VisIVOServer 2.0

User Guide

Introduction

Authors: Becciani U., Caniglia G., Costa A., Gheller C., Krokos M., Massimino P.

VisIVO Server is a suite of software tools for creating customized views of 3D renderings from astrophysical data tables. These tools are founded on the **VisIVO Desktop** functionality (visivo.oact.inaf.it) and support the most popular Linux based platforms (e.g. www.ubuntu.com). Their defining characteristic is that no fixed limits are prescribed regarding the dimensionality of data tables input for processing, thus supporting very large scale datasets.

VisIVO Server websites are currently hosted by the University of Portsmouth, UK (visivo.port.ac.uk), the INAF Astrophysical Observatory of Catania, Italy (visivo.oact.inaf.it) and in the near future by CINECA, Italy (visivo.cineca.it). These web sites offer data management functionality for registered users; datasets can be uploaded for temporary storage and processing for a period of up to two months. The sites can also be utilized through anonymous access in which case datasets can be uploaded and stored for a maximum of four days; to maximize available resources a limited dimensionality is only supported.

Assuming that datasets are uploaded, users are typically presented with tree-like structures (for easy data navigation) containing pointers to **files**, **tables**, **volumes** as well as **visuals**.

Files point to single, or possibly several (for distributed datasets), astrophysical data tables;
Tables are highly-efficient internal VisIVO Server data representations; they are typically produced from importing datasets uploaded by users using VisIVO Importer (see below);
Volumes are internal VisIVO Server data representations; they are produced either from direct importing of user datasets or by performing operations on already existing tables;
Visuals are collections of highly-customized, user-produced views of 3D renderings of volumes.

VisIVO Server consists of three core components: **VisIVO Importer**, **VisIVO Filter** and **VisIVO Viewer** respectively. Their functionality and usage is described in the following sections.

To create customized views of 3D renderings from astrophysical data tables, a two-stage process is employed. First, VisIVO Importer is utilized to convert user datasets into **VisIVO Binary Tables** (VBTs). Then, VisIVO Viewer is invoked to display customized views of 3D renderings. As an example, consider displaying views from only three columns of an astrophysical data table supplied in ascii form, say col_1, col_2 and col_3, by using the commands

```
  VisIVOImporter --fformat ascii UserDataSet.txt

  VisIVOViewer -x col_1 -y col_2 -z col_3 --scale --glyphs pixel VBT.bin
```

VisIVOServer is distributed with GPL V.2 License for NON COMMERCIAL use. VisIVOServer is hosted by sourceforge https://sourceforge.net/projects/visivoserver/ and its source code is downloadable via svn:
svn co https://visivoserver.svn.sourceforge.net/svnroot/visivoserver/branches/1.2 visivoserver
Disclaimer: user data integrity is never warranted.

VisIVO BINARY TABLE

A VisIVO Binary Table (VBT) is a highly-efficient data representation used by VisIVO Server internally. A VBT is realized through a header file (extension **.bin.head**) containing all necessary metadata, and a raw data file (extension **.bin**) storing actual data values. For example, the header may contain information regarding the overall number of fields and number of points for each field (for point datasets) or the number of cells and relevant mesh sizes (for volume datasets). The raw data file is typically a sequence of values, e.g. all X followed by all Y values.

Header

The header file contains the following fields:

```
float | double
n1
n2 [ GeoX GeoY GeoZ DX DY DZ ]
little | big
X
Y
Z
Vx
Vy
Vz
```

- **float | double** is the data type of the storage variables used;
- **n1** denotes the number of columns (fields) in the VBT;
- **n2** denotes the number of rows in the VBT;

- ***Geo*****X GeoY GeoZ DX DY DZ** are employed only if the VBT represents volumetric datasets. In that case GeoX, GeoY and GeoZ represent the mesh geometry, while DX, DY and DZ represent the x, y and z size of volumetric cells.
- **little | big** denotes the endianism employed in the VBT. After this field there exist n1 rows that indicate the VBT columns as positions (X, Y, Z) and velocities (Vx, Vy, Vz).

