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).

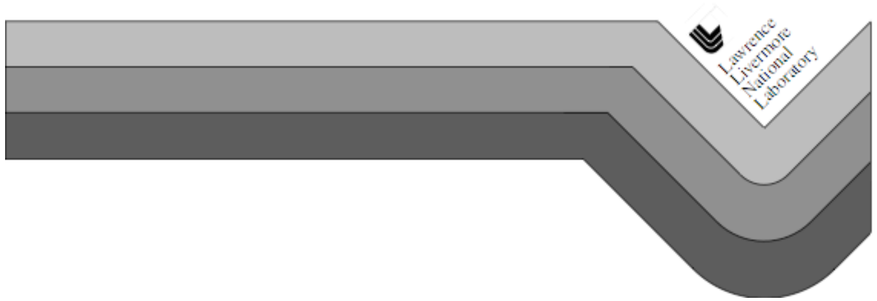
Matlab users: After version 7.3, Matlab can save *.mat files in HDF5 format.

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.
```

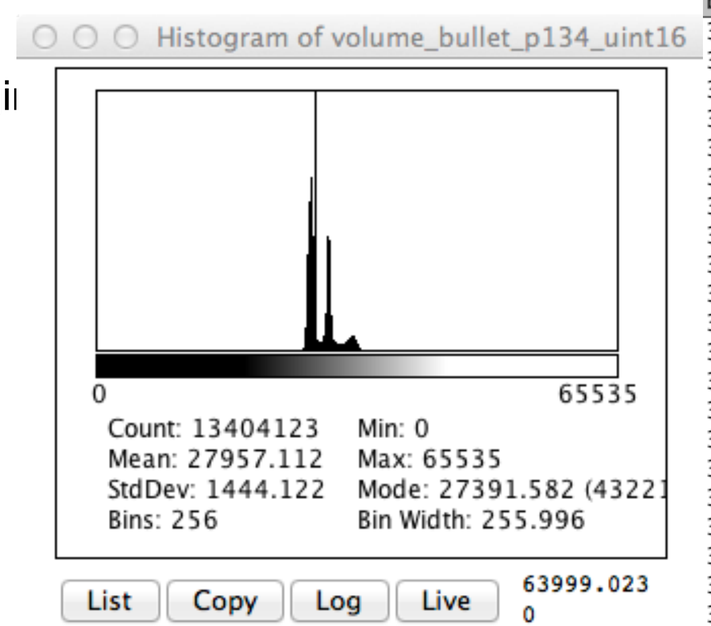
Additional BOV options:

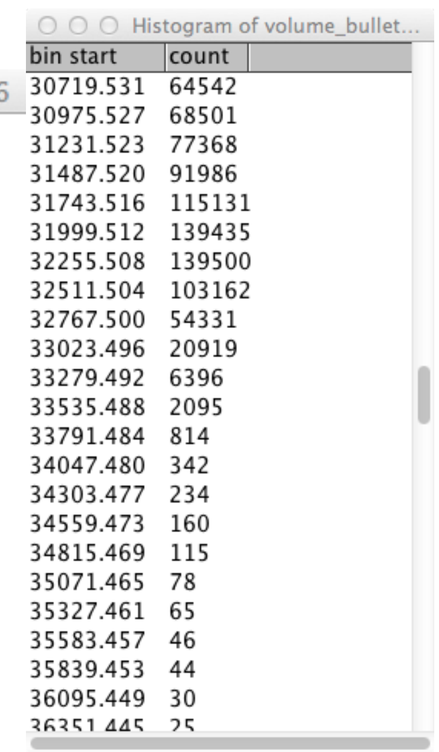
```
# BYTE_OFFSET: is optional and lets you specify some number of
# bytes to skip at the front of the file. This can be useful for
# skipping the 4-byte header that Fortran tends to write to files.
# If your file does not have a header then DO NOT USE BYTE_OFFSET.
BYTE_OFFSET: 4

# 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
# processed in parallel.
DIVIDE_BRICK: true

# 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_BRICKLETS: 5 5 5

# 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, INT, 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 origi
# the mesh that will be created to suit your data.
BRICK_ORIGIN: 1 1 1
# BRICK_SIZE lets you specify the size of the brick.
BRICK_SIZE: 227 243 243
```

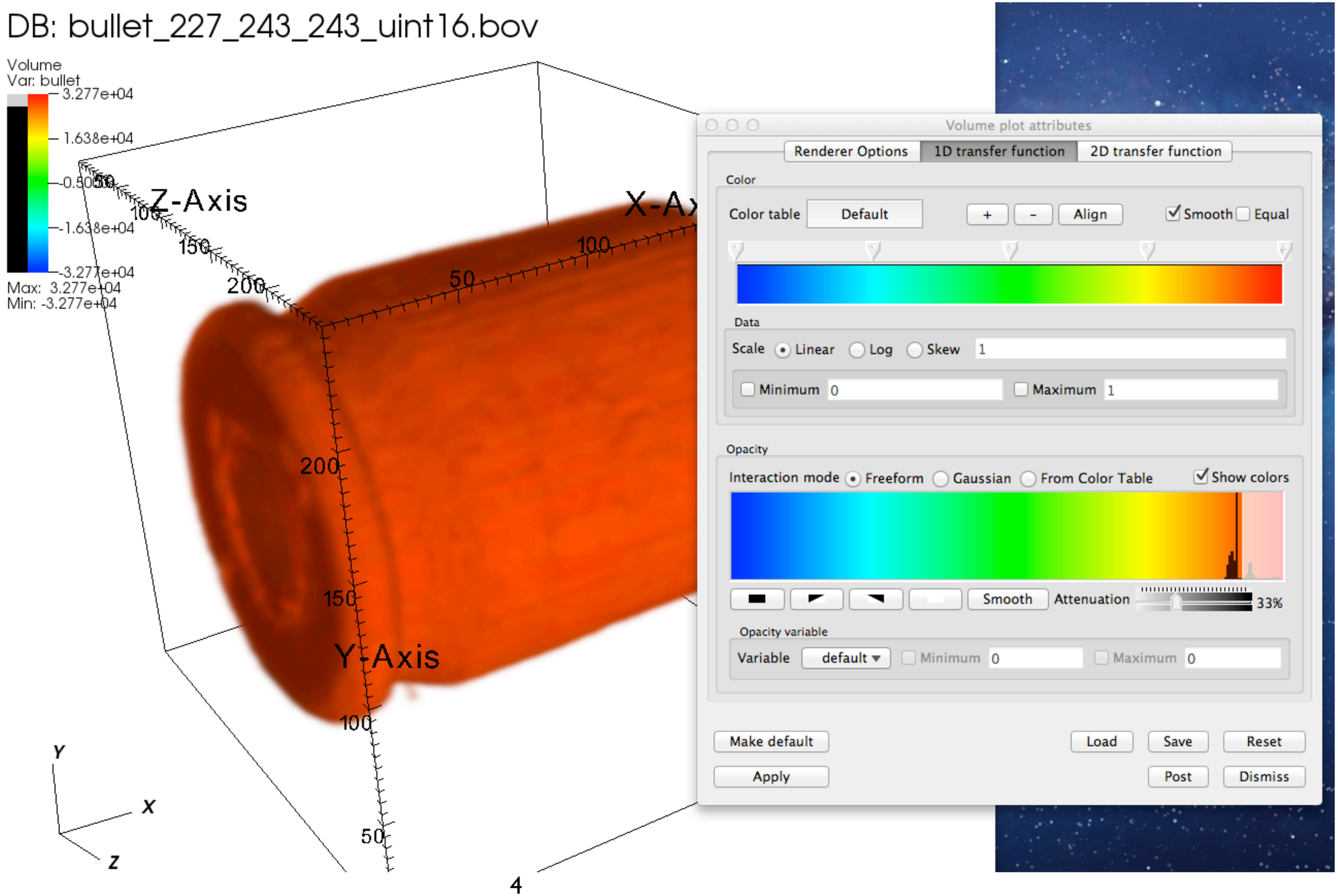
signed integer-16
Our data is unsigned.
