L15, 21 March: LLNL VisIT data import

1) Download Moodle/Week 10/VisIT: a sample BOV file for volume_bullet_p134_uint16.bin The name of this downloaded file is "bullet_227_243_243_uint16.bov". a) Find an editor (NotePad?, TextEdit?) for this file. b) Prepare to edit the first line "DATA_FILE: /Volumes/Sab-Data-1/t4581/wk10/volume_b..." c) Find the path on your computer to "volume_bullet_p134_uint16.bin" d) Update the first line with the path

Data file types: HDF5 (*.h5), NetCDF, TIF (or TIFF), DICOM, ANALYZE, and raw binary (block of bytes or block of values).

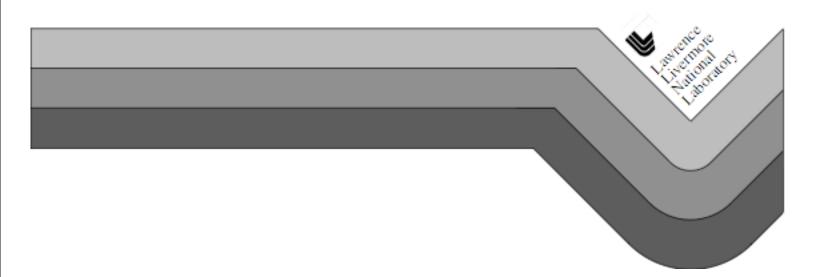
Matlab users: After version 7.3, Matlab can save *.mat files in HDF5 format.

Getting Data Into VisIt

July 2010

Version 2.0.0

Brad Whitlock



3.1 BOV file format

Example BOV header file:

TIME: 1.23456 DATA_FILE: file0000.dat # The data size corresponds to NX,NY,NZ in the above example code. DATA_SIZE: 10 10 10 # Allowable values for DATA FORMAT are: BYTE, SHORT, INT, FLOAT, DOUBLE DATA FORMAT: FLOAT VARIABLE: what I call the data # Endian representation of the computer that created the data. # Intel is LITTLE, many other processors are BIG. DATA ENDIAN: LITTLE # Centering refers to how the data is distributed in a cell. If you # give "zonal" then it's 1 data value per zone. Otherwise the data # will be centered at the nodes. CENTERING: zonal # BRICK_ORIGIN lets you specify a new coordinate system origin for # the mesh that will be created to suit your data. BRICK ORIGIN: 0. 0. 0. # BRICK_SIZE lets you specify the size of the brick. BRICK SIZE: 10. 10. 10.

Additional BOV options:

```
BYTE OFFSET: 4
# processed in parallel.
DIVIDE_BRICK: true
DATA_BRICKLETS: 5 5 5
```

BYTE_OFFSET: is optional and lets you specify some number of # bytes to skip at the front of the file. This can be useful for # skipping the 4-byte header that Fortran tends to write to files. # If your file does not have a header then DO NOT USE BYTE OFFSET.

DIVIDE_BRICK: is optional and can be set to "true" or "false". # When DIVIDE_BRICK is true, the BOV reader uses the values stored # in DATA_BRICKLETS to divide the data into chunks that can be

DATA_BRICKLETS: is optional and requires you to specify 3 integers # that indicate the size of the bricklets to create when you have # also specified the DIVIDE_BRICK option. The values chosen for # DATA BRICKLETS must be factors of the numbers used for DATA SIZE.

DATA_COMPONENTS: is optional and tells the BOV reader how many # components your data has. 1=scalar, 2=complex number, 3=vector,

1) Download Moodle/Week 10/VisIT: a sample BOV file for volume_bullet_p134_uint16.bin

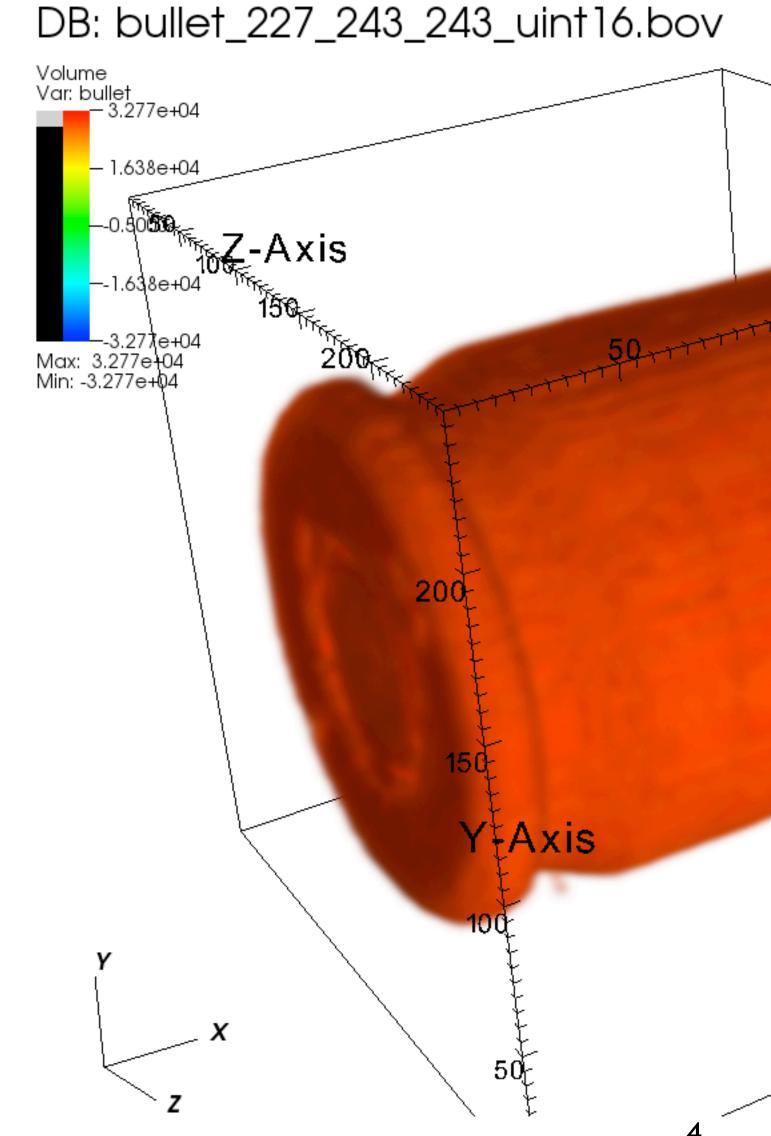
DATA_FILE: /Volumes/Sab-Data-1/t4581/wk10/volume_bullet_p134_uint16.bin # The data size corresponds to NX,NY,NZ in the above example code. DATA_SIZE: 227 243 243 # Allowable values for DATA_FORMAT are: BYTE,(SHORT,)NT, FLOAT, DOUBLE DATA FORMAT: SHORT VARIABLE: bullet # Endian representation of the computer that created the data. # Intel is LITTLE, many other processors are BIG. DATA_ENDIAN: LITTLE # Centering refers to how the data is distributed in a cell. If you # give zonal then its 1 data value per zone. Otherwise the data # will be centered at the nodes. **CENTERING**: zonal # BRICK_ORIGIN lets you specify a new coordinate system originate system o # the mesh that will be created to suit your data. BRICK ORIGIN: 111 # BRICK_SIZE lets you specify the size of the brick. BRICK_SIZE: 227 243 243

