



# NVIDIA IndeX

## Demo viewer

17 January 2025  
Version 2.3



---

## **NVIDIA IndeX – Demo viewer**

### **Copyright Information**

© 2023 NVIDIA Corporation. All rights reserved.

Document build number rev384410

---

# Contents

|       |   |    |
|-------|---|----|
| 1     | Introduction  | 1  |
| 2     | Installation and use                                      | 2  |
| 2.1   | Using the demo viewer                                     | 2  |
| 2.1.1 | To run the demo viewer on a single machine                | 2  |
| 2.1.2 | To run the demo viewer on multiple machines(cluster mode) | 2  |
| 2.2   | Building the demo viewer                                  | 2  |
| 2.2.1 | Building the flash user interface                         | 2  |
| 3     | View modes  | 4  |
| 3.1   | Full screen mode  | 4  |
| 3.2   | Edit mode   | 4  |
| 4     | Colormaps   | 6  |
| 4.1   | Colormap editor   | 6  |
| 4.2   | Colormap panel  | 7  |
| 5     | Region of interest  | 9  |
| 6     | Current volume  | 12 |
| 7     | Current horizon   | 13 |
| 8     | Current slice   | 16 |
| 9     | Display options   | 18 |
| 9.1   | Display bounding boxes                                    | 18 |
| 9.2   | Display regions of interest                               | 18 |
| 9.3   | Update view   | 19 |
| 10    | Performance statistics                                    | 21 |
| 11    | Questions and problems                                    | 23 |

---

# 1 Introduction

NVIDIA IndeX demo viewer provides a simple user interface showcasing the features of the NVIDIA IndeX software library.

The purpose of this document is to assist an end user who is new to NVIDIA IndeX and wants to explore the features of the library in a straightforward way. The following sections will give an overview of the demo viewer by explaining the various options it supports, followed by a section that answers a few basic questions.

---

## 2 Installation and use

### 2.1 Using the demo viewer

As a first step, the demo viewer can be used to verify if minimum hardware and software requirements for NVIDIA IndeX are met.

#### 2.1.1 To run the demo viewer on a single machine

- Navigate to the /demo directory and run the `nvindex-viewer.sh` shell script.
- Connect to `http://VIEWER_MACHINE_NAME:8080/` through a web browser to see the flash based user interface of the demo viewer running an artificial dataset with spheres.

#### 2.1.2 To run the demo viewer on multiple machines(cluster mode)

- Navigate to the /demo directory and run the `nvindex-remote-service.sh` shell script on all the machines except one which plays the role of a viewer machine.
- Once all the remote machines are connected and waiting, run the `nvindex-viewer.sh` shell script on the viewer machine.
- Connect to `http://VIEWER_MACHINE_NAME:8080/` through a web browser to see the flash based user interface of the demo viewer running an artificial dataset with spheres.

Should there be a problem loading the video stream, please make sure that both ports 1935 and 8080 are open on the viewer machine.

Scene description and other configuration options are described in the files `scene.prj` and `project.prj` under the /demo directory.

### 2.2 Building the demo viewer

For customer convenience, source code of the demo viewer is also shipped with the library which can be found under /src/demo/ directory accompanied with a Makefile. Running "make" in this directory compiles the demo viewer application. The default demo viewer uses some OpenGL functionality. However, there is another version of demo viewer that doesn't use OpenGL for no OpenGL environment. If the customer wants to build such No-OpenGL variant of the demo viewer, Running "make" with "-f Makefile-no-gl" arguments in this directory.

#### 2.2.1 Building the flash user interface

The source code of the user interface for the demo application is placed under /src/demo/flex/. Since the user interface is flash based it requires Flex SDK which can be downloaded from <URL> . Please note that the Flex SDK might require Java as well.

To build user interface, set the FLEX\_SDK variable in /src/demo/flex/Makefile to the path where the Flex SDK is installed. Running "make" under /src/demo/flex/ will then compile the user interface generating \*.swf files.

---

## 3 View modes

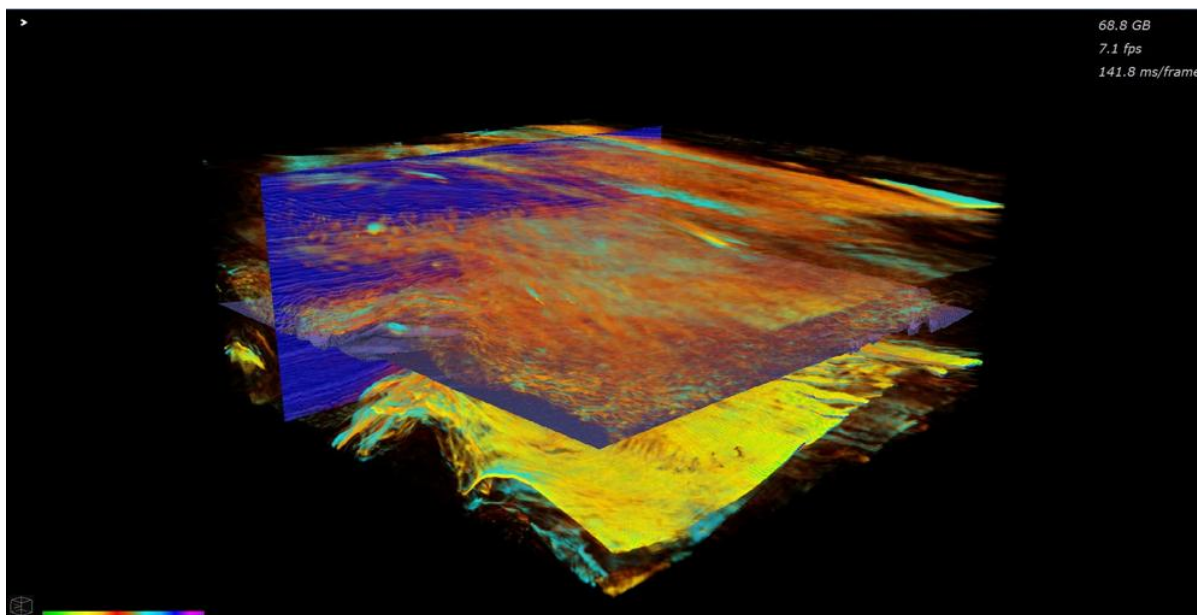
The NVIDIA IndeX demo viewer supports two view modes:

- Full screen mode
- Edit mode

### 3.1 Full screen mode

Full screen mode hides all the user interface options thus making it possible to visualize and interact only with the data. Minimal statistics such as the dataset size, frames per second and time per frame are shown at the top right corner of the screen.

Full screen and edit modes can be toggled by pressing the button at the top-left corner of the screen as seen in Figure 1.



*Fig. 3.1 - Figure 1: Full screen mode of the viewer*

### 3.2 Edit mode

Edit mode is the default mode of the user interface. The following section will describe the options showcasing various features of NVIDIA IndeX library.

Full screen and edit modes can be toggled by pressing the button at the top-left corner of the screen. Options with text in non-italics style are interactive. When the edit mode is toggled, the viewer displays the controls seen in Figure 2.

