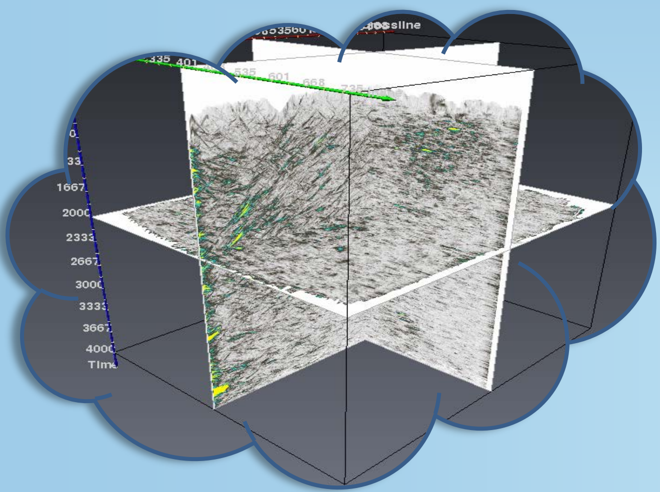
DKRZ
DEUTSCHES
KLIMARECHENZENTRUM



Tutorial

Creating 3D visualizations of seismic data cubes with Avizo Earth 8.1.1

Dela Spickermann
Michael Böttinger
DKRZ

Version 1.3, 20/04/2015

Contact: **Michael Böttinger**

Deutsches Klimarechenzentrum (DKRZ)
Bundesstrasse 45a
20146 Hamburg
Germany

Email: boettinger(at)dkrz.de
http://www.dkrz.de/

The tutorial datasets were kindly provided by:

Khawar Ashfaq Ahmed
Institute of Geophysics
University of Hamburg

The example data were acquired, generated and processed in the scope of the research project:

BMWi, Seismik im Kristallin
0325363C

Contents

1. About Avizo Earth and this tutorial

1.1 Introduction

Avizo is a general purpose 3D visualization and data analysis software. The Avizo Earth edition extends Avizo for applications in the geosciences, specifically for visualizing geophysical data such as volumetric seismic data acquired by the oil and gas industry.

The history of Avizo goes back to the mid-90s, when the Zuse Institute Berlin (ZIB) developed the scientific visualization and data analysis software Amira to aid tackling problems in medicine and biology. Computer tomography data, among others, can be used to create both slices through and 3D representations of the human brain, bones, tissue and tumors, helping physicians and scientists to better understand processes in the human body.
A few years after its initial development, the software was commercialized.Today, it is further developed and distributed by the FEI Visualization Sciences Group under the product names ***Amira*** and ***Avizo***. While ***Amira*** is customized for applications in life sciences, ***Avizo*** comes in various editions suitable for earth science, physical and material science data.

One of Avizo Earth's main features, which we will use in this tutorial, is its ability to generate expressive and insightful visualizations of 3D seismic data cubes using moving slices, camera rotations and many more. By carefully designing colormaps and using state-of-the-art 3D rendering techniques, geoscientists will be able to recognize the 3D spatial structures in their data cubes that may be masked by noise if one looks at 2D data slices only.

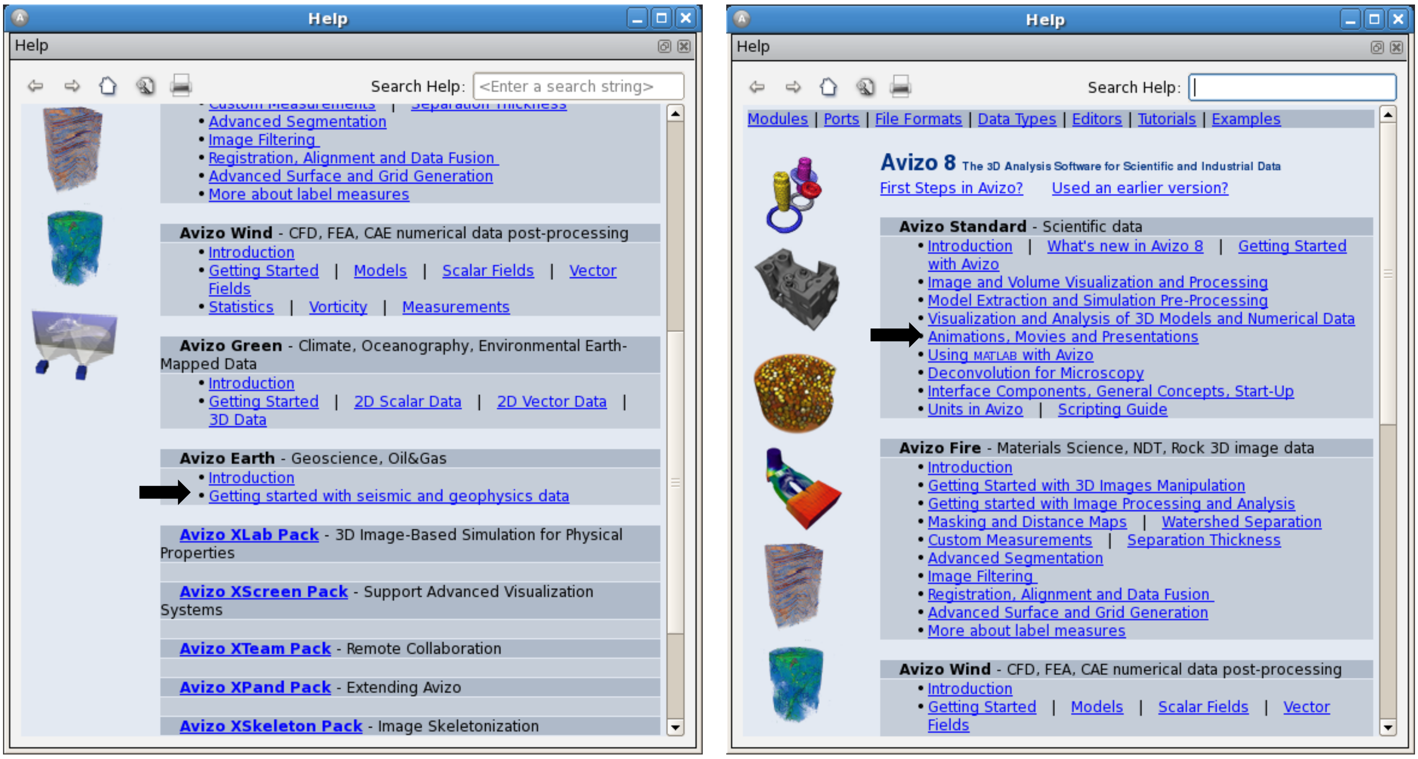
1.2 Motivation and scope of this tutorial document

The focus of this tutorial is on **creating 3D visualizations of 3D seismic data**. By using Avizo Earth on powerful workstations equipped with up-to-date 3D graphics cards, the visualizations can e.g. interactively be rotated on the screen in a way that the user can easily recognize the spatial structures in the seismic data. With respect to communicating such visualizations to a broader audience, we recommend to use animations to achieve the same degree of recognition even without interaction.

We assume that you are roughly familiar with the Avizo interface including the Animation Producer. Ideally, you have familiarized yourself with the Avizo Earth user's guide chapter ***Getting Started with Avizo Earth Edition – Visualizing Seismic and Geophysics Data***. The chapter allows you to test basic features of Avizo Earth using the small and free tutorial dataset ***BoonsVilleData.zip***.

Animations, however, are not covered. Thus we give a short introduction to the Animation Producer in chapter 9 of this tutorial document. For additional information see the ***Avizo Standard – Animations, Movies and Presentations*** section in the user's guide menu.

In this tutorial you will learn to create animations that go beyond simple 360° rotations – we will explain how to define camera paths and how to mark a specific (x,y) point with a moving sphere to keep track of it while a time slice animation is running. Volume rendering and transparent slices are also covered.



Avizo Help / User's Guide

1.3 Prerequisites and tutorial data

1.3.1 Access to DKRZ's visualization server halo

In order to use Avizo Earth at DKRZ you need:

- an account for the *halo* visualization server at DKRZ
- the TurboVNC Software

Information on how to get those and on making a reservation and starting a visualization session on a *halo* machine can be found on the DKRZ website:

http://www.dkrz.de/Nutzerportal-en/doku/vis/account

http://www.dkrz.de/Nutzerportal-en/doku/vis/reservation

1.3.2 Running Avizo Earth at DKRZ

As a DKRZ – or, more specifically - a *halo* user you will be assigned your own subdirectory on /work named /work/<yourProject>/<yourUserID> where you can save your Avizo projects, data etc.

To start a new session with Avizo Earth, first make a reservation and start a session by using the the halo reservation system https://halo.dkrz.de. The system will give you a one-time password. Start the TurboVNC Viewer, connect to the halo node for which you made your reservation and enter the one-time password. Once you see the virtual desktop of your halo machine, open a console window and navigate to your personal directory on /work.

To run Avizo Earth 8.1.1., simply type ***earth*** in the console and hit enter.

1.3.3 Tutorial data

All required files such as the tutorial data files ***Coherence_BMU0325363C.lda***[1], ***Cohmig_BMU0325363C.lda***, map ***Schneeberg_OSM1.jpg*** extracted from http://openstreetmap.org, colormaps ***coherence_col_WIT.am***, ***Cohmig_VolRen_orange.am*** and the completed example project files ***Tutorial.hx*** and ***Tutorial2.hx*** are available on DKRZ's *halo* system in the folder:

/work/kv0653/Tutorial_AvizoEarth

The data contained in ***Coherence_BMU0325363C.lda, which was extracted from the respective SEG-Y file,*** contains the coherence of seismic data traces instead of pure seismic amplitude data. ***Cohmig_BMU0325363C.lda*** contains migrated coherence. The data was computed using a 3D-CRS code and was kindly provided to DKRZ by Mr. Khawar Ashfaq Ahmed, Institute of Geophysics, University of Hamburg in the working group of Professor Dirk Gajewski.

The seismic field data was acquired in the scope of project ***BMWi, Seismik im Kristallin 0325363C*** in a crystalline hard rock environment in Saxony, Eastern Germany. Project partners are the Leibniz Institute for Applied Geophysics (LIAG) in Hannover, the University of Hamburg and the TU Freiberg. Field data acquisition was carried out by company DMT. More information (German only) can be found on the official project website:

http://www.liag-hannover.de/fsp/ge/seismik-im-kristallin-sachsen-siks.html

The project was publicly funded, thus you are free to use this tutorial data for educational purposes when stating the project name, project number and Mr. Khawar Ashfaq Ahmed as a reference.

---

[1] LDA is a technology and file format by FEI which allows interactively visualizing files larger than the system memory.

Seismic data from a hard rock environment poses a challenge regarding data processing because of steeply dipping faults and fractures causing diffractions. With respect to this application area we found coherence data to be more suitable than amplitude data for creating a meaningful tutorial.

2. SEG-Y data

For the sake of disk space and preprocessing time, the tutorial data is provided in the form of Avizo-generated “.lda” files with accompanying binary “.dat” files instead of the original SEG-Y data. These LDA files and their corresponding “.dat”-files are together by 10 to 20% smaller compared to the original SEG-Y files. LDA and their respective binary files are multi-resolution files, containing subsampled versions of the SEG-Y data. In the Avizo Earth Viewer, the data is either displayed as a full down sampled or as a locally refined view. In the latter case, the additionally required data is loaded from disk on demand.

Throughout this tutorial we will use LDA data, but if you prefer to work with your own data, it is important to have basic information on the SEG-Y format required by Avizo and on the data import using the ***SEGY Wizard***.

Importing ***Coherence_BMU03325363C.sgy*** and ***Cohmig_BMU03325363C.sgy*** was straightforward using the ***SEG-Y Wizard*** that is included in Avizo Earth. All header information was extracted automatically and default settings could be left as they were. However, in case you try to load your own SEG-Y file, you might face difficulties: We found that the SEG-Y wizard works fine for SEG-Y files created e.g. with Petrel software according to SEG-Y format revision rev 1 (as per year 2002), while it failed for data created with the Seismic Un*x software package developed by the Colorado School of Mines, which is based on SEG-Y revision rev 0.

Details on the SEG-Y rev1 specification can be found here:

http://www.seg.org/documents/10161/77915/seg_y_rev1.pdf

In contrast to the previous revision additional ***textual file headers*** have been introduced with revision 1. SEG-Y files according to the rev 1 standard thus consist of a 3200-byte textual and 400-byte binary file header, up to N 3200-byte extended textual file headers (optional) and M 240-byte data trace headers and M data traces respectively.

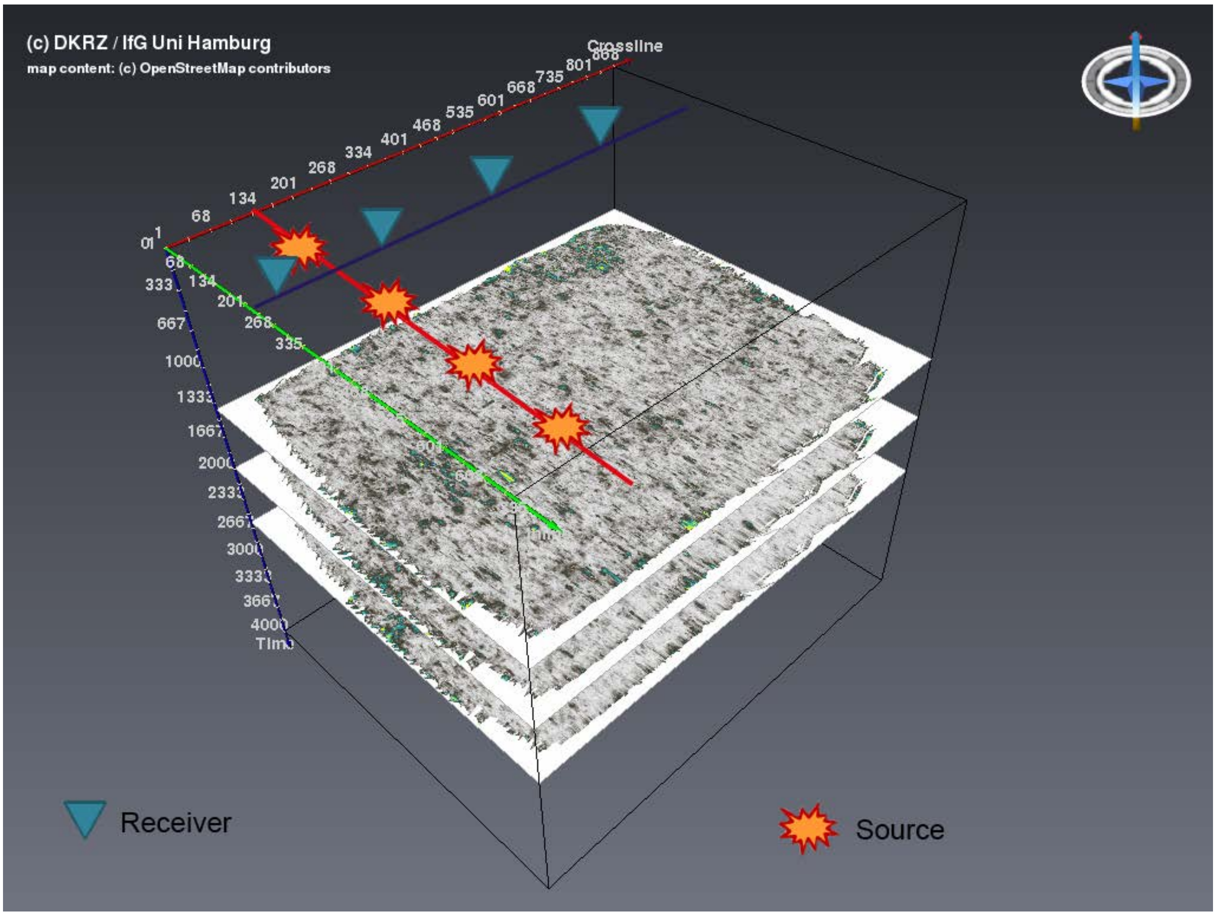
Seismic Un*x does not create this machine-readable extended textual file header and we assume that this header contains important information to extract data and geometry in Avizo Earth.

3. 3D data geometry

3D seismic data is stored in terms of ***inline***, ***crossline*** (or ***xline***) and ***time,*** and Avizo reads and displays the data accordingly. The ***inline*** direction is the direction in which the data have been acquired. This is depicted with a blue line and 4 symbolic receivers in the following figure. In our case, this receiver line stands for a land streamer with equidistant receiver groups. These receiver groups record the reflected and diffracted energy successively released into the subsurface by the seismic sources (here: 3 vibrator trucks). The other receiver lines not shown here are similarly parallel to the green inline axis.
The ***crossline*** direction is perpendicular to the inline direction in the horizontal plane. For the data acquisition of project BMWi 0325363C, the crossline direction corresponds with the vibrator source lines.
The z direction of the 3D data cube is actually time in milliseconds. Thus time slices, as shown in the figure, correspond to coherence amplitudes at all receivers for this specific recording time. In this exemplary figure time slices at 1400 ms, 2000 ms and 2600 ms are shown.

An example of marine 3D seismic data acquisition geometry can be found at:

http://www.glossary.oilfield.slb.com/en/Terms.aspx?LookIn=term%20name&filter=crossline



4. Getting Started with Avizo Earth

If you have successfully started your TurboVNC session and Avizo Earth, you are ready to start visualizing our seismic example data.

This tutorial consists of two parts based on the separate datasets ***Coherence_BMU0325363C.lda*** and ***Cohmig_BMU0325363C.lda***. For each of the datasets we will use distinct visualization techniques and colormaps – you should therefore create two separate Avizo projects, e.g. ***Tutorial1.hx*** and ***Tutorial2.hx***. For now we use ***Coherence_BMU0325363C.lda*** to build the first tutorial project.

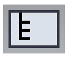
Prior to loading this first dataset we will briefly look at the Avizo GUI elements and adjust a few settings.

4.1 GUI panels overview

Avizo's graphical user interface (GUI) includes several panels, which can be selected and unselected via buttons in the menu bar:

As the highlighted buttons show, the ***Main Panel***, the ***Properties*** panel, the ***Console*** panel and the ***Animation Producer*** panel are active. The panels can be undocked and reorganized or resized by clicking and dragging with the mouse. However, as there is no ***reset*** button and as reinserting the panels requires some skill you may want to keep the default layout and only resize the panels. Your user interface could look similar to the picture on the following page.

A large part of the screen is taken up by the ***Viewer*** displaying the ***Bounding Box***, ***Seismic Axes*** and data, if selected. To the left of the viewer we have the ***Main Panel*** showing the ***Project View***. The aggregation of interconnected colored modules that you see is the ***Project Graph View***, with which we will work. To switch to the ***Project Tree View***, click on the far right symbol in the ***Main Panel's*** menu bar.

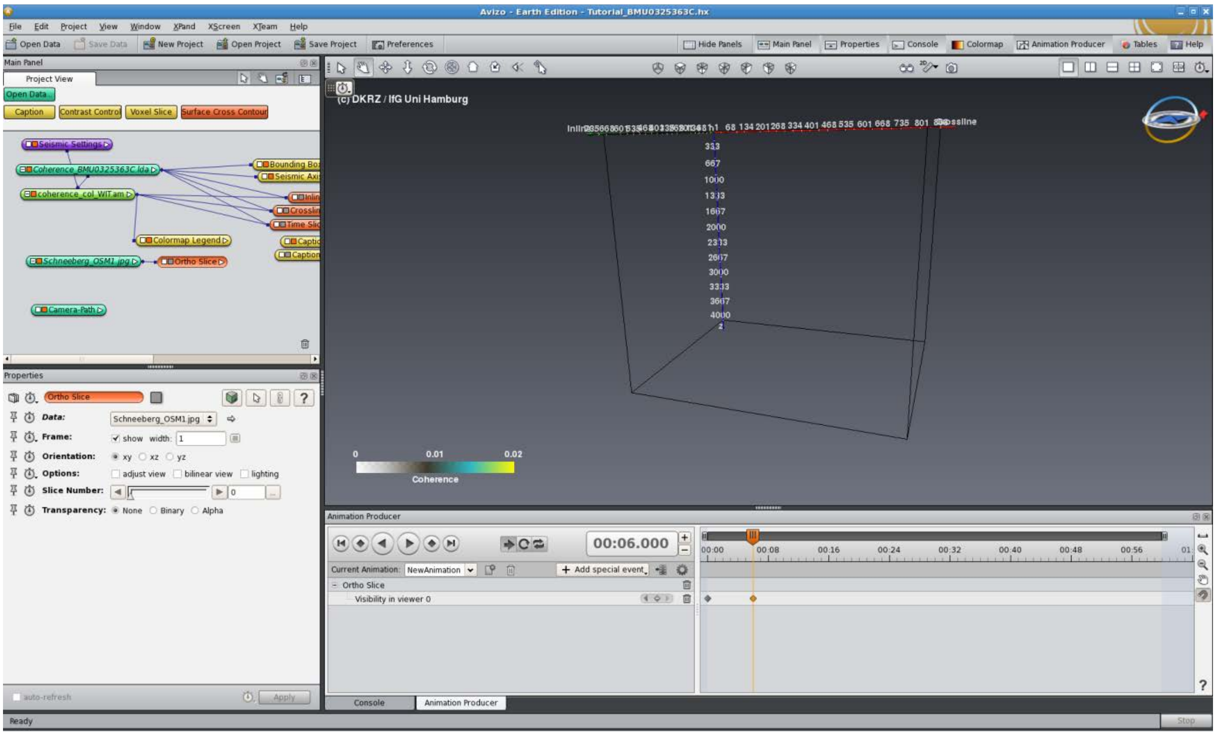


Depending on the modules in the ***Project View***, Avizo suggests new modules that you may want to add. These are displayed as colored buttons in the top section of the ***Project View***. In an empty project containing only the violet ***Seismic Settings*** module ***Open Data…*** is suggested as a green button. However, these suggestions are never a complete list of all available options and you may want to add other modules not listed here.

Individual modules can be selected by clicking with the left mouse button when the white arrow (***Project View Select Mode***) in the ***Main Panel's*** menu bar is active. Simultaneously press the ***Shift*** button to select multiple modules. Note how the color shading of the module is reversed once a module is selected. Selected modules in the ***Project View*** have a darker hue at the top than at the bottom.
For simultaneous vertical and lateral shift of all models, select the hand symbol (***Project View Pan Mode***).

The ***Properties*** panel below the ***Main Panel*** displays the properties of the module that has been selected in the ***Main Panel***. In this case it is the ***Ortho Slice*** module. If several modules are selected at the same time, they will be listed one below the other in the ***Properties*** panel.

Below the viewer, the ***Animation Producer*** is displayed. At its bottom one finds two tabs for switching between the ***Animation Producer*** and ***Console*** display. We will need the ***Console*** later on to query and set camera positions and to set the viewer size. The ***Colormap*** module and ***Table*** module are presently irrelevant for us.



Before we start with the initial settings for your Avizo Earth project, let us briefly look at the ".hx " project files, in which your Avizo Earth visualization sessions are saved.