> StdD Bins: List

- signed integer-16
- Our data is unsigned.
- What kind of error will this cause?

| | | | 0 0 0 His | stogram of | volume_bullet | |
|------------------|---------------|-------------|-----------|------------|---------------|--|
| | | | bin start | count | | |
| O Histogram of v | olume bullet | p134 uint16 | 30719.531 | 64542 | | |
| j | | | 30975.527 | 68501 | | |
| | | | 31231.523 | 77368 | | |
| | | | 31487.520 | 91986 | | |
| | | | 31743.516 | 115131 | | |
| | | | 31999.512 | 139435 | | |
| | | | 32255.508 | 139500 | | |
| | | | 32511.504 | 103162 | | |
| | | | 32767.500 | 54331 | | |
| | | | 33023.496 | 20919 | | |
| | | | 33279.492 | 6396 | | |
| | • | I | 33535.488 | 2095 | | |
| 0 | | 65535 | 33791.484 | 814 | | |
| Count: 13404123 | Min: 0 | 00000 | 34047.480 | 342 | | |
| | | | 34303.477 | 234 | | |
| Mean: 27957.112 | Max: 65535 | E 92 (42221 | 34559.473 | 160 | | |
| StdDev: 1444.122 | Mode: 27391 | | 34815.469 | 115 | | |
| Bins: 256 | Bin Width: 25 | 5.990 | 35071.465 | 78 | | |
| | | | 35327.461 | 65 | | |
| List Copy Lo | g Live | 63999.023 | 35583.457 | 46 | | |
| | | 0 | 35839.453 | 44 | | |
| | | | 36095.449 | 30 | | |
| | | | 36351 445 | 25 | | |

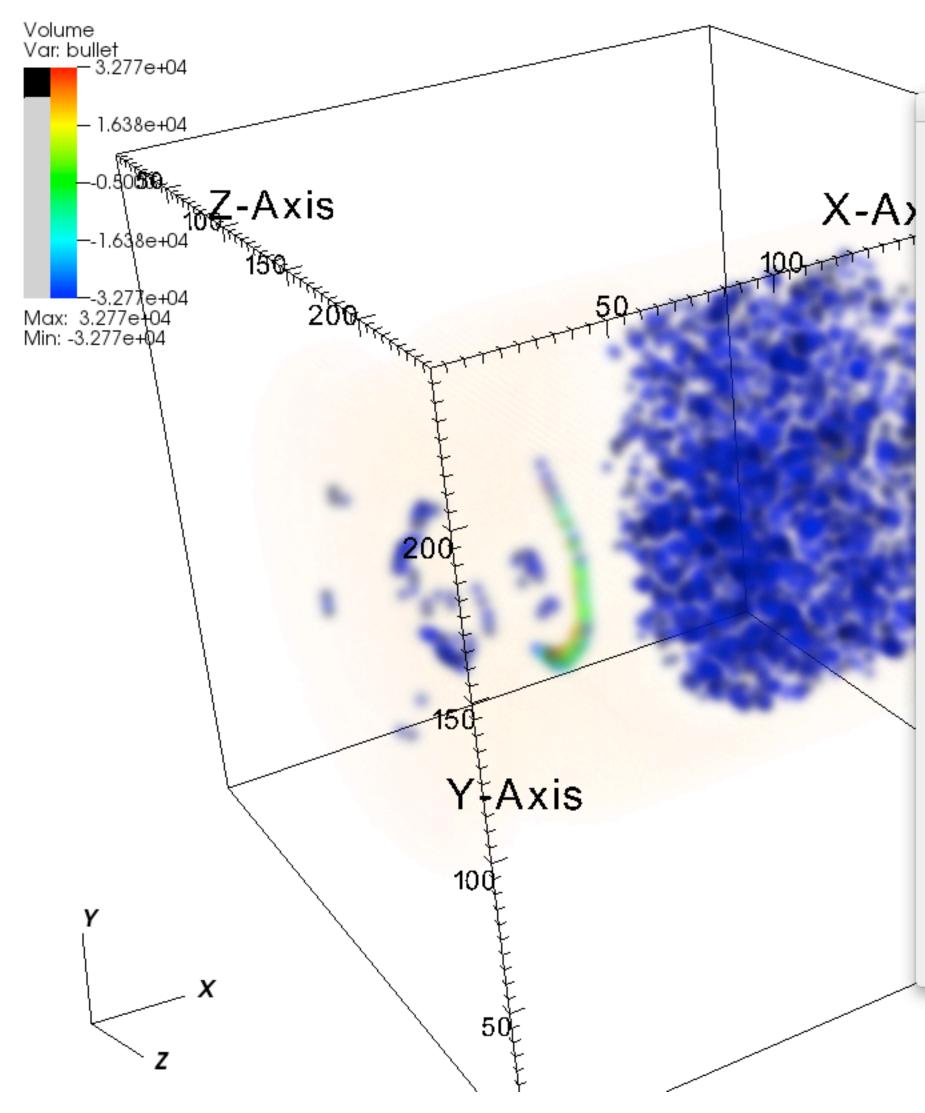
First, show the very high-intensity numbers. Looks ok at first.