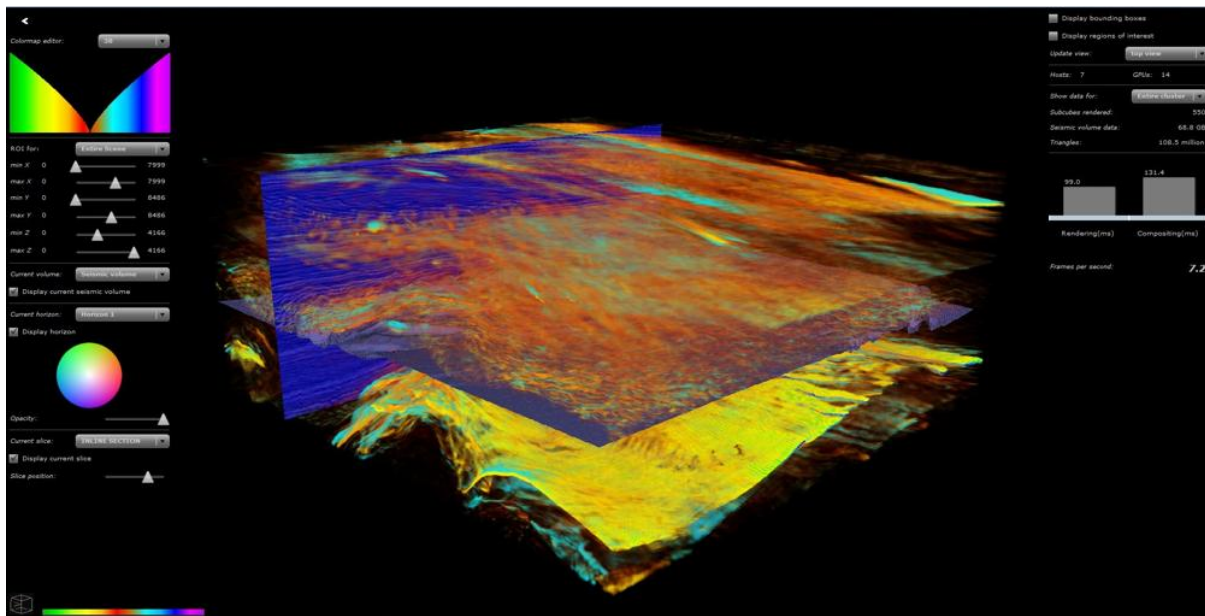


Fig. 3.2 - Figure 2: Edit mode of the viewer



## 4 Colormaps

Data is visualized by a color map that can be selected and modified in the demo viewer.

### 4.1 Colormap editor

The **Colormap editor** option is a drop-down menu which allows the user to select a colormap to visualize the data, as in the upper left corner of Figure 3.

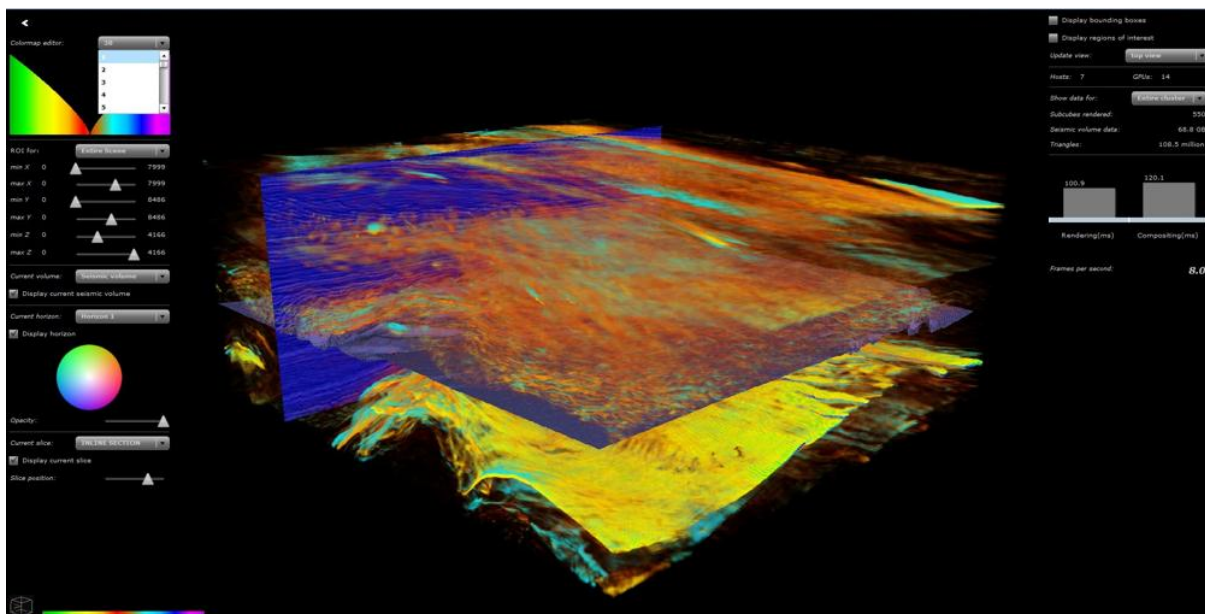


Fig. 4.1 - Figure 3: Selecting a color map

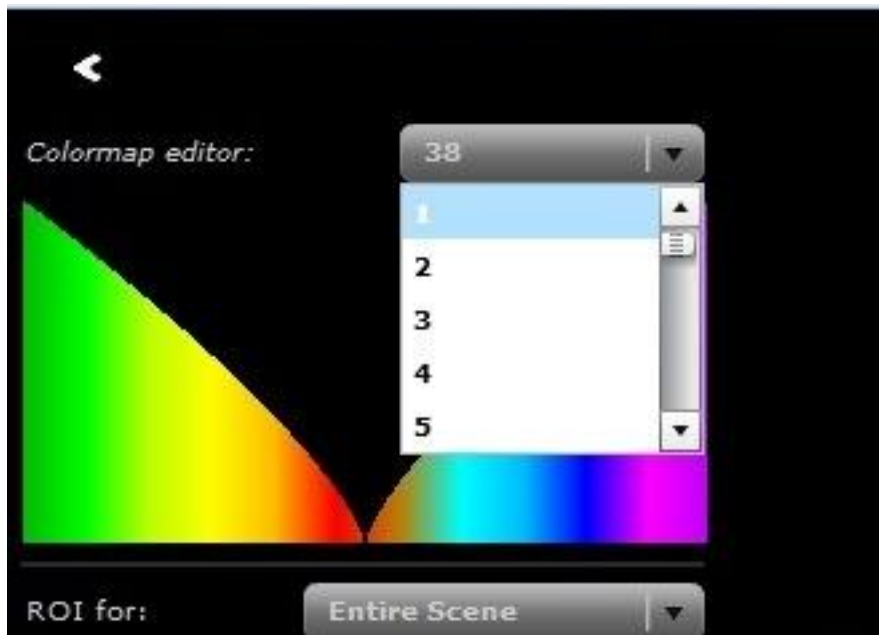


Fig. 4.2 - Figure 3a: The drop-down menu of the colormap editor in Figure 3

## 4.2 Colormap panel

Colormap panel displays the current colormap selected using the colormap editor (Figure 4). The user can press and drag anywhere in the panel to change the transparency values of the dataset being visualized.