Raw

The binary file is simply a sequence of *n1*n2* values. In the example shown in section 2.1 all X values, then all Y values and so on.

Note:

- **n1** represents the number of columns (fields) in the VBT (e.g. 6);
- **n2** represents the number of elements of each field in the VBT (e.g. 262144);
- **GeoX GeoY GeoZ** represent the number of the volumetric cells in each dimension of the mesh size (used only for Volumes) (e.g. 64 64 64);
- **DX DY DZ** represent the size of each cell (used only for Volumes) (e.g. 1.0 1.0 1.0)

VisIVOServer

VisIVO Viewer

VisIVOViewer creates views from the input data file. The input data file must be in the Internal Binary format. The input data file must fit the available RAM. VisIVOViewer produces five png images. The first four default images are generated with the following fixed values:

Azimuth	0	90	0	45
Elevation	0	0	90	45
Zoom	1			

The last image is given with values fixed by the user.

NOTE: the camera of the Viewer at the default position (Azimuth=0 Elevation=0) is looking the box from the top (z plane).

GENERAL SYNTAX

```
VisIVOViewer --help
```

produces a general help

```
VisIVOViewer <options>  inputFile
```

produces images from the input file.

The alternative command is:

```
VisIVOViewer parameterFile
```

GENERAL OPTIONAL OPTIONS

--out [filename] Default output filenames are VisIVOServerImage0.png, VisIVOServerImage1.png .... Note: the output name is always completed with *0.png 1.png 2.png and 3.png* if default images are produced. Anyway, the output file name (image) is completed with *.png* extension

--nodefault Default images are not created.

--cycle [filename] (optional) VisIVO viewer will produce a sequence of images reading data of azimuth elevation and zooming from the file given with this parameter. This option will prevent the production of default images while the camera position and zoom factor of the line command will be ignored. The file for the cycle must contain three ascii values for each row. Blank line or rows with less than three values will be ignored. VisIVO will produce one image for each valid row in the file. This file can be created with the tool *VisIVOUtils --op createpath* option. The output filenames will have the same root-name as those given in *--out* parameter plus a progressive number (with six digit) starting from 0 (or the value given in the --cycleoffset option), with png file extension. A new file listing all images is created. The file has the fixed name VSCycleImage#.txt in the output director, where # is the value of --cycleoffset option.

The cycle file for camera position must be of four types:
Four values for each line : Azimuth, Elevation, Zoom and Roll;
Seven values for each line : Azimuth, Elevation, Zoom , Focal Point (three values) ,and Roll;
Eight values for each line : Zoom , Camera Position (three values), Focal Point (three values) and Roll;
Ten values for each line : Azimuth, Elevation, Zoom , Camera Position (three values), Focal Point (three values) and Roll;

The keyword NULL is accepted for Camera position and/or Focal Point and/or Roll. In this case the previous camera setting is maintained. The NULL keyward cannot be used for Azimuth, elevation and Zoom.

Examples:
Azimuth, Elevation, Zoom and Roll

```
0.0     0.0     1.0  0
......
60.0   10.0   1.0  NULL
60.0   10.0   1.2  NULL
60.0   10.0   1.4  NULL
60.0   10.0   1.6  NULL
```

Azimuth, Elevation, Zoom Focal Point and Roll

```
0.0     0.0     1.0  35 35 35  0
......
60.0   10.0   1.0  20 20 20  0
60.0   10.0   1.2  20 20 20  0
60.0   10.0   1.4  20 20 20  0
60.0   10.0   1.6  20 20 20  0
```

Zoom, camera Position, Focal Point and Roll

```
1.0     35 35 200 35 35 35 0
......
1.0     35 35 70  NULL NULL NULL 60
```

Azimuth, Elevation, Zoom, camera Position, Focal Point and Roll

```
0 0 1.0 35 35 200 35 35 35 0
```

```
10 0 1.0 35 35 200 35 35 35 0
10 10 1.5 35 35 200 35 35 35 0
......
```

It is suggested to use this last type to move the camera on azimuth, elevation and zoom only, putting fix the other values

NOTE*:*
Fixing the camera position, the azimuth and the elevation are computed starting from the camera position set.