| | 000 | Volume plot attributes |
|--|-----------------------------|---|
| | Renderer Options | 1D transfer function 2D transfer function |
| Χ.Α. | Color | |
| X-A) | Color table Default | + - Align Smooth Equal |
| 10,0, | <u>v</u> v | <u>v</u> <u>v</u> |
| - | | |
| and the second | Data | |
| | Scale 💿 Linear 🔷 Log 🔷 S | kew 1 |
| | Minimum 0 | Maximum 1 |
| | | |
| the second s | Opacity | |
| | Interaction mode • Freeform | Gaussian ○ From Color Table Show colors |
| | | Smooth Attenuation 33% |
| | Opacity variable | |
| | Variable default 🔻 🗌 Mi | nimum 0 Maximum 0 |
| | Make default | Load Save Reset |
| | Apply | Post Dismiss |
| | | |

Second, show the low-intensity numbers. Problem: the propellant grains are supposed to be the highest intensity voxels. The numbers above 2¹⁵ were converted to negative values.

DB: bullet_227_243_243_uint16.bov



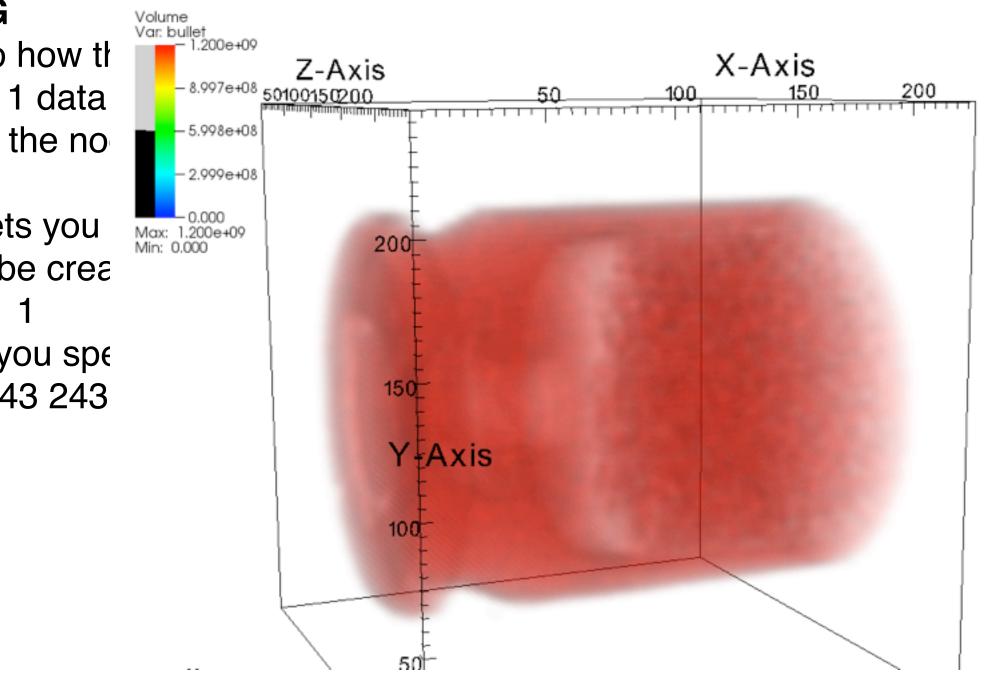
| 0.0 | Volum | e plot attributes | | |
|--------------------|------------------|-------------------|---------------------|-----------|
| Renderer | Options 1D trans | sfer function 20 | O transfer function | |
| Color | | | | |
| Color table De | fault | + - Aligr | 1 Smoot | h 🗌 Equal |
| 7 | ∇ | ∇ | ∇ | V I |
| | | | | |
| Data | | | | |
| Scale 💿 Linear 🤇 | Log Skew 1 | | | |
| | | | - | |
| Minimum 0 | | Maximum | 1 | |
| Que in | | | | |
| Opacity | Freeform Cours | cian O From Cold | Tabla Sh | ow colors |
| Interaction mode • | Freelorin O Gaus | sian O From Cold | | |
| | | Smooth Attenua | tion | 33% |
| Opacity variable | | | | |
| Variable defau | lt 🔻 🗌 Minimum | 0 | Maximum 0 | |
| | | | | |
| Make default | | | ad Cava (| Beset |
| Make default | | Loa | | Reset |
| Apply | | | Post | Dismiss |
| | | | | |

Third, I tried converting to integer-32 in ImageJ and make a new *.bin file. 1) ImageJ/Image/Type/32-bit

- 2) File/Save As/raw data volume_bullet_p134_int32.bin
- 3) Then, tested the file with Image/Import/Raw Data and learned the file is now big endian.