What kind of error will this cause?



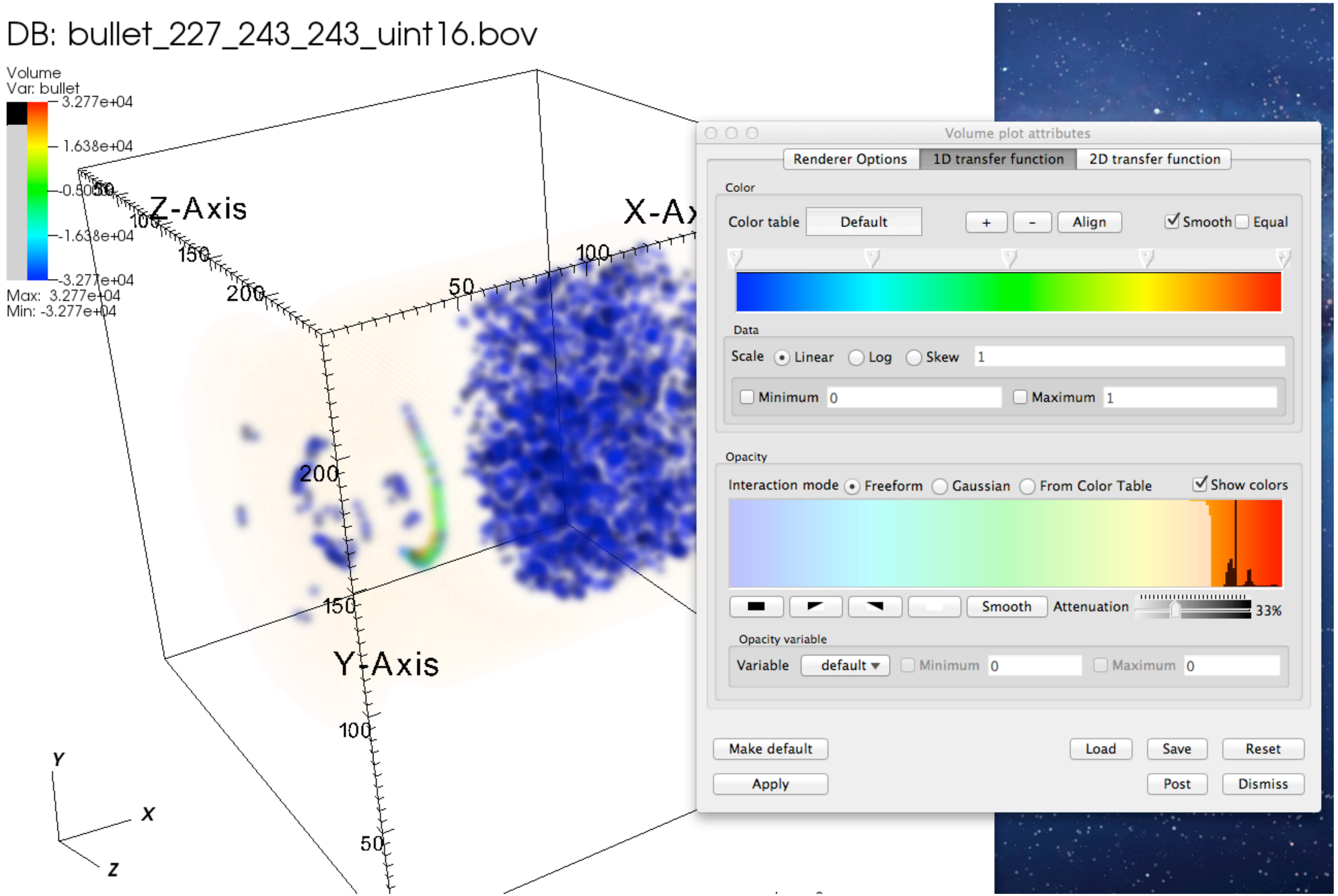


bin start	count
30719.531	64542
30975.527	68501
31231.523	77368
31487.520	91986
31743.516	115131
31999.512	139435
32255.508	139500
32511.504	103162
32767.500	54331
33023.496	20919
33279.492	6396
33535.488	2095
33791.484	814
34047.480	342
34303.477	234
34559.473	160
34815.469	115
35071.465	78
35327.461	65
35583.457	46
35839.453	44
36095.449	30
36351.445	25

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

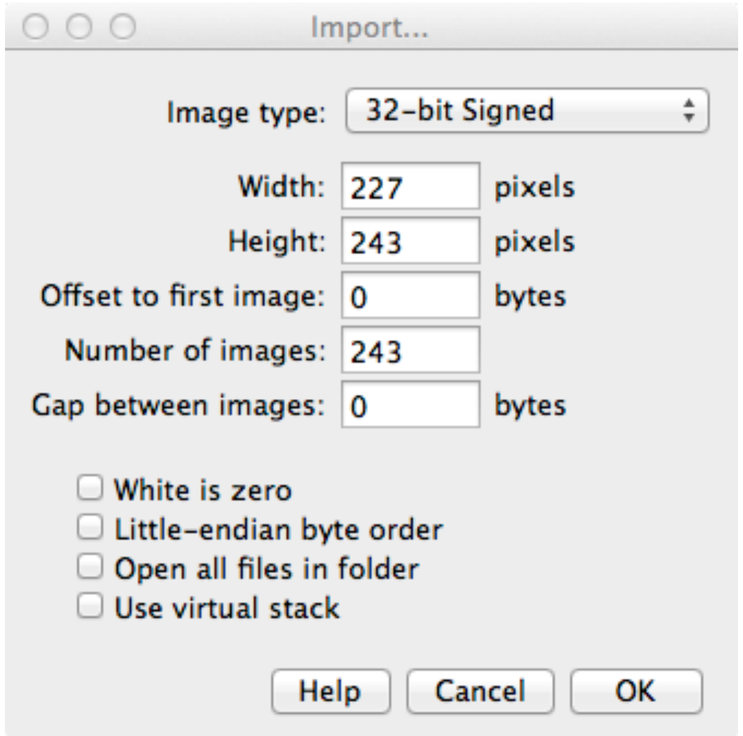