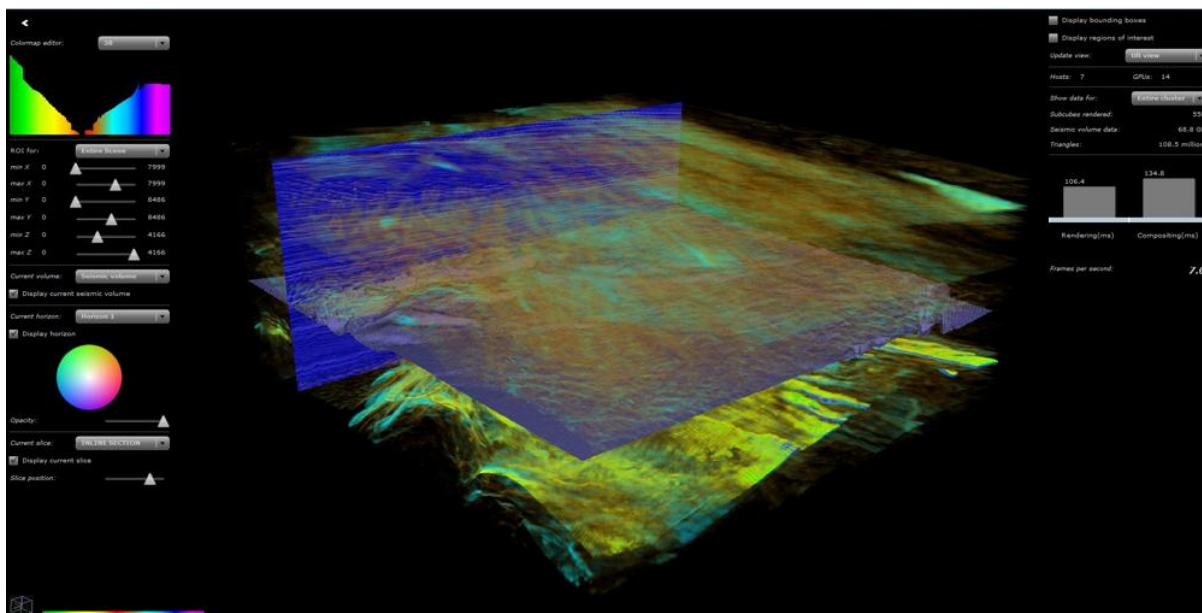


Fig. 4.3 - Figure 4: The colormap being edited in the colormap panel

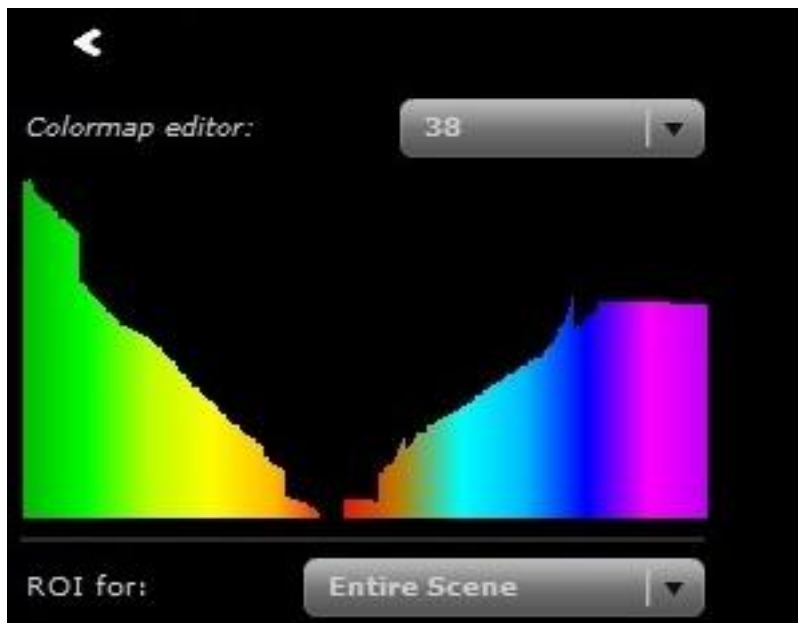


Fig. 4.4 - Figure 4a: The colormap editor in Figure 4

## 5 Region of interest

Region of interest (ROI) section allows the user to visualize only a specific portion of the data by changing all three axes(x, y, z) for both the volume and the horizon. The drop-down menu (Figure 5) enables the user to select either the entire scene or an individual volume or horizon whose region of interest shall be changed.

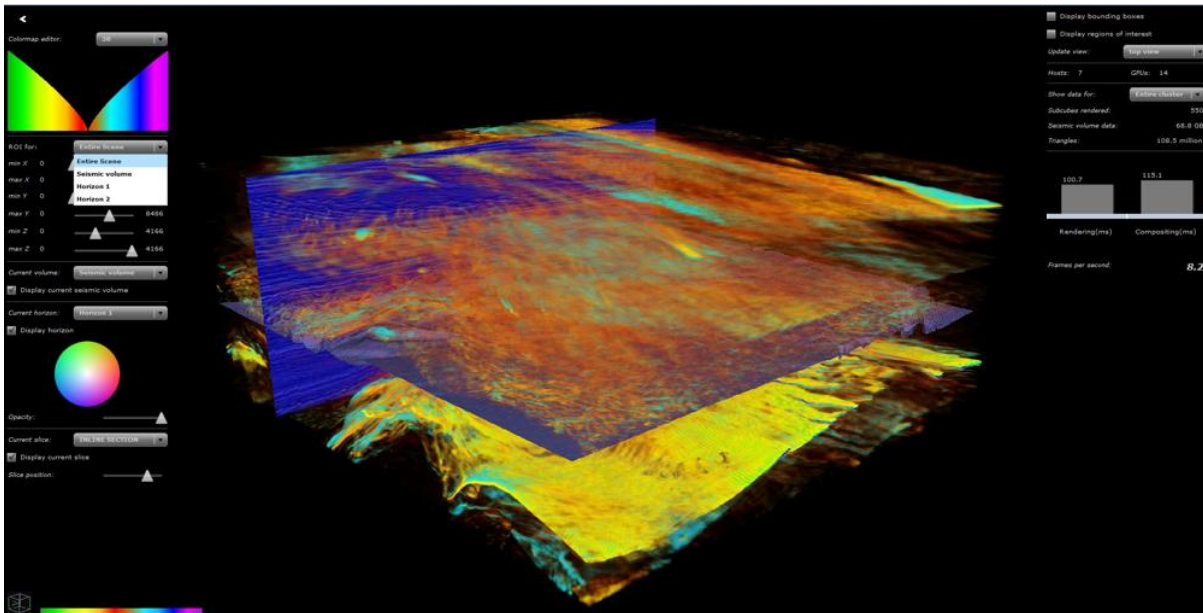


Fig. 5.1 - Figure 5: A drop-down menu showing different options for editing the region of interest



Fig. 5.2 - Figure 5a: Region editing options in Figure 5

Each axis is represented using two sliders, enabling the user to control both the minimum and maximum values for that particular axis. Using these sliders, the user can then select a particular region of interest to visualize. In Figure 6, the slider positions have been adjusted to create a smaller region of interest.

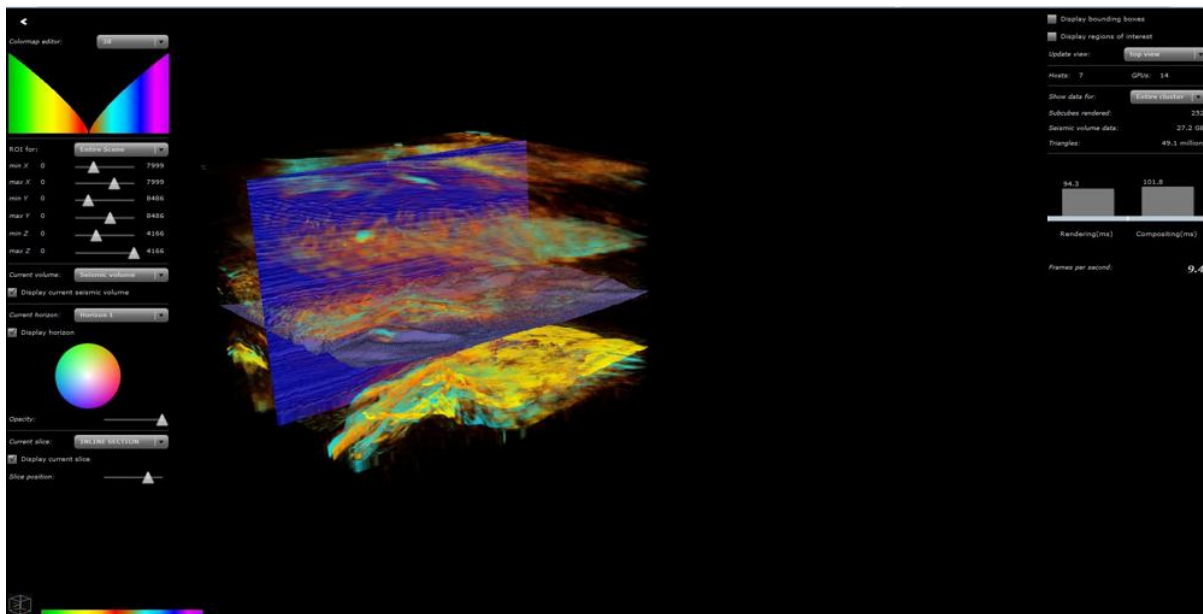


Fig. 5.3 - Figure 6: Changing the region of interest of the volume and horizons in the scene