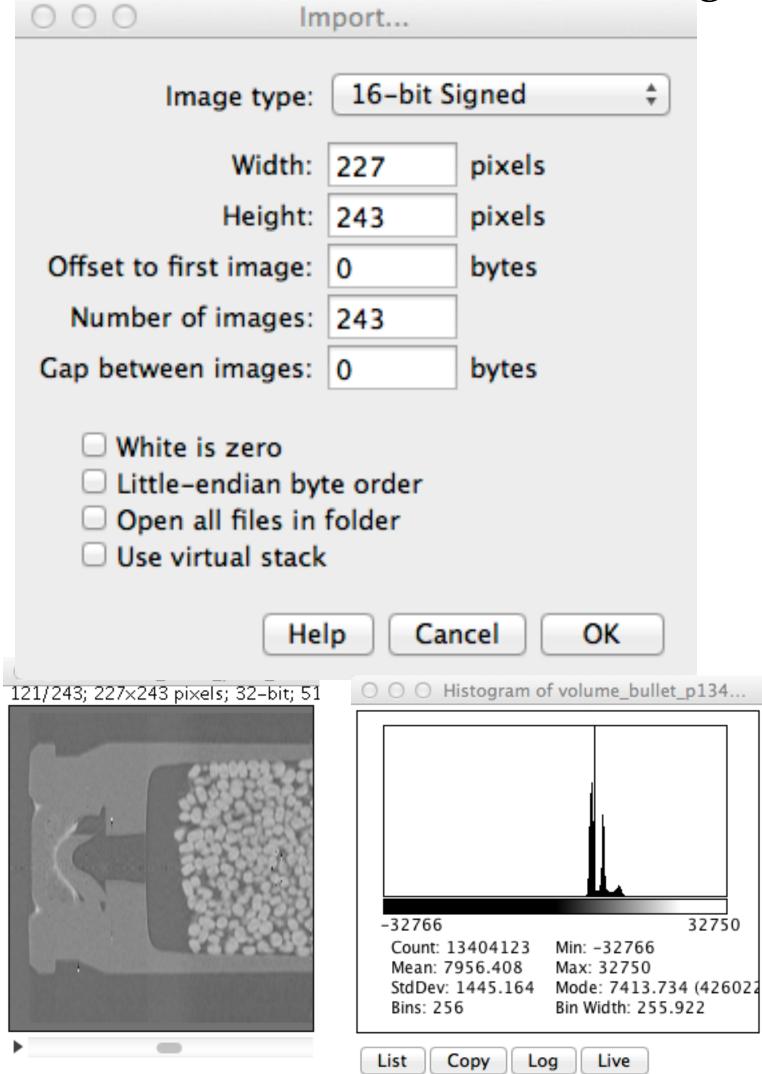
| 000 Im | | | | |
|--|------------|------------------------|--|--|
| Image type: | 32-bit S | igned 🗘 | | |
| Width: | 227 | pixels | | |
| Height: | 243 | pixels | | |
| Offset to first image: | 0 | bytes | | |
| Number of images: | 243 | | | |
| Gap between images: | 0 | bytes | | |
| White is zero Little-endian byte order Open all files in folder Use virtual stack Help Cancel OK | | | | |
| ○ ○ ○ volume_bullet_p134_i 121/243; 227×243 pixels; 32-b | | ○ ○ Histogram of volum | ne_bullet_p134 | |
| | 75: 75: | | 1185513219 (125 dth: 4685823 11199116: | |
| | | List Copy Log L | live 0 | |

DATA_FILE: /Volumes/Sab-Data-1/t4581/wk10/volume_bullet_p134_int32.bin # The data size corresponds to NX,NY,NZ in the above example code. DATA_SIZE: 227 243 243 # Allowable values for DATA_FORMAT are: BYTE, SHORT, INT, FLOAT, DOUBLE DATA_FORMAT: INT VARIABLE: bullet # Endian representation of the computer that created the data. # Intel is LITTLE, many other processors are BIG. DATA_ENDIAN: **BIG** Var: bullet 1.200e+09 # Centering refers to how the Z-Axis # give zonal then its 1 data 3.997e+08 50100150200 - 5.998e+08 # will be centered at the no **CENTERING**: zonal -2.999e+08 # BRICK_ORIGIN lets you Max: 1.200e+09 Min: 0.000 200 # the mesh that will be crea BRICK_ORIGIN: 1 1 1 # BRICK_SIZE lets you spe BRICK_SIZE: 227 243 243 150



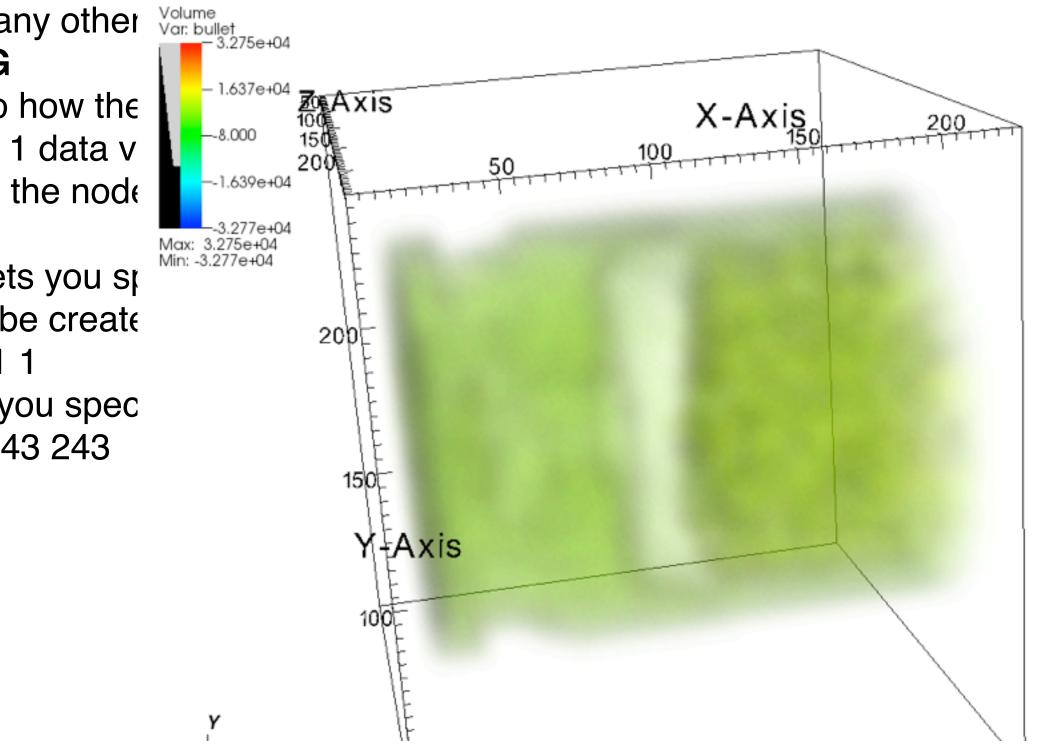
Fourth, I tried 20,000 from the data in ImageJ and make a new *uint16_v2.bin file. 1) ImageJ/Process/Math/Subtract 20,000

- 2) File/Save As/raw data volume_bullet_p134_uint16_v2.bin
- 3) Then, tested the file with Image/Import/Raw Data and learned the file is now big endian.