4.2 ".hx" project files & TCL language

The GUI may not be giving it away but Avizo Earth is in fact a script-based software. All the adjustments you make in your project, if it is creating and connecting modules or changing settings, are translated into a human-readable script file. That is the .hx project file, and the script is updated whenever you choose File ➡ Save Project. Caution: changes and adjustments since your last project save cannot be undone! You can only return to the project state that was last saved.

The scripting language of the .hx project file and also of the Avizo console is TCL. For a short introduction and for links to further information, enter ***TCL*** into the search field of the Avizo Help. TCL commands can also be entered into the Animation Producer, as shown in chapter 11 on volume rendering and transparent slices.

However, unless you are very experienced with both TCL and Avizo scripting, we do not recommend making adjustments in the project file itself. Changing the path to data, colormaps etc. when reusing a copy of an old project is straightforward though. The following figure shows an exemplary beginning of an .hx project file:

Tutorial_BMU0325363C.hx

```
# Avizo Project
# AvizoEarth
# Generated by Avizo 8.0.1
remove -all
remove "grayScale.am" "Depth_Rainbow.am" "coherence_col_WIT.am" "Coherence_BM
"SphereOct.surf" "Seismic Settings" "WellsSettings" "WellTopsSettings" "Bound
"Caption 3" "Caption 4" "Display Time" "Caption 5" "Display Time 2" "Display

# Create viewers
viewer setVertical 0

viewer 0 setBackgroundMode 1
viewer 0 setBackgroundColor 0.145098 0.152941 0.176471
viewer 0 setBackgroundColor2 0.435294 0.45098 0.498039
viewer 0 setTransparencyType 5
viewer 0 setAutoRedraw 0
viewer 0 show
mainWindow show

set hideNewModules 1
[ load ${AMIRA_ROOT}/data/colormaps/grayScale.am ] setLabel "grayScale.am"
"grayScale.am" setIconPosition 0 0
"grayScale.am" setNoRemoveAll 1
"grayScale.am" setVar "CustomHelp" {HxColormap256}
"grayScale.am" master disconnect
"grayScale.am" Datafield disconnect
"grayScale.am" fire
"grayScale.am" setMinMax 0 255
"grayScale.am" flags setValue 1
"grayScale.am" shift setMinMax -1 1
"grayScale.am" shift setButtons 0
```

.

4.3 Initial settings for animating slices

In Avizo Earth there is usually a tradeoff between the rendering quality of data volumes and slices. As the first and larger part of this tutorial focuses on animating slices, we start by optimizing the rendering of slices.

To do this, click on the ***Preferences*** button 

in the toolbar just above the main panel and viewer window and navigate to the tab ***LDA***. In the line for setting the ***Loading Priority***, drag the slider to the left side to choose ***High Slice/Low Volume***. Click ***Apply*** and leave with ***OK***. This setting will remain even if you quit and restart Avizo Earth during your current TurboVNC session.

Even though your final video file should only take up a few hundred MB of disk space, you may want to increase the ***Video Memory Amount*** here to a value well above 1000 MB: The graphic card’s video memory is then used to store a portion of the seismic data in order to allow for its interactive 3D rendering. On the DKRZ’s visualization server *halo* you will currently have at least 2.5 GB graphics memory available.

For your convenience you may check or uncheck some options in the ***Layout*** tab of the ***Preferences*** menu. For now uncheck ***Show viewer in top-level window*** to see the viewer next to the project view and properties window as shown in the previous figure. You may change this later on for watching or exporting your whole animation. In case you cannot see the compass in your viewer you will find ***Viewer gadgets*** and the ***Compass*** tab in the right section of the ***Preferences*** window.

In the ***Project View*** of the main panel, click on the ***Seismic Settings*** module to see its properties. Check that ***Coordinate System: Crossline/Inline/Time*** is selected and set the ***Time Scale Factor***, which stretches or shrinks the time axis, to value **3**. The result is, compared to the default settings, a vertical exaggeration by the factor 3.

For projects like the first part of this tutorial, where you have a convenient colormap ready to load, it is best to load this colormap prior to the actual data. Via ***Open Data,*** select ***coherence_col_WIT.am*** and see the new light green colormap module appear in the ***Project View***.

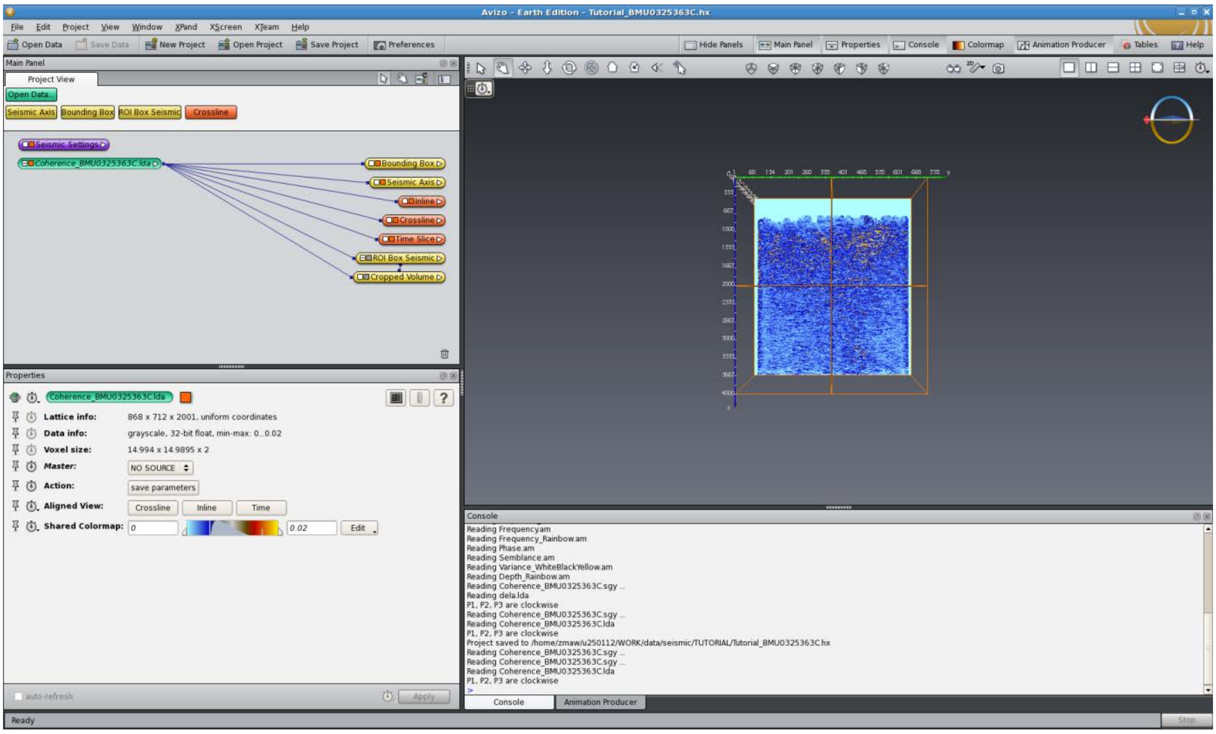
To set it as the default colormap for the data and for later display modules, click on the white square in the violet ***Seismic Settings*** module and select ***DefaultSeismicColormap***. Note that a dark blue ‘rubber band’ has appeared between the ***Seismic Settings*** module and the tip of your mouse pointer. Drag it onto the colormap module until the rubber band turns light blue and click to connect the modules. In their respective ***Properties*** both modules now show the colormap ***coherence_col_WIT.am*** with a lower and upper limit of 0 and 0.02.

Subsequently loaded data modules will automatically be connected to this colormap.

5. Data loading

Avizo Earth treats data loaded from SEG-Y files differently from LDA data in that it automatically connects a number of display modules to the dark green data module:

Bounding Box, Seismic Axis, Inline, Crossline, Time Slice, ROI Box Seismic and ***Cropped Volume*** as shown in the following figure.



When loading LDA files, as we do in this tutorial, we need to connect the desired modules manually. This is described in the following section ***Load LDA file***. Please read the section on loading SEG-Y files as well because it contains valuable information on the headers and coordinates of the tutorial data.

5.1 Load LDA file

If you have loaded and connected ***coherence_col_WIT.am*** to the ***Seismic Settings*** module, the next step is loading ***Coherence_BMU0325363C.lda***. You may do this via the suggested ***Open Data…*** button in the ***Project View*** or via the toolbar just above the panels. A dark green data module of the same name has appeared and is connected to the colormap. Clicking on the small white square of this module you will see that ***coherence_col_WIT.am*** is connected as a ***SharedColormap***.

In the ***Properties*** of this data module you find additional information: The data consists of 868 x 712 x 2001 values given on uniform coordinates, which is 868 crossline entries, 712 inline entries and 2001 time samples. The sampling interval is

2 ms, as will be shown in the next section on loading SEG-Y files. Minimum and maximum values of the data are 0 and 0.02.

To use modules that are not suggested as colored buttons by Avizo, right-click on the dark green data module in the ***Project View***:

Right-click *on* ***LDA module*** → Display → Bounding Box → Create
→ Seismic Axis → Create
→ Inline → Create
→ Crossline → Create
→ Time Slice → Create

For a snapshot of what your viewer should look like with these modules connected, go to ***chapter 6***.

5.2 Load SEG-Y file

We advise you to choose this option only when working with your own SEG-Y data <u>for the first time</u>. If you already have an LDA file, take the steps described in the previous subsection.

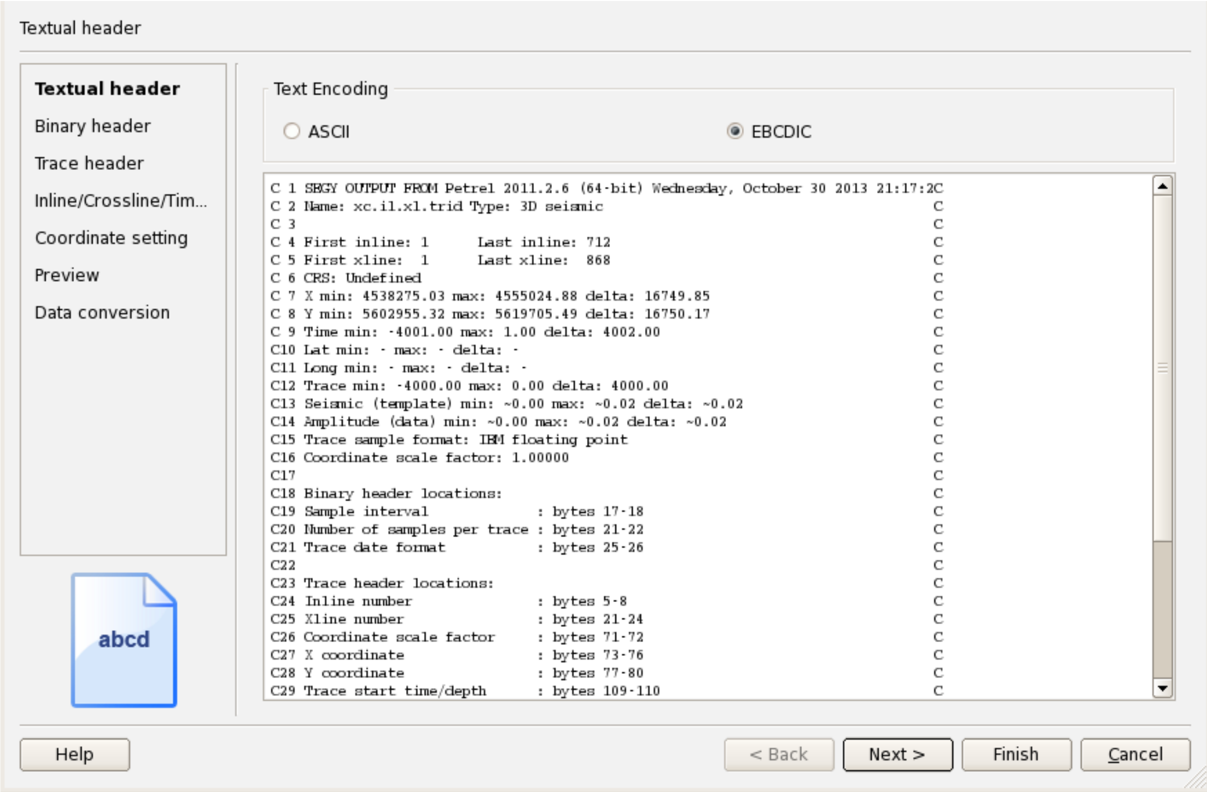
Starting the ***SEGY Wizard*** immediately creates the respective LDA file for your SEG-Y data as <your filename>.lda. If <your filename>.lda already exists, it will be instantly and irreversibly overwritten! This may not be too tragic but you will lose precious time by once more waiting for the SEG-Y to LDA conversion. On a halo system, the conversion of 4.8 GB from SEG-Y to LDA takes about 15 minutes - plenty of time for a tea or coffee break!

To open a SEG-Y data file click on ***Open Data*** in the Avizo Earth menu bar and navigate to the folder containing the data file and select it for opening. When the ***SEGY Data Import*** window pops up, choose the ***3D*** option and start the ***SEGY Wizard***. It will convert the SEG-Y input to a somewhat smaller data file in the LDA format, which is then directly loaded by Avizo Earth whenever you work on your project. Additionally, a binary .dat file with hierarchical multiresolution data will be created, from which Avizo loads higher resolution data on demand - that is, when zooming in on a portion of the data. If you would like to start a second project with the same data, you only need to open the existing .lda file and connect your desired modules.

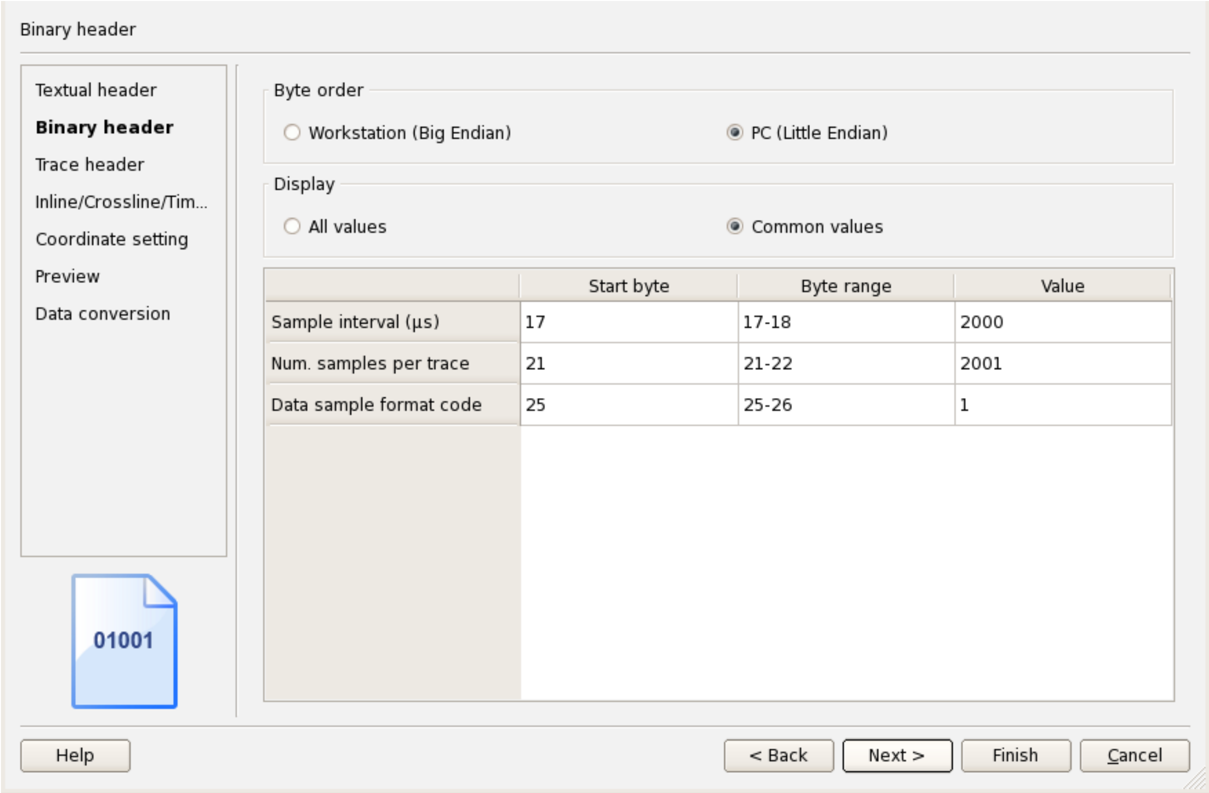
The textual header as well as the binary and trace header information displayed in the SEGY Wizard provide us with useful information on the geometry and sampling of the data and with its geographical referencing. The following pictures show a selection of SEGY Wizard screens for ***Coherence_BMU0325363C.sgy.*** If the data is provided in the correct SEG-Y revision format, one moves from screen to screen by simply clicking the ***Next*** button. The actual data conversion will start after one has pressed the ***Finish*** button.

The red bar at the bottom of the Avizo screen allows you to keep track of the conversion progress.

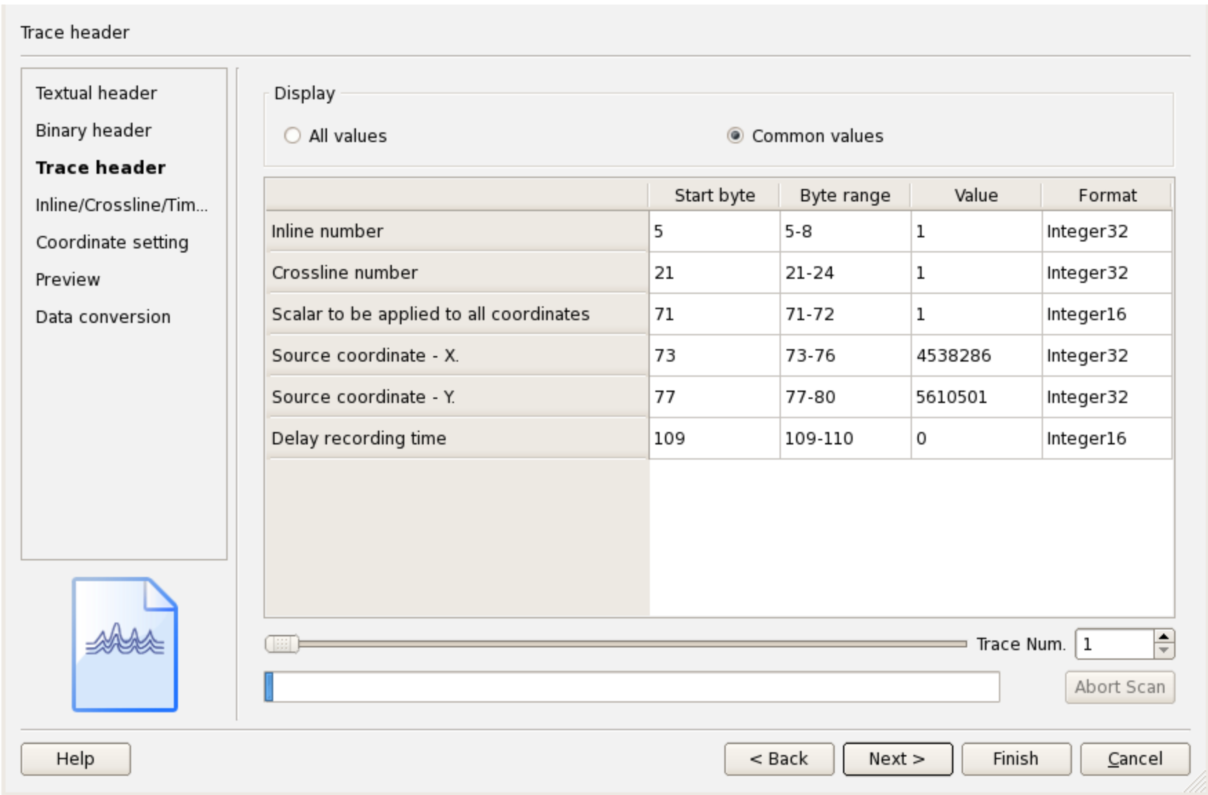
5.2.1 Header and geometry information



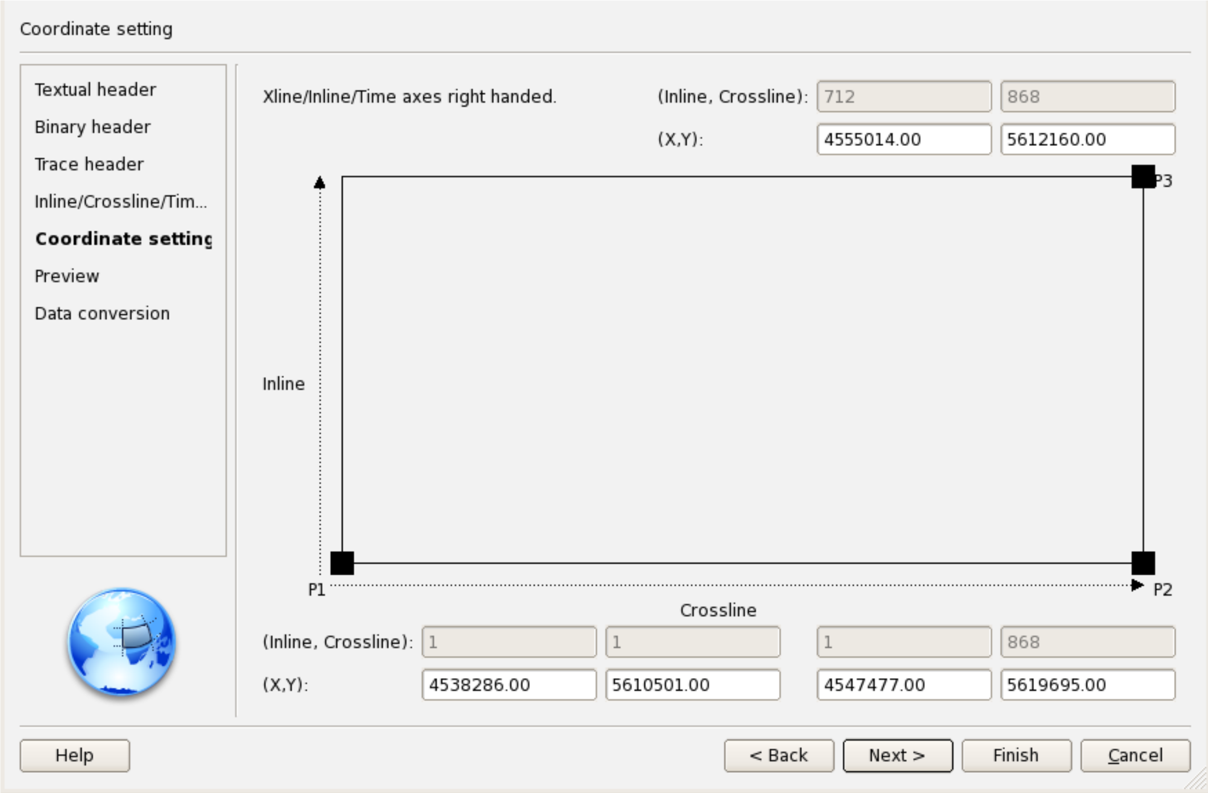
We can see when (October 30, 2013) and using which software (Petrel 2011.2.6 64 Bit) the SEG-Y data file ***Coherence_BMU0325363C.sgy*** was created and that it is comprised of 712 inlines and 868 crosslines. Furthermore, the headers contain important information such as sampling interval, the number of samples per trace, line numbers, coordinates etc. Avizo Earth automatically displays the correct bytes for the header entries on the following screens.