If slices are required, the cycle file must contain the sequence of positions in the volume of the planes. This file can be created with the tool *VisIVOUtils --op createslices* option.

--cycleoffset [value] (optional) is the value for the progressive number of files produced with the cycle option. The default value is 0.

--cycle_skip_from [value] (optional) skips the first number of lines in the cycle file
--cycle_skip_to [value] (optional) reads up to the given line in the cycle file
(E.g.: --cycle_skip_from 30 --cycle_skip_to 34, read lines 31,32,33 and 34 (included))

--camazim [double] (optional) Fixes the camera azimuth position from the camera position
--camelev [double] (optional) Fixes the camera elevation position from the camera position
The allowed range for the camera elevation is [-90 , 90].
If the given camera elevation value is out of the valid range, the elevation is set at the boundary. Ex: --camelev 100 will automatically be changed with --camelev 90.
--zoom [double] (optional) Zooming factor. A value greater than 1 is a zoom-in, a value less than 1 is a zoom-out *(--zoom 2.0).*
--camfov [value] (optional) Fixes the zooming factor.
--campos [value] (optional) Fixes the camera position in the system coordinate: 3 values (x, y and z) must be given
--camfp [value] (optional) Fixes the focal point position in the system coordinate: 3 values (x, y and z) must be given. The default focal point is the center of the sytem.
--camroll [double] (optional) Fixes the camera roll factor.

--imagesize (optional) fixs the size of the image. It may assume the following values: *small, medium, large*. Default value is medium.
--backcolor (optional) fixs the background color. It may assume one of the following values : y*ellow, red, green, blue, white, black, cyan, violet.* Default value is black.
--onecolor (optional) fixs the color of points and isosurfaces (ignored if –-color option is given). It may assume one of the following values : y*ellow, red, green, blue, white, black, cyan, violet* . Default color is white.
--color (optional) Uses the palette
--colortable [name] (optional) Selects the palette *(--colortable default) or (--colortable paletteFilename)*

The following predefined palette can be given

default default_step efield glow gray min_max physics_contour pure_red pure_green pure_blue run1 run2 sar temperature tensteps volren_glow volren_green volren_rg volren_twolevel,all_yellow,all_red,all_green,all_blue,all_white,all_black,all_cyan,all_violet

--colorrangefrom (optional) sets the lower limit of the palette
--colorrangeto (optional) set supper limit of the palette
--stereo (optional) produces stereoscopic images. May assume values **RedBlue**: it produces an image for use with red-blue glasses. **CrystalEyes**: it uses frame-sequential capabilities available in OpenGL to drive LCD shutter glasses and stereo projectors; it produces two images with suffixes _r and _l for the Right and Left eyes. **Anaglyph**: it is a superset of RedBlue mode, but the color output channels can be configured using the anaglyphmask option and the color of the original image can be (somewhat) maintained using anaglyphsat option; the default colors for Anaglyph mode is red-cyan. Stereoscopic visualization option is ignored if the *slice* view is required.
--anaglyphsat (optional) Sets the anaglyph color saturation factor. This number ranges from 0.0 to 1.0: 0.0 means that no color from the original object is maintained, 1.0 means all of the color is maintained. The default value is 0.65. Too much saturation can produce uncomfortable 3D viewing
--anaglyphmask (optional) Sets the anaglyph color mask values. These two numbers are bits mask that control which color channels of the original stereo images are used to produce the final anaglyph image. The first value is the color mask for the left view, the second the mask for the right view. If a bit in the mask is on for a particular color, that color is passed on to the final view; if it is not set, that channel for that view is ignored. The bits are arranged as r, g, and b, so r = 4, g = 2, and b = 1. By default, the first value (the left view) is set to 4, and the second value is set to 3.