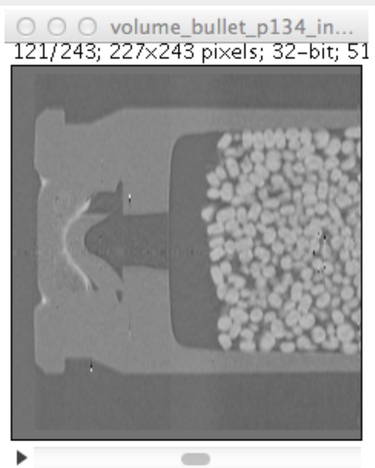
Second, show the low-intensity numbers. Problem: the propellant grains are supposed to be the highest intensity voxels. The numbers above 2^{15} were converted to negative values.

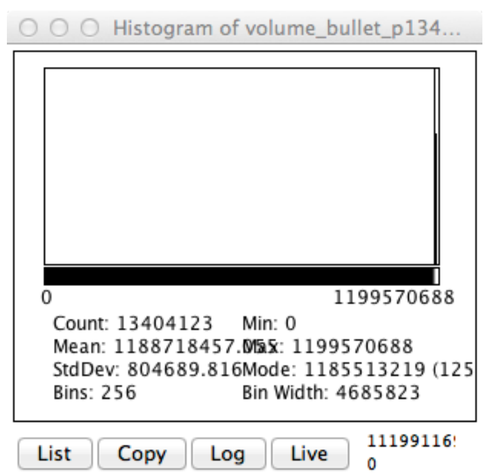


Third, I tried converting to integer-32 in ImageJ and make a new *.bin file.

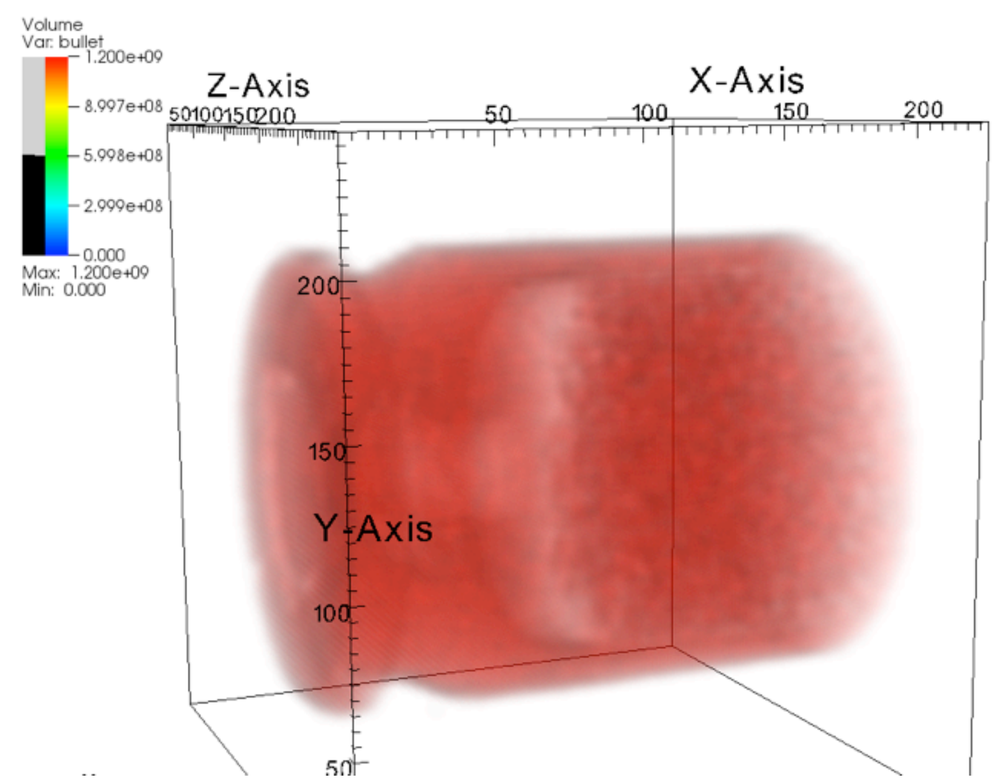
1) ImageJ/Image/Type/32-bit
2) File/Save As/raw data volume_bullet_p134_int32.bin
3) Then, tested the file with Image/Import/Raw Data and learned the file is now big endian.