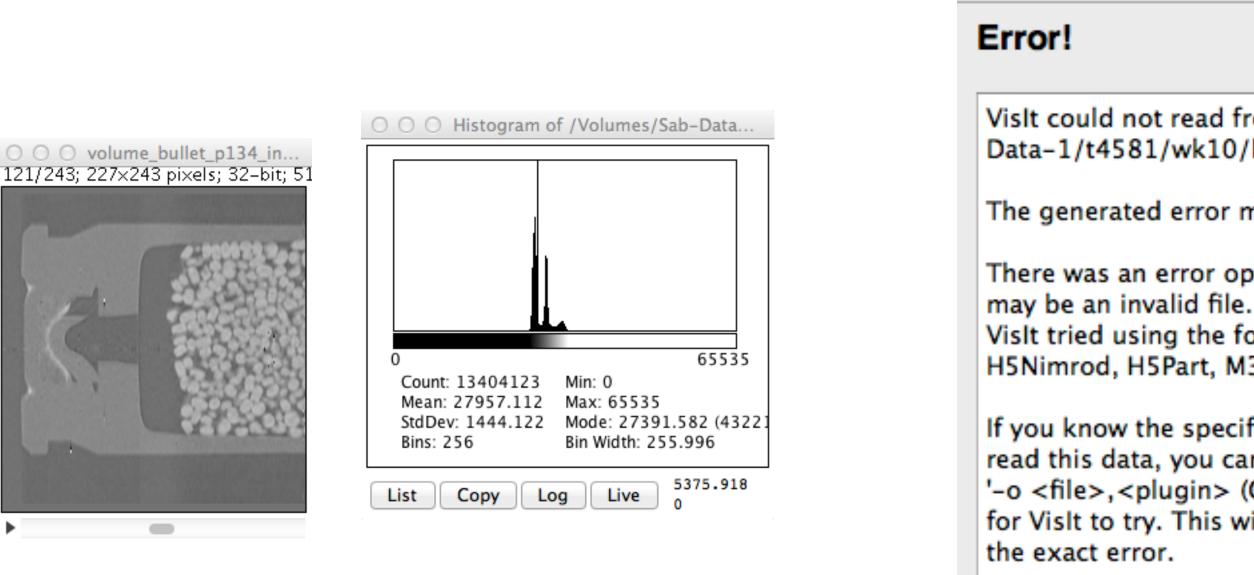


The data size corresponds to NX,NY,NZ in the above example code. DATA_SIZE: 227 243 243 # Allowable values for DATA_FORMAT are: BYTE, SHORT, INT, FLOAT, DOUBLE DATA_FORMAT: SHORT VARIABLE: bullet # Endian representation of the computer that created the data. # Intel is LITTLE, many other Volume Var. bullet DATA ENDIAN: **BIG** # Centering refers to how the Axis # give zonal then its 1 data v -1.639e+04 # will be centered at the node **CENTERING**: zonal Max: 3.275e+04 Min: -3.277e+04 # BRICK_ORIGIN lets you sp # the mesh that will be create BRICK_ORIGIN: 111 # BRICK_SIZE lets you spec BRICK_SIZE: 227 243 243

DATA_FILE: /Volumes/Sab-Data-1/t4581/wk10/volume_bullet_p134_uint16_v2.bin



Fifth, I tried creating an HDF5 file in ImageJ from the uint16 file 1) ImageJ/Plugins/HDF5/Save HDF5 as bullet_uint16.h5 2) dataset name is bullet-uint16 3) Then, tested the file with Image/Plugins/HDF5/Load HDF5



000

Information

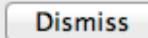
Vislt could not read from the file "/Volumes/Sab-Data-1/t4581/wk10/bullet_uint16.h5".

The generated error message was:

There was an error opening /Volumes/Sab-Data-1/t4581/wk10/bullet_uint16.h5. It

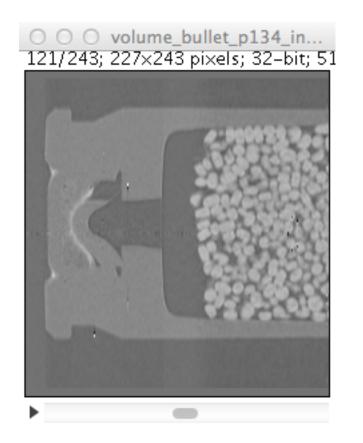
Visit tried using the following file format readers to open the file: Chombo, GTC, H5Nimrod, H5Part, M3DC1, M3D, PFLOTRAN, Pixie, Tetrad, UNIC, Vs, Silo

If you know the specific format reader Vislt should use to read this data, you can use Open As... (GUI) or '-o <file>, <plugin> (CL arg.) and identify that specific reader for Vislt to try. This will possibly give more information on

