ImarisReader for MATLAB

Contents

License

Introduction

ImarisReader is a MATLAB class that facilitates access to the image DataSet and Surpass object data stored in an ims file. Imaris ims files are based on the HDF5 file format, which can be read using the built-in HDF5 access functions in MATLAB. For more information about the HDF5 file format, see the HDF Group website, the MATLAB HDF5 Help, and the Imaris Open website.

The ImarisReader class does not directly read Imaris data. Rather, ImarisReader objects encapsulate a set of objects with methods to read the DataSet and Spots, Surfaces, and other objects in a Surpass scene. These additional classes are included with ImarisReader and are independent of ImarisReader—you do not need to create an ImarisReader object to create an instance of one of the reader classes (see Usage example).

Setup

To use ImarisReader, download the ImarisReader .zip file from the MATLAB Central File Exchange or from the Imaris Open website. Once you have downloaded ImarisReader, extract the contents of the zip file to a folder, and add the extracted folder to the MATLAB path. ImarisReader was designed using MATLAB 2015b and should work in version 2009b or later. ImarisReader does not require any additional MATLAB toolboxes. However, you might find a number of toolboxes, such as the Image Processing toolbox, valuable when working with the contents of ims files.

Notation conventions

Imaris Surpass objects are indicated by an initial upper case letter, e.g., Surfaces, while individual objects within a Surpass object are indicated by an initial lower case letter, e.g., the first surface in a Surfaces object.

ImarisReader MATLAB class components

ImarisReader

ImarisReader objects are containers for additional reader objects that read the HDF5-based contents of ims files.

Properties

FID

HDF5 File identifier for the file

DataSet

A **DatasetReader** class object

Cells

A 1×n array of **CellsReader** class objects

Filaments

A 1×n array of **FilamentsReader** class objects

Scene

A **SceneReader** class object

Spots

A 1×n array of **SpotsReader** class objects

Surfaces

A 1×n array of **SurfacesReader** class objects

Methods

delete

Close all open HDF5 Identifiers and invalidate the object.

DatasetReader

DatasetReader objects access the Imaris DataSet stored in an ims file. Note that the Imaris DataSet is stored as an HDF5 Group.

Properties

GID

HDF5 Group identifier to access the DataSet properties

DataType

String representing the data type for the data set. Possible values are:

- 'uint8' (Corresponds to 'eTypeUint8' in Imaris)
- 'uint16' (Corresponds to 'eTypeUint16' in Imaris)
- 'single' (Corresponds to 'eTypeFloat' in Imaris)

ExtendMinX

Minimum extent of the data set along the x dimension (in µm)

ExtendMinY
Minimum extent of the data set along the y dimension (in µm)

ExtendMinZ
Minimum extent of the data set along the z dimension (in µm)

ExtendMaxX
Maximum extent of the data set along the x dimension (in µm)

ExtendMaxY
Maximum extent of the data set along the y dimension (in µm)

ExtendMaxZ
Maximum extent of the data set along the z dimension (in µm)

SizeX
Number of voxels in the x dimension

SizeY
Number of voxels in the y dimension

SizeZ
Number of voxels in the z dimension

SizeC
Number of channels in the data set

SizeT
Number of time points in the data set

ChannelInfo
A struct containing image channel information. Field names are generated from the HDF5 attribute names. Possible fields include:

- Color
- Description
- Name

as well as many others.

Timestamps
A t×1 cell containing the image acquisition time points as strings in 'YYYY-MM-DD HH:MM:SS.FFF…' format

Methods

delete
Close the DataSet HDF5 Group Identifier and invalidate the object.

GetData
```
data = obj.GetData
```

Return the entire Imaris dataset as a five-dimensional array in xyzct order.

GetDataSlice

```
dataSlice = obj.GetDataSlice(zIndex, cIndex, tIndex)
```

Return the xy slice from the dataset represented by the zero-based indices zIndex, cIndex, and tIndex.

GetDataVolume

```
dataVolume = obj.GetDataSlice(cIndex, tIndex)
```

Return the xyz volume from the dataset represented by the zero-based indices cIndex and tIndex.