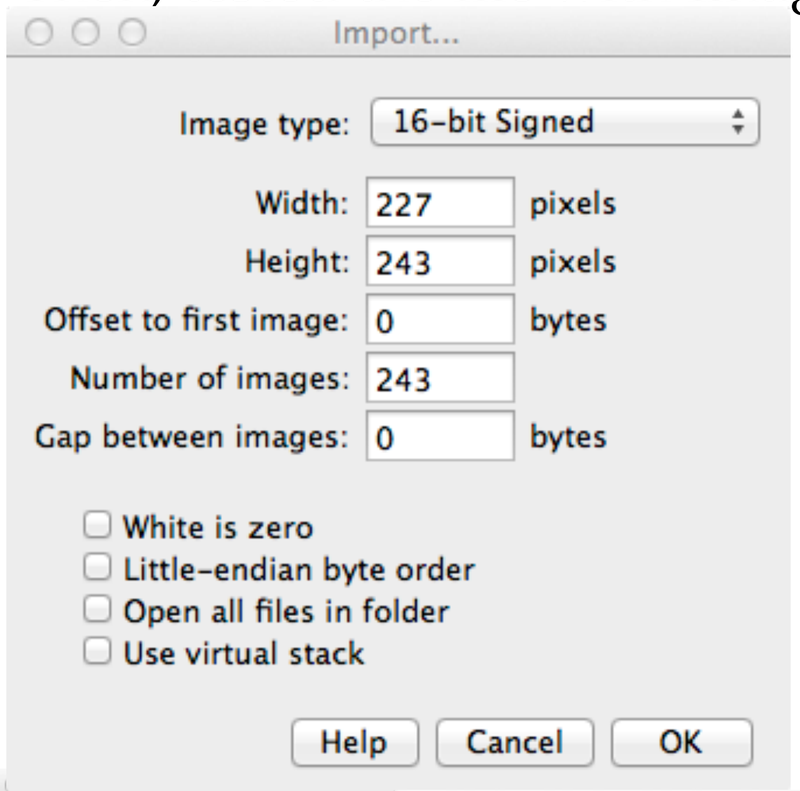


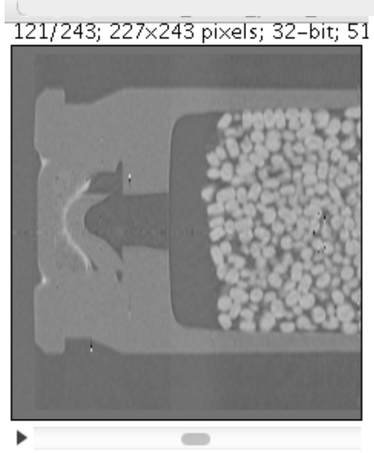
DATA_FILE: /Volumes/Sab-Data-1/t4581/wk10/**volume_bullet_p134_int32.bin**
The data size corresponds to NX,NY,NZ in the above example code.
DATA_SIZE: 227 243 243
Allowable values for DATA_FORMAT are: BYTE, SHORT, INT, FLOAT, DOUBLE
DATA_FORMAT: **INT**
VARIABLE: bullet
Endian representation of the computer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

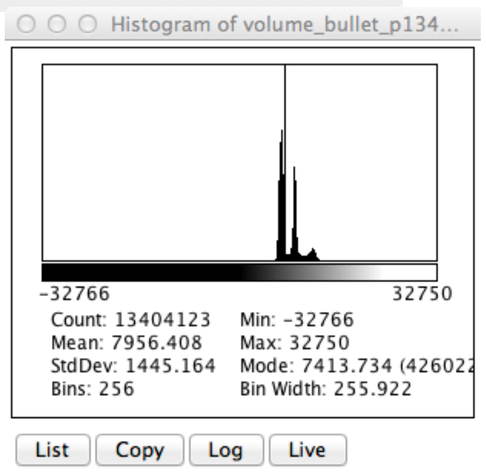


Fourth, I tried 20,000 from the data in ImageJ and make a new *uint16_v2.bin file.

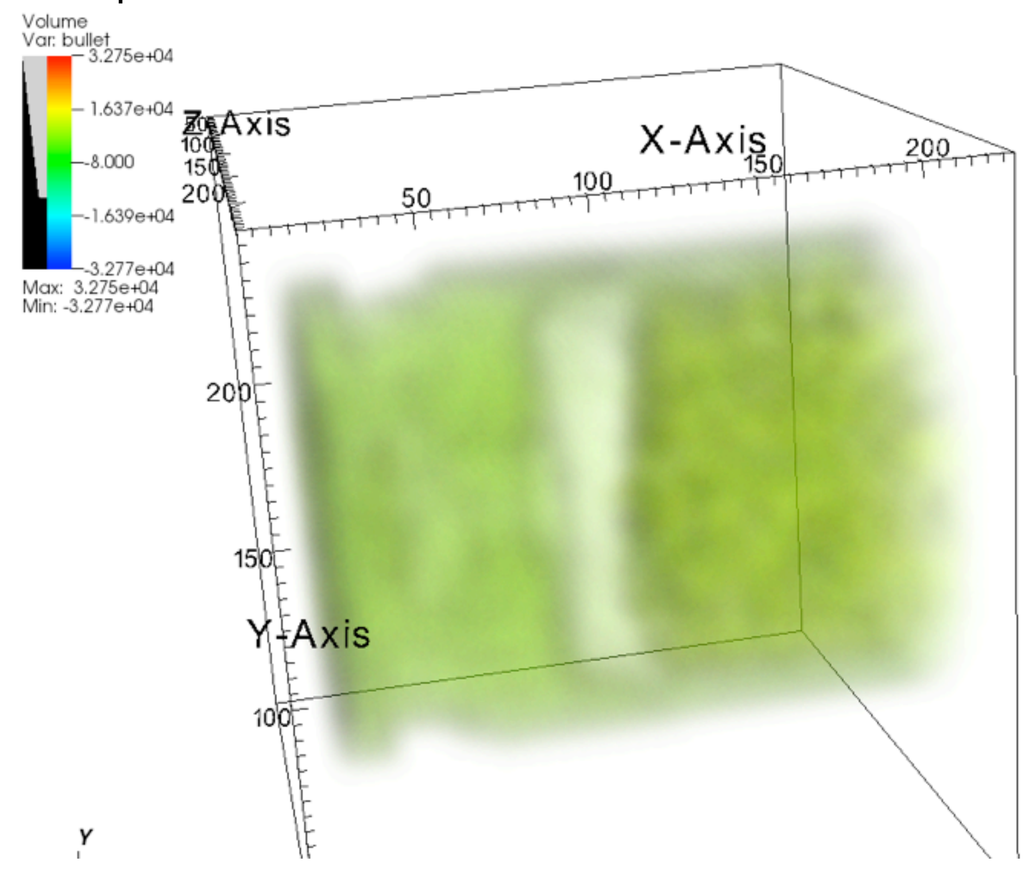
1) ImageJ/Process/Math/Subtract 20,000
2) File/Save As/raw data volume_bullet_p134_uint16_v2.bin
3) Then, tested the file with Image/Import/Raw Data and learned the file is now big endian.