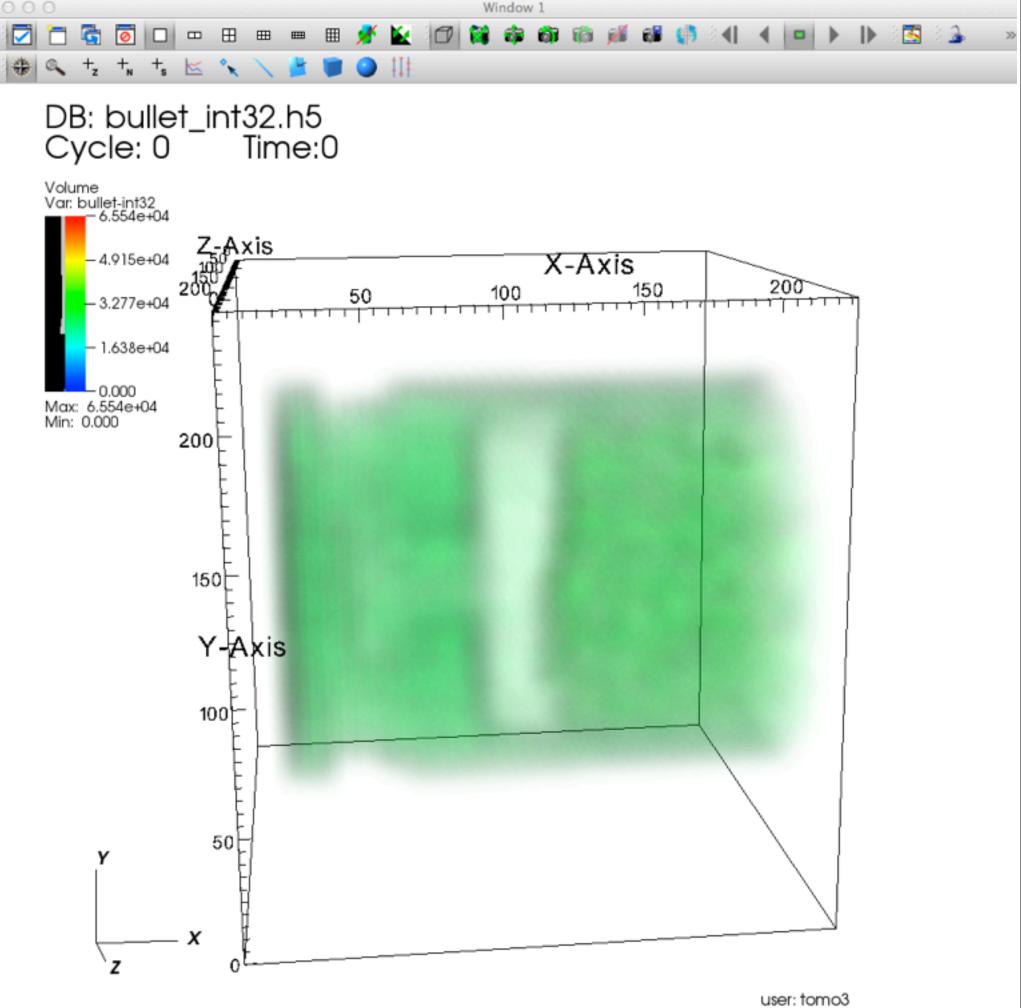


Sixth, I tried creating an HDF5 file in ImageJ from 32-bit data

- 1) ImageJ/Image/Type/32-bit
- 2) ImageJ/Plugins/HDF5/Save HDF5 as bullet_uint16.h5
- 3) dataset name is bullet-int32
- 3) Then, tested the file with Image/Plugins/HDF5/Load HDF5