The temporal resolution of the data is 2001 samples with a sampling interval of 2 milliseconds. Thus every trace is 4 seconds long.



By dragging the *Trace Number* slider below the table in the ***Trace Header*** section of the SEGY Wizard, one can for different trace numbers observe how the respective inline and crossline numbers and their associated geographical coordinates change.



The ***Coordinate settings*** show three out of four corner points of the survey area with their respective Gauss-Krueger coordinates.

5.2.2 Gauss-Krueger coordinates

The Gauss-Krueger coordinate system is a transverse Mercator map projection usually used in Europe to map smaller areas in metric coordinates while preserving angles. It is based on the Bessel or the Krassowski ellipsoid and divides the area to be mapped into Meridian stripes of 3° extending from the north to the south pole, parallel to a central meridian. For more details, see

http://de.wikipedia.org/wiki/Gau%C3%9F-Kr%C3%BCger-Koordinatensystem

The website of the German state of Saxony provides a good graphic comparison of Gauss-Krueger coordinates ('GK', marked in black) and the globally used ***Universal Transverse Mercator (UTM)*** projection ('UTM', marked in light blue):

http://www.landesvermessung.sachsen.de/inhalt/etrs/grund/gross/gk-utm.html

In the Gauss-Krueger system, the x coordinate, denoted here by ‘H’, which stands for ‘Hochwert’ or ‘Northing’ is the distance in m along the central meridian of the 3° map stripe under consideration with value 0 at the equator (coordinate origin) and positive values to the north. The y coordinate, referred to as ‘R’ for ‘Rechtswert’ or ‘Easting’, consists of the ***zone identification number*** (here: zone ***4*** for central meridian at 12° of eastern longitude) and the (distorted) distance from the central meridian along the y axis in m. To avoid negative Easting values for points located west of the central meridian, a value of 500000 m is added.

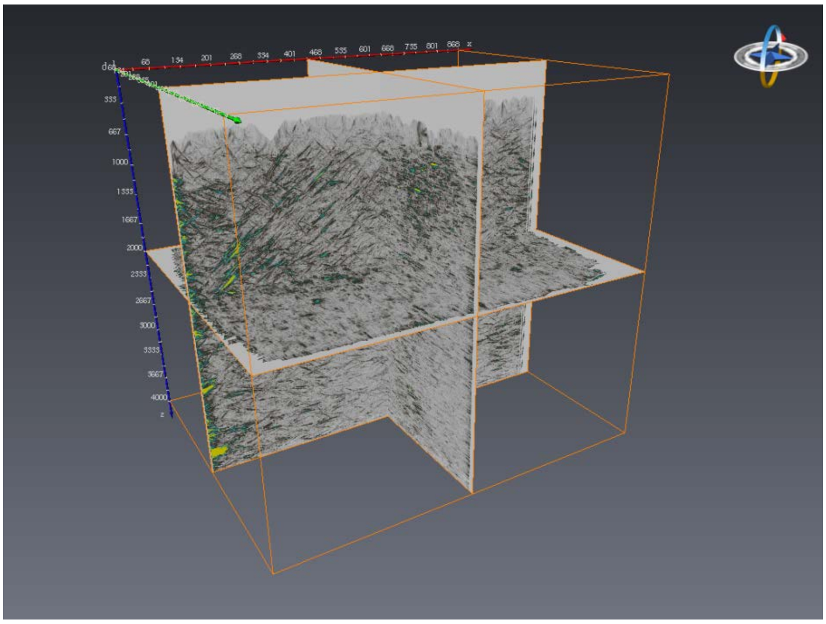
Feel free to convert the coordinates displayed in the ***Coordinate settings*** view to lat-lon-coordinates and to use free mapping tools to create your own map of the survey area.

To complete the ***SEGY Wizard*** and start the data conversion, press ***Finish*** when you have reached the final ***Data conversion*** screen.

Once the SEG-Y to LDA data conversion has finished, a green module representing the LDA data file is visible in the ***Project View*** panel of Avizo. Furthermore, Avizo should have created and connected a ***Bounding Box*** and ***Seismic Axis*** module as well as an ***Inline***, ***Crossline*** and ***Time Slice,*** which are all active ( ), and grayed out modules ( ) ***ROI Box Seismic*** and an attached ***Cropped Volume***. You may delete those latter two, as we will not use them in the first part of this tutorial.

6. Getting the overview and basic settings

The following snapshot shows the viewer with ***Inline***, ***Crossline*** and ***Time Slice*** as you should be seeing them on your computer screen now.



Before we start with the (partly tedious) work of positioning a map on top of the 3D data cube and setting the camera positions, let us have a look at the data and adjust the settings to improve the visualizations of inline, crossline and time slices.

6.1 Customize axes and bounding box

Our aim is to adjust the ***Seismic Axes*** and ***Bounding Box*** in a way that we can easily identify the inline and the crossline axis and that we are able to read the axis annotations. We also want to make sure that the bounding box interferes as little as possible with the view of the slices. We therefore

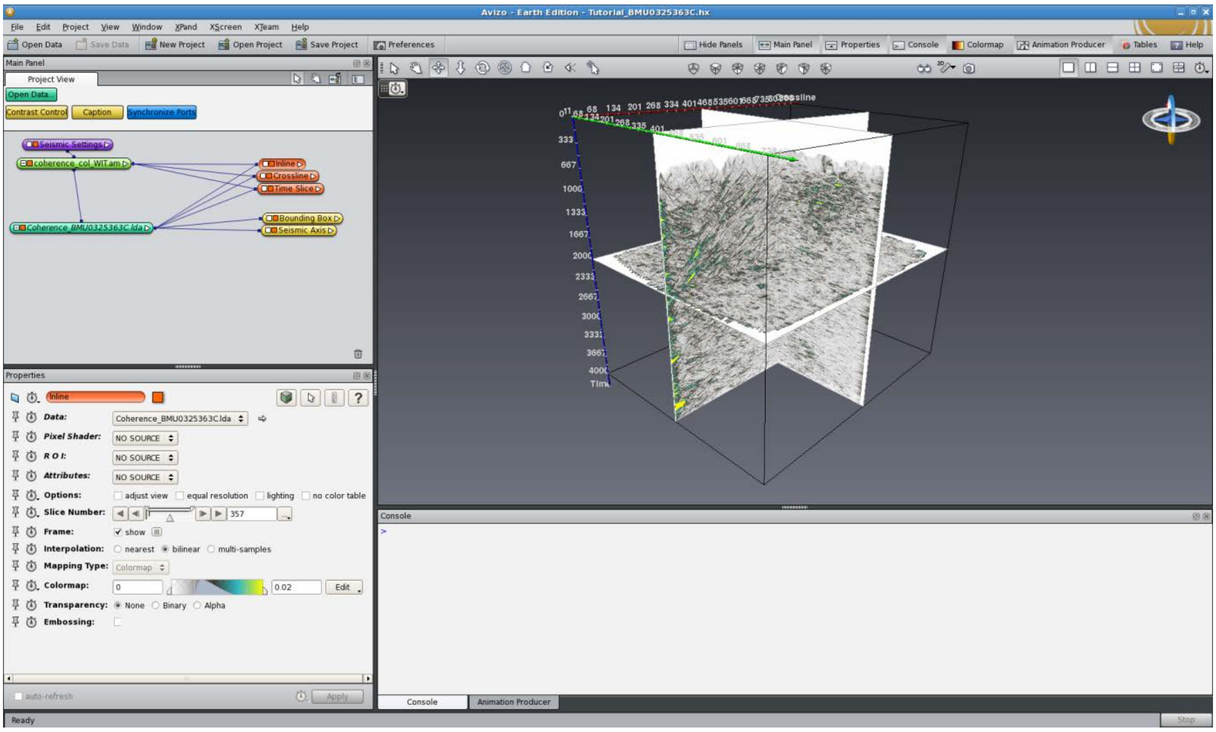
- Change axis titles, font and font size
- Change bounding box color

Click on the ***Seismic Axis*** module to see the module properties. In the second last line you find the option ***Titles.*** Type ***Crossline*** for x, ***Inline*** for y and ***Time*** for z. By clicking on the Select button in the ***Font*** line, you can select a different font, style, and size. With respect to good readability in videos we suggest using ***Nimbus Sans L*** in ***Bold*** type for all annotations and 14 pt. for the axis labels.

In the ***Bounding Box*** module / properties go to ***Options*** and click on the orange (current color) square left of ***color*** to open the Color Dialog. Choose, e.g. black and leave the dialogue window with ***OK***.

6.2 Adjust slice settings

Before we go on to the next step, change the frame color of all three slices from orange to a less striking color such as light gray via the color button next to ***Frame: show***. For all slices uncheck the ***lighting*** option to render all slices independently of the camera viewing angle. The following picture shows what the data cube and the Avizo interface should look like.



Note: Remember to save your project occasionally and keep in mind that Avizo does not have an 'undo' option!

6.3 Create legends and annotations

It is time to add some annotations to our visualization! We will now create a colormap legend with annotation, a copyright remark and a caption for the data view that we will set up next.

Colormap Legend

Right-click on the ***Coherence_col_WIT.am*** module ⟶ Annotate ⟶ Colormap Legend ⟶ Create

A ***Colormap Legend*** module has appeared in the project view and is displayed in the viewer. We will now change the appearance and position of this colormap legend.

- Check ***custom text*** option and change ***Custom Text*** line to 0/0 0.01/0.01 0.02/0.02

- Check ***font*** and change font to ***Nimbus Sans L Bold 14 pt*.**

- In the ***Size*** options, set ***length*** to **250**.

- Set the ***Position*** to x=50, y=50.

The latter option tells Avizo to position the colormap legend relative to the left (x) and lower (y) border of the viewer window. Negative position values refer to the right (x) and upper (y) border of the viewer.

Colormap title

The easiest way to add a title to the colormap legend is to fill out the ***Title*** field in the ***Colormap Legend*** module. The title will be centered on the middle of the legend. For creating a title that is not centered or has a larger spacing below the color bar you should annotate the colormap legend with a caption:

Right-click on the ***Colormap Legend*** module ➡ Annotate ➡ Caption ➡ Create

- ***Text: Coherence***

- ***Font:*** Nimbus Sans L **Bold** (**16 pt.**)

- ***Absolute Position:*** x=140, y=25

Copyright notation

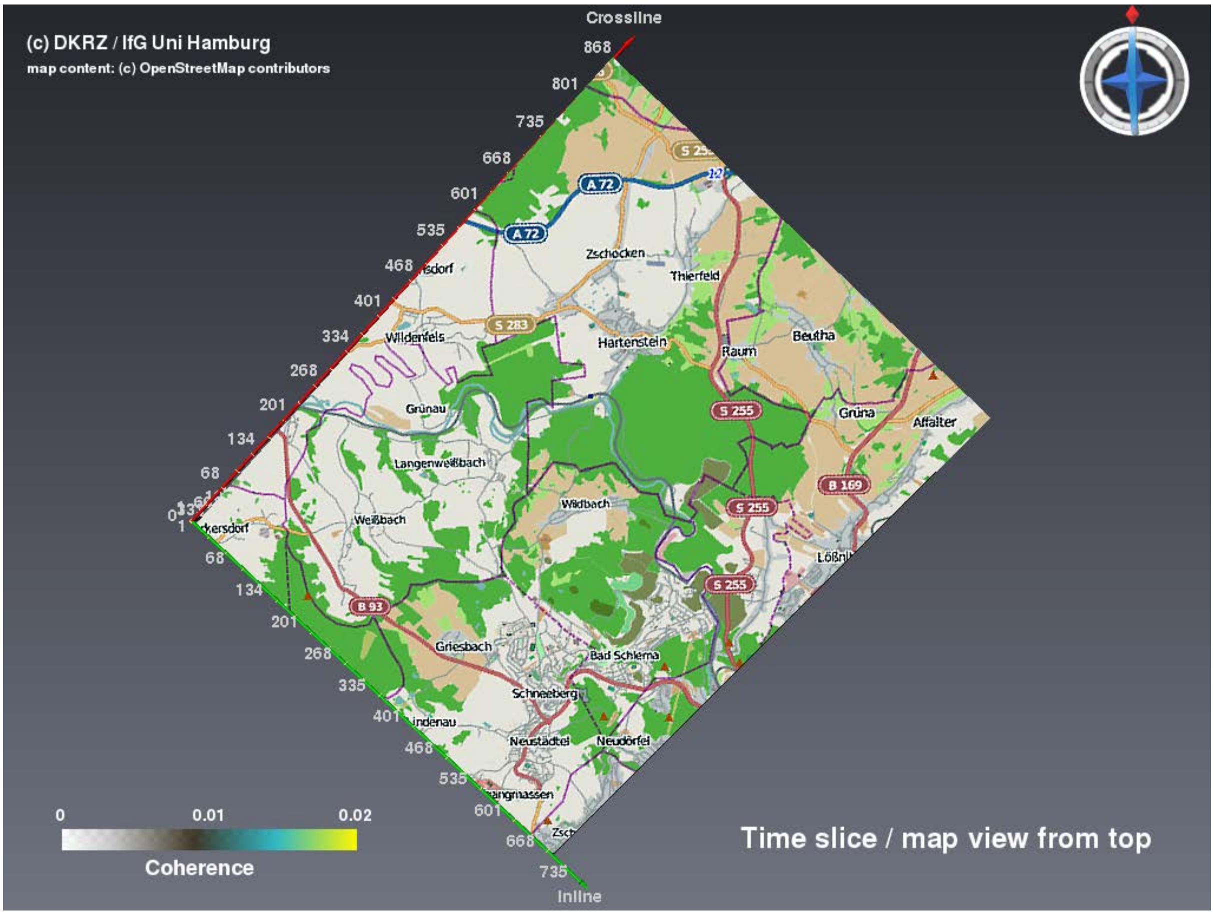
If you do not create animations with Avizo Earth for fun but intend to include visualizations of your work in publications or presentations, you should add a copyright notation. We suggest doing this by inserting a caption in the upper left corner of the viewer.

Right-click in the ***Project view*** ➡ Create Object… ➡ Annotations ➡ Caption ➡ Create

You can keep it at the default absolute position **(20,-20)** and only change the font and font size, e.g. to ***Nimbus Sans L Bold 16 pt*.**, and the text to **(c) your name / institution**.

7. Setting up a top view with a map

We would like the coherence data animation to start with a top view of a street map showing the geographic context of the data cube. The map should be oriented vertically towards north. Our goal is to achieve a view like this:

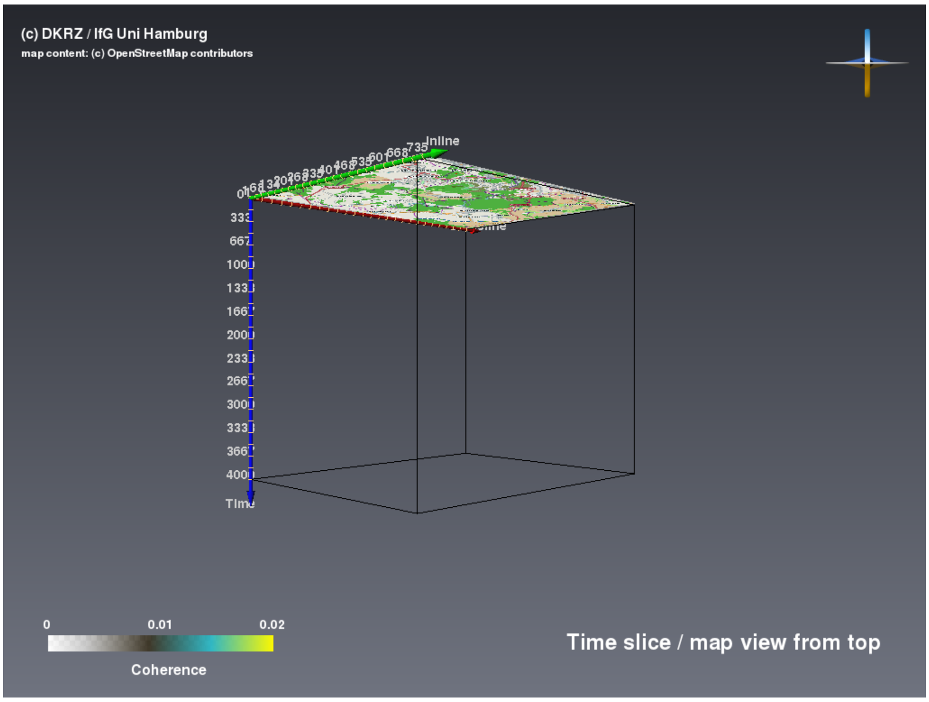


Set up camera position

- Switch off ***Inline***, ***Crossline*** and ***Time Slice***
- Check option ***Show viewer in toplevel window*** under ***Preferences – Layout***
- Type in the ***Console*** window: **viewer setSize 1024 768**

Now your viewer should have the size 1024 x 768, which will also be our video output resolution later on. Type ***viewer getSize*** in the ***Console*** window to check if your command was successful. Note: as long as the viewer is not shown in a top level window, the size and aspect ratio of the viewer is restricted by the size of Avizo's main application window and the minimum sizes of various GUI elements.

- In the ***Viewer*** toolbar, select ***View from south (Z)***  to obtain the view shown in the next figure, but so far without the map.



- Use the camera trackball (hand symbol) to tilt the southern end of the map downwards until the desired top view is achieved.

- If necessary, zoom in on the map and axes.

If you are satisfied with your top view, type ***viewer getCamera*** to get the current camera settings. Copy those into a text file and save them for later. By copying them into the ***Console*** and pressing ***enter*** you can always return to these camera settings.
It might also be a good idea to set this view as the ***home view*** by clicking .

In case you are not successful or want to be on the safe side with the map, go to appendix A to find the camera positioning that we used for the example project.

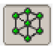
Load and position map

Open data ➡ Schneeberg_OSM1.jpg ➡ OK

A new green module for ***Schneeberg_OSM1.jpg*** has appeared in the project view. To visualize it in the viewer we need to connect an ***Ortho Slice*** module.

Right-click on the ***Schneeberg_OSM1.jpg*** module ➡ Display ➡ Ortho Slice ➡ Create

Note that a tiny map has appeared at the bottom of the time axis. We now need to scale it properly and bring it to the top. In the next step we will position the map a little above the first time slice and rescale it.

Right-click on the ***Schneeberg_OSM1.jpg*** module ➡ Transform Editor 

➡ ***Manipulator: Transformer*** Dialog... **.**

Note the blue square in the map module showing that the editor is active. Try the following settings:

- **Absolute** tab: Translation 0 0 4040
- Scale factor 13 13 1

This should position the map above the seismic axis and enlarge it in x- and y-direction so that it almost fits the lateral extent of the ***Bounding Box***. However, the map is flipped upside down and displayed at the wrong angle relative to the horizontal axes. This is where the really tedious work begins! There are two strategies you can follow to get the correct mapping: graphical interaction and/or numerical entries:

For interactively manipulating the map position, orientation, and size: In the ***viewer***, switch to ***interact mode*** by clicking on the arrow symbol or by pressing ***ESC***, then:

- Grab the green button in the middle of the slice or on the sides to flip the slice bottom-to-top (Press ***shift*** button for rotation around horizontal).

- Use the white cubes to adjust the size of the slice; be careful not to press shift while dragging because this causes unilateral instead of diagonal stretching or compression.

- Grab the green handles outside of the slice for
  a) Rotation around the vertical axis
  b) +**shift:** Rotation around an axis through the middle of the slice, which is either parallel to the inline or crossline axis
  c) +**Ctrl:** Commands as before but with rotation radius doubled

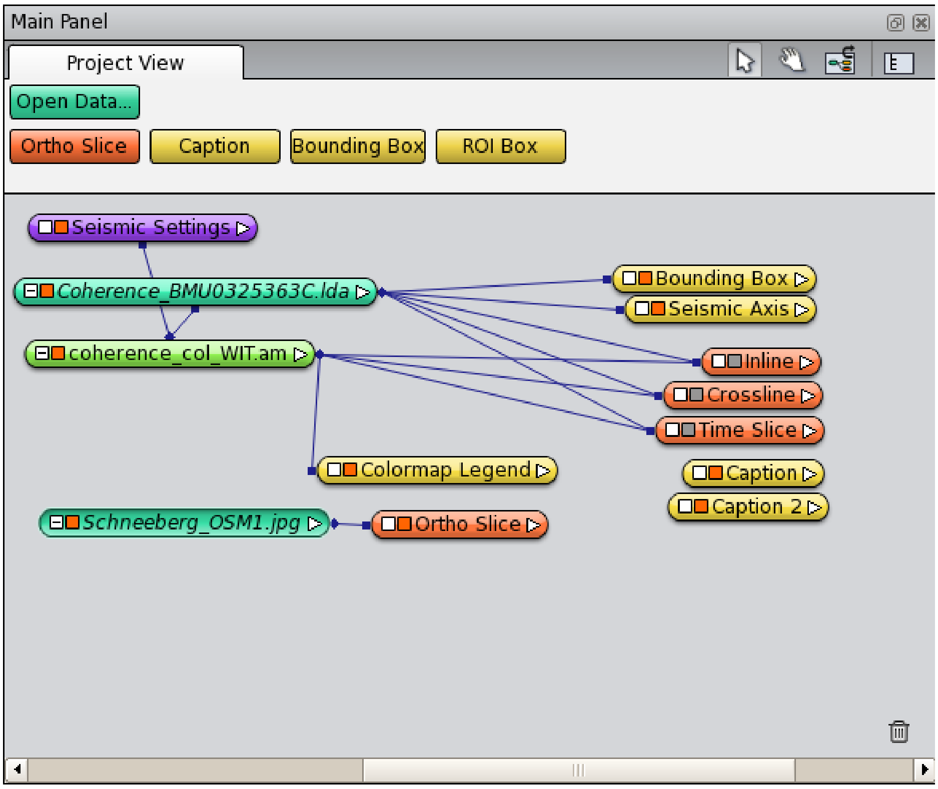
For a numerical manipulation of the mapping: for 'fine tuning' try to adjust the values in the ***Transform Editor Dialog***, <u>***Relative Local***</u> tab by entering values for ***Rotation around the local z-axis*** and ***Translation***. To reverse a translation or rotation you need to enter the same value but with a negative sign – to ‘undo’ a 90° rotation, type ***-90*** and click on ***Apply***.
Click on ***Close*** to leave the ***Transform Editor*** window. The green handles remain visible in the viewer until you click on the ***Transform Editor*** symbol again. To check how well your map is positioned, switch back to the ***camera trackball*** (hand symbol) in the viewer to tilt the bounding box, axes, and map. Last but not least change the color of the Ortho Slice frame to a light gray.

Create annotation for this view

The last thing missing in comparison with the previous figure is the caption reading **‘Time slice / map view from top’**. We will show this caption again at a later point.