-- showlut (optional) visualizes the colorbar
-- showbox (optional) visualizes the box
-**- showaxes** (optional) visualizes the axes

NOTE:
If the **--colortable** option does not contain a predefined palette, VisIVOViewer assumes that an external filename is given as for this parameter. The file must exist in the current directory or the path must be specified. The palette is an ASCII file. The table must contain [Id+] RGB [+A] or Id+HSV+A comma or space separated values. HSV values are converted into RGB. The Id is an integer that represents the number of points in the palette. RGB (or HSV) and A must be given in [0.0,1.0] range.

The palette file can have one of the following formats:

1) Only RGB values (or HSV) are given. Opacity (A) is equal to 1.0
RGB *[optionally HSV]*
0.2 0.1 0.7
0.7 0.5 1.0
0.4 1.0 0.2

The palette table will have the number of points (3 in this case) equal to the number of rows

2) RGB+A values (or HSV+A) are given.
RGB *[optionally HSV]*
0.2 0.1 0.7 0.2
0.7 0.5 1.0 0.8
0.4 1.0 0.2 1.0

The palette table will have the number of points (3 in this case) equal to the number of rows

3) Id+RGB+A values (or id+HSV+A) are given.
RGB *[optionally HSV]*
0 0.2 0.1 0.7 0.2
100 0.7 0.5 1.0 0.8
299 0.4 1.0 0.2 1.0

The palette table will have the number of points equal to the last Id+1 (300 in this case). Intermediate values (not given) are automatically generated with a linear interpolation between the given values. The table must have increasing Ids. Tables with not ordered Ids are discarded. If the starting 0 point is not given (first row) it is assumed (by default) given as follows:

0 0.0 0.0 0.0 0.0

The first row indicates the table RGB or HSV. In the case of the RGB table this row could not be given.

The following kinds of visualizations are available: data points, isosurfaces, volumes and vectors.

The alternative command allows VisIVOViewer to read all options (including the Splotch options) from a parameter file. Lines starting with # are comments. Splotch parameter are included in this file.
An example of this file is the following:

```
############## VisIVO SECTION
######## Sect 1 General
volume=no                ← no|yes default value: no
vector=no                ← no|yes default value: no
splotch=yes              ← No parameter splotch file must be given: only no (default) or yes
input=u2.bin
out=outFilename
showbox=yes
showaxes=no
imagesize=medium
#cycle=cicleFilename
#cycleoffset=0
#cycle_skip_from=0
#cycle_skip_to=0
```

```
######## Sect 2 Points and vectors
x=X
y=Y
z=Z
#vx=VX
#vy=VY
#vz=VZ
#scale=yes
#######
########### Sect 3 Volume
#vrendering=yes
#isosurface=no
#slice=no
#shadow=no
#vrenderingfield=ColumnName
#slicefield=ColumnName
#sliceplane=x
#sliceposition=0
#sliceplanenormal= 1 1 1
#sliceplanepoint= 10 10 10
#isosurfacefield=ColumnName
#isosurfacevalue=120
#wireframe=no
#isosmooth=none
################# Sect 5 Camera
camazim=20
camelev=20
campos= 35 35 200
camfp= 35 35 35
zoom=1.5
nodefault=yes                ← put yes if you do not want the default images
#largeimage=no
################# Sect 4 Colour
color=yes
colorscalar=X
colortable=default
#colorrangefrom=0
#colorrangeto=100
#onecolor=white
#backcolor=black
#showlut=yes
opacity=0.666
#logscale=no
###########  Sect 5 Glyphs
#glyphs=sphere
#scaleglyphs=no
#radius=1.0
#radiusscalar=ColumnName
#height=1.0
#heightscalar=ColumnName
#vectorline=yes
#vectorscalefactor=1.0
#vectorscale=1
##############  SPLOTCH SECTION
# Visualization properties for gas
```

```
(see the above parameter description)
```

VisIVO Viewer on gLite systems

VisIVO may be compiled on a gLite system. In this case the option *GLITE* must be given for compile the tool. The main purpose is to read a VBT from catalogue and/or create the images and movies in the catalogue.
When running on gLite system:

the option −-VO is to specifying the virtual organization (VO) name.