*Fig. 5.4 - Figure 6a: Region of interest adjustment in Figure 6*

## 6 Current volume

The **Current volume** option is a drop-down menu which allows the user to select a particular volume when there are multiple volumes available.

The check box **Display current seismic volume** (the center of the left menu of Figure 7) allows the user to choose if the volume currently selected from the drop-down menu should be visualized or not.

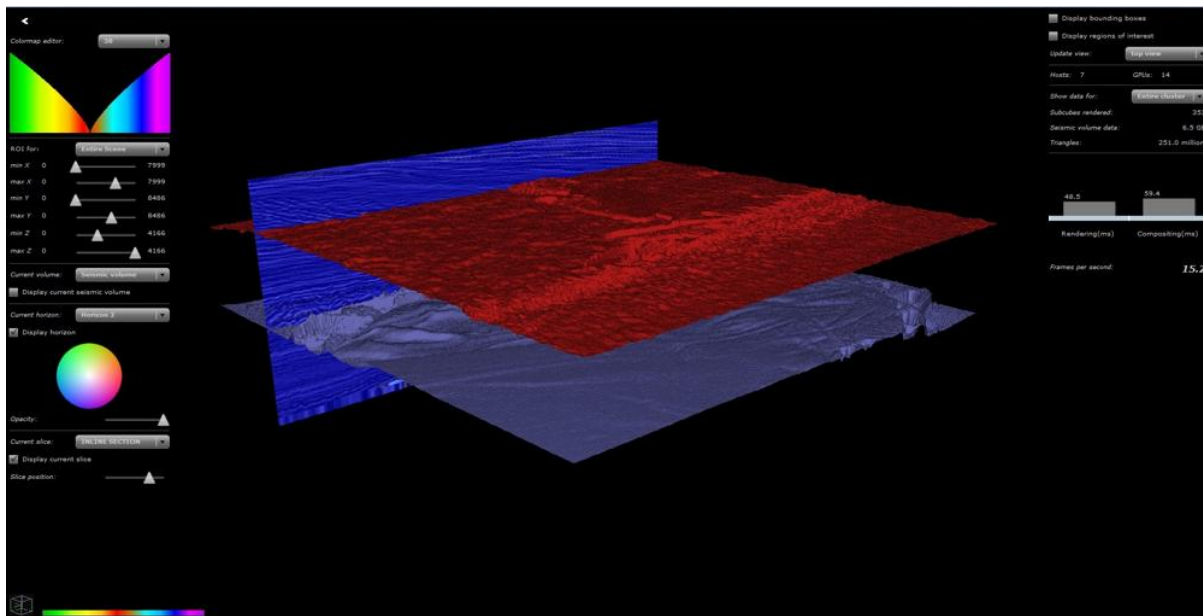


Fig. 6.1 - Figure 7: The **Display current seismic volume** option being unchecked, disabling volume visualization



Fig. 6.2 - Figure 7a: Display of the current seismic volume disabled in Figure 7

## 7 Current horizon

The **Current horizon** option is a drop-down menu (Figure 8) which allows the user to select a particular horizon when there are multiple horizons available.

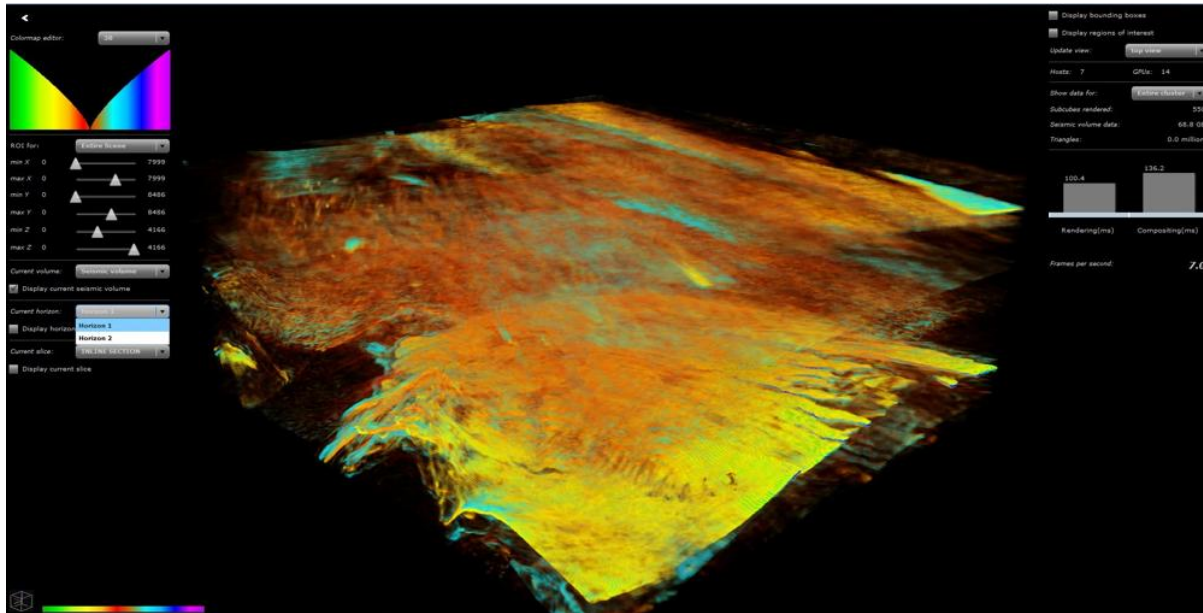


Fig. 7.1 - Figure 8: Selecting a horizon for display

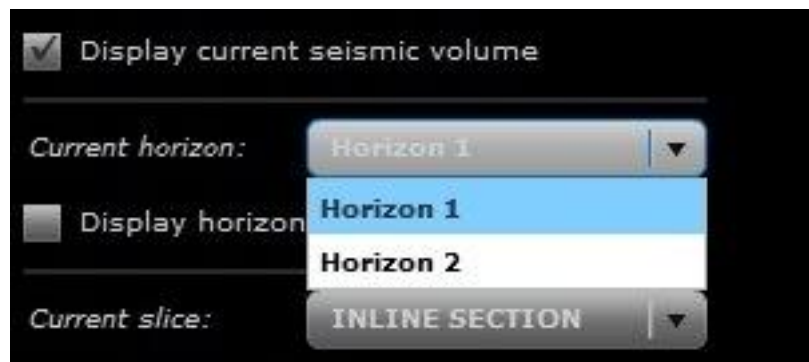


Fig. 7.2 - Figure 8a: The drop-down menu of available horizons

The check box **Display horizon** allows the user to choose if the horizon currently selected from the drop-down menu should be visualized or not.

