Importing 3D Images

You've seen how to import a series of images as a 3D volume using the Volume Viewer app. You've also performed custom segmentations using the Volume Segmenter app. The Volume Viewer and Segmenter apps are useful tools to gain familiarity with your 3D images and perform common operations.

However, if you need to process many 3D images or perform highly customized operations, it might be more useful to analyze your volumes with code.

In this reading, you'll learn how to import, create, and view 3D objects in your scripts.

Table of Contents

Common Medical Image Formats

Many medical scanners use file types specific to medical imaging. In addition to common image file formats, MATLAB also supports many medical imaging formats such as:

- DICOM ("Digital Imaging and Communications in Medicine")
- IMA ("Image", a disk image very similar to DICOM)
- NIfTI ("Neuroimaging Informatics Technology Initiative")
- Analyze 7.5

Let's take a closer look at the images we used in the "Analyzing 3D Images" video:

```
ls("./T2 MRI Scan")
```

```
.           T2_12.IMA   T2_142.IMA  T2_165.IMA  T2_188.IMA  T2_37.IMA   T2_6.IMA    T2_82.IMA
..          T2_120.IMA  T2_143.IMA  T2_166.IMA  T2_189.IMA  T2_38.IMA   T2_60.IMA   T2_83.IMA
T2_1.IMA    T2_121.IMA  T2_144.IMA  T2_167.IMA  T2_19.IMA   T2_39.IMA   T2_61.IMA   T2_84.IMA
T2_10.IMA   T2_122.IMA  T2_145.IMA  T2_168.IMA  T2_190.IMA  T2_4.IMA    T2_62.IMA   T2_85.IMA
T2_100.IMA  T2_123.IMA  T2_146.IMA  T2_169.IMA  T2_191.IMA  T2_40.IMA   T2_63.IMA   T2_86.IMA
T2_101.IMA  T2_124.IMA  T2_147.IMA  T2_17.IMA   T2_192.IMA  T2_41.IMA   T2_64.IMA   T2_87.IMA
T2_102.IMA  T2_125.IMA  T2_148.IMA  T2_170.IMA  T2_2.IMA    T2_42.IMA   T2_65.IMA   T2_88.IMA
T2_103.IMA  T2_126.IMA  T2_149.IMA  T2_171.IMA  T2_20.IMA   T2_43.IMA   T2_66.IMA   T2_89.IMA
T2_104.IMA  T2_127.IMA  T2_15.IMA   T2_172.IMA  T2_21.IMA   T2_44.IMA   T2_67.IMA   T2_9.IMA
T2_105.IMA  T2_128.IMA  T2_150.IMA  T2_173.IMA  T2_22.IMA   T2_45.IMA   T2_68.IMA   T2_90.IMA
T2_106.IMA  T2_129.IMA  T2_151.IMA  T2_174.IMA  T2_23.IMA   T2_46.IMA   T2_69.IMA   T2_91.IMA
T2_107.IMA  T2_13.IMA   T2_152.IMA  T2_175.IMA  T2_24.IMA   T2_47.IMA   T2_7.IMA    T2_92.IMA
T2_108.IMA  T2_130.IMA  T2_153.IMA  T2_176.IMA  T2_25.IMA   T2_48.IMA   T2_70.IMA   T2_93.IMA
T2_109.IMA  T2_131.IMA  T2_154.IMA  T2_177.IMA  T2_26.IMA   T2_49.IMA   T2_71.IMA   T2_94.IMA
T2_11.IMA   T2_132.IMA  T2_155.IMA  T2_178.IMA  T2_27.IMA   T2_5.IMA    T2_72.IMA   T2_95.IMA
T2_110.IMA  T2_133.IMA  T2_156.IMA  T2_179.IMA  T2_28.IMA   T2_50.IMA   T2_73.IMA   T2_96.IMA
T2_111.IMA  T2_134.IMA  T2_157.IMA  T2_18.IMA   T2_29.IMA   T2_51.IMA   T2_74.IMA   T2_97.IMA
T2_112.IMA  T2_135.IMA  T2_158.IMA  T2_180.IMA  T2_3.IMA    T2_52.IMA   T2_75.IMA   T2_98.IMA
T2_113.IMA  T2_136.IMA  T2_159.IMA  T2_181.IMA  T2_30.IMA   T2_53.IMA   T2_76.IMA   T2_99.IMA
T2_114.IMA  T2_137.IMA  T2_16.IMA   T2_182.IMA  T2_31.IMA   T2_54.IMA   T2_77.IMA
T2_115.IMA  T2_138.IMA  T2_160.IMA  T2_183.IMA  T2_32.IMA   T2_55.IMA   T2_78.IMA
T2_116.IMA  T2_139.IMA  T2_161.IMA  T2_184.IMA  T2_33.IMA   T2_56.IMA   T2_79.IMA
T2_117.IMA  T2_14.IMA   T2_162.IMA  T2_185.IMA  T2_34.IMA   T2_57.IMA   T2_8.IMA
T2_118.IMA  T2_140.IMA  T2_163.IMA  T2_186.IMA  T2_35.IMA   T2_58.IMA   T2_80.IMA
T2_119.IMA  T2_141.IMA  T2_164.IMA  T2_187.IMA  T2_36.IMA   T2_59.IMA   T2_81.IMA
```