To create a caption as seen in the figure, right-click in the ***Project View*** ➡ Create Object… ➡ Annotations ➡ Caption ➡ Create

Choose ***x= -50*** and ***y=53*** and set ***Nimbus Sans L Bold 24 pt.*** Now your project view should look similar to the following picture:

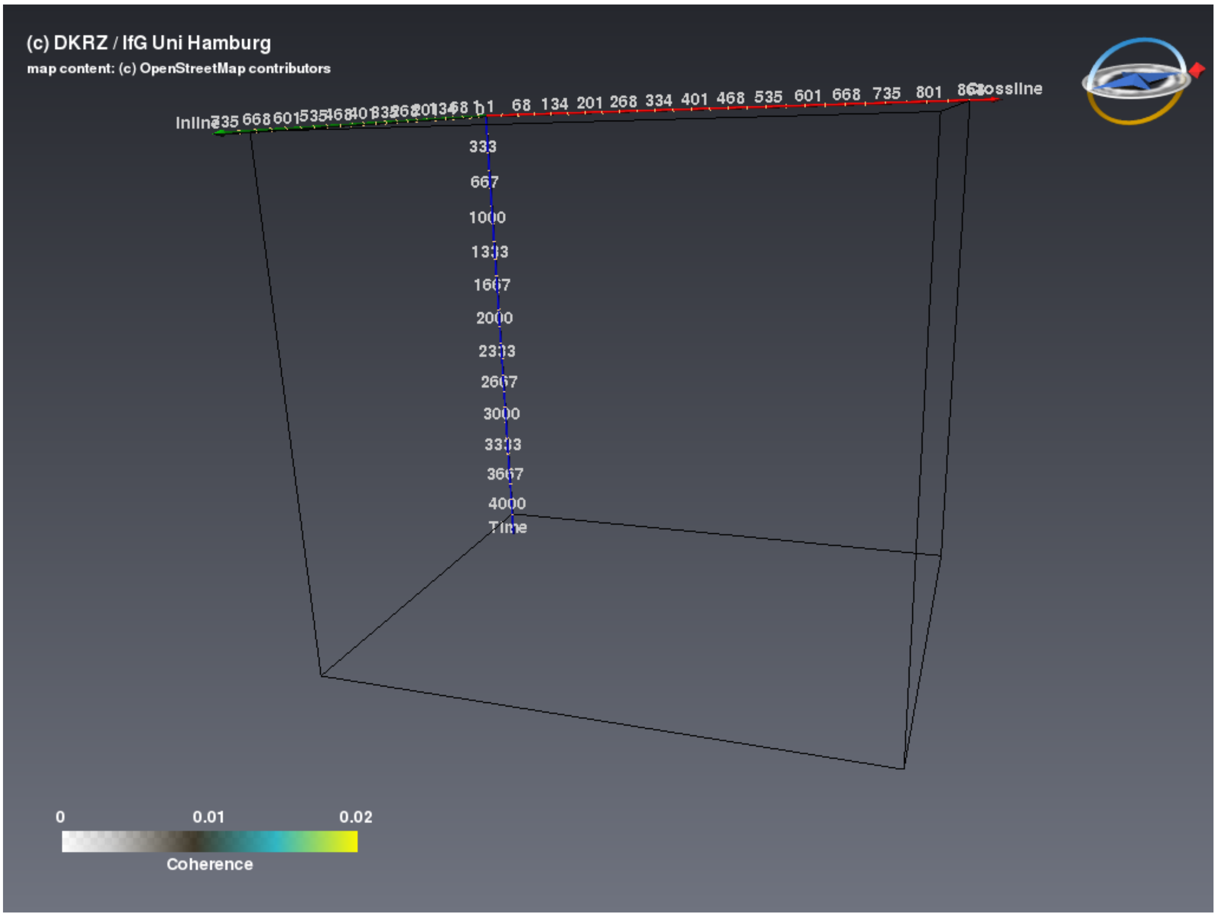


8. Create a camera path to rotate data cube

After showing the top view with the map we would like to animate moving inline slices. To do this, we need to change the camera perspective by creating a ***Camera Path***. This is especially simple when we have the camera settings at the beginning and at the end of the path.

Finding a starting position for animating the inline slice

Our data cube should be oriented more or less like in the following figure in order to have a good view of the moving inline slices.



Use ***trackball***, ***translate*** and ***zoom*** to set up this view or simply peek at appendix A for the camera settings we used. If you have set up the camera yourself, don't forget to save the camera settings in a text file.

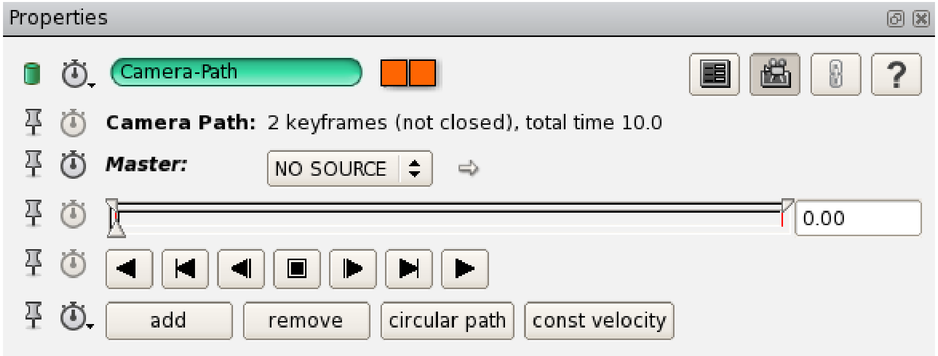
Creating the camera path

In the ***Project View,*** right-click ➡ Create Object… ➡ Animations And Scripts ➡ Camera Path ➡ Create

A new green Camera Path module has been added to the project view. In the ***Properties*** of this module click on the ***Camera Path Editor*** , which will change the appearance of the ***Camera Path Properties*** window and will open a new viewer window (Viewer 5) to show the camera trajectory created by the ***Camera Path***. Note that all modules have received an additional square toggle for this second viewer.

To get started, copy your camera settings for the initial top-north view into the console and hit enter. Then click on the 

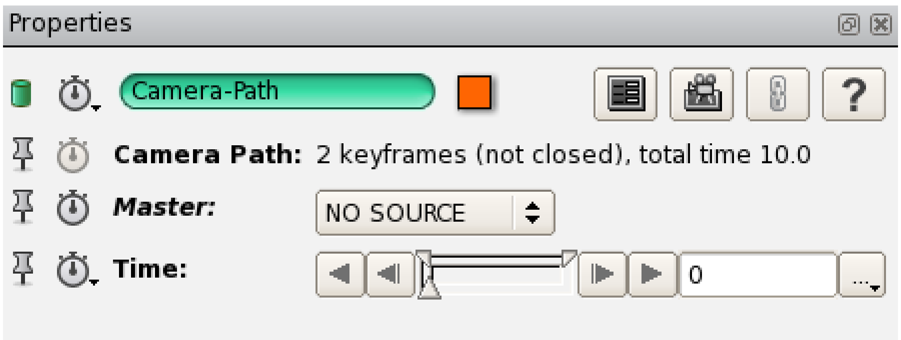
button in the Camera Path Properties.



A thin red bar marking the first keyframe has appeared. Now copy the camera settings for the start of the moving inline slice into the ***Console***, hit enter again and add the second keyframe. Press the play  button in the ***Camera Path*** module to see the resulting camera movement in the main viewer (Viewer 0).

This is the easiest way to create a smooth camera path between two subsequent camera settings. Alternatively you could start from your initial camera position, rotate the cube with the ***trackball*** and adjust the view with ***translate*** and ***zoom,*** and add keyframes bit by bit. When you have reached your final camera setting, delete the keyframes in between to achieve a smooth transition.

The time value that you see in the ***Camera Path*** module is not of importance for the duration of the animation that we want to create. We will adjust this later in the ***Animation Producer***. When you are done with creating your Camera Path, click on the ***Camera Path Editor*** symbol again to close Viewer 5 and to return to the initial ***Camera Path Properties*** view:

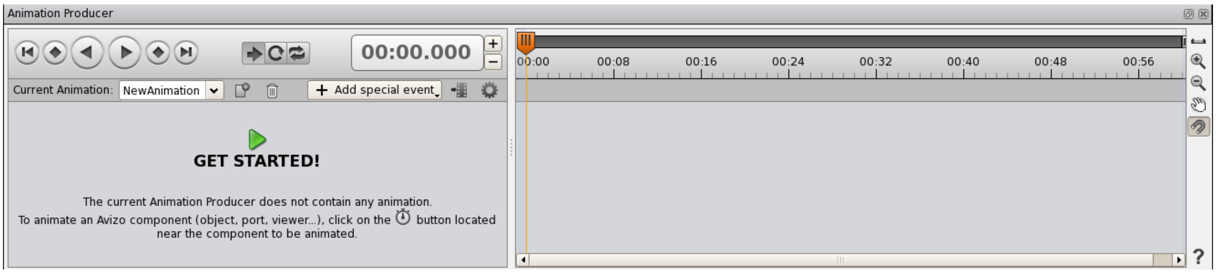


Next time you save your project, Avizo Earth will ask you to save this camera path as a file in .civ format. Be sure to save it in your own folder!

9. Introduction to the Animation Producer

We have decided to include a chapter on the basic handling of the Animation Producer for those of you who are new to creating animations with Avizo. You may want to skip this chapter if you are already familiar with its usage.

Click on the Animation Producer button in the menu bar to open the Animation Producer and to start creating a first animation.



The topmost menu bar, the ***Animation Producer control panel***, contains buttons for steering the orange ***master time slider***, play buttons for the animation and the time window showing the time value corresponding to the current master time slider position. Let us examine these three elements in detail.

Steering buttons

- Jump to Start or End of animation
- Jump to previous (*left*) or next (*right*) keyframe
- Play animation backward (*left*) or forward (*right*)

Play buttons

- Play once
- Play loop
- Play swing (back and forth)

Time window

00:00.000

The time window does not only tell the temporal position of the master time slider but can also be used to move the slider more precisely than by dragging it with the mouse cursor. Simply overwrite the displayed time to move the slider to the desired position. If Avizo does not accept your time input, check and maybe increase the **End** time of the animation. This is done in the Animation Producer's ***More options*** (see menu bar below the steering and play buttons and the time window). By modifying the ***Snapping Resolution*** you can change the smallest time steps you can address by moving the time slider.

In the same menu bar you will also find the ***Current Animation*** (named ***NewAnimation*** by default – can be overwritten with a title of your choice) and buttons for adding more ( ) and for deleting animations ( ) in the current Avizo Earth project.

Getting started

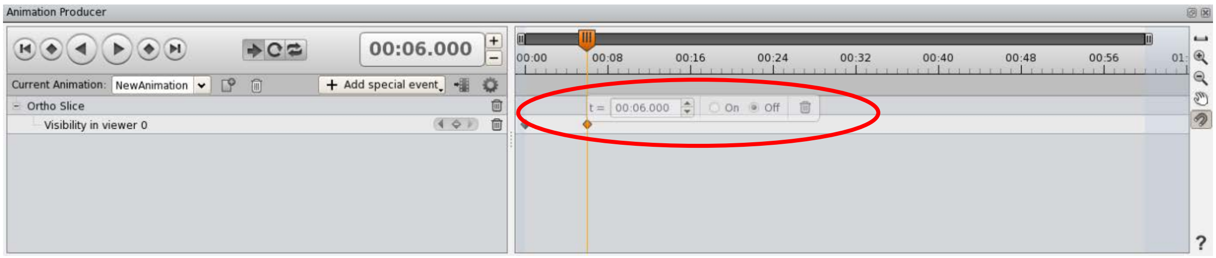
Many ways end in Rome. – Wise words applying to the Animation Producer as well. The same results can be accomplished in different ways: Some of us may prefer one way over the other or the problem at hand is a decisive factor for the way to go. In the following paragraphs we will therefore present several techniques for adding keyframes to the animation and for navigating between those.

Add events to the event list

A new animation project starts with an empty event list in the left sub-window of the Animation Producer. Events can be added by selecting a module in the ***Main Panel***'s ***Project View*** and by clicking on one of the ***stopwatch buttons*** in the module's ***Properties***.
All module ports with a stopwatch symbol can be animated. The stopwatch symbol with a small downward pointing arrow on the side indicates that there is a drop-down menu with several 'sub-ports' to be selected. For colormap animations there are, e.g., options for a colormap source as well as for a minimum and a maximum value.

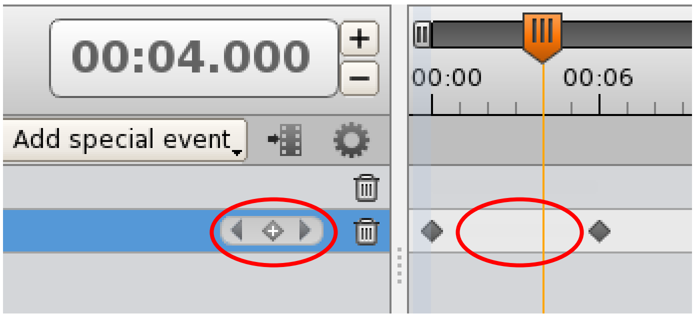
A module such as a slice, caption, etc. is added to and made visible in the Animation Producer by clicking on the stopwatch next to the ***module name*** in the ***Properties***. From the opening drop-down menu, ***Visibility in Viewer 0*** is chosen. This creates one ***main event for the module*** (*in this figure:* ***Ortho Slice***) and a lighter-shaded ***sub-event for the module port*** in the event list. Additionally, a first diamond-shaped keyframe is added at time 00:00.000 s.

When keyframe and master time slider coincide in the time line, the keyframe is active and highlighted in orange color. Its toggle – ***on*** or ***off*** – depends on whether the module was active ( ) or inactive when the stopwatch symbol was clicked. To check the toggle, hover over the keyframe symbol with your mouse cursor until you see a small input field appear (see next figure). The field states the keyframe time and whether the module is switched on or off at that time. These settings can be changed. To change the time value of the keyframe, either overwrite the numbers or use the up and down arrow keys.
Alternatively, you can shift a keyframe in time by clicking on it with the left mouse button pressed down (note keyframe symbol turning red) and dragging it to the desired time line position.



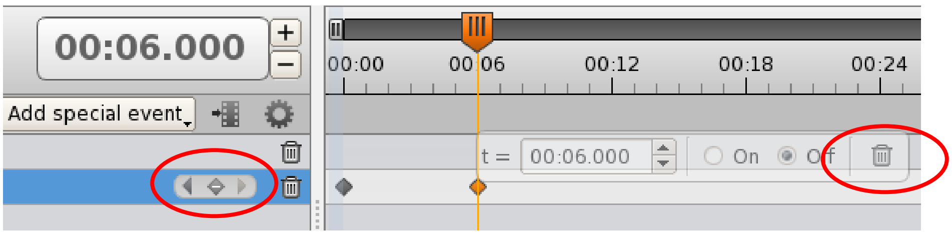
Add new keyframes by:

- Clicking the *stopwatch button* of a module port
- Clicking on the *'Add keyframe' button* (diamond symbol with **+**) in the event list
- Double-clicking with left mouse button at desired position in time line



Delete keyframes by:

- Clicking on the *'Delete Current Keyframe' button* (diamond symbol with **-**) in the event list. This button is only shown for active keyframes as shown below.
- Clicking on a keyframe and then hitting the Delete key of your keyboard.
- Clicking on the trashcan symbol in the *small input field* shown by hovering with your mouse cursor.



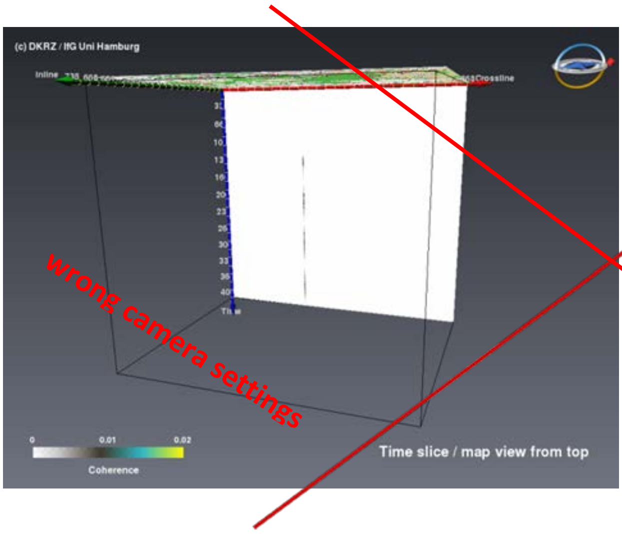
To delete individual keyframes be sure to use the trashcan symbol in the small semi-transparent input field which pops up when the mouse gets over the keyframe. The trashcan symbol next to 'Add keyframe' (or 'Delete Current Keyframe', respectively) deletes the whole module port or sub-port with all its keyframes from the event list. The trashcan symbol in the module line, e.g. for Ortho Slice, removes the whole module with all its ports and sub-ports from the event list.

Most of your modules need not only be switched ***on*** but also switched ***off*** at a later time. The same applies to module ports such as slice number, time, etc. – You will need more than one keyframe for each of those. Therefore we suggest adding events to the event list when the slider is positioned on a ***non-zero time value***. This automatically creates a second keyframe at 00:00.000 s. If the port is about ***Visibility***, the automatically created keyframe will have the ***opposite*** toggle. For example: If you switch ***on*** a module, e.g. the Ortho Slice module, at 00:03.000 s via the stopwatch button, the keyframe at 00:00.000 s will have the toggle ***off*** – and vice versa. If the port is about setting a value, such as slice number, or a source, such as the colormap used, both keyframes will have the same values or source, respectively.

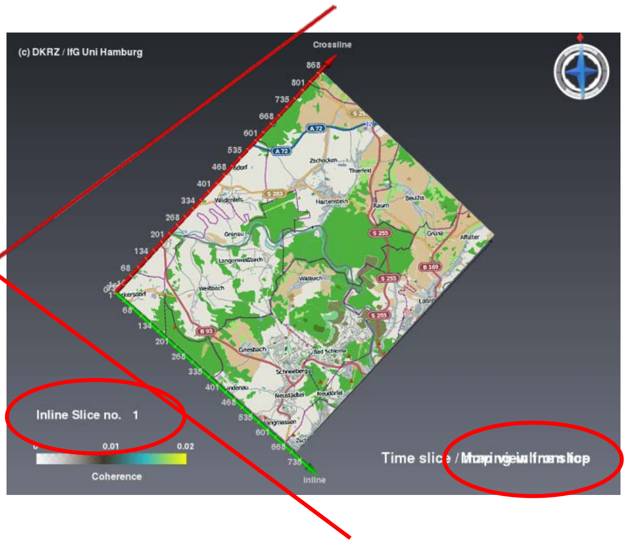
Warning:
For module ports which modify values or sources, the keyframe at 00:00.000 s can be shifted to a desired later time on the time line. This is ***not advisable*** for the ***visibility ports*** of most modules. Returning the master time slider to 00:00.000 s will not switch off captions, slices etc. that are to be shown later in the movie. For a clean initial state, the startup values need to be set on the very first frame.

The following figures show what could happen when the initial visibility keyframes at 00:00.000 s are missing.

Return to 00:00.000 s with ***Jump to start*** button :



Return to 00:00.000 s with the ***master time slider***:



This is clearly not desirable.

For the ***Visibility in Viewer 0*** of ***Camera Path*** and ***Camera Orbit*** we also suggest setting the keyframe with toggle ***off*** at 00:00.000 s.

Tip:
If you press the ***Jump to Start*** button in the Animation Producer menu, hoping to

replay your animation from 00:00.000 s, you may find that the ***Camera Path*** has not been reversed. This problem can be overcome by:

- Using the ***Jump to previous keyframe*** button to return to 00:00.000 s
- Using the master time slider to return to 00:00.000 s
- Adding the initial camera position as a TCL command at 00:00.000 s

Change time values of a port or entire module in the event

You may want to add new elements to your animation – e.g. not at the end but maybe at the beginning or in between. Shifting each and every keyframe manually would certainly take a long time. Fortunately Avizo provides the options of simultaneously shifting all keyframes per port or even per entire module in the event list. This is what the following two paragraphs are about.

To shift all keyframes of a ***port*** in parallel:

Position the mouse cursor on the time line, on the space between the keyframes to be shifted. Just hover there, without clicking, and observe that all keyframes have become active (orange) and are connected by a dark gray bar. Hold down the left mouse button and drag with the mouse to shift all keyframes simultaneously.

To shift all keyframes of a ***module*** in parallel:

Position the mouse cursor on the time line ***above*** the keyframes, in the same line as the ***module name*** (e.g. Ortho Slice). This turns the space between keyframes in the subordinated ports light blue and shows the connecting dark gray bar above, in the line of the module name. Click and drag to shift all keyframes in all subordinate ports in parallel.

Run-time / Playing speed of animation

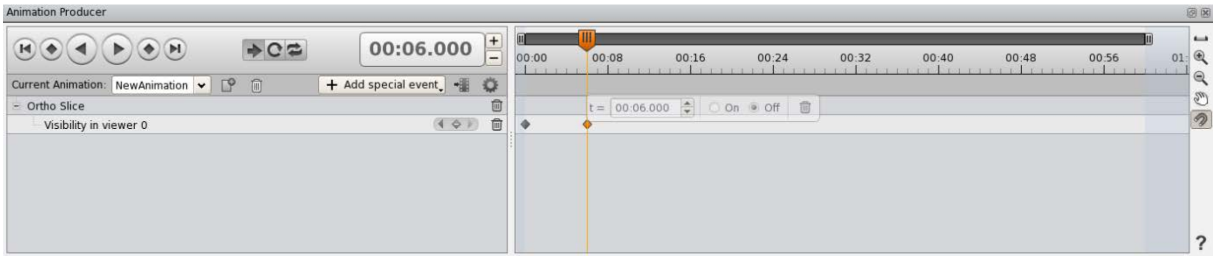
When running your first animation you will realize that the animation runs more slowly than expected. 1 second of movie time may correspond to 2 or more seconds in real time. Do not let this disturb you. – The final, exported video will run as per your Animation Producer settings.

10. Create tutorial animation with moving slices

Prior to building the animation, return to the initial top-north view by using the ***Home*** symbol or by entering the camera settings in the ***Console***. Switch off all slices except the ***Ortho Slice*** module displaying the map.

We plan to show the map for 6 seconds – 3 seconds in the top view and in the next 3 seconds while the camera path is played. Thus we switch ***off*** the ***Ortho Slice*** module toggle and add the two required keyframes at 00:00.000 s and 00:06.000 s. One way to do this is by navigating the orange main slider to 00:06.000 s and by clicking on the topmost stopwatch symbol in the ***Ortho Slice*** module, choosing ***Visibility in viewer 0*** from the drop-down menu.

Hover over the keyframes' diamond symbols to see if both have the correct toggles.



Next, colormap legend and captions should be animated starting from 00:00.000 s. Note that the colormap legend and copyright caption should remain active during the whole animation. Only the caption reading **'Time slice / map view from top'** should be switched off at 00:06.000 s.