SurpassObjectReader

This is the base class for creating Surpass object readers. SurpassObjectReader is abstract—instances cannot be created.

Properties

CreationParameters

Parameters used to create the objects in Imaris

GID

HDF5 Group identifier for the object data

Name

Surpass scene name (Note that this is not the HDF5 group name)

Methods

delete

Close the object's associated HDF5 Group Identifier and invalidate the object.

GetStatistics

```
stats = obj.GetStatistics
```

Return the statistics for the objects as a structure array. For p statistical properties, each element of the returned 1×p structure array has fields for the Name, associated object IDs and Values of the property.

```
stats = obj.GetStatistics('Stats', {str1, str2, str3})
```

Return the statistics matching the cell array of strings str1, str2 and str3. The matching is not case-sensitive.

```
stats = obj.GetStatistics('Type', str)
```

Return the statistics that match str, a case-sensitive string from the list:

- 'all' (returns all statistics)
- 'id' (returns statistics for all objects with a positive identifier: individual objects and tracks)
- 'singlet' (returns statistics for individual objects, e.g., spots and surfaces)
- 'track' (returns statistical aggregates for tracks, e.g., track mean speed)
- 'summary' (returns summary statistics, e.g., "Total number of spots")

```
stats = obj.GetStatistics(…, 'ReturnUnits', 1)
```

Return the statistics with a field for the units of measurement for each statistic.

CellsReader

CellsReader objects access Cells objects stored in an ims file.

Properties

GID

Inherited from SurpassObjectReader

Name

Inherited from SurpassObjectReader

NumberOfCells

Total number of cells in the Cells object

NumberOfNuclei

Total number of nuclei in the Cells object

NumberOfVesicleTypes

Number of vesicle types in the Cells object

Methods

delete

Close the associated HDF5 Group Identifiers and invalidate the object.

GetCell

```
mask = obj.GetCell(cIdx)
```

Return a Boolean mask for the cell indicated by the zero-based index cIdx.

GetIDs

```
ids = obj.GetIDs
```

Return a vector of IDs for all cells.

GetIndicesT

```
idx = obj.GetIndicesT
```

Return a vector of zero-based indices indicating the time index for each cell.

GetNucleus

```
nucleus = obj.GetNucleus(nIdx)
```

Return a mask for the nucleus represented by the zero-based index nIdx.

GetNucleiIDs

```
ids = obj.GetNucleiIDs
```

Return a vector of IDs for all the nuclei in the Cells object.

GetNucleiIndicesT

```
idx = obj.GetNucleiIndicesT(nIdx)
```

Return a vector of zero-based indices indicating the time index of each nucleus.

GetNucleiPositions

pos = obj.GetNucleiPositions

Return an m×3 array of nuclei centroids.

GetNucleiTrackEdges

```
edges = obj.GetNucleiTrackEdges
```

Return an m×4 array with the track connections between nuclei. Each row of the returned array represents a connection between a cell and associated nucleus:

[cellIndexA, nucleusIndexA, cellIndexB, nucleusIndexB]

GetNucleiTrackIDs

```
ids = obj.GetNucleiTrackIDs
```

Return a vector containing the track IDs for the nuclei track edges.

GetPositions

```
pos = obj.GetPositions
```

Return an m×3 array with the xyz positions (centers of mass) for all cells in the Cells object.

GetStatistics

Inherited from SurpassObjectReader

GetTrackEdges

```
edges = obj.GetTrackEdges
```

Return an m×2 array containing the track edges for the cells. Each row represents an edge, where an edge represents a tracking connection between two objects.

GetTrackIDs

```
ids = obj.GetTrackIDs
```

Return a vector containing the track IDs for the cell track edges.

GetVesiclesIDs

```
ids = obj.GetVesiclesIDs(vIdx)
```

Return the IDs for all vesicles represented by the zero-based index vIdx. The vIdx argument corresponds to Type A (0), Type B (1), etc., vesicles.

GetVesiclesPositions

```
pos = obj.GetVesiclesPositions(cIdx, vIdx)
```

Return an m×3 array of vesicle positions for the cell represented by the zero-based index cIdx and the vesicle type represented by the zero-based index vIdx.