The colorwheel displayed when the **Display horizon** option is selected allows the user to choose a color for the selected horizon (Figure 9).



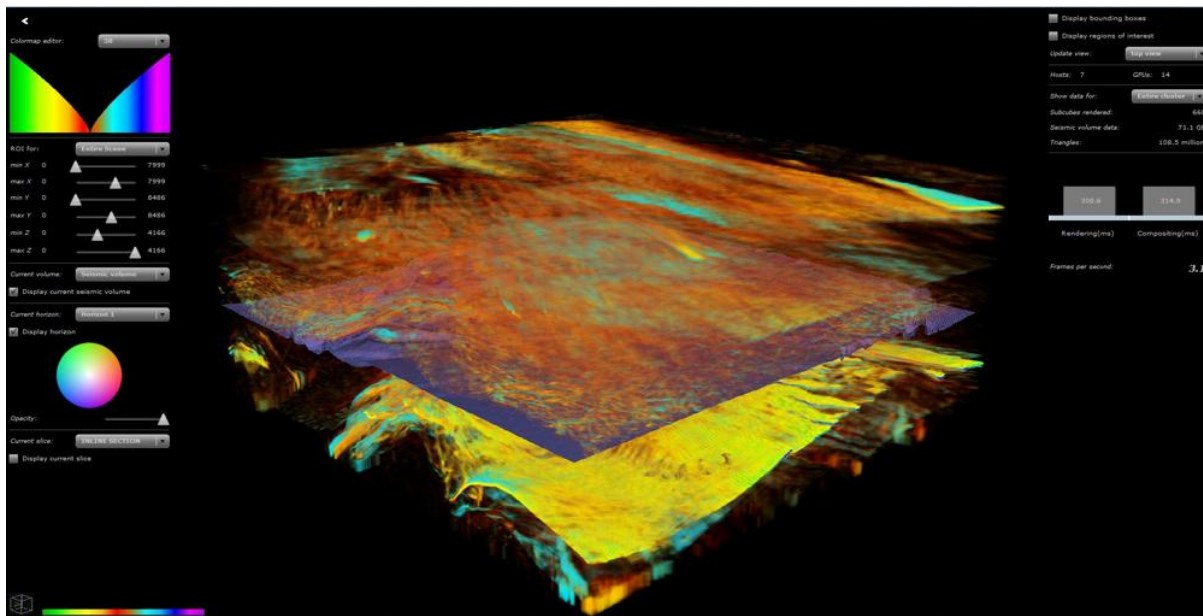


Fig. 7.3 - Figure 9: A blue-colored horizon visualized along with its colorpicker

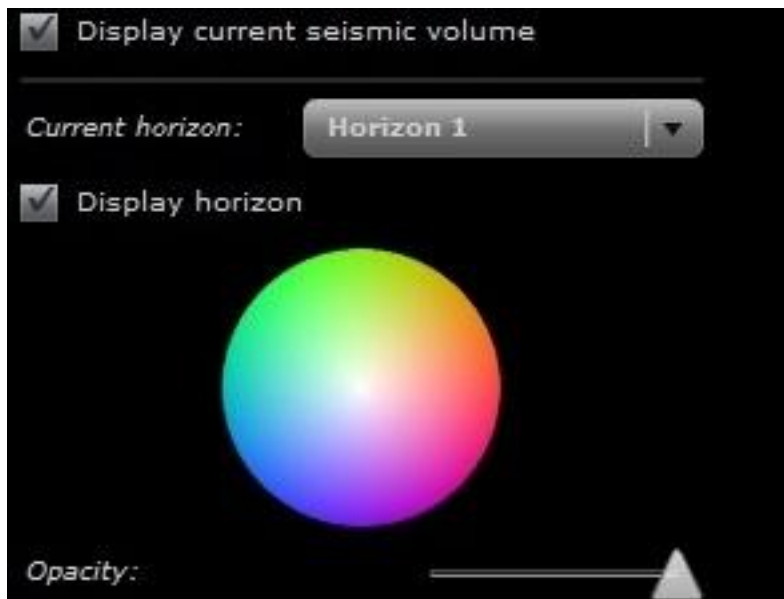


Fig. 7.4 - Figure 9a: Horizon and display color selection in Figure 9

The **Opacity** slider allows the user to control the transparency of the selected horizon (Figure 10).

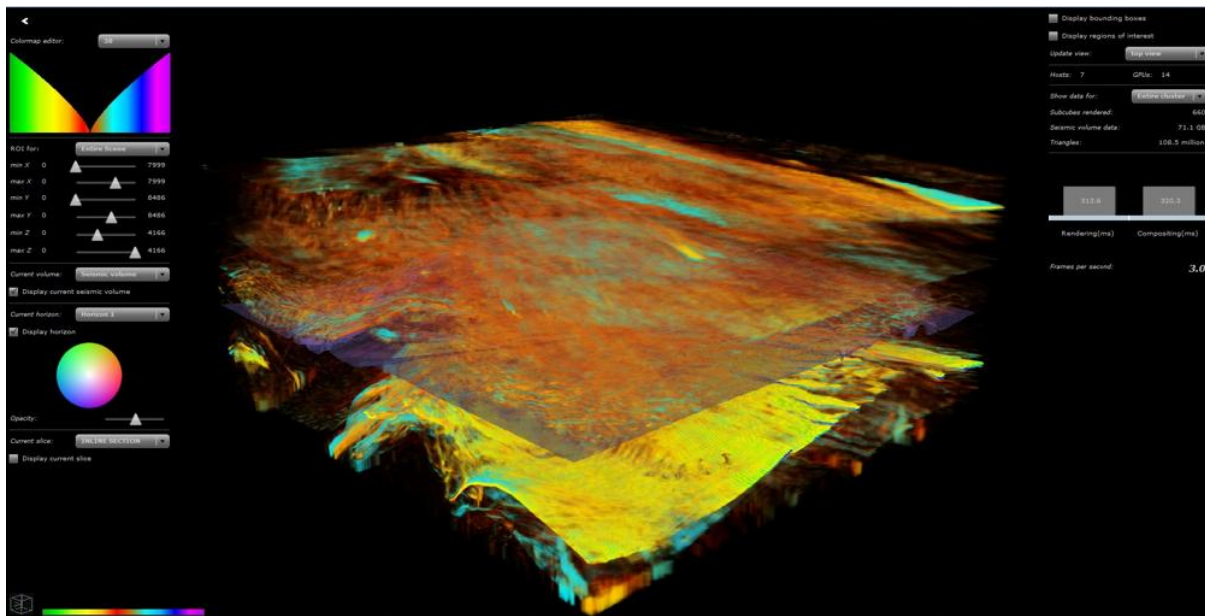


Fig. 7.5 - Figure 10: Changing opacity for the selected horizon



Fig. 7.6 - Figure 10a: The opacity control slider in Figure 10

## 8 Current slice

The **Current slice** option is a drop-down menu (Figure 11) which allows the user to select a slice to visualize.

The check box **Display current slice** allows the user to choose if the slice currently selected from the drop-down menu should be visualized or not.

The **Slice position** slider displayed when **Display current slice** is selected allows the user to move the slice position.