Animating the Camera Path

Next we will animate the camera path created before, starting at 00:03.000 s. Before you proceed make sure that your ***Camera Path*** module is active ( ) and that the ***Time*** port shows the value 0.
To add the ***Camera Path*** to the the Animation Producer event list, click on the topmost stopwatch symbol in the ***Camera Path Properties*** and select ***Visibility in viewer 0*** from the drop-down menu. Two Camera Path keyframes have been added to the event list: A keyframe with toggle ***off*** at 00:00.000 s and with toggle ***on*** at 00:03.000 s. Add a third keyframe with toggle ***off***, marking the end of the ***Camera Path***, at 00:06.000 s.

Return to 00:03.000s and click on the stopwatch button next to the ***Time*** section in the ***Properties*** panel of the ***Camera Path*** module and choose ***'Time: value'*. This produces two keyframes as before, both with time value 0 at 00:00.000 and 00:03:000s and time value 0. Move the first keyframe to time 00:06.000s and change the time value in the ***Camera Path*** module to **10** (total time).

To see the result of your first animation efforts, return to animation time 00:00.000 s and press the **play** button in the **Animation Producer**.

Animating the inline slice

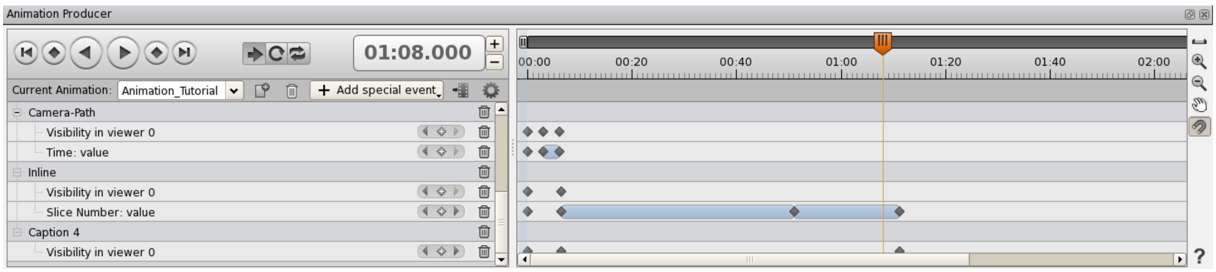
We will now show how to animate the position of the inline slice. Starting with **inline no. 1** we will go all the way through to **inline no. 712** and then return to **inline 390**. The intersection of inline 390 and crossline 310 marks the position of a borehole and is therefore of special interest to us.

From experience we suggest showing at least 16 slices per second, which makes the passage from inlines 1 to 712 44.5 seconds long. For returning to inline 390, let us add another 20.125 seconds. The inline animation should start at 00:06.500s and finish 64.625 seconds later, i.e. at 01:11:125 s. As this exceeds the default movie end time of 1 minute, click on the ***'More options'*** gear wheel in the **Animation Producer** and extend the end time to 3 minutes.

Along with the moving inline slice we would like to display a caption similar to ***Caption 2*** , which reads 'Time slice / map view from top'. Click on this module and press Ctrl+C and Ctrl+V for copying and pasting. Overwrite the newly created caption text with ***'Moving inline slice'***.

In the Animation Producer, navigate to time 00:06:500 s and switch on the ***Inline*** toggle. Set the ***Slice Number*** value to our starting value of **1** and choose ***'Visibility in Viewer 0'*** via the ***Inline Animations*** stopwatch button. With the ***Slice Number Animations*** stopwatch button add ***'Slice Number: value'*** to the current animation***.*** Additionally, add the new caption to the animation by toggling it ***on*** and choosing ***'Visibility in Viewer 0'*** from the respective ***Caption Animations*** (in our case Caption 4).
Inline 712 shall be reached at 00:51.000s. – Thus we need a keyframe in the ***Slice Number*** port at this time. We suggest moving the keyframe from 00:00.000 s to 00:51.000 s and overwriting the slice value of 1 with 712. For slice number 390 at time 01:11:125 s, create a new keyframe by a method of your choice, e.g. via the 'Add Keyframe' button, the stopwatch button in the module properties or by double-clicking in the time line. To switch off the inline slice caption at this time, create a new keyframe with toggle ***off***. The inline slice itself, now at number 390, should remain visible.

If you scroll down to the bottom of the Animation Producer, you should see the following view. Note that for this example all keyframes at 00:00.000 s have been maintained, which is not mandatory for ***value*** sub-ports.

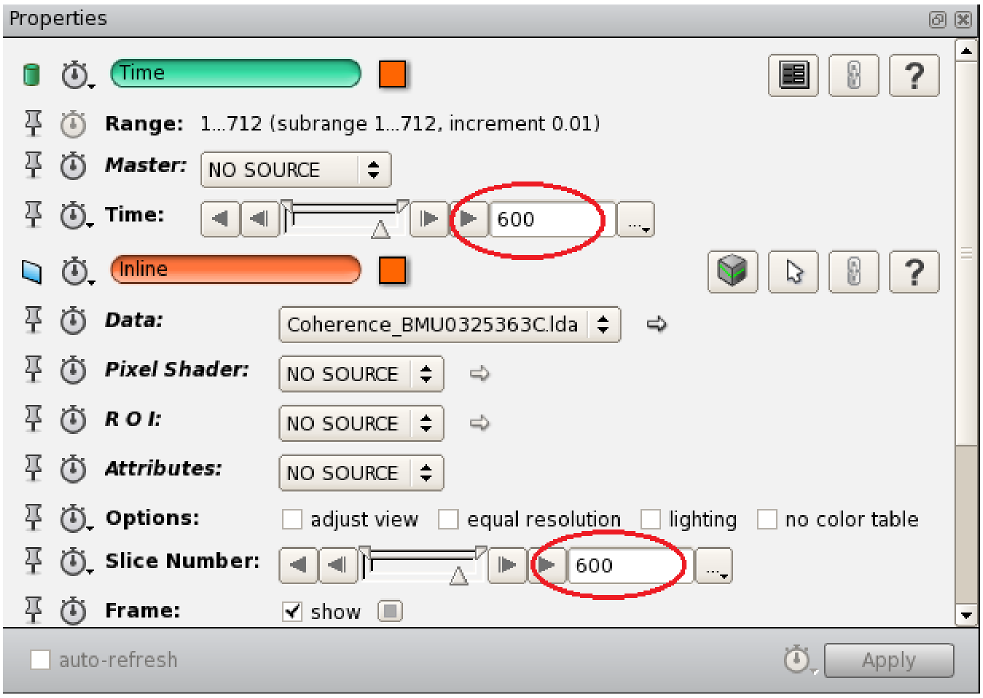


Displaying the current slice number

In order to identify the positions of interesting features in the inline data, we should annotate the number of the slice currently shown. Since this is not supported directly in Avizo Earth, we need a little workaround: This is can be accomplished by creating a ***Time*** module with ***Time*** representing ***Inline no.*** and connecting it with a ***Display Time*** module.

Move the main slider back to 00:06.500s, the start of the inline animation. In the ***Project View,*** right-click → Create Object… → Animations And Scripts → Time → Create. In the ***Project View***, click on the white square of the ***Time*** module, and choose ***Time*** from the drop-down menu. Connect the blue rubber band to the ***Inline*** module. This automatically adjusts the min and max values of the time module to the first and last inline slice numbers. Additionally, the time slider is set to the current inline number.

The following figure exemplarily shows how ***Time*** value and ***Inline*** number match.



To connect the ***Display Time*** module to the ***Time*** module, right-click on the ***Time*** module → Annotate → Display Time → Create. Along with the new module in the project view, a blue bar with the values 1 and 712 at its end points has appeared in the ***viewer***. Let us now change the layout of the time display because we only want to see the current slice number as an alphanumeric display.

In the ***Display Time*** module, select ***text only***, choose ***x*** position ***50*** and ***y*** position ***120***, delete <unset> in the ***Custom Text*** field, and fill the ***Format*** field with ***'Inline Slice no. %3.f'***. Once you have changed the font to ***Nimbus Sans L Bold 20 pt.*** we can include this display in the Animation Producer.

The ***Time*** module must be active (■) and we need to set a keyframe for ***Time:value*** in the ***Time*** module. With the ***master time slider*** at 00:00.000 s and 00:06.500 s, add keyframes for ***Visibility in Viewer 0*** in the ***Display Time Animations*** of the ***Display Time*** module and for ***Time:value*** in the ***Time*** module. At 01:11.125 s, when the inline animation stops, only ***Visibility in Viewer 0*** of the ***Display Time*** module needs to be toggled ***off***.

Switching on the crossline and rotating to a different viewing angle

To get a better view of the crossline slices, which we will animate next, it is necessary to rotate the data cube first. For this purpose we will switch on the crossline slice and rotate the cube anti-clockwise until the viewing angle of the image on the following page is reached.

Switch on ***Crossline Slice*** with ***Slice no. 1*** at 01:11.250 s, just after the inline slice has stopped moving. To achieve a simple rotation around the vertical axis, the ***Camera-Orbit*** module is most convenient. Create it by right-clicking in the ***Project View*** ➡ Create Object… ➡ Animations And Scripts ➡ Camera-Orbit ➡ Create.
Clicking on the ***More options*** **…** button of the ***Camera-Orbit*** module and choosing ***Configure*** shows that the default values range from 0 to 360 (degrees) with an increment of 1. We can thus close this dialogue window and make the ***Camera-Orbit visible*** at 01:11.250 s and set the ***Time value*** to ***360***. 3 seconds later, at 01:14:250 s, we want the time value to reach ***273*** and then switch the ***Camera-Orbit off***. For the correct rotation we need to execute the ***recompute*** command before the rotation starts. To do this, go e.g. to time 01:11:000 s and choose ***Action: recompute*** in the ***Camera-Orbit*** Properties panel.

Animating the crossline slice

Knowing how to animate (and annotate) the position of the inline slice, the transfer to animating and annotating the position of the crossline slice is straightforward. What we want to do is:

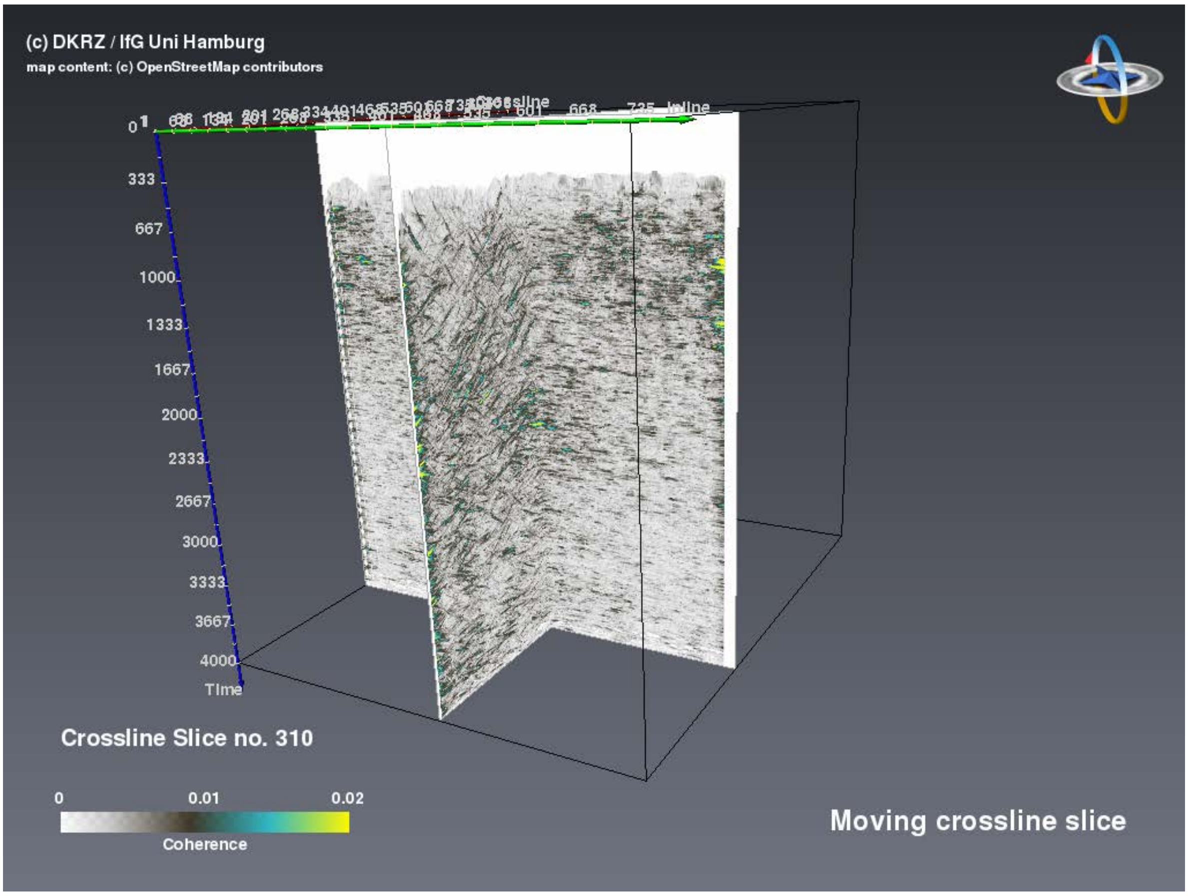
- Show the moving crossline slice from slice no. 1 to 310
- Perform a 360° rotation around the vertical axis
- Continue with moving the crossline slice from 310 to 845.

As before, we will display 16 slices per second. For the crossline animation starting at 01:14.500 s, crossline 310 is reached 19.375 seconds later, at 01:33.875 s.
Do not forget to set an extra ***Slice Number: value*** keyframe for slice no. 1 at the starting time! Otherwise the crossline animation will start too early, at 01:11.250 s, when it is first shown.

The subsequent 360° rotation should have a duration of 7.5 seconds, which corresponds to 48°/s. The same ***Camera-Orbit*** module as before can be used and in order to get the correct starting parameters, we need a keyframe for ***Action: recompute (most vertical)*** just before the rotation is performed, e.g. at 01:33.000 s. ***Time: value*** can then range from ***0*** at 01:33.875 s to ***360*** at 01:41.375 s.

Annotate the moving crossline slice with a caption, ***Time*** module and connected ***Display Time*** module as you did for the inline slice.

Just before the rotation, your viewer should look like this:



During the rotation, switch off both ***Display Time*** and the caption reading ***Moving crossline slice***. At 01:42.000 s, the crossline slice should continue to move and finally reach position 845 at 02:15.438 s.

Rotating the data cube for a top view of time slices

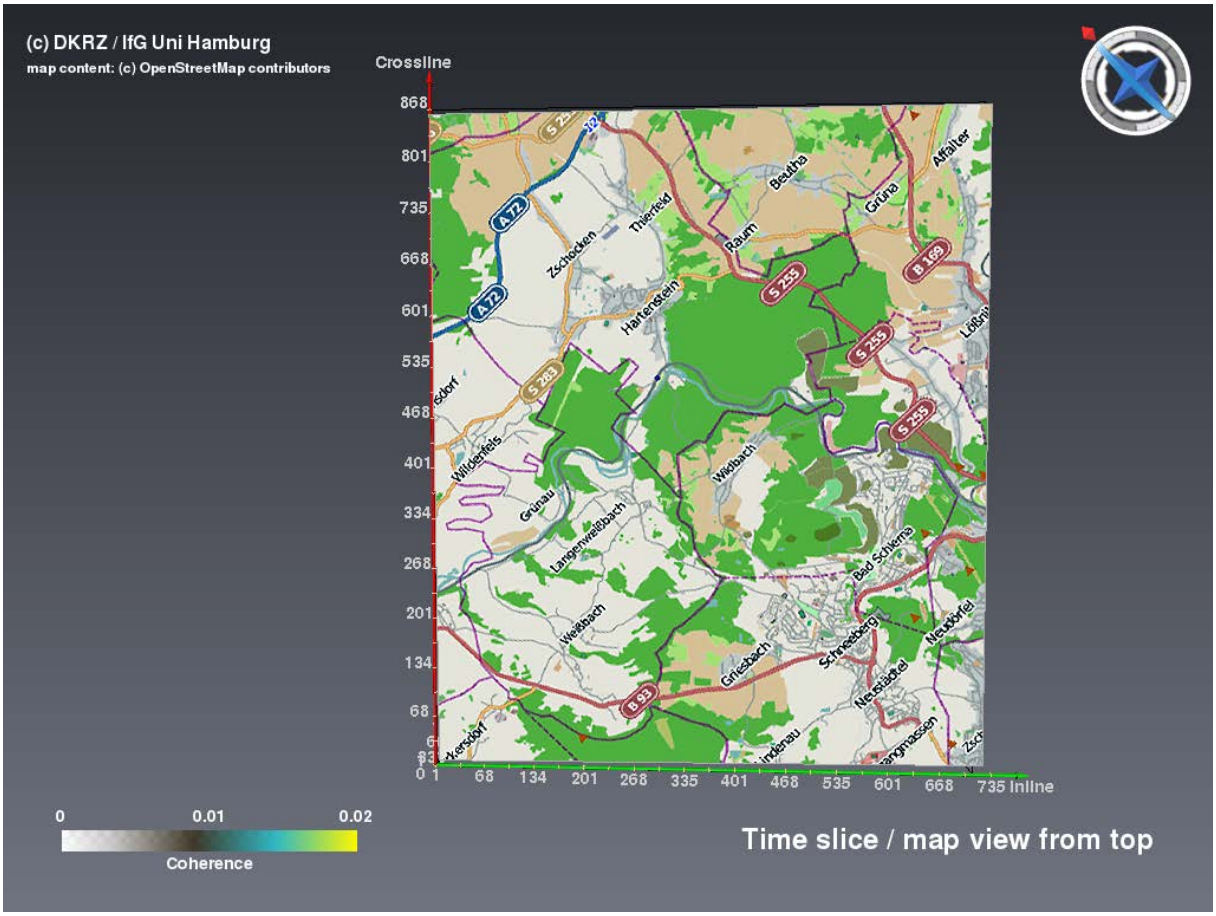
Having already looked at inline and crossline slices, it is time to animate time slices as well. To get a good view of the time slices, we will first rotate the data cube anti-clockwise until we have a perpendicular look at the inline slice using ***Camera-Orbit***. By creating a second ***Camera Path*** module, we can then rotate the camera to achieve a top view.

Before the new ***Camera-Orbit*** begins, e.g. at 02:15.000 s, add a keyframe to ***Camera –Orbit Action: recompute***. Our rotation should start at 02:16.000 s, thus we switch **on** its ***Visibility in Viewer 0*** at this time and **off** 2 seconds later, at 02:18.000 s. The respective ***Time: value*** settings are ***360*** and ***318***°.

At the end of the Camera-Orbit animation, switch off both inline and crossline slice and simultaneously switch on the ***Ortho Slice*** module connected to map module ***Schneeberg_OSM1.jpg*** again.

For the new ***Camera Path*** we suggest to use the same procedure as before: Manually rotate, shift and zoom the data cube until you have found a good setting for the time slice top view. Get the camera settings with ***viewer getCamera*** from the ***Console*** and save them in a text file. Your Camera Path could run 3 seconds, from 02:18.500s to 02:21.500s.

After the successful implementation of the Camera Path and with the time slice / top view caption switched on again, you should see the following image:



Animating the time slice

At 02:21.900s, just before we start animating the time slice (02:22.000s), switch **off** the ***Ortho Slice*** of the map. Instead switch **on** the ***visibility*** of the ***time slice***. Interesting features in the data mostly appear between 1400 ms (time slice 700) and 2400 ms (time slice 1200). This is the time period we are now going to visualize by animating the position of the time slice. According to our experience, one can go through the time slices more quickly than through inline and crossline slices. We will therefore display 30 ms of data per second, finishing at 02:55.333 s. This will also be the end time of the movie clip.

Displaying the current time slice number

Create ***Time module*** by:

Project View, right-click ➡ Create Object… ➡ Animations And Scripts ➡ Time ➡ Create

Time module ➡ ***More options*** ➡ ***Configure***

Connect ***Time*** port of the module to the ***Time Slice*** module

Create ***Display Time module*** by:

Right-click on the ***Time*** module ➡ Annotate ➡ Display Time ➡ Create

In the ***Display Time*** module, select ***text only***, choose ***x*** position ***50*** and ***y*** position ***120***, delete <unset> in the ***Custom Text*** field, and fill the ***Format*** field with ***'Time= %4.f ms'***. Also change the font to ***Nimbus Sans L Bold 20 pt.***

Then add the two keyframes for the ***Time*** value and one keyframe for ***Display Time*** to the Animation Producer as you did before for the inline and crossline slices.

Mark borehole position with a red sphere

While the time slice animation is running we would like to mark the intersection of inline 390 with crossline 310, the location of the previously mentioned borehole. In order to do this, we will make use of the option ***Create Sphere*** and change its vertical position synchronized with the time slice animation so that it is vertically centered in the <u>moving</u> time slice.
You can switch on the sphere on top of the data cube anytime you like, but our suggestion is doing it just in time for the 360° rotation of inline 390 and crossline 310. Thus it will clearly show the position it represents. In our animation this is time 01:33.875s.

Navigate to this time in the Animation Producer and create the sphere by right-clicking in the ***Project View*** ➡ Create Object… ➡ Surfaces And Grids ➡ Create Sphere ➡ Create. A red ***Create Sphere*** module has appeared. Choose ***octahedron*** as ***Sphere Type***, increase the maximum value of the ***Radius*** to at least 120 and choose 120 as current radius value.

The ***Point Coords*** coordinate entries of the sphere are not straightforward: the coordinate space used there is different from the coordinate system of the seismic data. As a result the x and y axes, to which the point coordinates refer, do not coincide with the seismic axes for inline and crossline; instead they are rotated against those by some degree. To overcome this problem, we tried to find the according x and y coordinates by trial and error. The 'z coordinate' does not match with the time samples either. To get started, try setting the following values:

x: 7420 y: -854 z: 4300

No. Of Points Per Unit^2: 0.1

When done, click on the green Apply button. A new module named ***SphereOct.surf*** has appeared and is connected to the ***Create sphere*** module. In the new module, click on the ***Transform Editor*** button and open the ***Dialog…*** in the ***Manipulator*** line. Go to the ***Scale factor*** fields at the bottom of the ***Transform Editor***, ***Absolute*** tab and change the third value from 1 to ***0.330434***. This removes the vertical exaggeration and reshapes the sphere from an ellipsoid to a round sphere.

Close the editor and click on the ***Surface Editor*** symbol . This has created and connected a ***Surface-View*** module. Moreover, a small sphere has appeared in the viewer. Next time you save your project you will be asked to save this surface. Be sure to choose ***Auto Save***. This will create a directory named ***Tutorial-files*** containing ***SphereOct.surf***.

Next, select the ***Surface-View*** module and set ***Draw Style*** to ***outlined*** in order to change the color of the sphere. For this dataset and colormap we suggest, e.g., bright red.