GetVesiclesRadii

```
radii = obj.GetVesicleRadii(cIdx, vIdx)
```

Return the radii for the vesicles in the cell represented by the zero-based index cIdx and the vesicle type represented by the zero-based index vIdx.

GetVesiclesTrackEdges

```
edges = obj.GetVesiclesTrackEdges(vIdx)
```

Return an m×4 array containing the edges for the vesicles represented by the zero-based index vIdx. Each row of the edges array represents the track connection between a cell and the associated vesicle:

[cellIndexA, vesicleIndexA, cellIndexB, vesicleIndexB]

GetVesiclesTrackIDs

```
ids = obj.GetVesiclesTrackIDs(vIdx)
```

Return a vector containing the track IDs for the edges for the vesicles represented by the zero-based index vIdx.

FilamentsReader

FilamentsReader objects access Filaments objects stored in an ims file.

Properties

GID

Inherited from SurpassObjectReader

Name

Inherited from SurpassObjectReader

NumberOfFilaments

Total number of filaments in the Filaments object

Methods

delete

Close the associated HDF5 Group Identifiers and invalidate the object.

GetBeginningVertexIndex

```
idx = obj.GetBeginningVertex
```

Return the index of the initial vertex for the filaments represented by the zero-based index fIdx.

GetEdges

```
edges = obj.GetEdges(fIdx)
```

Return an m×2 array containing the edges (connections) between vertices for the filament represented by the zero-based index fIdx.

GetEdgesSegmentID

```
ids = obj.GetEdgesSegmentID(fIdx)
```

Return the ids of the segments for the corresponding edges.

GetIDs

```
ids = obj.GetIDs
```

Return a vector containing the IDs of all the filaments.

GetIndicesT

```
idx = obj.GetIndicesT
```

Return the zero-based time indices for all filaments in the Filaments object.

GetNumberOfEdges

```
number = obj.GetNumberOfEdges
```

Return a vector containing the number of edges for each filament in the Filaments object.

GetNumberOfPoints

```
number = obj.GetNumberOfPoints
```

Return a vector containing the number of points/vertices for each filaments in the Filaments object.

GetPositions

```
pos = obj.GetPositions(fIdx)
```

Return an m×3 array of xyz coordinates for the vertex positions of the graph for the filament represented by the zero-based index fIdx.

GetRadii

```
radii = obj.GetRadii(fIdx)
```

Return a vector containing the radii of the vertices of the graph for the filament represented by the zero-based index fIdx.

GetStatistics

Inherited from SurpassObjectReader. For Filament objects, calling GetStatistics with the type parameter set to track will return both filament and point track statistics. Calling GetStatistics with the type parameter set to singlets will return statistics for dendrites, filaments, points and spines.

GetTrackEdges

```
edges = obj.GetTrackEdges
```

Return an m×2 array containing the edges between filaments. This method corresponds to GetFilamentTrackEdges in ImarisXT.

GetTrackIDs

```
ids = obj.GetTrackIDs
```

Return a vector containing the track IDs for the corresponding filament track edges. This method corresponds to GetFilamentTrackIDs in ImarisXT.

GetTypes

```
types = obj.GetTypes(fIdx)
```

Return a vector with the vertex type for all the vertices for the filament represented by the zero-based index fIdx. The values in the returned vector represent Dendrites (0) and Spines (0).

GetVertexTrackEdges

```
edges = obj.GetVertexTrackEdges
```

Return an m×4 array of edges representing track connections for the filament vertices. Each row of the returned array represents a filament and associated vertex connection:

[filamentIndexA, vertexIndexA, filamentIndexB, vertexIndexB]

GetVertexTrackIDs

```
ids = obj.GetVertexTrackIDs
```

Return the ID of the track to which each vertex edge belongs.

SceneReader

SceneReader objects access the general Scene data in an ims file.

Properties

GID

HDF5 Group identifier to access the Scene properties

NumberOfChildren

The total number of child objects in the Scene

NumberOfCells

The number of Cells objects in the Scene

NumberOfFilaments

The number of Filaments objects in the Scene

NumberOfSpots

The number of Spots objects in the Scene

NumberOfSurfaces

The number of Surfaces objects in the Scene

SpotsReader

SpotsReader objects access Spots objects stored in an ims file.