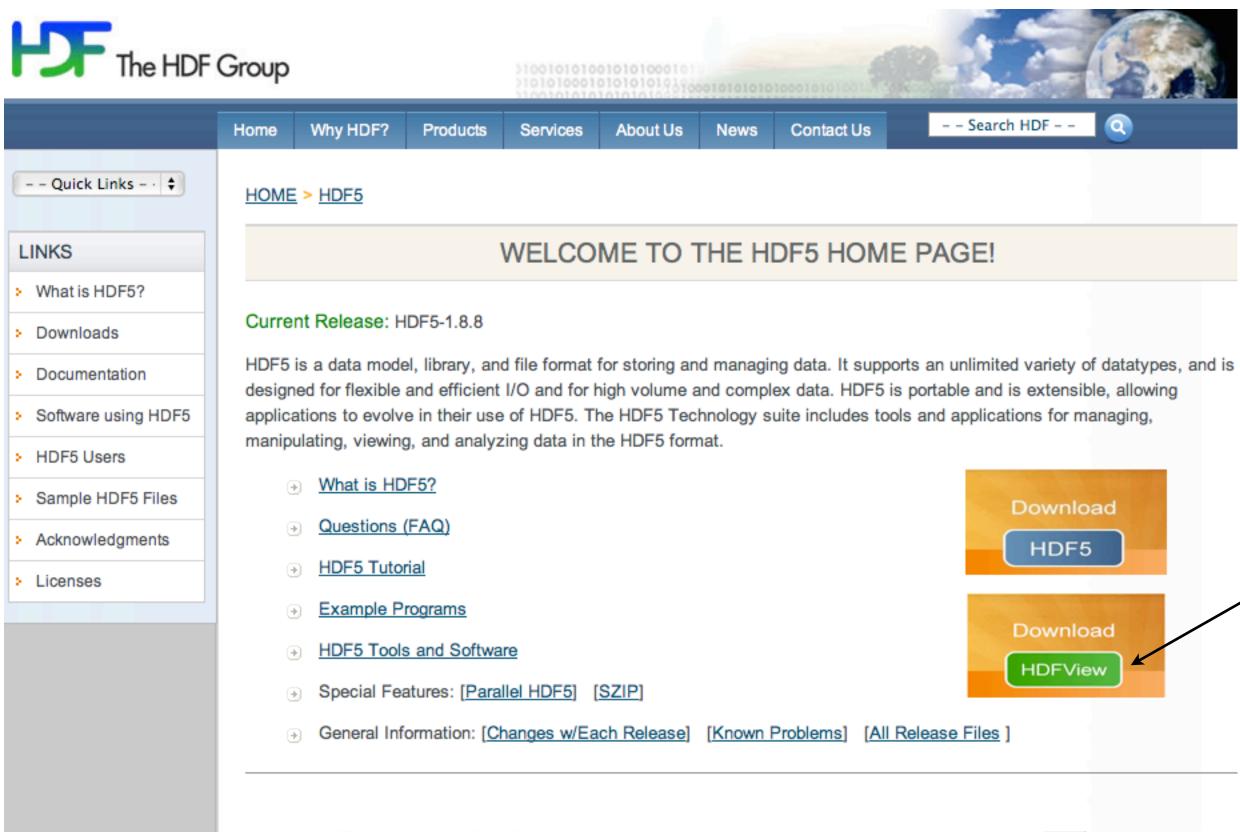


| ○ ○ ○ Histogram o | f /Volumes/Sab-Data |
|-------------------|------------------------|
| | |
| 0 | 65535 |
| Count: 13404123 | Min: 0 |
| Mean: 27957.112 | Max: 65535 |
| StdDev: 1444.122 | Mode: 27391.582 (43221 |
| Bins: 256 | Bin Width: 255.996 |
| List Copy Lo | g Live 36607.441 |



Wed Mar 21 10:07:05 2012

http://www.hdfgroup.org/



- - Last modified:November 15th 2011



HDFVIEW

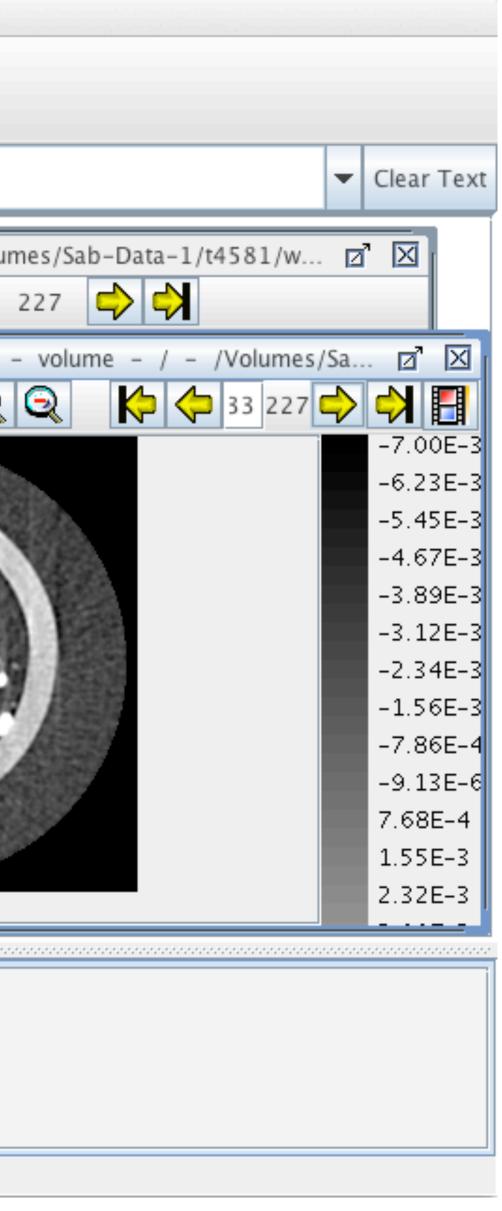
HDFView and HDF-Java Products

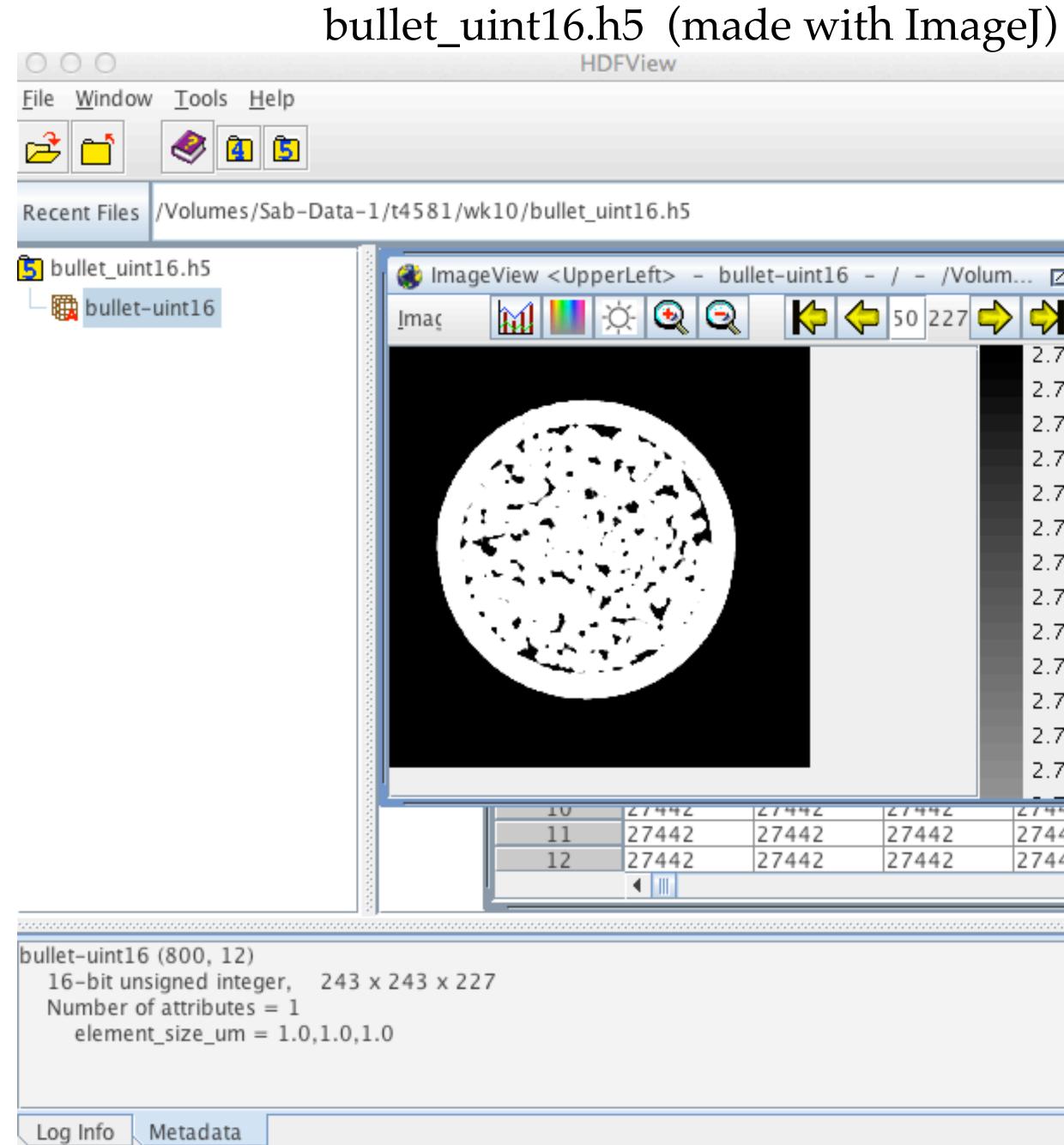
| Downloar | HDFView is packaged with an installer for easy download and installation. |
|----------|---|
| HDFVitw | |
| naries | The "bin/" directory includes the pre-built binaries for all of the HDF-Java product. |
| ource | The "src/" directory has all of the source code for the HDF-Java product. |