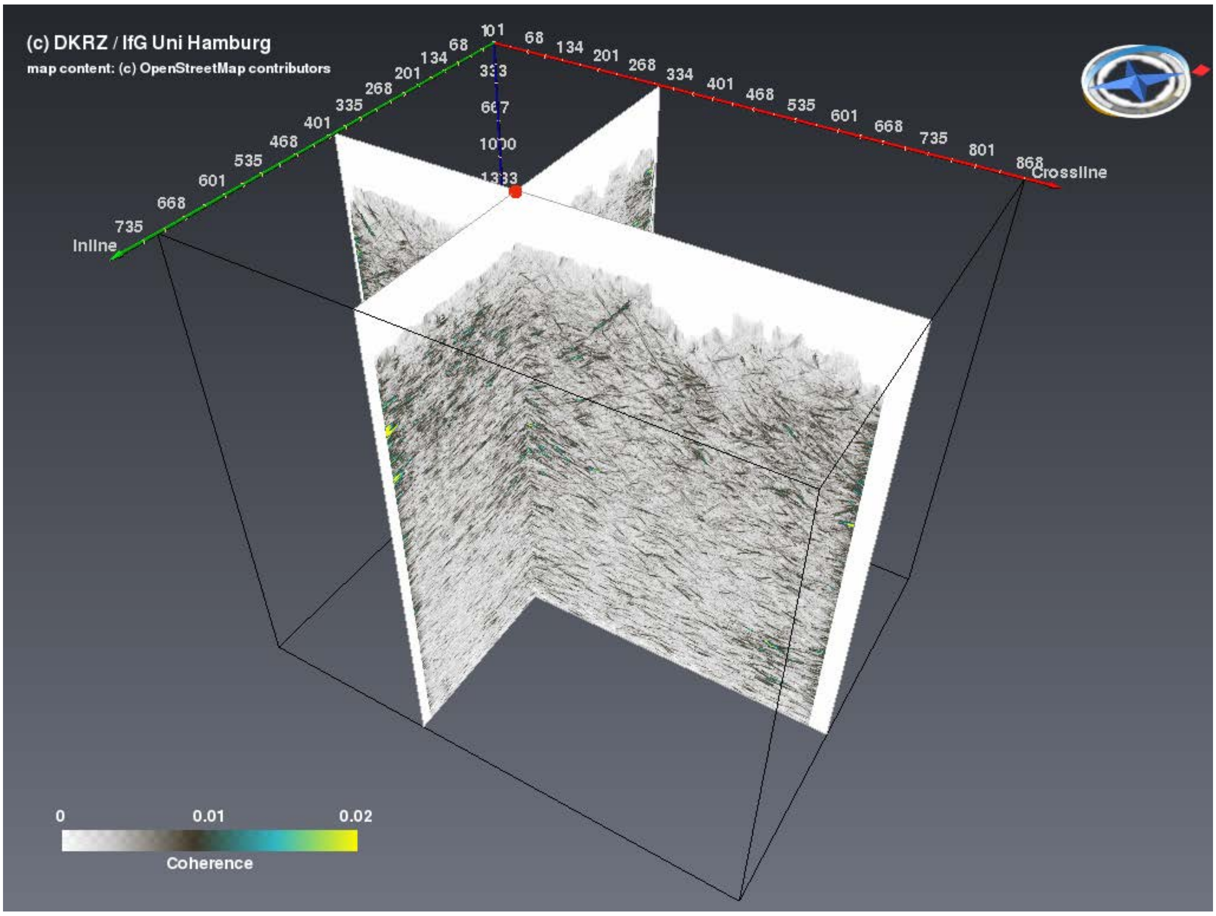
If, for some reason, the sphere does not sit with its center on top of the inline-crossline-intersection, select the ***Create Sphere*** module again, check ***auto-refresh*** and adjust the coordinates. When your cursor is in the respective coordinate field, use the mouse wheel to increase or decrease the values while observing the changes in real time.

Animating the sphere along with the time slice

In the following we will animate the red sphere that marks the intersection point of inline 390 with crossline 310.

We will first switch it on at 01:33.875s, when it shall sit on top of the intersection of the slices. Please set the current coordinates in the ***Animation Producer*** by choosing ***Point Coords Animations*** and selecting all three coordinates respectively. The sphere should remain visible at this position position until the next Camera-Orbit starts, e.g. till 02:17.000s.

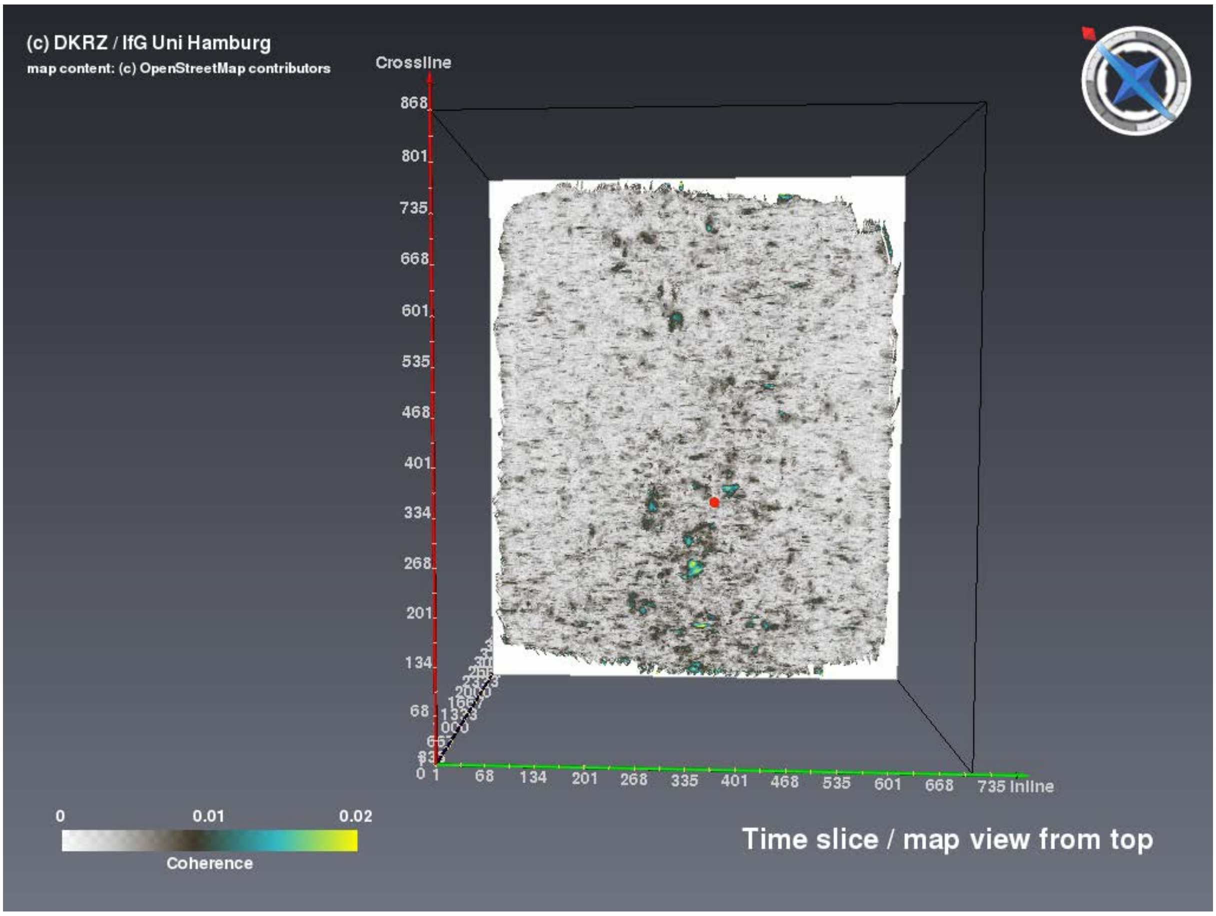
The following figure shows an exemplary view of the sphere sitting on top of the intersecting slices.



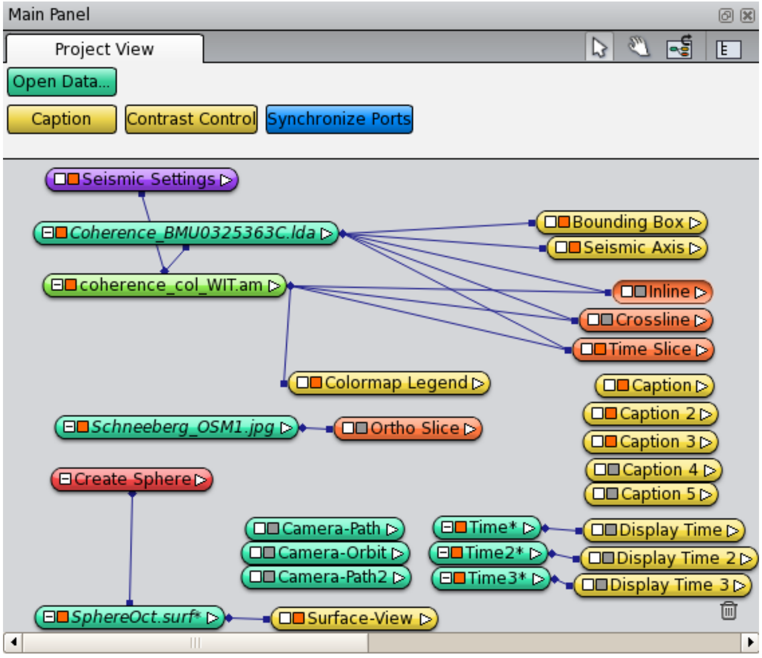
Then we briefly switch it off while the coordinates remain the same. Just before the map is shown, at 02:17.250s we should switch on the ***Surface-View*** again with a new z coordinate so that the sphere now intersects the map. Try a ***z coordinate*** of ***4680***.

From 02:22.000s on we should animate the sphere along with the time slices so that the sphere is always, 'cut in half' by the currently displayed slice. For this purpose we switch off the ***Surface-View*** at 02:21.900s while the ***z*** value is still ***4680*** and switch it on again at 02:22:000s with ***z coordinate*** set to ***70***.

Now navigate to the end time of your animation, to the final position of the time slice. If you have followed all timing instructions given here, this should be 02:55.333s. Set the ***z coordinate*** to a suitable value. We suggest ***-2954***. Now choose ***Show viewer in top-level window*** from the ***Preferences*** settings again and set the viewer size to 1024 x 768 again. Play your entire animation to check if it is ready for video export. At the end of the animation your ***Viewer*** should look like this:



The ***Project View*** should contain the following elements:



Movie creation

Once your animation is completely prepared and working, click on the ***Movie Creation***  button in the ***Animation Producer,*** and use the following settings:

Viewer:	**Viewer 0**	**AntiAlias 2 (or 3)**
File Format:	**MPEG movie**	
Filename:	**<yourdirectory-path/yourmovie.mpeg>**	
Frame Rate:	**25**	
Quality:	**0.4**	
Type:	**monoscopic**	
Size:	**Custom**	
Resolution:	**X 1024**	**Y 768**

The resolution setting used above is a resolution supported by almost every projector today. Of course you may change this setting to higher resolutions according to your needs.

When ready prepared, press the ***Create Movie*** button and wait until the movie creation is finished. Be careful that no other windows overlap the rendering area until the rendering of the movie is completed! The single images of the movie are first rendered in the frame buffer of the GPU and then read back and saved, so if you bring something else in front of Avizos' ***Viewer*** window, this will get a part of your movie.

Error messages

The next times you load your project, Avizo may present an error message regarding ***SphereOct.surf***:

Error in script <yourscript.hx>, line <line no.>:

Invalid command name ""
while executing
"["Create Sphere" create
] setLabel "SphereOct.surf" "

There is nothing to worry about – just click on ***Continue*** and your project should work fine was before.

11. Volume rendering and transparent slices

For the animation including volume rendering and transparent slice we need the second dataset, ***Cohmig_BMU0325363C.lda***. This file contains migrated coherence data and allows us to see fault-like structures, even though we are still in the time domain. As we will also use a different colormap, we suggest creating a separate Avizo project file. You can either:

- Begin from scratch with a new project or
- Save a copy of your tutorial project under a new name, e.g. ***Tutorial2.hx*** and delete / change existing modules .

This chapter describes the latter procedure, an adaptation of the previous project.

Visualization aim

In this chapter we want to combine the volume rendering technique and slices with transparency to better visualize and study structures in the migrated coherence data. You can imagine volume rendering as a 3D version of ***painting by numbers*** – every voxel has an RGB color value and an additional opacity value.

Our aim in this chapter is to create a visualization consisting of:

- Two 360° rotations while volume rendering is displayed (Camera Orbit)
- Rotate to a front view of the inline slices (Camera Path)
- Show animated inline slices 1 to 712 (Inline & Time)
- Rotate to a front view of the crossline slices (Camera Orbit)
- Show animated crossline slices 1 to 868 (Crossline & Time)
- Show inline 390 & crossline 420 with 360° rotation (Inline, Crossline, Camera Orbit)
- Rotate to a top view of the time slices (Camera Path)
- Show animated time slices for 600 to 2500 ms (Time Slice & Time)

For a video example, see ***Tutorial_Comig.mpg***.

Avizo settings

To make sure that the volume rendering is always completed before the frames are captured, please set the environment variable LDA_WAIT_TILL_VOLUME_IS_LOADED. This needs to be done prior to starting your Avizo session. So if you have started Avizo already, shut it down and type the following command in your console (in this example: csh/tcsh):

setenv LDA_WAIT_TILL_VOLUME_IS_LOADED 1

To check if the variable is set to value 1, type:

echo $LDA + tab (auto completion)

After that, start Avizo Earth as usual, open your previous project and save it under a new name **in the same folder**. If it is in a different folder, the paths to the respective elements (colormaps, data etc.) need to be adjusted manually in the .hx file.

Next we need to adjust the LDA preferences in Avizo Earth. For a fairly good rendering of slices, go to Preferences, ***LDA*** tab and position the ***Loading priority*** slider in the middle of High Slice/Low Volume and Low Slice/High Volume. Click on ***Apply*** and leave with ***OK***.

Disconnect and delete superfluous modules:

- Delete the previous animation in the ***Animation Producer*** using the ***trashcan symbol***

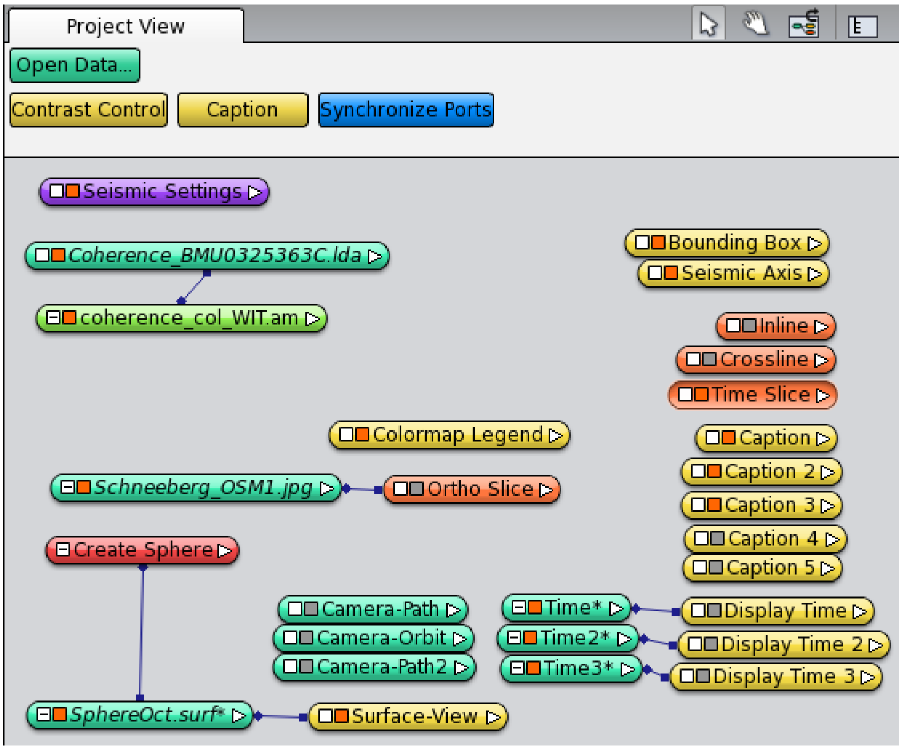
Disconnect the following modules by clicking on their white square and by drawing the blue 'rubber band' back on the module itself:

- Bounding Box module: Data
- Seismic Axis module: Data
- Seismic Settings module: DefaultSeismicColormap
- Colormap Legend module: Data
- Inline, crossline, time slice: Data & Colormap

Now you should have a ***Project View*** similar to the figure on the next page. Delete the following modules that we won't need any more:

- Coherence_BMU0325363C.lda module
- Coherence_col_WIT.am
- Schneeberg_OSM1.jpg & Ortho Slice modules

- Create Sphere, SphereOct.surf & Surface-View modules
- Camera Path modules
- Caption 3: map content…



Load and connect new colormap and data

Start by loading the suitable colormap ***Cohmig_VolRen_orange.am*** via Open Data… Define it as the ***DefaultSeismicColormap*** of the ***Seismic Settings module*** and set its upper and lower limits to ± 0.03 respectively.

In the next step, load ***Cohmig_BMU0325363C.lda***. You will find that it has automatically connected to ***Cohmig_VolRen_orange.am*** as ***SharedColormap*** and that it has taken over the correct colormap limits.

Establish new data connections with ***Cohmig_BMU0325363C.lda*** for the following modules:

- Bounding Box
- Seismic Axis
- Inline, Crossline, Time Slice

Note that connecting the ***Data port*** of the slices has automatically built a connection with ***Cohmig_VolRen_orange.am***. Colormap limits have been taken over as well.

However, we will change the colormap limits for the slices later in the Animation Producer.

Adjustments

Now it is necessary to adjust some module properties, such as the colormap legend and the captions.

Colormap Legend: change ***Custom Text*** line to -0.02/-0.02 -0.01/-0.01 0/0 0.01/0.01 0.02/0.02

Captions:
Caption 2: Change “Time Slice / map view from top” to ***“Animated time slice”***
Caption 4: Change “Moving inline slice” to ***“Animated inline slice”***
Caption 5: Change “Moving crossline slice” to ***“Animated crossline slice”***

At this point, you may already create two additional captions by duplicating one of the captions above. Select one of captions and click Ctrl-C, followed by a Ctrl-V. In the new Caption, which appears, write:

“Volume rendering of migrated coherence” and **“Inline 390 & Crossline 420”**

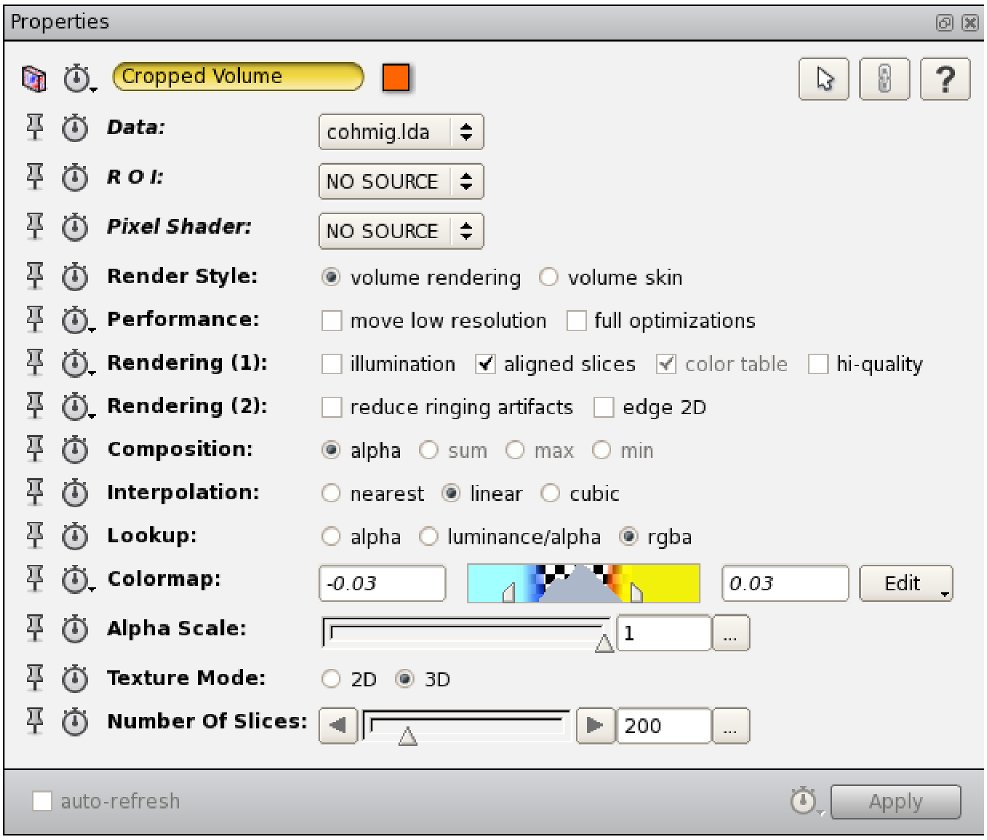
Volume rendering & ROI Box

The last missing puzzle piece for starting the animation is the ***Cropped Volume module***. For now, switch off all slice modules to have a good view of the Bounding Box and the contained volume.

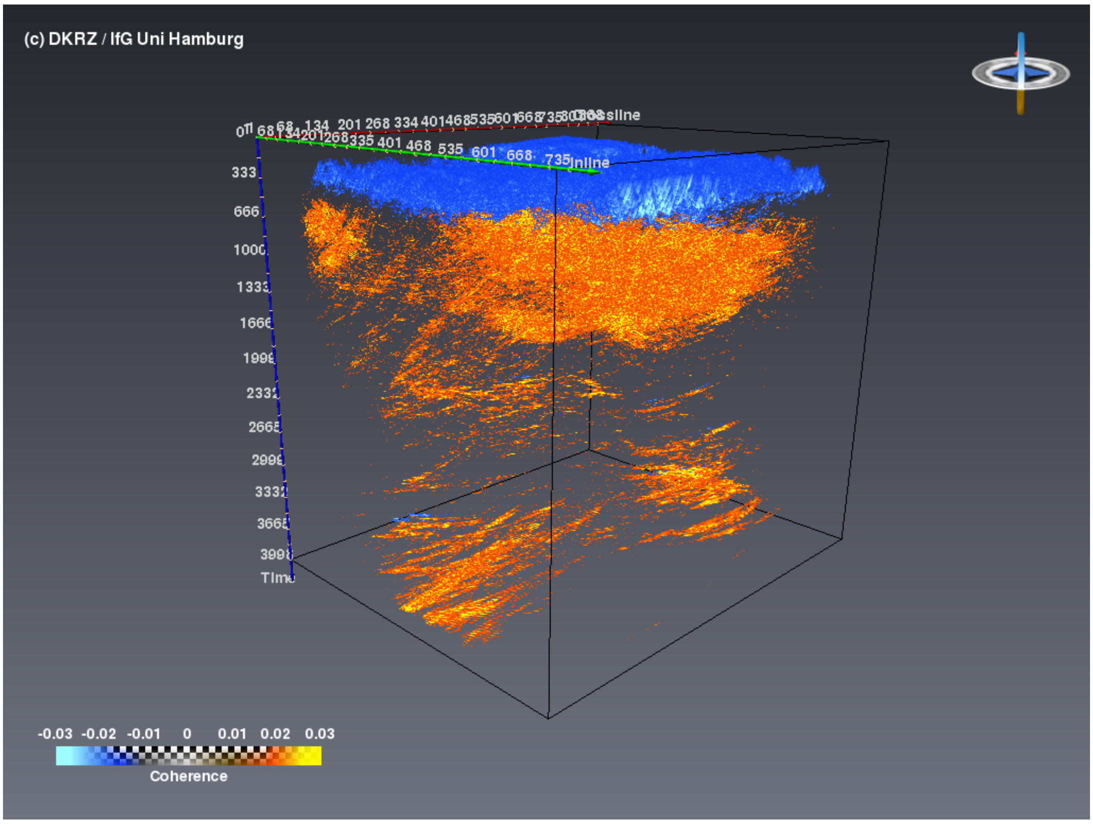
You create the Cropped Volume module by right-clicking on the ***Cohmig_BMU0325363C.lda*** module ➡ Display ➡ Cropped Volume ➡ Create

It is automatically connected to the colormap but its colormap limits are set to the data min and max values. We adjust those by hand to ± 0.03 and uncheck ***Auto adjust range*** in the Colormap port: Edit ➡ Options ➡ Auto adjust range to prevent automatic resetting of the colormap limits.
So far, nothing is visible inside the Bounding Box. To change this, go to the ***Render Style*** port in the ***Cropped Volume Properties*** and switch from volume skin to ***volume rendering***.