Properties

CreationParameters

Inherited from SurpassObjectReader

GID

Inherited from SurpassObjectReader

Name

Inherited from SurpassObjectReader

NumberOfSpots

Total number of spots in the Spots object

Methods

delete

Close the associated HDF5 Group Identifier and invalidate the object.

GetIDs

```
ids = obj.GetIDs
```

Return the IDs of all spots in the Spots object.

GetIndicesT

```
idx = obj.GetIndicesT
```

Return the zero-based time indices for all spots in the Spots object.

GetPositions

```
pos = obj.GetPositions
```

Return the xyz positions for all spots in the Spots object.

GetRadii

```
radii = obj.GetRadii
```

Return the radii of all spots in the Spots object. For ellipsoid spots, the radii are calculated as the average of the lateral and axial radii: $r = \langle r_x, r_y, r_z \rangle$.

GetRadiiXYZ

```
radiiXYZ = obj.GetRadiiXYZ
```

Return the ellipsoid xyz radii of all spots in the Spots object.

GetStatistics

Inherited from SurpassObjectReader

GetTrackEdges

```
edges = obj.GetTrackEdges
```

Return an m×2 array containing the edges between spots.

GetTrackIDs

```
ids = obj.GetTrackIDs
```

Return a vector containing the track IDs for the corresponding spot track edges.

SurfacesReader

SurfacesReader objects access Surfaces objects stored in an ims file.

Properties

CreationParameters

Inherited from SurpassObjectReader

GID

Inherited from SurpassObjectReader

Name
Inherited from SurpassObjectReader

NumberOfSurfaces
Total number of surfaces in the Surfaces object

Methods

delete
Close the associated HDF5 Group Identifier and invalidate the object.

GetIDs
```
ids = obj.GetIDs
```

Return the IDs of all surfaces in the Surfaces object.

GetIndicesT
```
idx = obj.GetIndicesT
```

Return the zero-based time indices for all surfaces in the Surfaces object.

GetNormals
```
normals = obj.GetNormals(sIdx)
```

Return the normals for the surface represented by the zero-based index sIdx.

GetPositions
```
posXYZ = obj.GetPositions
```

Return the xyz positions (centers of mass) for all surfaces in the Surfaces object.

GetMask
```
mask = obj.GetMask(sIdx)
```

Return a Boolean voxel mask for the surface represented by the zero-based index sIdx.

GetStatistics
Inherited from SurpassObjectReader

GetTrackEdges
```
edges = obj.GetTrackEdges
```

Return an m×2 array containing the edges between surfaces.

GetTrackIDs
```
ids = obj.GetTrackIDs
```

Return a vector containing the track IDs for the corresponding surface track edges.

GetTriangles
```
triangles = obj.GetTriangles(sIdx)
```

Return an m×3 array containing the triangles for the surface represented by the zero-based index sIdx.

GetVertices

```
vertices = obj.GetVertices(sIdx)
```

Return an m×3 array containing the vertices for the surface represented by the zero-based index sIdx.

Usage examples

Create an ImarisReader object

You can create an ImarisReader object to access any of the example files included with Imaris. For example, if Imaris 8.1.2 is installed:

```
xPath = 'C:\Program Files\Bitplane\Imaris x64 8.1.2\images\celldemo.ims';
imsObj = ImarisReader(xPath);
```

Get the x, y and z dimensions of the dataset in an ims file

If 'file.ims' is a file on the MATLAB path, you can access the xyz dimensions of the dataset:

```
imsObj = ImarisReader('file.ims');
xSize = imsObj.DataSet.SizeX;
ySize = imsObj.DataSet.SizeY;
zSize = imsObj.DataSet.SizeZ;
```

Get the image volume for a specified channel at a specified time point

```
cIdx = 1; % Second channel index
tIdx = 1; % Second time point
dataVolume = imsObj.DataSet.GetDataVolume(cIndex, tIndex);
```

Get the xyz positions of the first Spots object in an ims file

```
posXYZ = imsObj.Spots(1).GetPositions;
```

