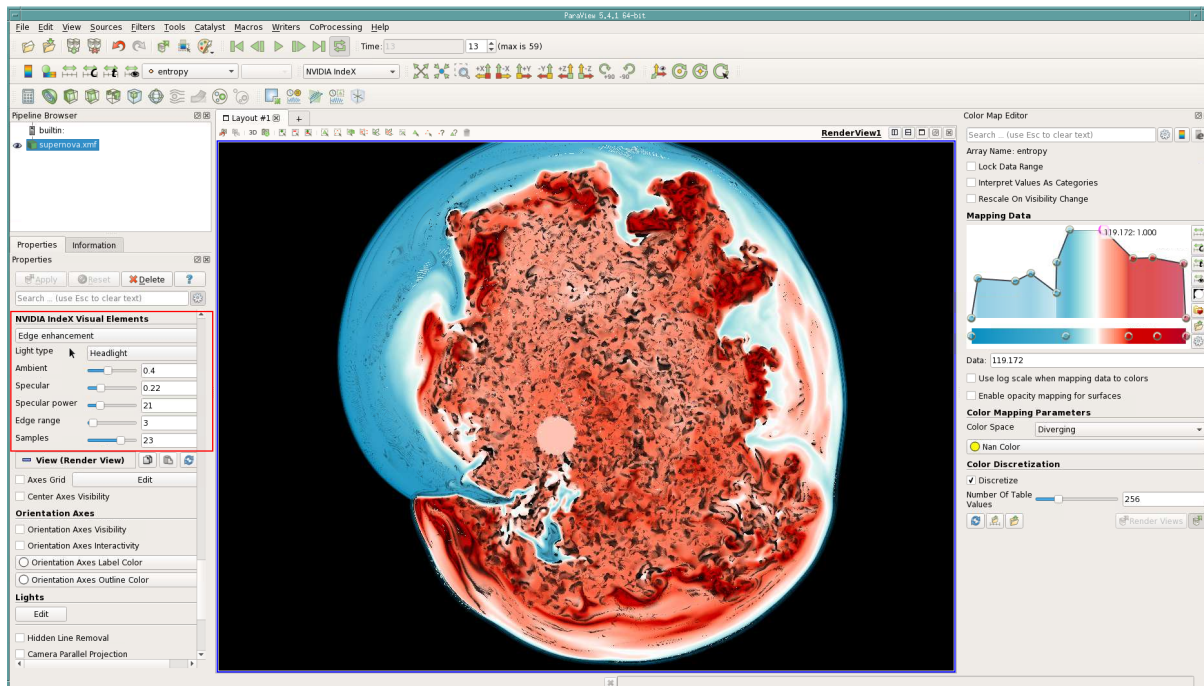


Fig. 5.25 – Supernova SASI visualized using edge enhancement preset

5.9 Mixing ParaView primitives

One of the unique features of the plugin is to mix volume rendering from NVIDIA IndeX along with other primitives such as Glyphs, Streamlines and Surfaces rendered by ParaView.