Previous Releases

volume_bullet_p134.h5 (made with Mathematica)

| 000 | HDFView |
|---|---|
| Eile Window Tools Help | |
| Recent Files /Volumes/Sab-Data- | 1/t4581/wk10/volume_bullet_p134.h5 |
| volume_bullet_p134.h5 | TableView - volume - / - /Volu Table 104, 11 104, 11 ImageView <upperleft> Image 97</upperleft> |
| | 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 |
| volume (976, 6) 64-bit floating-point, 243 x 2 Number of attributes = 0 | 43 x 227 |
| Log Info Metadata | |

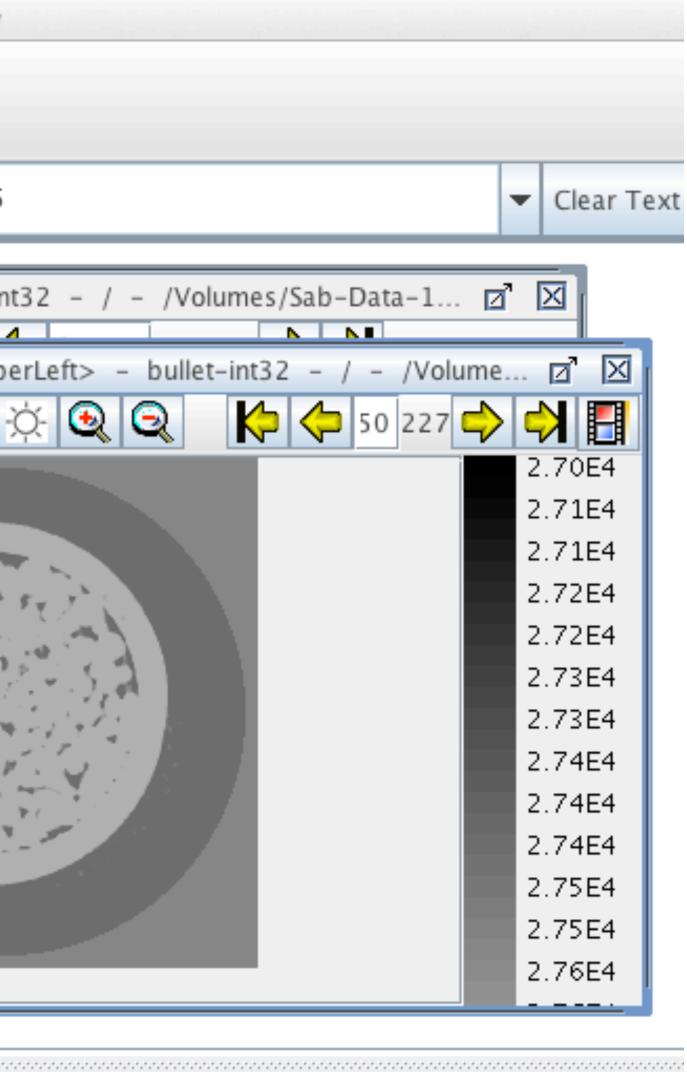




| | | | - | Clear T | ext |
|-------------|-------------------|---|---|--|-----|
| | / - /Vo 50 227 | Ium Image: Comparison of the second s | 24 24 24 24 24 24 24 24 24 24 24 24 24 2 | 2744 2744 2744 2744 2744 2744 2744 2744 | |
| 27442 27442 | 27442 27442 | 27442 27442 | | 2744 2744 | • |
| | | | | ••••••••••••••••••••••••••••••••••••••• | |

bullet_int32.h5 (made with ImageJ)

| 000 | HDFView |
|---|--|
| <u>File Window Tools H</u> elp | |
| 2 1 🧶 🗈 🖻 | |
| Recent Files /Volumes/Sab-Data- | 1/t4581/wk10/bullet_int32.h5 |
| bullet_int32.h5 bullet-int32 bullet-int32 (800, 10) | TableView – bullet-ir Tat Imaç Imaç |
| 32-bit floating-point, 243 x 2 Number of attributes = 1 element_size_um = 1.0,1.0,1 | |



Summary

VisIt raw file import supports signed integer.

VisIt HDF5 supports integer-32 bit.

VisIT and ImageJ make a nice exploration pair.

Now, let's try the MAS_rotor.h5 1) import as *.h5 into VisIt then,

2) convert *.h5 to int16 in ImageJ, save as *_int16.bin, and verify in ImageJ.
3) write a *.bov file to import this *_int16.bin file.