Re-reference a Reader object to simplify syntax

```
imsObj = ImarisReader('file.ims'); % Read a file with a Surfaces object
xSurfaces = imsObj.Surfaces(1); % Reference the first Surfaces object
posXYZ = xSurfaces.GetPostions; % Same as imsObj.Surfaces(1).GetPositions
```

Create a SpotsReader object without creating an ImarisReader object

SurpassObjectReader objects can be created without constructing an ImarisReader object. To create a component reader object, you need an HDF5 Group ID for the object you want to access:

```
FID = H5F.open('file.ims'); % Open a file with a Spots object
GID = H5G.open(FID, '/Scene/Content/Points1'); % Open the first Spots
spotsObj = SpotsReader(GID); % Create the reader object
```

Differences between ImarisReader and the XT interface

As ImarisReader family objects directly access the data stored in an ims file, they cannot read unsaved changes made to open Imaris data and objects. Also, ImarisReader is a reader class only and cannot edit the contents of an ims file.

The ImarisReader class is designed to reproduce much of the reading functionality of the XT interface. As a result, many of the methods mimic the XT interface syntax. In certain cases, though, where it made sense, methods were changed to produce similar behavior for different Surpass object types. For example, in the XT Interface:

```
ISurfaces.GetTimeIndex(sIdx)
```

returns the time point index for the single surface represented by sIdx. To get the time points for all the surfaces in the Surfaces object, you loop for all surface indices. In contrast, for Spots objects:

```
ISpots.GetIndicesT
```

returns the time indices for all spots in the Spots object. To make Spots and Surfaces act consistently, the ISurfaces.GetTimeIndex method was replaced with a GetIndicesT method that returns the time indices for all the surfaces.

Accessing Property values versus get methods

DataSet and other object parameters can be accessed as MATLAB object properties. For example, to get the number of channels in a dataset, you simply reference the SizeC property of a DataSetReader object:

```
nChannels = obj.SizeC;
```

You can also access properties by directly calling get():

```
nChannels = get(obj, 'SizeC');
```

Data types

ImarisReader returns data according to the type specified by the HDF5 Type field, while ImarisXT sometimes converts data to a smaller bit size integer, e.g., int64 to int32. As a result, the data type of the data returned by ImarisReader sometimes differs from the data type of the data returned by the corresponding method in the ImarisXT interface. This does not affect the actual values, but can require a type conversion to perform arithmetic operations when mixing data generated using ImarisReader and ImarisXT.

GetData methods

Rather than having separate 'GetData' methods for DataSet objects for each of the possible data types (unsigned 8-bit integer, unsigned 16-bit integer, and single precision floating point), ImarisReader relies on MATLAB's built-in, automatic determination of the appropriate data type. For example, to get the data volume for channel 0 at time point 0:

```
volume = obj.DataSet.GetDataVolume(0, 0);
```

The three-dimensional array returned by GetDataVolume will be class 'uint8', 'uint16' or 'single' based on the data type in the Imaris DataSet. You can then cast the data to an alternative data type as needed.

GetMask method for Surfaces

Like ImarisXT, the SurfacesReader class includes a method to generate voxel masks for the isosurfaces in a Surfaces object. However, the mask creation algorithms in ImarisXT and ImarisReader are not identical, and often produce different results for a given surface. Therefore, if your analysis requires that you generate masks that are identical to those created by Imaris, you should use the XT interface.

Indexing

Imaris uses zero-based indexing, while MATLAB uses one-based indexing. Because individual component readers in an ImarisReader object are stored as properties of a MATLAB object, use one-based indexing to address them. For example, to get the first Surfaces object with an ImarisReader object use:

```
iSurfaces = obj.Surfaces(1);
```

However, to get the vertices for the first surface of the first Surfaces object, use:

```
vertices = obj.Surfaces(1).GetVertices(0);
```

Notes

In addition to the matlab.mixin.SetGet class, ImarisReader and all component readers are derived from the built-in dynamicprops class. As a result, you can define new properties as needed for reader objects.

Version history

Release date	Version	Notes
April 5, 2016	1.0	Initial release