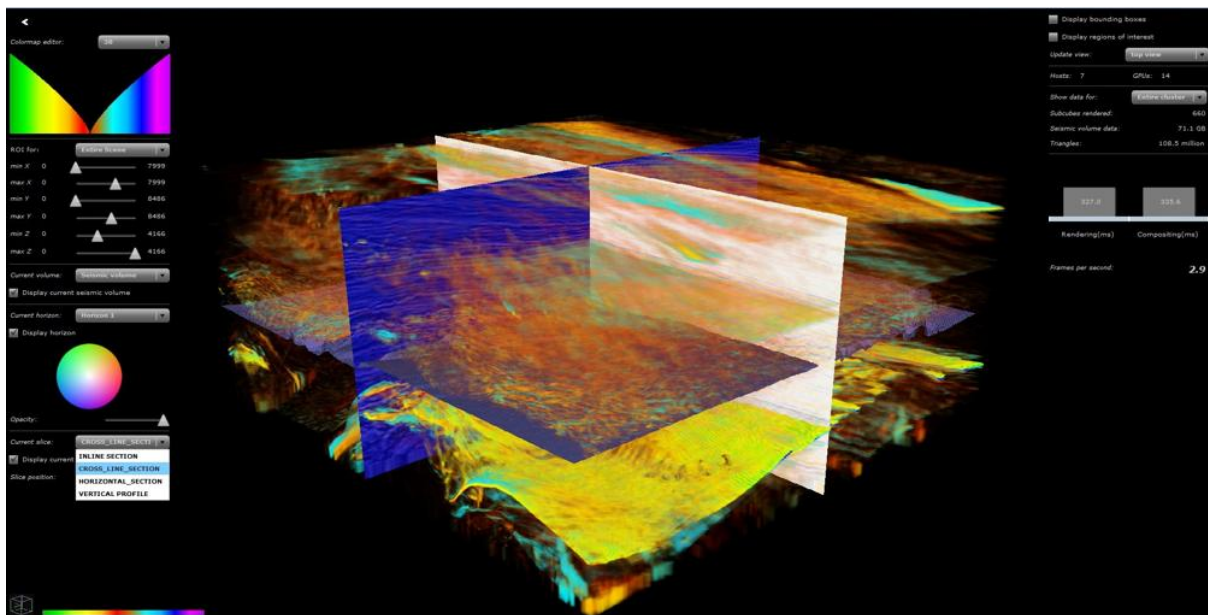
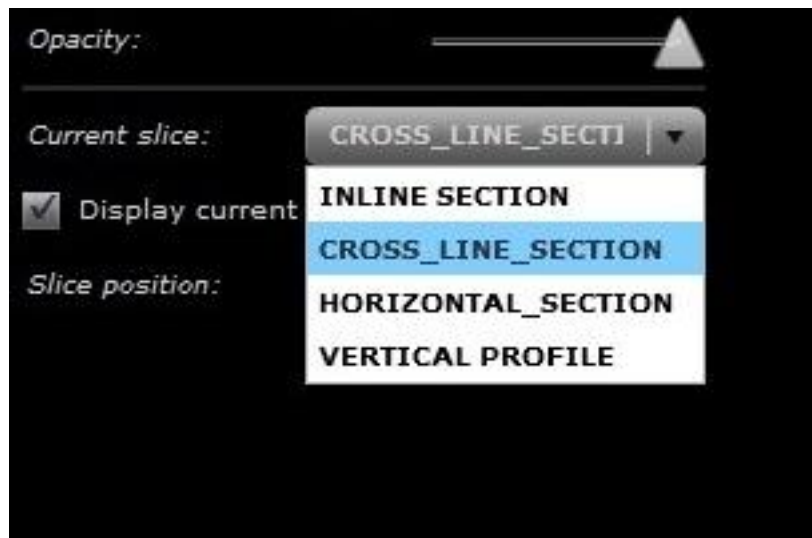


Fig. 8.1 - Figure 11: A drop-down menu showing different slice options available



*Fig. 8.2 - Figure 11a: The slice options drop-down menu in Figure 11*

## 9 Display options

The demo viewer can control the display of bounding boxes and region of interest. The view parameters can also be reset.

### 9.1 Display bounding boxes

When the **Display bounding boxes** checkbox is selected, bounding boxes are drawn that enclose the entire scene, including the volume and horizon (Figure 12).

### 9.2 Display regions of interest

When the **Display regions of interest** checkbox is selected, bounding boxes are drawn that contain the region of interest currently selected by the user (Figure 12).

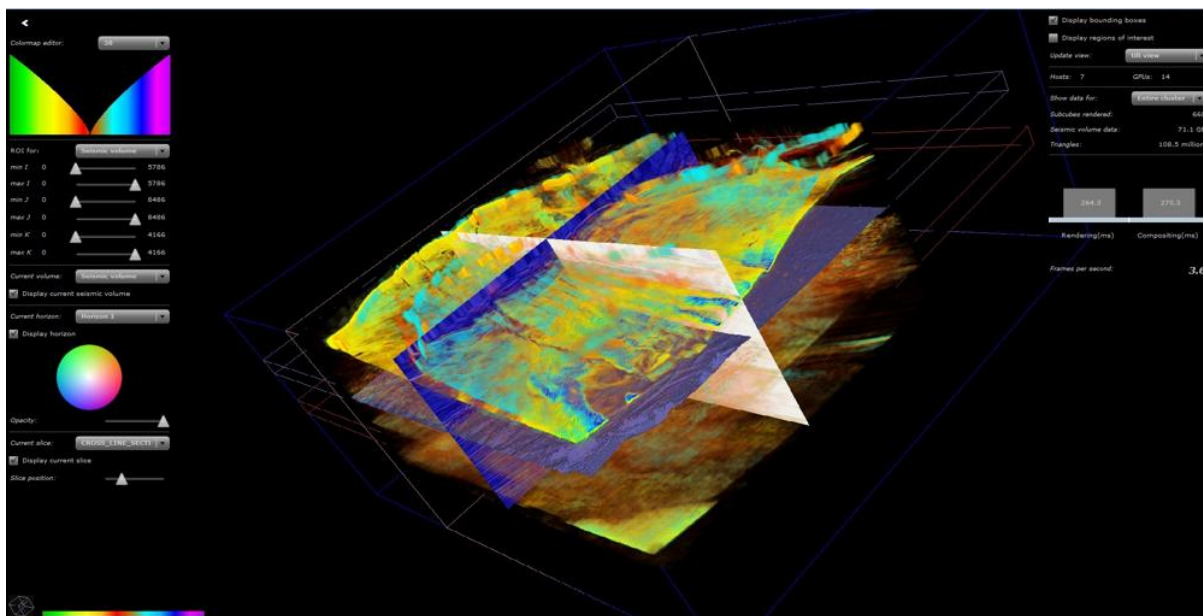


Fig. 9.1 - Figure 12: Bounding boxes drawn in the scene

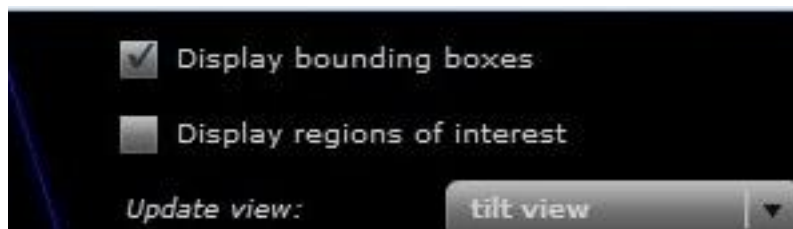


Fig. 9.2 - Figure 12a: Enabling bounding-box drawing in Figure 12

### 9.3 Update view

The **Update view** option is a drop-down menu which allows the user to reset the current view and select a particular view on the dataset. The menu is visible in the upper right corner of Figure 13.