2) the input VBT file, to be imported, can be a logical filename (lfn) and must start with *lfn:// .*

3) the output VBT, can be local (-- out option) or the output lfn can be given. But in any case the −-out is the local filename where a temporary images will be created

4) - -lfnout: is the output logical filename. In this case an image will be saved in the grid catalogue and deleted from local system.

5) the option −-se is to specifying the Storage Element (SE)

I
The following options can be applied to all filetrs:

--VO [value] (optional) is used to set the virtual organization (VO) when running on gLite grid. It is mandatory when running on gLite using catalogue.
--se [value] (optional) is used to set the Storage Element (SE) when running on gLite grid. Default value is DPM_HOST (site default storage element).
--lfnout [value] (optional) is used to set the logical filename (lfn) when running on gLite grid.

VisIVO Viewer

Data Points

VisIVOViewer creates data points views from the input data file. The Input data file must be in Internal Binary format. The input data file must fit the available RAM.

SPECIFIC DATA POINTS OPTIONS

--x [field] (optional) Selects the first coordinate *(--x x-coordinate)*
--y [field] (optional) Selects the second Coordinate *(--y y-coordinate)*
--z [field] (optional) Selects the third Coordinate *(--z z-coordinate)*
--scale (optional) Enables data normalization. It always allows you to visualize a cubic region even if the coordinates system has different scales.

The field names containing *X,Y,Z* or *RA,DE* and *Mag* are assumed to be default values for the x y z system, or the first three table columns, if these options are not given. A warning message will be given. NOTE: it is strongly recommended to fix these parameters to prevent unpredictable behavior.

--colorscalar [field] (optional) Selects the field for the palette *(--colorscalar scalar0)*
--logscale (optional) Uses the logarithmic scale for the palette. If the select field has values <=0 this option is ignored and the linear scale will be used.

--glyphs [name] (optional) Data points are displayed with different geometrical form. The following forms are available: *pixel sphere cone cylinder cube (--glyphs sphere)*. This option has no effect if the data point number is more than 1000.

--radius (optional) Radius of the geometrical form *(--radius 2.0)*
--height (optional) Height of the geometrical form (where applicable) *(--height 5)*
--mode (optional) If the smooth keyword is given a *smoothed* points data representation is given. In this case colors are set from the *scenario* parameter that gives a prefixed color mode representation *(--mode smooth)*

--opacity [double] (optional) Data points opacity. Default value 0.66 *(--opacity 0.03)*
--opacityTF [three double values] (optional) Data smoothed points opacity representation. They fix the curve slope for opacity transfer function when smoothed representation is given. The three values must not be negative. Suggested ranges are [3-10] [1-5] [2-5]. Default values are 5.0 3.0 2.5 *(--opacityTF 3 4 4)*

--scaleglyphs (optional) Enables the geometrical form to be scaled with a scalar field.

--scenario (optional) In a smoothed representation it gives the colors for data point. Each scenario is represented by a string name. The active current scenario is called *etna*. This is the default value.

--radiusscalar [field] (optional) Sets the scalar field for radius scaling *(--radiusscalar scalar0)*
--heightscalar [field] (optional) Sets the scalar field for height scaling *(--heightscalar scalar1)*

Examples

palette usage

```
VisIVOViewer --x X --y Y --z Z  --color --colorscalar scalar0  --colortable temperature --logscale /home/user/inputFile.bin
```

Normal GLIPHS

```
VisIVOViewer --x X --y Y --z Z --glyphs cone --radius 1 --height 2 /home/user/inputFile.bin
```

Scaled GLYPHS

```
VisIVOViewer --x X --y Y --z Z  --glyphs cone --scaleglyphs --radiusscalar scalar0 --heigthscalar scalar1 /home/user/inputFile.bin
```

VisIVO Viewer

Volumes

VisIVOViewer creates a volume view of data points from the input data file that contains a volume. The input data file must be in Internal Binary format and must have the number of mesh elements on each dimension. The input data file must fit the available RAM. A volume can be visualized with the volume rendering technique, with an isosurface or with slices.