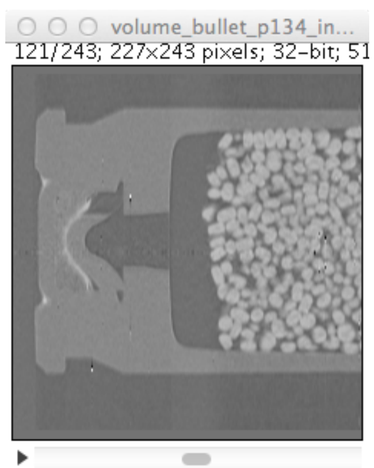


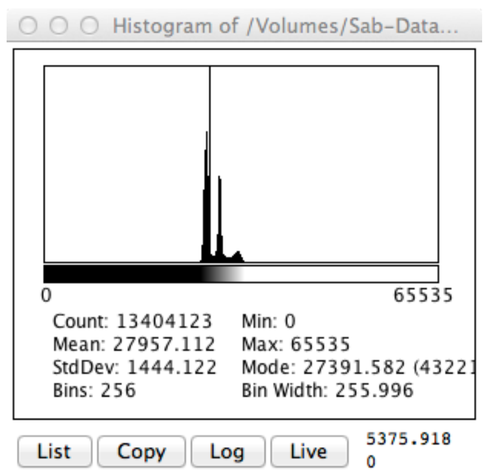
DATA_FILE: /Volumes/Sab-Data-1/t4581/wk10/**volume_bullet_p134_uint16_v2.bin**
The data size corresponds to NX,NY,NZ in the above example code.
DATA_SIZE: 227 243 243
Allowable values for DATA_FORMAT are: BYTE, SHORT, INT, FLOAT, DOUBLE
DATA_FORMAT: **SHORT**
VARIABLE: bullet
Endian representation of the computer that created the data.
Intel is LITTLE, many other
DATA_ENDIAN: **BIG**
Centering refers to how the
give zonal then its 1 data v
will be centered at the node
CENTERING: zonal
BRICK_ORIGIN lets you sp
the mesh that will be create
BRICK_ORIGIN: 1 1 1
BRICK_SIZE lets you spec
BRICK_SIZE: 227 243 243

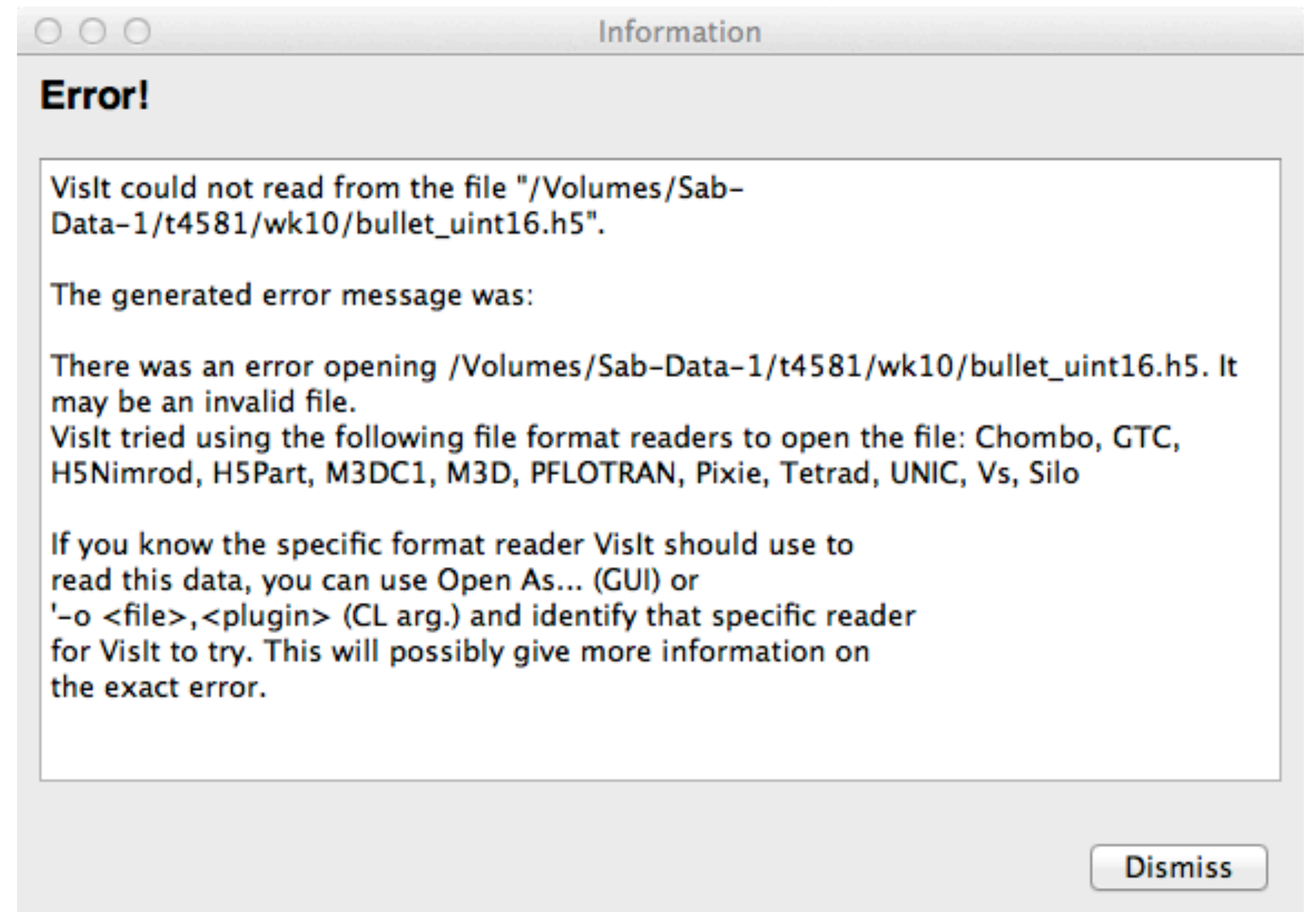


Fifth, I tried creating an HDF5 file in ImageJ from the uint16 file

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

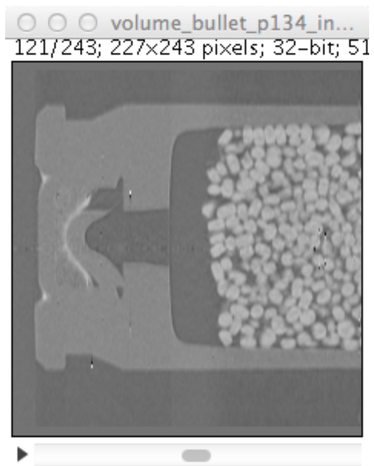


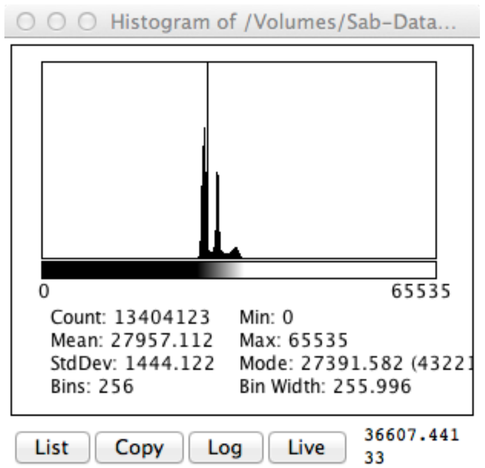