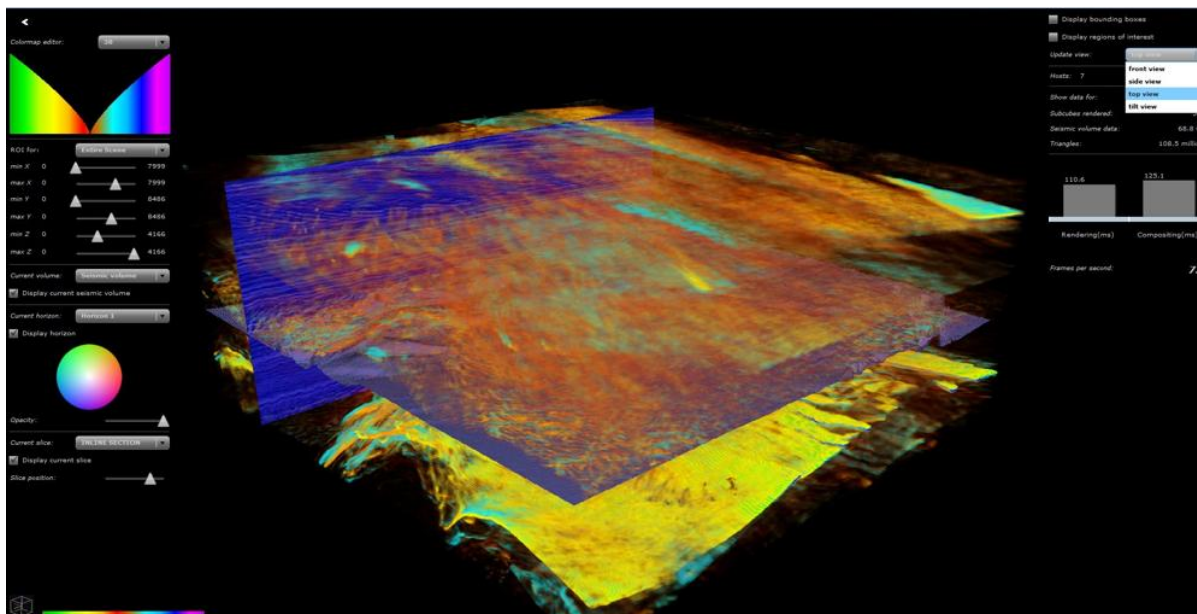


Fig. 9.3 - Figure 13: A drop-down menu showing different options for resetting the view

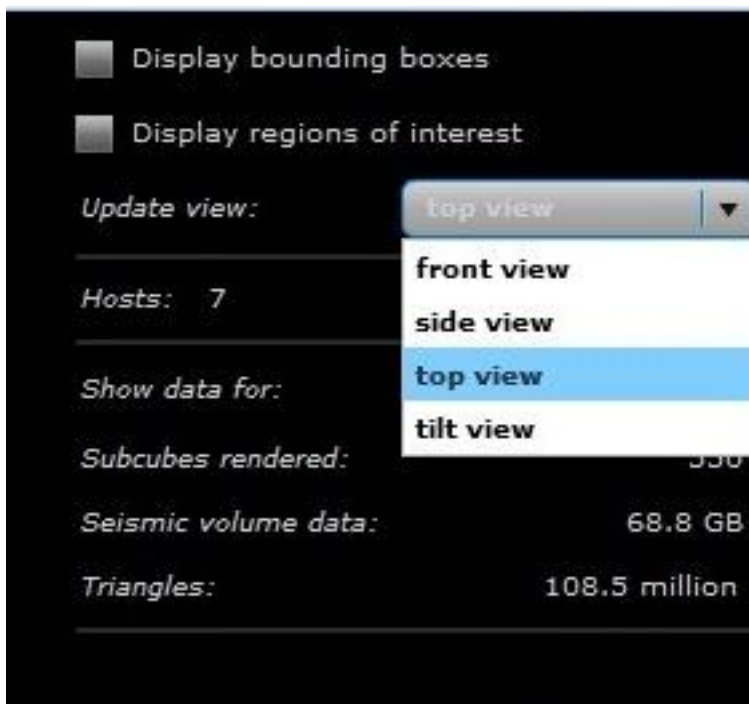
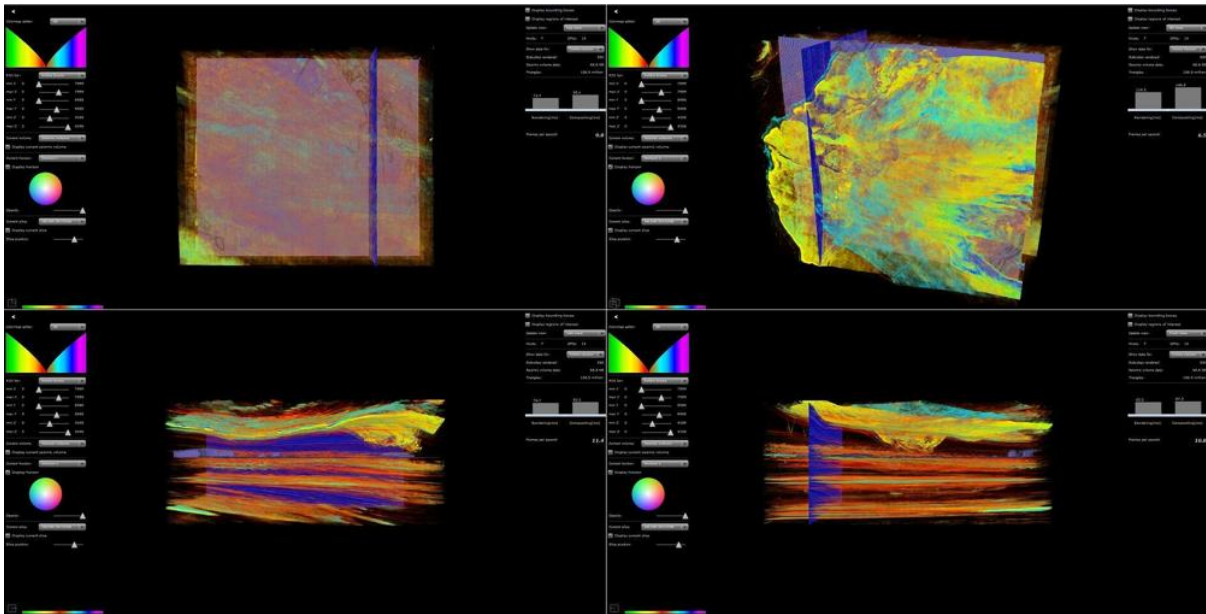


Fig. 9.4 - The view resetting options in Figure 13

Currently, the viewer supports four different view reset options: **front view**, **side view**, **top view**, **tilt view** (Figure 14).



*Fig. 9.5 - Figure 14: Four different options for resetting the view. Clockwise from top left: top view, tilt view, front view, side view*

## 10 Performance statistics

The demo viewer can describe performance statistics with a visual display.