SPECIFIC VOLUME OPTIONS

--volume (optional) enables volume visualization

VOLUME RENDERING Visualization OPTIONS

A color table must be given. The default color table will be used if the colortable option is not given.

--vrendering (optional) enables volume rendering view. The volume rendering view is the default when --volume is given.
--vrenderingfield [field] sets the scalar to be represented in the view. *(--vrenderingfield scalar0)*
--shadow (optional) enables shadow view in the rendering view.

Example

VisIVOViewer --volume --vrendering --vrenderingfield density –colortable temperature /home/user/inputFile.bin

ISOSURFACE Visualization OPTIONS

--isosurface (optional) enables isosurface view
--isosurfacefield [field] (optional) sets the scalar to be represented in the view *(--isosurfacefield scalar0)*
--isosurfacevalue [field] (optional) fixes the isocontur value: from 0 to 255 *(--isosurfacevalue 200)*
--wireframe (optional) visualizes the isosurface with wireframe.
--isosmooth (optional) smoothes the isosurface visualization. It may assume the following values: *none* (default), *medium, high*

Example

VisIVOViewer --volume --isosurface --isosurfacefield density --isosurfacevalue 200 /home/user/inputFile.bin

SLICER Visualization OPTIONS

A color table must be given. The default color table will be used if the *colortable* option is not given.

--slice (optional) enables slice view
--slicefield [field] (optional) sets the scalar to be represented in the slice view *(--slicefield scalar0)*
--sliceplane [plane] (optional) sets the plane to be represented in the view *(--sliceplane y).* It must be one of the following: *x,y, z.* The camera is always positioned in front of the plane.
--sliceposition [position] (optional) sets the plane coordinate position to be represented in the view *(--sliceposition 5).* It must be an integer value from 0 to the maximum number of cells in the selected direction*.* Ignored if cycle option is given.

VisIVOViewer can also visualize oblique planes. In this case sliceplane and sliceposition options must not be given. The camera is positioned using azimuth and elevation options.

--sliceplanepoint (optional) sets the three coordinates of a point in the plane. It is ignored in case of cycle file.
--sliceplanenormal (optional) sets the three coordinates of a point belonging to the normal axes to the slice. The sliceplanepoint and the sliceplanenormal fix the points and an

the axis in the space. The slice is normal to this axis and the point in sliceplanepoint is a point of this plane. It is ignored in case of cycle file

Important Remarks:
One of the following options must be specified: sliceplane, sliceplanenormal and/or sliceplanepoint. If sliceplane is selected orthogonal slices will be produced. If sliceplane is not given but sliceplanenormal and/or sliceplanepoint are given, generic slices will be produced. In case of cycle the specific values of sliceposition, sliceplanenormal and sliceplanepoint are ignored and the cycle file values will be used, even if the options must be given to select the type of slice visualization.

Note:

1) The stereoscopic visualization is ignored in case of slice.
2) Cycle can be given for Orthogonal Normal planes (x, y or z). In this case the cycle file must contain a sequence of integers (one for each row) inside the volume range (e.g 0-64). See also VisIVOUtils --op slices
3) Cycle can be given for point-planenormal slice. In this case the cycle file must contain a sequence of six values: three point coordinates and three plane normal coordinates. Lines with less than 6 values are ignored. In this case the showbox option is recommended. See also *VisIVOUtils --op slices*

Example

VisIVOViewer --volume -–slice --slicefield density--sliceplane x –sliceposition 3 –color --colortable default /home/user/inputFile.bin

VisIVO Viewer

Vectors

VisIVOViewer creates a view of vectors created from the input data file that contains data points. The input data file must fit the available RAM.

SPECIFIC VECTOR OPTIONS

--vector (optional) enables vector visualization
--x [field] (optional) First component of the vector application point.
--y [field] (optional) Second component of the vector application point.
--z [field] (optional) Third component of the vector application point.

--vx [field] (optional) First component of the vector.
--vy [field] (optional) Second component of the vector.
--vz [field] (optional) Third component of the vector.