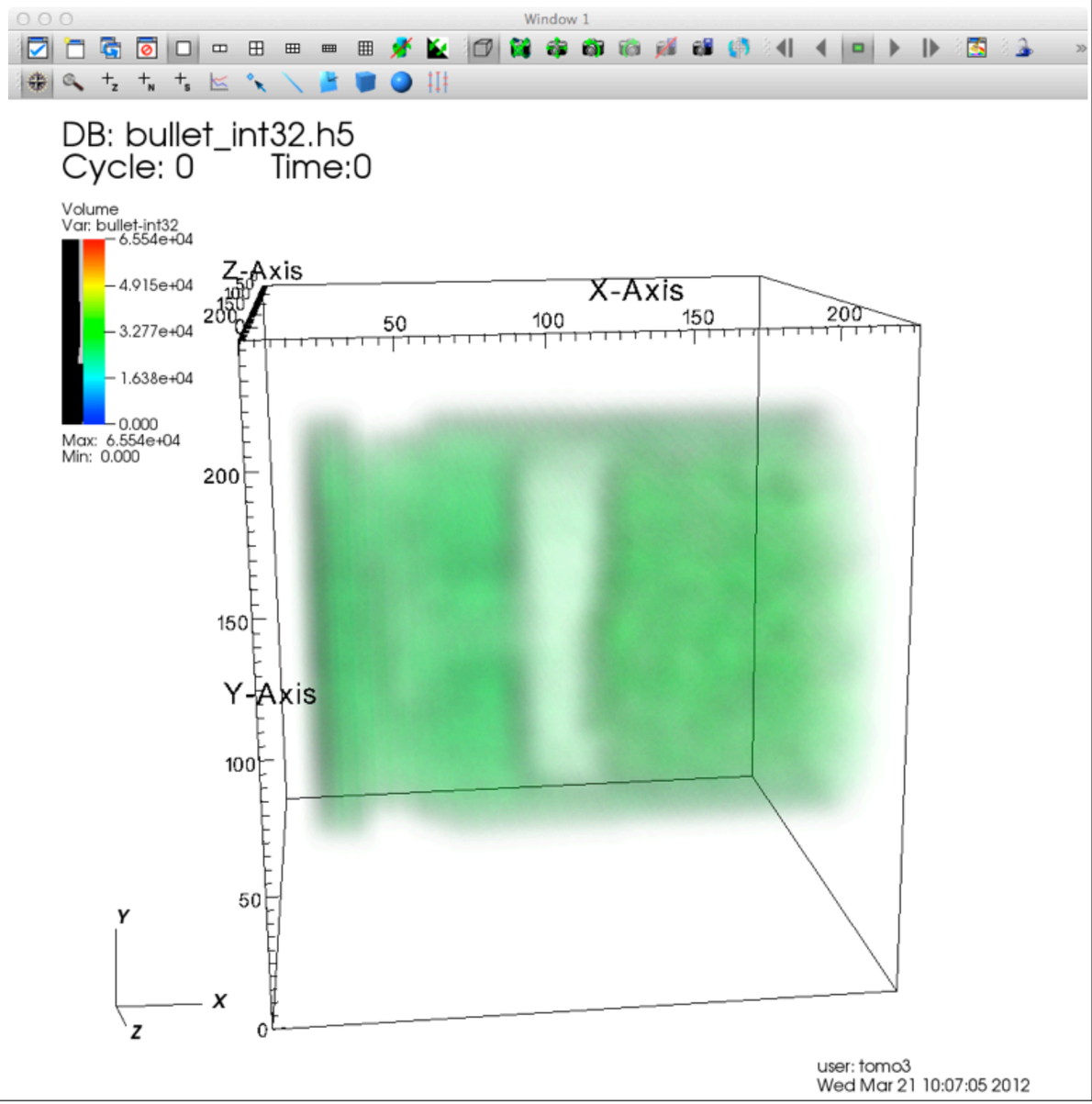


Sixth, I tried creating an HDF5 file in ImageJ from 32-bit data

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







http://www.hdfgroup.org/

The HDF Group

Home | Why HDF? | Products | Services | About Us | News | Contact Us | - - Search HDF - -

- - Quick Links - -

LINKS

- What is HDF5?
- Downloads
- Documentation
- Software using HDF5
- HDF5 Users
- Sample HDF5 Files
- Acknowledgments
- Licenses

HOME > HDF5

WELCOME TO THE HDF5 HOME PAGE!

Current Release: HDF5-1.8.8

HDF5 is a data model, library, and file format for storing and managing data. It supports an unlimited variety of datatypes, and is designed for flexible and efficient I/O and for high volume and complex data. HDF5 is portable and is extensible, allowing applications to evolve in their use of HDF5. The HDF5 Technology suite includes tools and applications for managing, manipulating, viewing, and analyzing data in the HDF5 format.