With the other port entries on default values, your ***Cropped Volume Properties*** window should look like the next figure. These settings produce a first rough rendering of the migrated coherence data. However, the data still looks very shiny and somehow noisy. Zooming in reveals that we are still not seeing surfaces but individual ‘strips’ of data.
To improve the rendering, you should increase the ***Number of Slices*** port to a value of at least ***500*** slices. ***1000*** slices gives an even better result but slows down the interactive rendering. In a next step, decrease the ***Alpha Scale*** value for the overall transparency of the data cube to a value of about 0.4. You may also try activating ***hi-quality*** in the ***Rendering (1)*** port.



Now features of relatively high amplitudes can be seen in the data volume. However, the negative coherence amplitudes on top (rendered in blue) are an undesirable feature and obstruct the view of the relevant data part (see picture below).

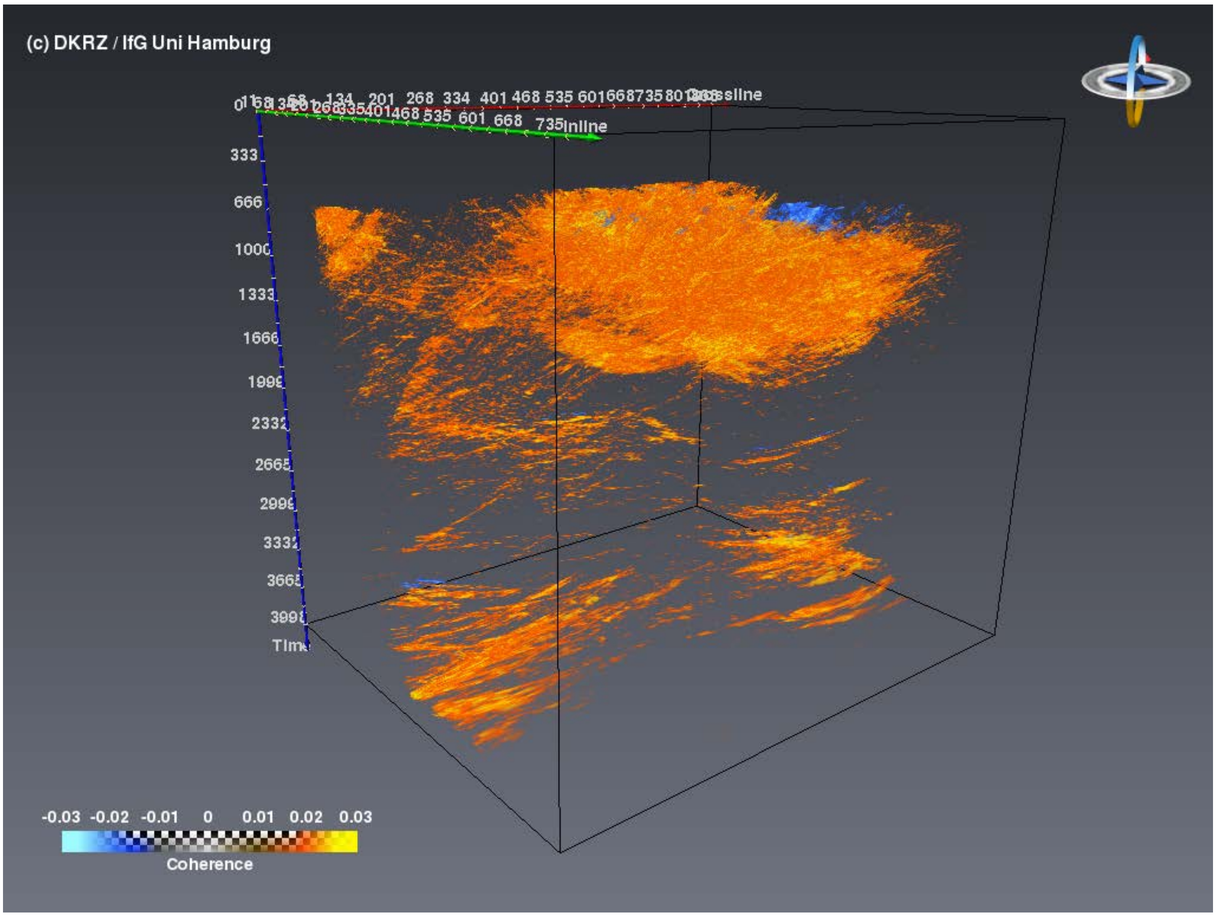


ROI Box Seismic

To display only the coherence amplitudes we are interested in, we define a ***ROI (Region of Interest) Box***: Right-click on the ***Cohmig_BMU0325363C.lda*** module ➡ Display ➡ ROI Box Seismic ➡ Create

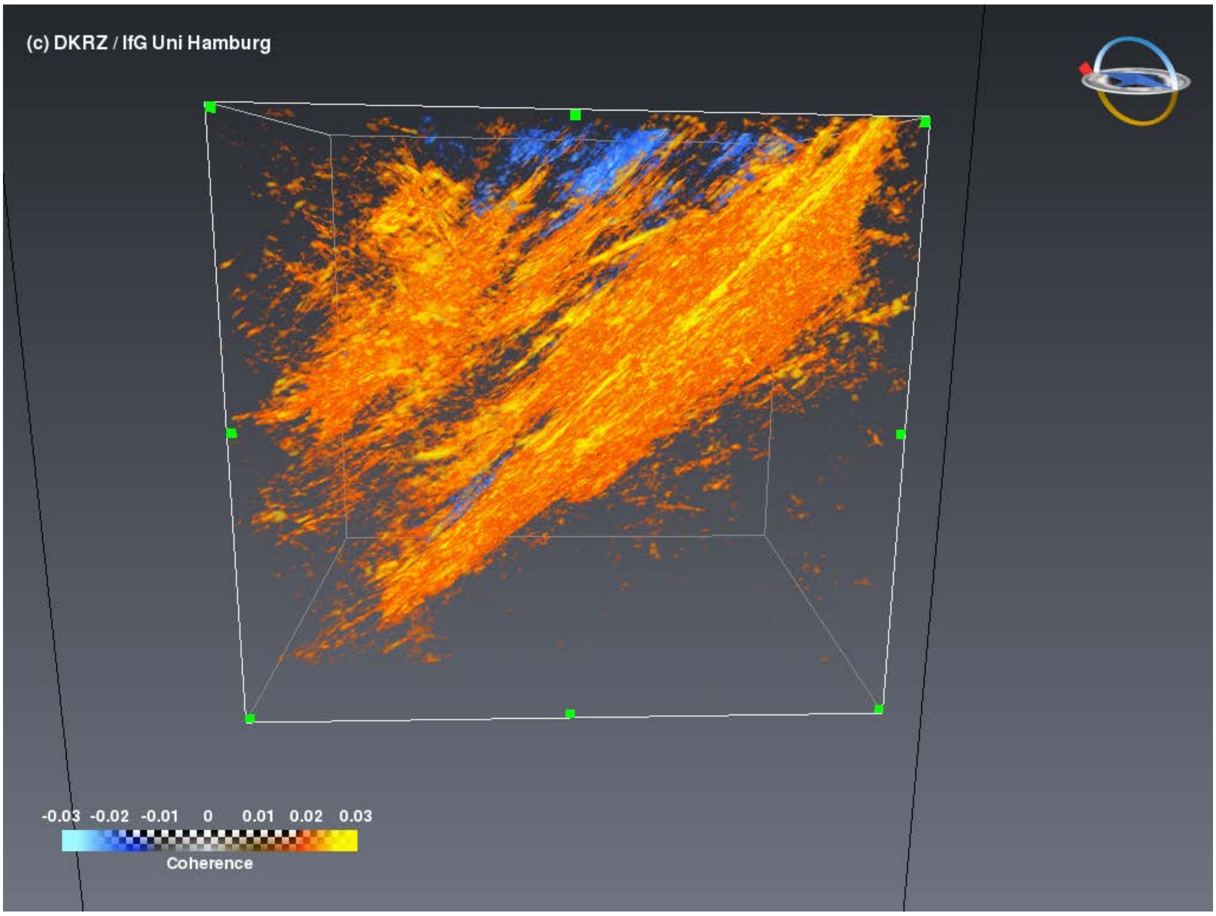
This has created an active ROI Box with green adjustable corners with the dimensions of the data cube and Bounding Box. Link it to a visualization element by, e.g., clicking on the white square of the ***Cropped Volume module*** and choosing the ***ROI port***. Then connect the dark blue rubber band to the ***ROI Box Seismic*** module.

To 'clip away' the negative blue amplitudes on top either reshape the ROI Box by hand in ***interact mode*** or enter the value ***650*** in the third field of the ***Minimum port*** in the ***ROI Box Seismic*** module. Having switched off the ***ROI Box Seismic***, you should see the following data view:



Before this subset of the data is used for creating the animation, try exploring the data with a even smaller ROI Box, which you shift freely through the data volume. Have a look at the next figure for an example with the ROI Box Seismic switched on.

Resize the ROI Box Seismic now to all inline and crossline slices and time slices starting from 650 on. Also connect the slices to the ROI Box.



Start animating: 360 ° rotations

As an initial view for your new animation you may reuse the camera setting of your your first movie or you can set up a new view. To position your data in the middle of the viewer and concentrically around the center of rotation, switch on the ***Rotation axes***:

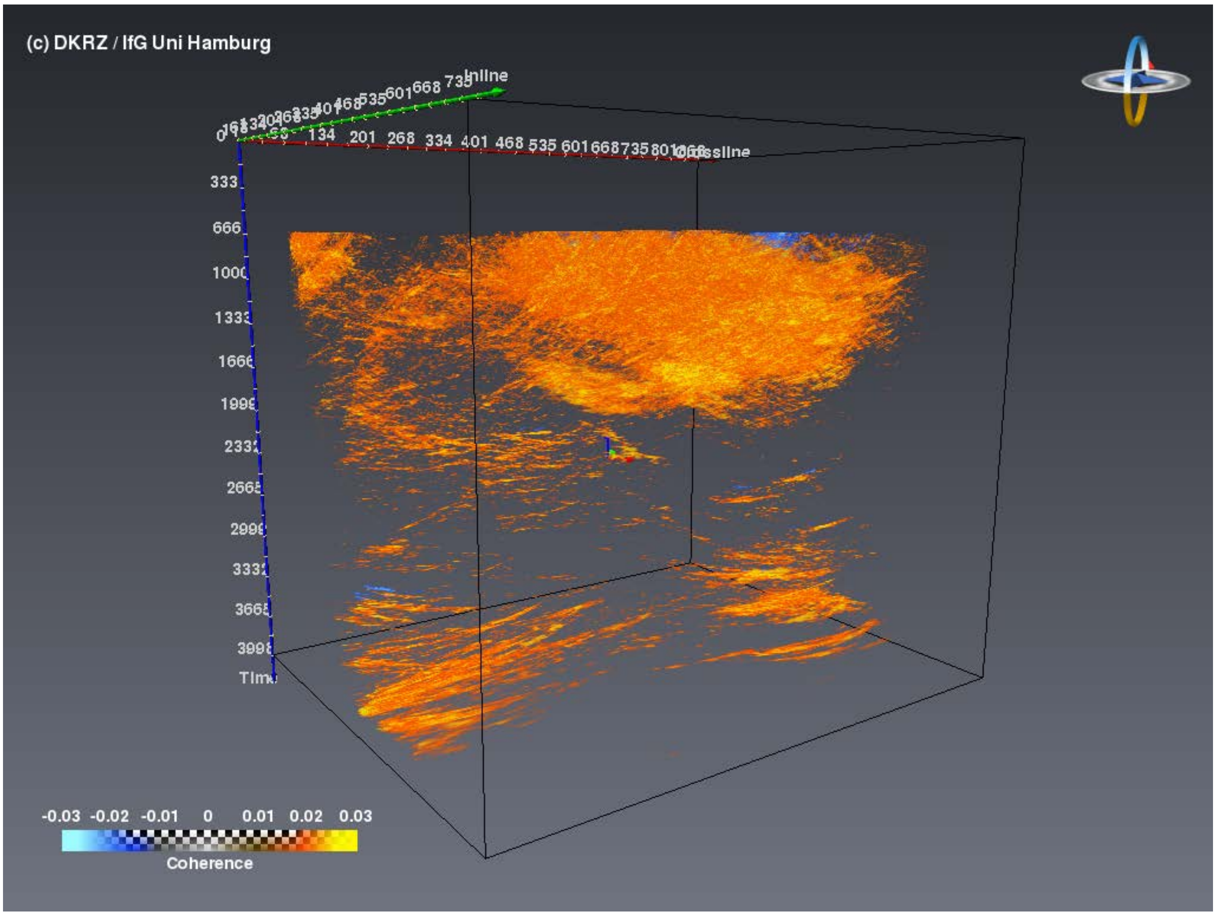
Right-click in the ***Viewer*** ➞ Preferences ➞ Rotation axes

Look at the next figure to see the rotation axes symbol for our initial camera position. You will find this camera position in part A.5 of the appendix. When done, hide the rotation axes again.

At 00:00.000 s, switch ***on*** the Cropped Volume with ***visibility in viewer 0*** in the Animation Producer and also set the correct ***Colormap: source*** and ***min*** and ***max*** values. In addition, make the ***Colormap Legend*** and ***Caption 3*** for the volume rendering visible. If you want to make 100% sure that jumping to the beginning of the animation gives you the correct starting view, click on the ***+ Add special event*** button of the Animation Producer and choose ***Add TCL Command***. Hover over the ***TCL house symbol*** between time line and keyframes and copy the initial Camera settings from the text file into the input window. This will always reset the camera to the correct start view at the beginning of your animation.

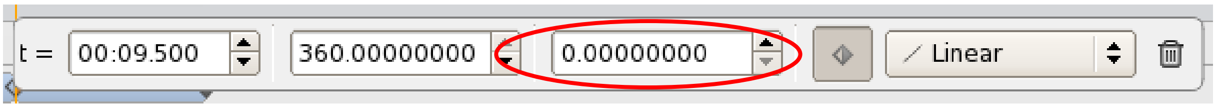
Also set the ROI Box Seismic time minimum value of ***650*** as a keyframe in the Animation Producer and change the ***interpolation curve*** to ***none.***

After two seconds the first 360° rotation out of two shall begin. Each rotation should have a duration of 7.5 seconds as for the previous tutorial movie. To avoid a pause between the two rotations and to have a smooth transition, we will use a ***Make transition keyframe***.



With the master time slider at 00:00.000 s, press the ***recompute*** button in the ***Camera-Orbit Action*** port to recompute the center of rotation in the viewer. Also add a keyframe for the ***Action: recompute port*** in the Animation Producer.
At 00:02.000 s the animation should start: switch ***on*** the ***Camera-Orbit - visibility in viewer 0*** and set ***Time: value*** to ***0***. At 00:09.500 s set a keyframe for ***Time: value*** **360**. Hover over this new ***Time: value*** keyframe until you see the semi-transparent input field. To the right of the field for entering the time value (degree of rotation) you find the symbol for the ***transition keyframe***.

If you click on this symbol, the previously normal keyframe is transformed into a transition keyframe and an extra field for the time value appears:



This second value field should be set to 0.00 because we want the second rotation to be from 0° to 360°. Complete the ***Camera-Orbit*** at 00:17.000 s by a ***time: value*** keyframe of ***360***. Play the animation with undocked viewer at beamer size 1024 x 768 to check if the two rotations are performed correctly and if the Bounding Box does not interfere with the colormap legend.

Animated inline

To animate the inline slice we first need to rotate the Bounding Box to a front view of the inline axis. For this purpose we will use the ***Camera-Orbit*** module again. Simultaneously with the onset of this rotation, which should last 1.5 s, the volume rendering ***Caption 3*** should be switched off. We suggest the following procedure:

- Change the ***time: value*** keyframe for 360° at 00:17.000 s into a ***transition keyframe*** as well with transition from 360 to 0°.
- Switch ***off visibility in viewer 0*** of ***Caption 3*** at 00:17.000 s.
- At 00:18.500 s, add a ***time: value*** keyframe with value ***35.65*** and toggle the ***Camera-Orbit off***.
- Switch ***off*** the ***Cropped Volume*** as well.

Next we should switch on the ***transparent*** inline slice. Be sure that the ***Transparency port*** is set to ***binary*** and that the slice frame is switched ***off****.* We suggest doing this 0.250 s after the end of the ***Camera-Orbit***, at 00:18.750 s. At this time the ***Slice Number port*** should be set to 1. For completion, also set ***Colormap: source***, and ***min*** and ***max*** values to **± *0.025***. This allows us to see more features in the data than for **± *0.03***. The length of the ***Colormap Legend*** in the viewer will be adjusted automatically. In addition, switch on the respective caption (***Caption 4***).

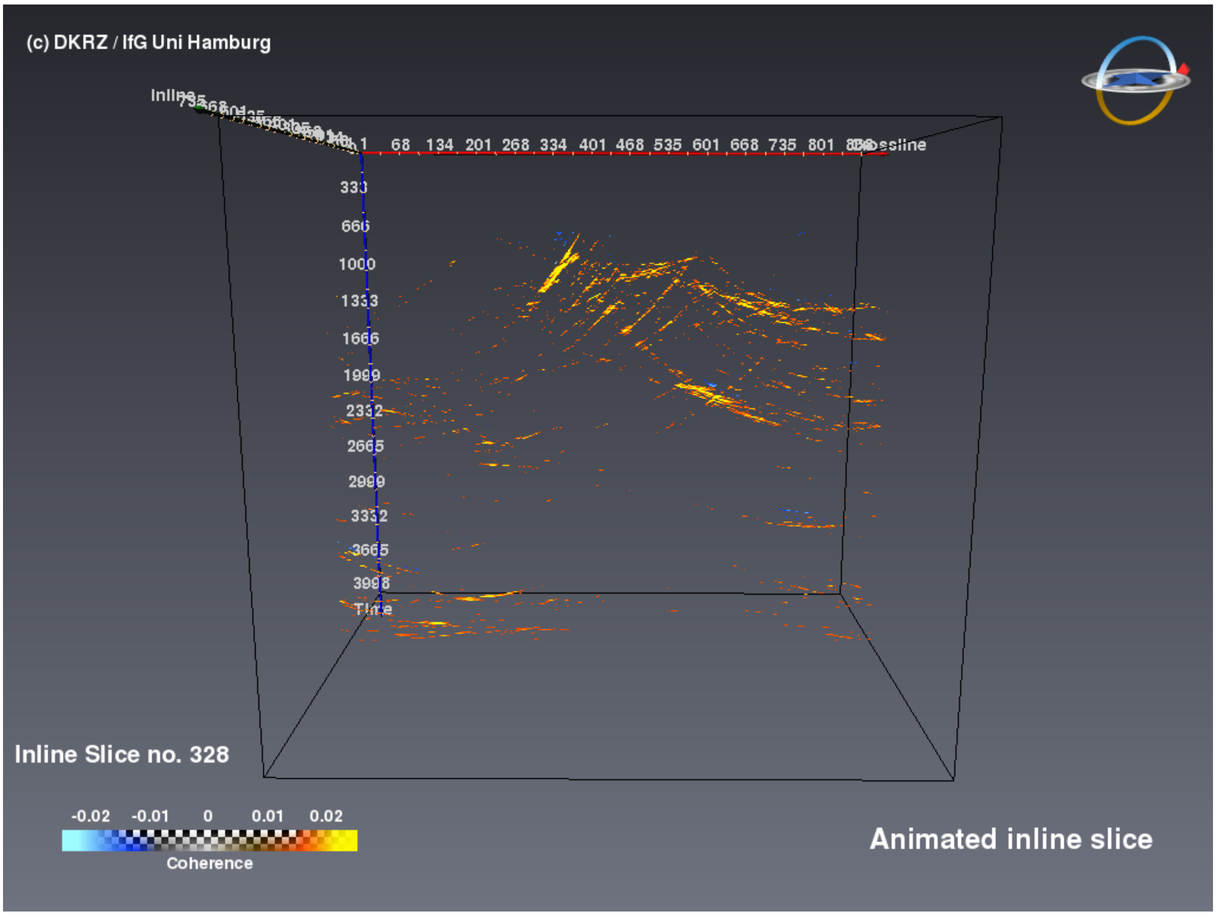
As for the first video we want the inlines to be animated with 16 slices per second, thus the whole inline passage takes 44.5 s. Starting at 00.18.750 s, inline 712 should be displayed at 01:03.250 s and then switched ***off***. Do not forget to extend the animation time via ***More options***. Likewise, ***Caption 4*** should be switched ***off.***

For displaying the slice number we employ the same method as in the first tutorial project. We simply reuse the existing ***Time module*** and ***Display Time module*** in the Animation Producer. If necessary, adjust the screen ***position*** of the ***Display Time module***. The next figure shows one exemplary transparent inline slice.

Animated crossline

Next the Bounding Box should be rotated about 90° anti-clockwise to get a front view of the crossline slice.

To do this, insert a ***Camera-Orbit – Action: recompute*** keyframe before the new rotation shall take place (e.g. at 01:03.000 s). Make the Camera-Orbit ***visible*** at 01:03.500 s and set ***Camera-Orbit – time: value*** to ***360*** at this time. We want to perform the 90° rotation in 2 seconds, thus ***Camera-Orbit – time: value*** should be ***270*** at 01:05.500 s. At this time, switch the ***Camera-Orbit visibility off***.



After a brief pause of 0.5 seconds the crossline animation shall start, with 16 slices per second as for the inline. This gives a length of 54.1875 seconds, starting at 01:06.00 s. As the 02:00.000 s mark is close, let us use 02:00.000 s as the end time of the crossline animation. The animation steps are:

- Switch ***on Caption 5*** at 01:06.000 s and ***off*** at 02:00.000 s.