--colorscalar [field] (optional) Selects the field of the VBT to be used for the palette *(--colorscalar scalar0).* The vectors are displayed with the color palette based on the value of the active scalar given in this option. If this option is not given, the palette is based on is the magnitude of the vector. This option set the active scalar.

--vectorline (optional). Enables the vector representation with a line. Default is arrows.

--vectorscalefactor [field] (optional). Scale factor for vector representation.

--vectorscale [field] (optional). Assumes the following values. Value **0**: the scale of the vector dimension is given by the active scalar (colorscalar option). Value **1**: the scale of the vector dimension is given by the vector magnitude. Value **-1** (default): the vectors are not scaled.

Example

```
VisIVOViewer --x X --y Y --z Z –vx Vx –vy Vy –vz Vz --color --colorscalar scalar0  --colortable temperature --vectroscalefactor 1.3 --vectorscale 0  /home/user/inputFile.bin
```

VisIVO Viewer

Splotch

Splotch is a ray tracing rendering tool, used to represent several species. When using -–splotch options the images are produced with Splotch.

VisIVOViewer integrates the main basic Splotch functionalities (http://www.mpagarching.mpg.de/~kdolag/Splotch/). VisIVO uses, up to now, only one specie visualization of Splotch and cannot be used for Volume visualization. This option can be used only to visualize Data Points and all the options of that section can be used (e.g. --x, --y, --z, --camazim, –camelev, --zoom, --camfov, --out, --cycle .. etc.)

SPECIFIC SPLOTCH OPTIONS

--splotch *splparafile*

The splparafile is the same file of the Splotch program but some news are included and some lines are ignored. The Splotch option can also be used with --cycle option.

Example

VisIVOViewer --splotch /home/user/splotchfile.par --x X --y Y --z Z --color --colortable mypalette.pal --cycle /home/user/mypath.file /home/user/inputFile.bin

The Splotch parameter file has many options as documented by Splotch authors.

Splotch can read and display several species in the same image. VisIVO (up to now) limits the visualization to only one specie. The following parameters can be given in the parameter file.

The following parameters can be given using Splotch. More detail can be found in the Splotch documentation. In brackets there are the default values square

size_fix0= value [0.0] It is the fixed smoothing length. If this value is >0 use this value as radius for particles.

If size_fix0 = 0
size_label0= label Column name of the quantity to use for the smoothing of particles.

size_fac0= value It is the scale factor applied for the smoothing of particles.

color_is_vector0= TRUE/FALSE [FALSE] When this value is set to TRUE, colors are RGB (read from file) otherwise a palette must be used. Suggested value: FALSE

r_col= colname The name of column to be used for smoothing

I_col= colname The name of column to be used for Intensity
C1_col=colname The name of a column to be used for R color if color_is_vector0= TRUE. Otherwhise it is the column name used to give the color to the particles using the palette.
C2_col=colname The name of a column to be used for G color. Ignored if color_is_vector0= FALSE
C3_col=colname The name of a column to be used for B color.Ignored if color_is_vector0= FALSE

xres, yres= value [800] Image size in pixel

gray_absorption= value [0.2] It is the opacity factor.
brightness0=value [1] factor for Intensity (increase the brightness of each point). Try to increase it in case of black image)

intensity_log0 =TRUE/FALSE [TRUE]. If true use the logarithm scale for Intensity field
color_log0=TRUE/FALSE [TRUE]. If true use the logarithm scale for color field
color_asinh0= TRUE/FALSE [FALSE] color is computed with asinh function.

The following parameters fix the extreme values for intensity and color. Default values are the minimum and the maximum values read from the table column.

intensity_min0= value
intensity_max0=value
color_min0=value
color_max0= value

EyeSeparation=value [0] if >0 produce a stereoscopic image (2 frame for each image)

colorbar= TRUE/FALSE [FALSE]. If true the color bar is displayed

More details can be found on http://www.mpagarching.mpg.de/~kdolag/Splotch/ .