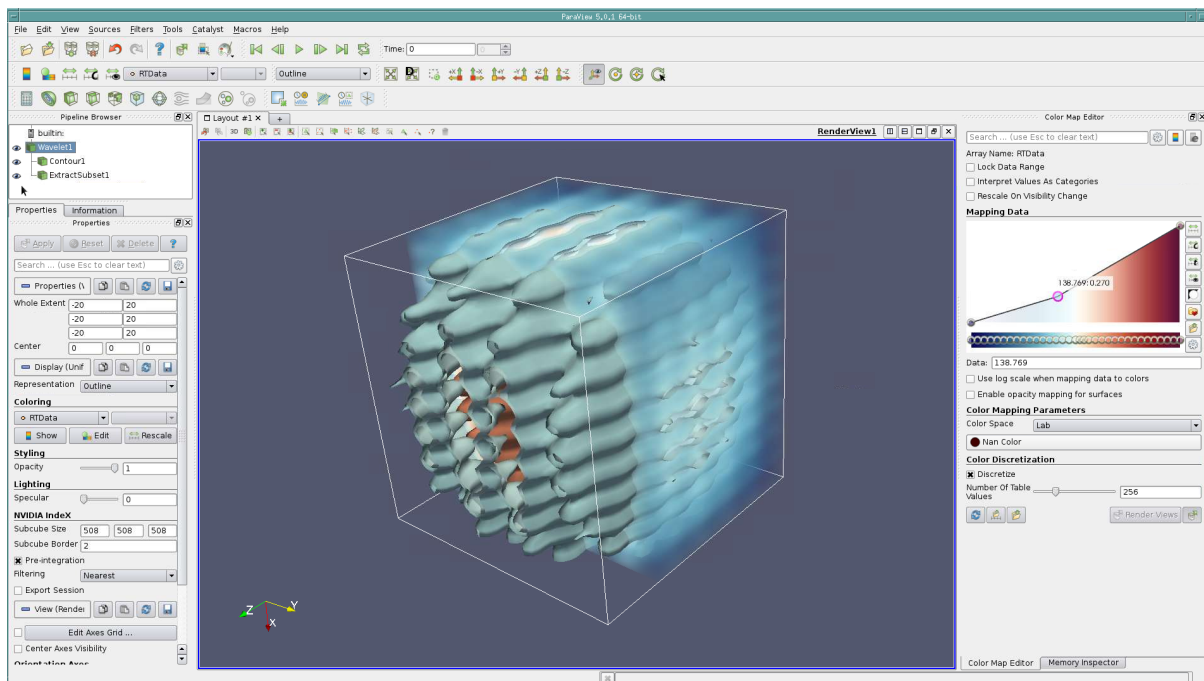


Fig. 5.26 – Wavelet data rendered as a Surface by ParaView and as a volume by NVIDIA IndeX

6 Frequently Asked Questions

Q: Do I need to install CUDA or any other libraries for using the plugin?

A: There is no need to install CUDA separately as the plugin package is bundled with all the required libraries. However, you need to have an appropriate NVIDIA display driver for your graphics card.

Q: When I load the plugin from ParaView's Manage Plugins window, libpvNVIDIAIndex or pvNVIDIAIndex does not show up as loaded.

A: Make sure you have no errors in ParaView's console or on your terminal where you started ParaView from. These error messages will give you additional information about what the issue might be.

Q: Plugin is loaded successfully without any errors but *NVIDIA Index* as a representation does not show up in ParaView's representation dropdown box.

A: Make sure you have loaded a regular-grid volumetric data and it is selected in ParaView's scene graph, ParaView will automatically show representations based on the input data format.

Q: There is an error saying "Failed loading NVIDIA Index library" and viewport is empty.

A: This error message is usually printed when NVIDIA Index library (in `libnvindeX`.) is not found. Make sure you have `libnvindeX`. and `libdice`. in your `LD_LIBRARY_PATH` (or `PATH` on Windows). You can also copy all the libraries from the plugin directly into ParaView's library directories. Refer section-2 for more information.

Q: Viewport is blank when I choose *NVIDIA Index* as a representation.

A: Select appropriate *Scalar Array* for the dataset instead of *Solid Color*.

Q: Viewport is blank when I choose *NVIDIA Index* as a representation with a *Scalar Array* and not with *Solid Color*

A: This is most likely because of an old NVIDIA display driver, update your display drivers to the recommended versions.

Q: Why does my rendering look down-sampled when I interact?

A: NVIDIA Index does not down-sample the data and renders at full resolution. By default ParaView optimizes for high latency networks and enables compression and level of detail, you can disable this from [Edit ► Settings] option and turn off LOD Resolution, Image Reduction Factor and Image Compression.

Q: Can I render multiple volumes at once in the same scene graph in ParaView?

A: While the NVIDIA Index library itself supports multi-volume rendering, the ParaView plugin does not yet have this feature integrated so you can only render one volume at a given time.

Q: Can I use NVIDIA Index library in my own application without ParaView?

A: Sure you can, contact us for more details.

Q: What if I want to have a feature that is part of NVIDIA IndeX but not integrated in the ParaView plugin?

A: Full set of NVIDIA IndeX features are described on this [webpage](#)⁴. If there is a feature that is important for you please contact us, we are happy to take workflow and feature requests.

7 Useful Links

- [NVIDIA IndeX for ParaView plugin website](#)⁵
- [NVIDIA IndeX for ParaView plugin forum](#)⁶
- [NVIDIA IndeX website](#)⁷
- [ParaView binaries and source code download](#)⁸
- [ParaView documentation](#)⁹
- [ParaView user guide](#)¹⁰
- Contact email: indexintegration@nvidia.com

⁴<https://developer.nvidia.com/index>

⁵<http://www.nvidia.com/object/index-paraview-plugin.html>

⁶<https://forum.nvidia-arc.com/forumdisplay.php?210-NVIDIA-IndeX-for-ParaView-Plug-in>

⁷<https://developer.nvidia.com/index>

⁸<http://www.paraview.org/download/>

⁹<http://www.paraview.org/documentation/>

¹⁰<http://www.paraview.org/paraview-guide/>