- What is HDF5?
- Questions (FAQ)
- HDF5 Tutorial
- Example Programs
- HDF5 Tools and Software
- Special Features: [Parallel HDF5] [SZIP]
- General Information: [Changes w/Each Release] [Known Problems] [All Release Files]

Download HDF5

Download HDFView

HDFVIEW

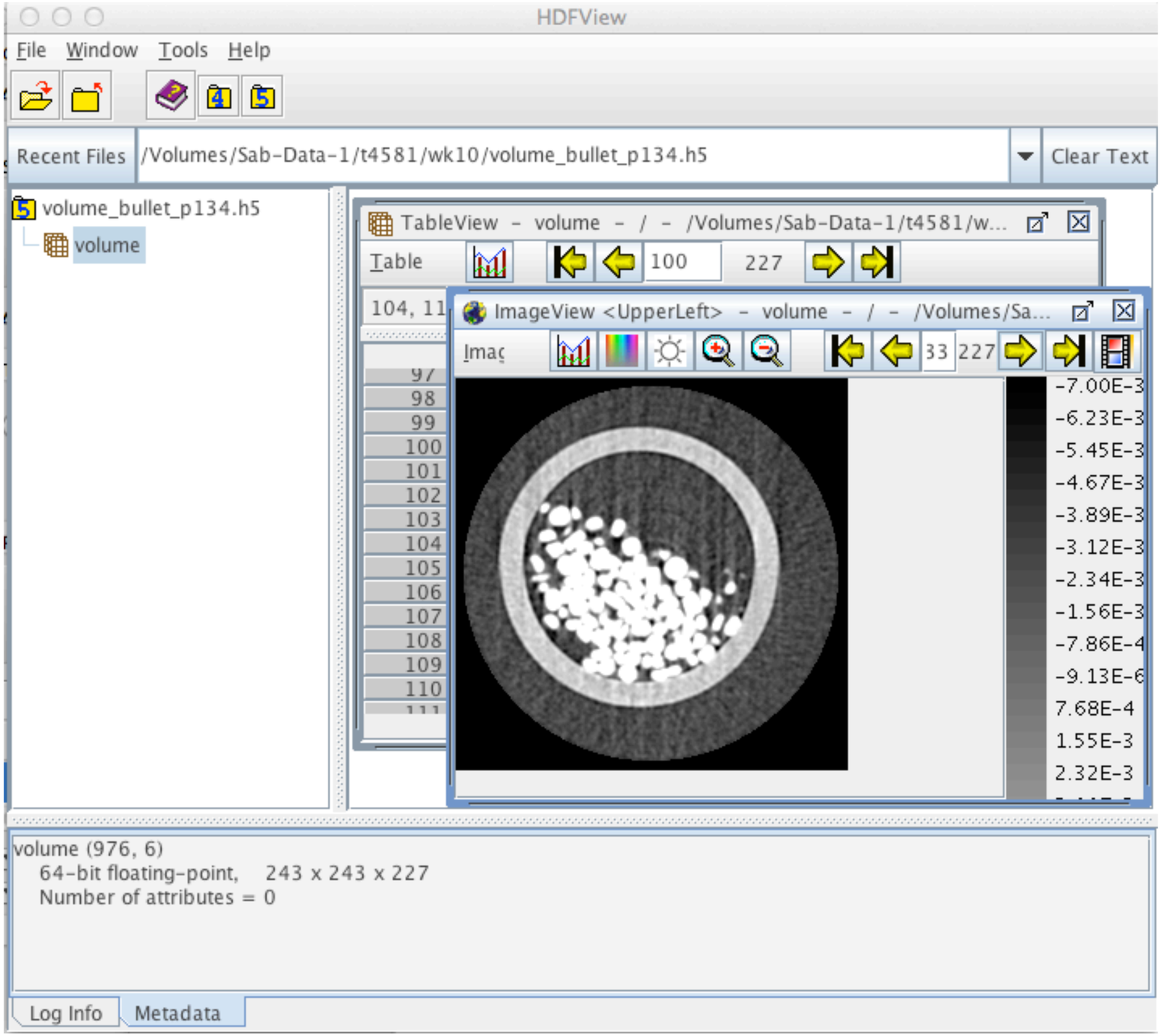
- - Last modified:November 15th 2011

HDFView and HDF-Java Products

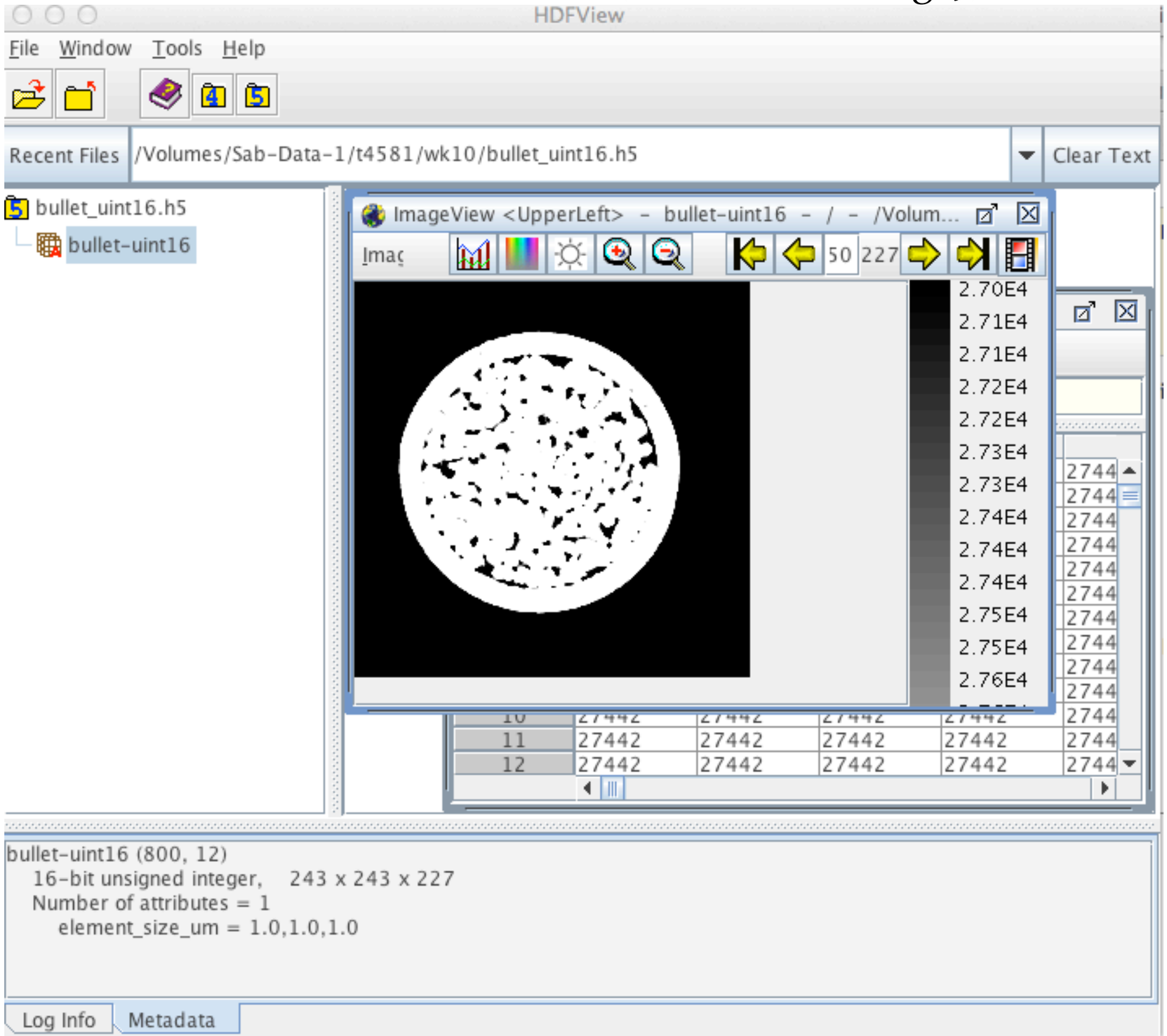