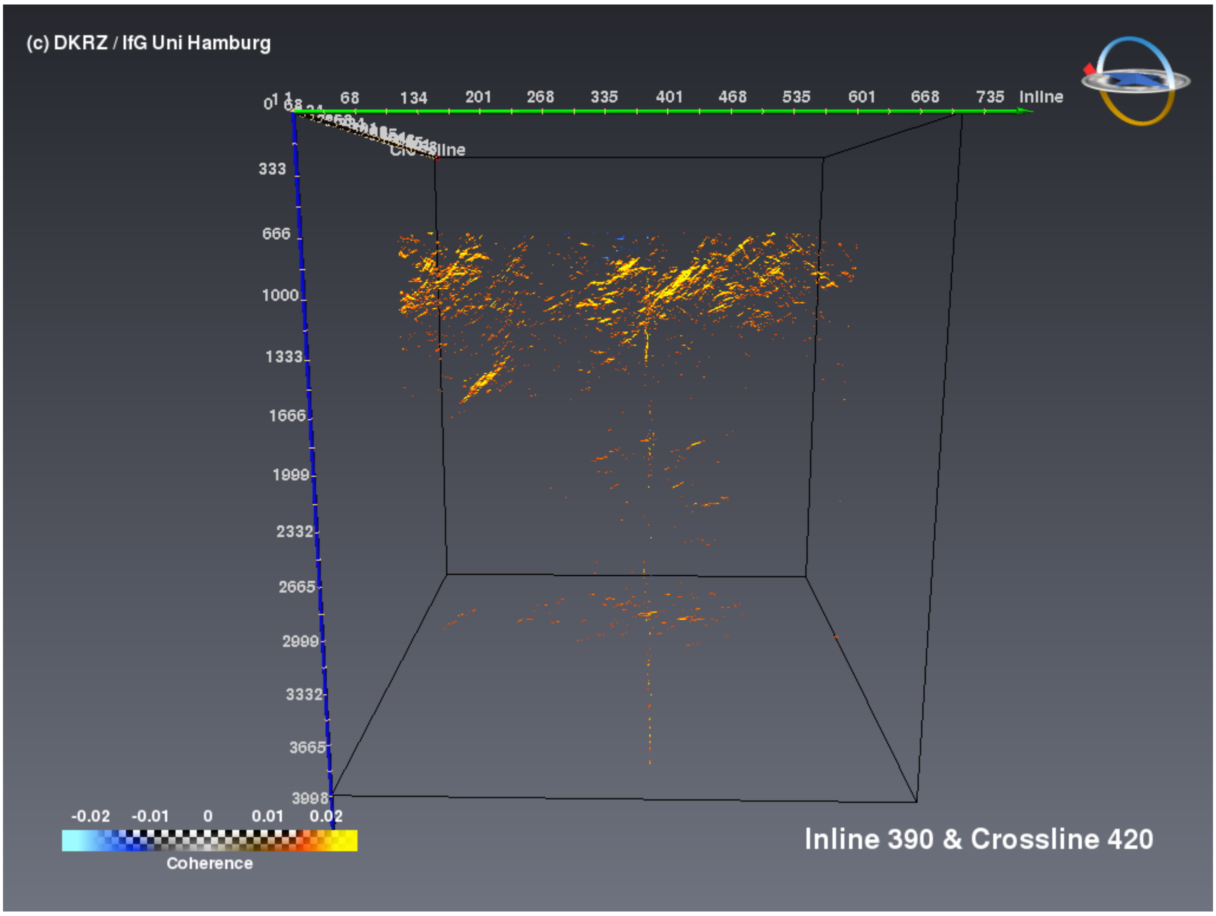
- Switch ***on visibility*** of ***crossline*** at at 01:06.000 s; ***Slice Number: value*** 1; ***Colormap: source*** Cohmig_VolRen_orange.am, ***Colormap: min/max*** ± 0.025. Frame off, ***Transparency***: Binary. At 02:00.000 s; ***Slice Number: value*** 868 and ***visibility off***

- ***Time 2*** module (connected to ***Crossline*** module): Add keyframes for ***Time: value*** with ***value 1*** at 00:00.000 s and at 01:06.000 s.

- ***Display Time 2*** module: ***on*** at 01:06.000 s and **off** at 02:00.000 s. Adjust **x position** to e.g. 10.

Inline 390 & Crossline 420

Next we would like to show a 360° rotation of inline 390 and crossline 420. Shortly after the animated crossline, e.g. at 02:00.500 s, switch ***on inline*** and ***crossline***. Note that modules ***Time*** and ***Time 2*** automatically display the correct time values / slice numbers of ***390*** and ***420***.Next, switch ***on Caption 6***. This should give you the view on the following page.



To achieve the rotation, ***recompute*** the ***Camera-Orbit***, e.g. at 02:00.500 s. Then switch the module ***on*** at 02:01.000 s and ***Time: value*** to 0.

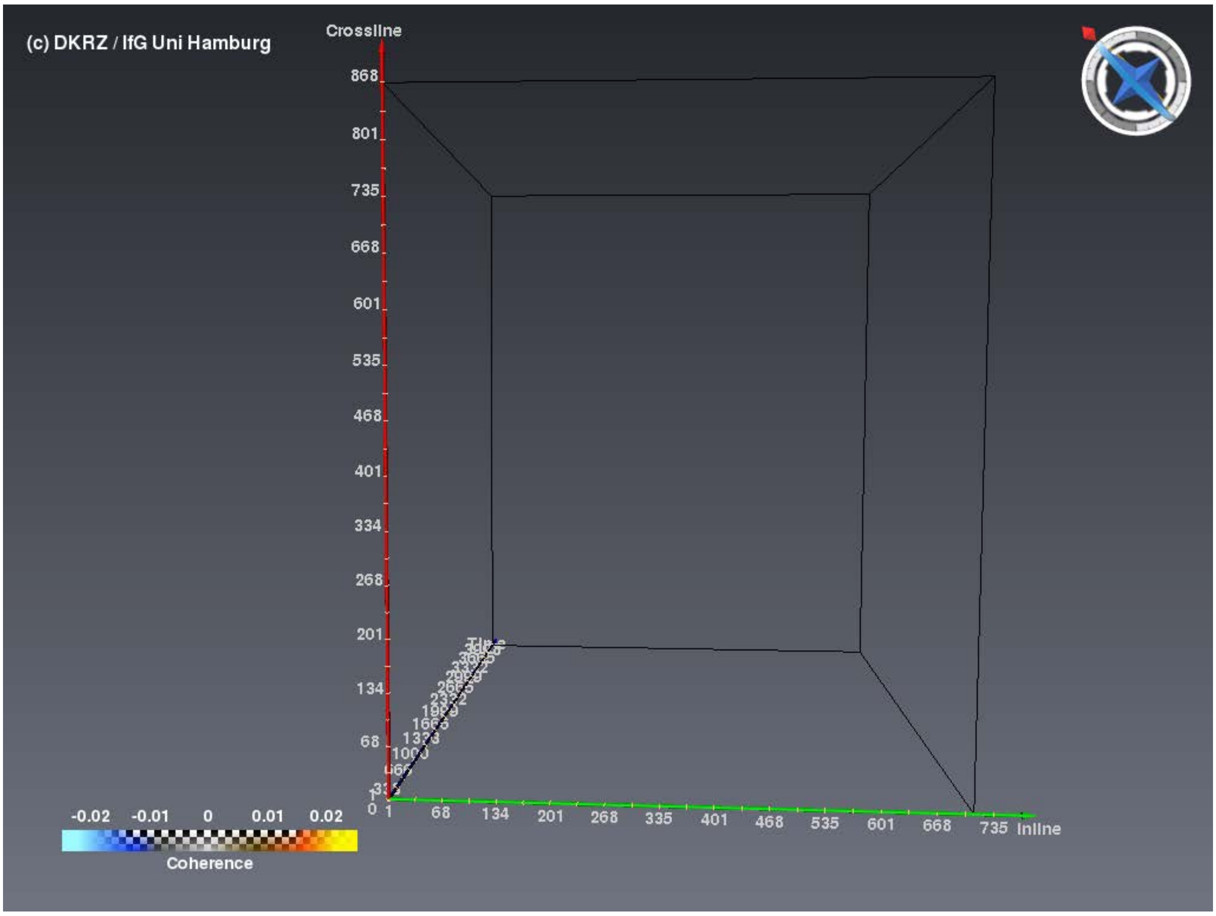
7.5 seconds later, at 02:08.500 s set ***Camera-Orbit Time: value*** to 360 and switch it ***off***. Shortly afterwards, e.g. at 02:09.000 s, the ***slices*** and ***Caption 6*** should be switched ***off*** as well.

Camera Path

Finally, we would like to see the animated transparent time slices (600 to 2500 ms) in a top view. For this purpose, we first have to create a new ***Camera Path*** from the current camera position to the top view depicted in the following figure. Have your camera orientations ready in a text file!

In the ***Project View,*** right-click ➡ Create Object… ➡ Animations And Scripts ➡ Camera Path ➡ Create

In the ***Properties*** of this module click on the ***Camera Path Editor*** to see the extended ***Camera Path Properties*** with steering buttons for the ***Camera Path*** and ***Viewer 5*** for the trajectory. As for the previous movie, copy your camera settings for the current camera position into the console and hit enter. Then click on the add button to create the first keyframe. Repeat this procedure for the time slice start camera settings. Close the ***extended Properties*** by clicking once more on the symbol . The next time you save your project, click on the ***Auto save*** option for the ***Camera Path***. Your ***Camera Path*** thus gets saved in the folder **<your project file name>-files**, which prevents it from being overwritten by other projects.



Animate the ***Camera Path*** by making it ***visible*** at 02:09.000 s and by setting its ***Time: value*** port to 0. At 02:11.000 s, ***Time: value*** reaches its maximum value, e.g. 10, and the ***Camera Path*** should be switched ***off***.

Animated time slice

Half a second later, the inline slice animation should start. Like for the previous movie, 30 ms of data per second should be displayed, giving the time slice animation a length of 01:02.333 s. These are the necessary steps:

- Extend the animation end time to at least 03:14.000 s.
- Set a ***ROI Box Seismic*** minimum time value keyframe of ***600*** and choose ***interpolation: none***
- Switch ***on Caption 2*** at 02:11.500 s and ***off*** at 03:13.833 s. If necessary, adjust its position to e.g. **y value** 33.
- Switch ***on visibility*** of ***time slice*** at 02:11.500 s; ***Slice Number: value*** 300 (equals 600 ms); ***Colormap: source*** Cohmig_VolRen_orange.am, ***Colormap: min/max*** ± 0.025. Frame off, ***Transparency***: Binary. At 03:13.833 s; ***Slice Number: value*** 1250 (equals 2500 ms) and ***visibility off***.
- ***Time 3*** module (connected to ***Time Slice*** module): Add keyframes for ***Time: value*** with ***value 600*** at 00:00.000 s and at 02:11.500 s.
- ***Display Time 3 module***: ***on*** at 02:11.500 s and **off** at 03:13.833 s. Adjust **x position** to e.g. 10.

When done, run your animation. If everything works well, export the video using the same settings as in the ***Movie creation*** section of the previous chapter.

A Cheat sheets

Here you find example camera positions and map adjustments which you can use in case you are in a hurry or too lazy to make the adjustments yourself.

A.1 Top view towards north

viewer setCameraOrientation -0.886862 0.396482 -0.237229 0.107493

viewer setCameraPosition 10317.9 3353.74 35087.4

viewer setCameraFocalDistance 30968.9

viewer setCameraNearDistance 22112.8

viewer setCameraFarDistance 36930.4

viewer setCameraType perspective

viewer setCameraHeightAngle 44.5904

A.2 Transform Editor absolute values for map

Translation:	8008.87	366.541	4039.43
Rotation:	180.48 degrees		
	0.924897	-0.38021	0.00229198
Scale Factor:	13.09	12.8347	0.993925

A.3 Start view for moving inline slice

viewer setCameraOrientation 0.680921 0.444933 0.581705 1.63994

viewer setCameraPosition 30994.9 -10979.7 11916.9

viewer setCameraFocalDistance 25686.8

viewer setCameraNearDistance 16128.4

viewer setCameraFarDistance 35422.6

viewer setCameraType perspective

viewer setCameraHeightAngle 44.5904

A.4 Top view for moving time slices

viewer setCameraOrientation -0.136682 0.0061768 -0.990596 0.782671

viewer setCameraPosition 10066.9 2274.15 35201.1

viewer setCameraFocalDistance 30968.9

viewer setCameraNearDistance 22112.8

viewer setCameraFarDistance 36930.4

viewer setCameraType perspective

viewer setCameraHeightAngle 44.5904

A.5 Start view for volume rendering movie

viewer setCameraOrientation -0.990249 -0.0864768 -0.109223 4.9136

viewer setCameraPosition 12397.5 -23388.5 10291.6

viewer setCameraFocalDistance 25420.6

viewer setCameraNearDistance 14633.7

viewer setCameraFarDistance 34947.1

viewer setCameraType perspective

viewer setCameraHeightAngle 44.5904

A.6 Start view for transparent inline slice

viewer setCameraOrientation 0.833405 0.346541 0.430517 1.54126

viewer setCameraPosition 25700.5 -16411.6 10291.6

viewer setCameraFocalDistance 25420.6

viewer setCameraNearDistance 17172.6

viewer setCameraFarDistance 31791.5

viewer setCameraType perspective

viewer setCameraHeightAngle 44.5904

A.7 End view of inline 390 & crossline 420 rotation

viewer setCameraOrientation 0.836349 -0.345603 -0.425534 1.5362

viewer setCameraPosition -9470.72 -16554.3 10291.6

viewer setCameraFocalDistance 25420.6

viewer setCameraNearDistance 17283.5

viewer setCameraFarDistance 33839.3

viewer setCameraType perspective

viewer setCameraHeightAngle 44.5904

A.8 Start view of time slice

viewer setCameraOrientation -0.0554944 0.101722 -0.993264 0.78327

viewer setCameraPosition 9095.85 1845.4 31750.7

viewer setCameraFocalDistance 26777.1

viewer setCameraNearDistance 18770.7

viewer setCameraFarDistance 33290.9

viewer setCameraType perspective

viewer setCameraHeightAngle 44.5904

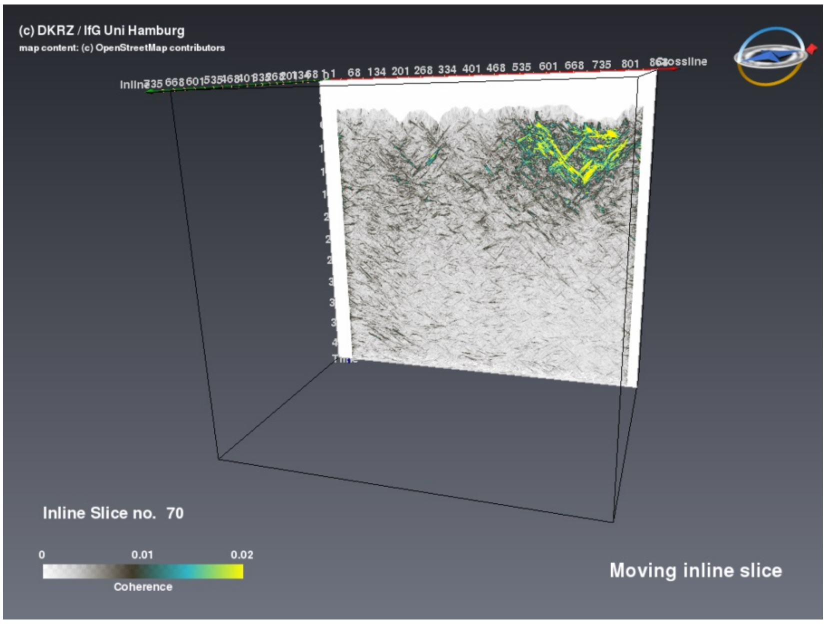
B More information on the coherence data cube

If you would like to learn more about the significance of the coherence data content you have come to the right section.

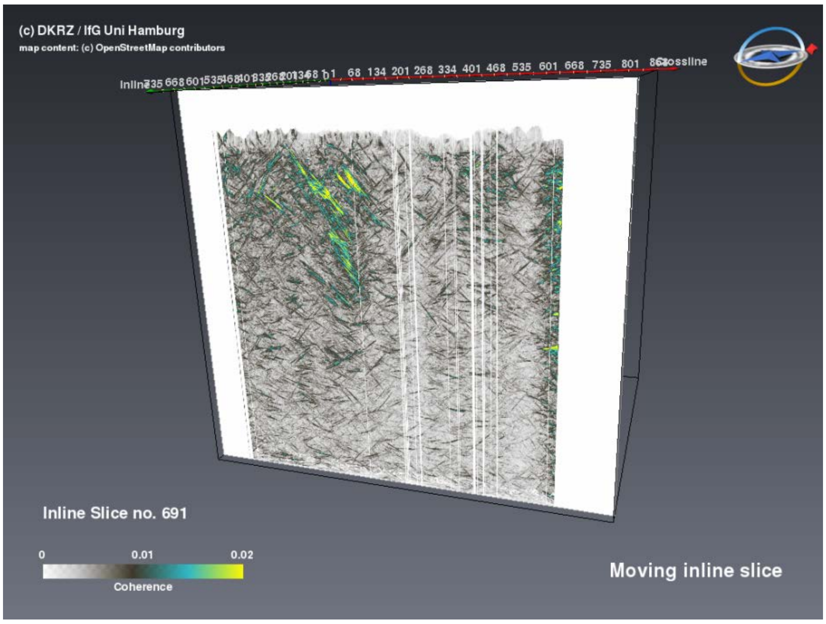
As mentioned in the introductory section, the 3D data set that you have been working with does not show the amplitudes of seismic traces but their coherence, which is basically trace similarity. For hard rock and crystalline environments and in the absence of tailor-made specialized processing techniques, coherence may be a better indicator of discontinuities than unmigrated or even time-migrated seismic reflection data. At least this is what we experienced when trying to visualize the unmigrated seismic data, coherence data, migrated coherence and migrated seismic amplitude data. Natural faults and fracture zones were best visible in migrated coherence data.

In the coherence dataset used for this tutorial, the coherence values are comparatively low, varying from 0 to 0.02 only. When speaking of 'high' coherence in this section we mean relatively high values of 0.015 or more.

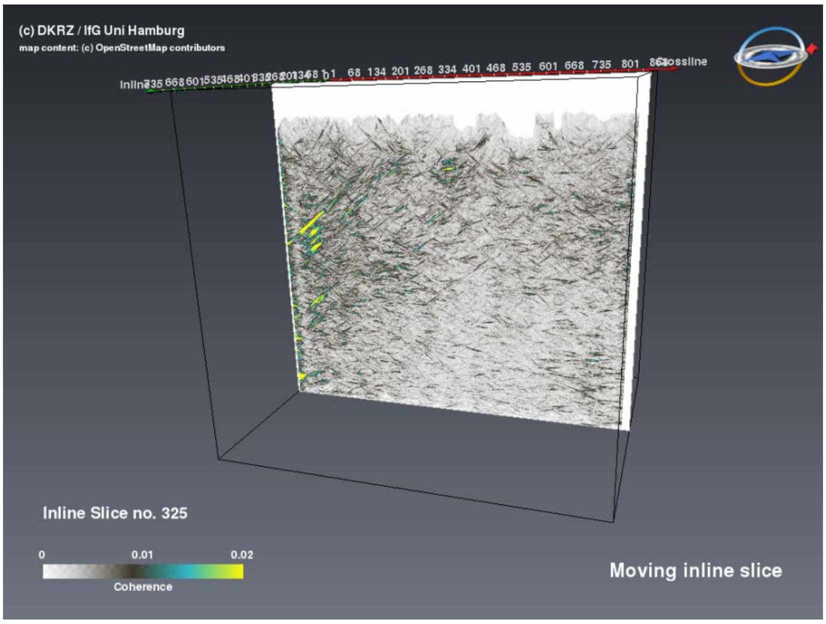
Looking closely at the animated slices, the first high coherence values appear in the northern part of the data cube, in the area around the towns ***Zschocken*** and ***Hartenstein***, and are visible on the first 180 inline slices. This is not surprising because this area has a long mining history. If you visit the website (available in German only) http://www.bergbau.sachsen.de/8159.html about mining in Saxony and enter either of the two town names into the 'Hohlraumkarte' (cavity map) of the sächsisches Bergamt, you will see a large area with known cavities northeast of Hartenstein.



The same applies to a more or less oval area stretching from south of Schneeberg to the northeast, covering Bad Schlema and extending almost to Lößnitz. However, the high amplitudes for the cavities in the Schneeberg area only stick out on the last visible inline slices between slices 670 and 700.



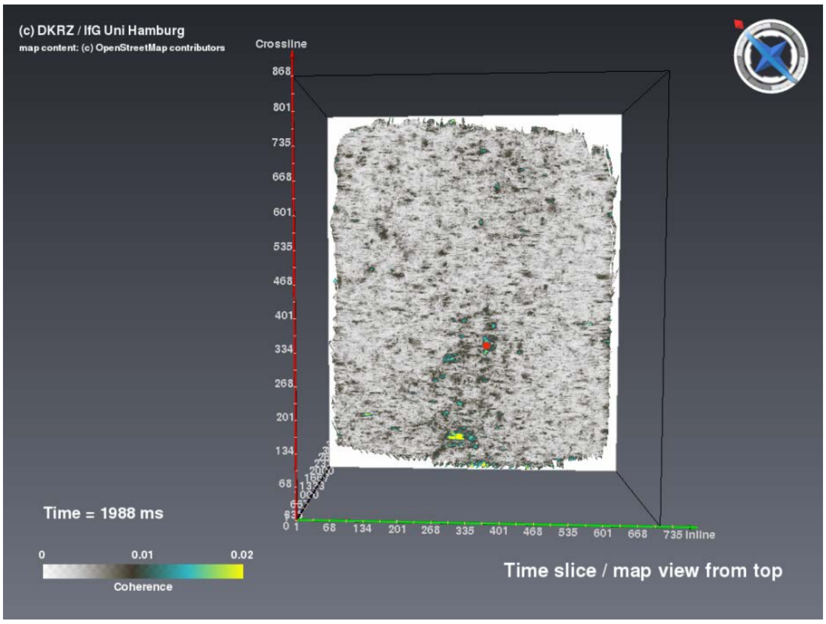
Beyond inline slice 280, in the area around Griesbach, we also observe higher coherence amplitudes, although they appear weaker and more isolated than in the Hartenstein area. Their strike seems to be from northeast to southwest and they may outline geological faults.



Due to the processing technique applied, the crossline slices show less meaningful features than the inline slices and cannot be interpreted reasonably without the respective inline slices at hand. That is because we do not see continuous structures extending into the crossline direction. We only observe ‘spots’ of high amplitudes. Nevertheless these spots help us to get a feeling for the extent of structures into the second spatial dimension.

At the intersection point of inline 390 and crossline 310, a borehole was been drilled and one would expected to see higher coherence values in this area. This is partly the case. The best viewing directions are from the south and from the west. See page 26 for a view of these two intersecting slices from the southwest.

We see a first higher amplitude spot on crossline 310 at 1000 ms and more prominent features with amplitudes of up to 0.02 in the time window of approx.1800 to 2200ms. This is also visible in animated time slices and in the following figure.



We hope that this tutorial has been helpful. In case you have any questions or comments please contact us.