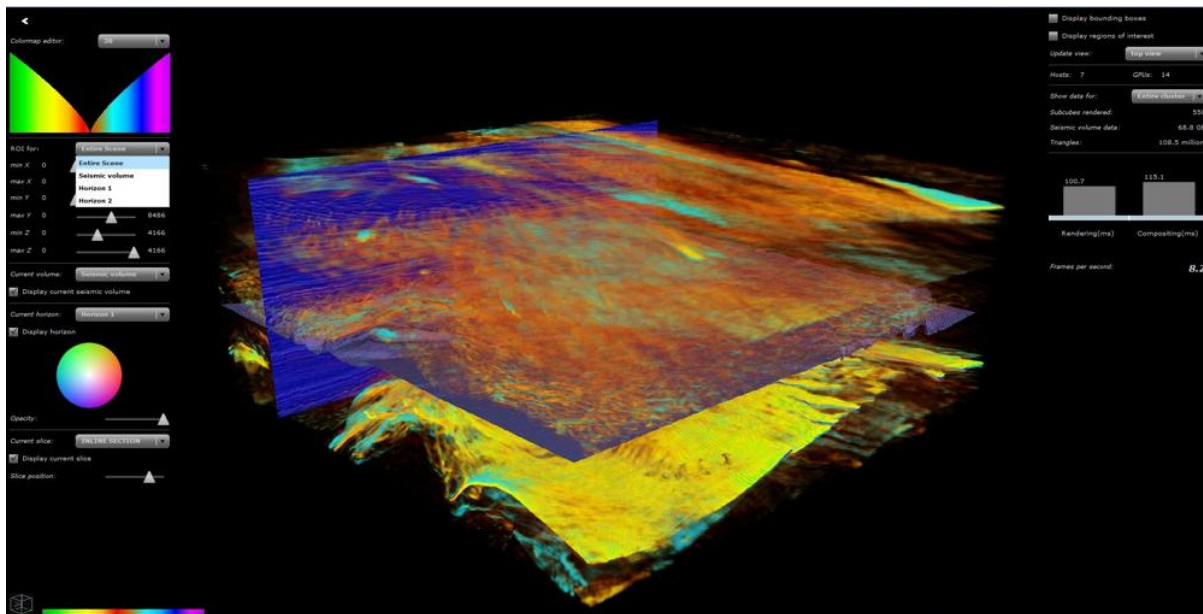


Fig. 10.1 - Figure 15: Performance statistics are displayed on the right

### Show data

The **Show data** option is a drop-down menu which allows the user to select either a particular machine or the entire cluster whose statistics shall be shown. Currently the viewer displays only a minimal subset of the statistics available in the library.

### Performance bars

The two gray colored performance bars show the total rendering and compositing time in milliseconds representing the overall load of the cluster.

### Frames per second

Displays the number of frames currently being rendered.



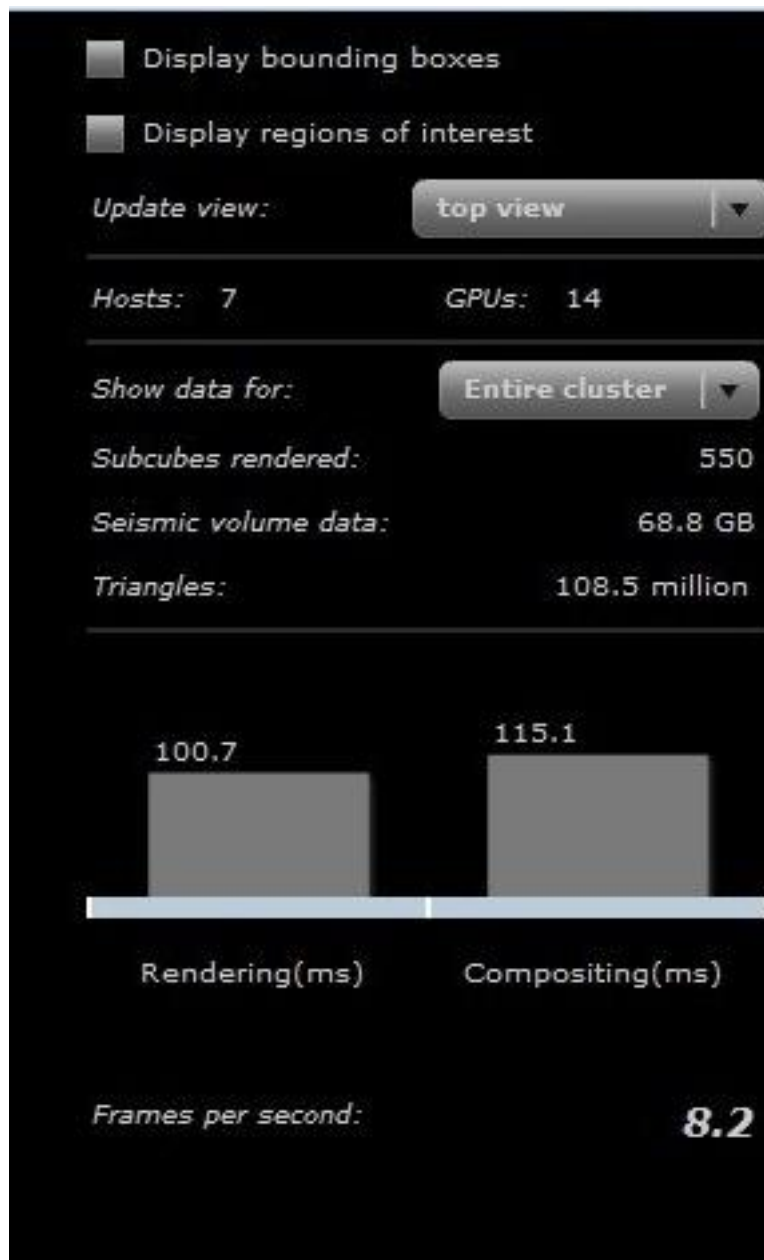


Fig. 10.2 - Figure 15a: Performance statistics in Figure 15

---

## 11 Questions and problems

**Q:** What are the different display devices on which the demo viewer will run?

**A:** Currently the viewer is built for screens ranging from tablets to Quad-HD displays. The viewer will run on any browser and on any Android device.

**Q:** A valid username/password pair is not recognized by the demo viewer.

**A:** Please contact NVIDIA ARC.

**Q:** I received and installed the APK file of the viewer on an Android device. What's next?

**A:** Open the application from the Android menu and you should immediately see the demo if it's running on the cluster.

**Q:** Screen is blank and values are static after I open the viewer application on my Android device.

**A:** Kill the viewer application process and re-install the application again with the same APK file. Wait for a few seconds for the video stream to load.

**Q:** Screen is blank but I still see some values and performance bars responding.

**A:** Press the **Reconnect** button at the bottom-right corner of the screen once or twice and wait for a few seconds for the video stream to load.

**Q:** There is an error with a URL, producing a message that a stream is not found.

**A:** It is very likely that the NVIDIA IndeX library is not running on the cluster or the device has no Internet connection to the cluster running the demo.

**Q:** What ports need to be opened for the video stream?

**A:** Usually firewalls need to enable the HTTP protocol over port 8080 and RTMP video stream over port 1935. If the network uses a proxy, a site exception might be required as well.

On all render nodes the firewall should be disabled:

```
/sbin/chkconfig iptables off
/sbin/service iptables save
/sbin/service iptables stop
/sbin/chkconfig ip6tables off
/sbin/service ip6tables save
/sbin/service ip6tables stop
```

On the viewer node additionally firewalling for the private cluster network should be disabled and to the WAN side the following ports need to be open:

- 7050/tcp to SPM license server on all nodes

- 8080/tcp on the headnode for HTTP
- 1935/tcp on the headnode for RTMP streaming protocol
- 8081/tcp for admin console on all nodes (optional)

**Q:** How is a network interface chosen when there is more than one?

**A:** Multiple network interfaces exist, for example, an Gigabit Ethernet as well as an InfiniBand, and connections fail between cluster nodes.

Go through the following checklist:

1. Use the Infiniband network, not the GbE network.
  - For each cluster node (one cluster may use ib0, others may use ib1, so specify proper Infiniband interface): `route add -net 224.0.0.0 netmask 240.0.0.0 ib0`
  - Verify multicast works over Infiniband using `net_test` and `net_test -m` on cluster nodes. Check that proper Infiniband addresses are reported during multicast test.
2. Edit `project.prj` file.
  - Add line with proper `ipaddr/netmask` (use `sbin/ip addr` to query proper address/netmask): For instance, `dice::network::cluster_interface_address = 172.20.1.0/24:10001`

**Q:** Frames are dropped and the quality is very low.

**A:** It is very likely that the Internet connection is too slow and the video stream bitrate is too high for this connection.