Download HDFView	HDFView is packaged with an installer for easy download and installation.
Binaries	The "bin/" directory includes the pre-built binaries for all of the HDF-Java product.
Source	The "src/" directory has all of the source code for the HDF-Java product.

Previous Releases

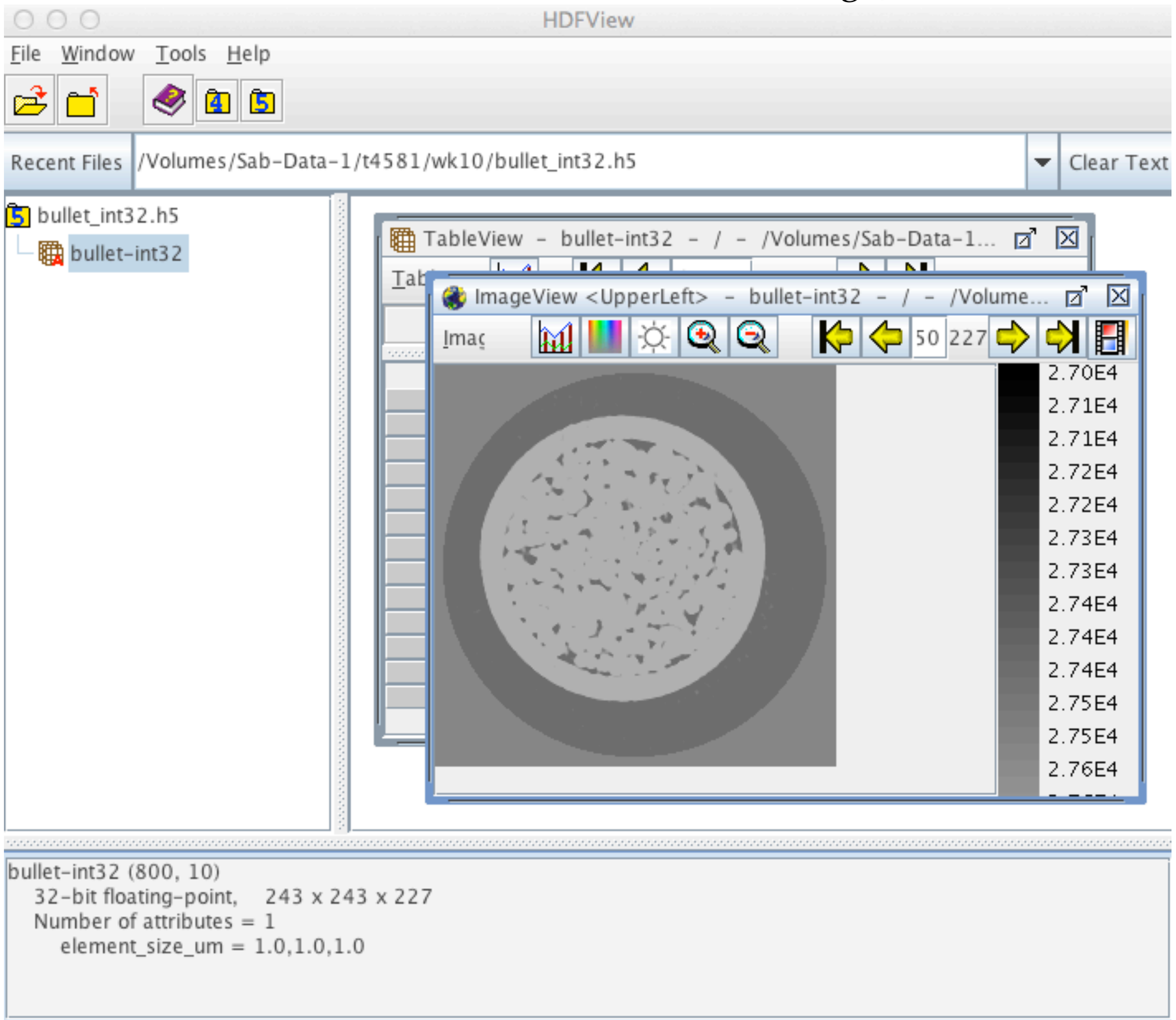
volume_bullet_p134.h5 (made with Mathematica)



bullet_uint16.h5 (made with ImageJ)



bullet_int32.h5 (made with ImageJ)



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.