We see that our 3-dimmensional brain scan is composed many individual IMA files.

Reading Medical Images

Many of the functions we've used in the image processing toolbox have equivalents for DICOM and IMA files.

```
brain76Meta = dicominfo("T2_76.IMA")    % Display meta data for DICOM/IMA image
```

```
brain76Meta = struct with fields:
                           Filename: 'D:\Mina\Document\Coursera\Image Processing for Engineering and Science\Data'
                       FileModDate: '04-Jan-2022 17:37:10'
                           FileSize: 133222
                             Format: 'DICOM'
                      FormatVersion: 3
                              Width: 256
                             Height: 256
                           BitDepth: 16
                          ColorType: 'grayscale'
    FileMetaInformationGroupLength: 210
        FileMetaInformationVersion: [2×1 uint8]
           MediaStorageSOPClassUID: '1.2.840.10008.5.1.4.1.1.4'
        MediaStorageSOPInstanceUID: '1.3.6.1.4.1.9590.100.1.2.383125523312033013016153209983059058139'
                 TransferSyntaxUID: '1.2.840.10008.1.2.1'
            ImplementationClassUID: '1.3.6.1.4.1.9590.100.1.3.100.9.4'
         ImplementationVersionName: 'MATLAB IPT 9.4'
              SpecificCharacterSet: 'ISO_IR 100'
                         ImageType: 'ORIGINAL\PRIMARY\M\ND\NORM'
              InstanceCreationDate: ''
              InstanceCreationTime: ''
                      SOPClassUID: '1.2.840.10008.5.1.4.1.1.4'
                   SOPInstanceUID: '1.3.6.1.4.1.9590.100.1.2.383125523312033013016153209983059058139'
                        StudyDate: '19390825'
                       SeriesDate: '19390825'
                  AcquisitionDate: '19390825'
                      ContentDate: '19390825'
                        StudyTime: '163228.526000'
                       SeriesTime: '172035.762000'
```

```
             AcquisitionTime: '165659.237500'
                 ContentTime: '172035.920000'
             AccessionNumber: ''
                    Modality: 'MR'
                Manufacturer: ''
             InstitutionName: 'Shiz University'
          InstitutionAddress: '1 Yellow Brick Road, Emerald City, Oz'
      ReferringPhysicianName: [1×1 struct]
                 StationName: 'PoppyField'
           StudyDescription: 'IfOnlyaBrain'
          SeriesDescription: 'T2_tse_vfl_wave_1mm_R9_blip_RR'
InstitutionalDepartmentName: 'Department'
    PerformingPhysicianName: [1×1 struct]
               OperatorsName: [1×1 struct]
     ManufacturerModelName: ''
                 PatientName: [1×1 struct]
                   PatientID: '99.99.99-99:99:99-DST-9.9.99.9.9999.9.9.99.99999'
           PatientBirthDate: '19040115'
                 PatientSex: 'M'
                 PatientAge: '107Y'
                PatientSize: 1.7907
              PatientWeight: 61.2350
          BodyPartExamined: 'BRAIN'
          ScanningSequence: 'SE'
           SequenceVariant: 'SK\SP'
               ScanOptions: 'PFP'
         MRAcquisitionType: '3D'
              SequenceName: 'spcR_260ns'
                 AngioFlag: 'N'
             SliceThickness: 1
             RepetitionTime: 3200
                   EchoTime: 406
          NumberOfAverages: 1
          ImagingFrequency: 123.2540
             ImagedNucleus: '1H'
               EchoNumbers: 1
     MagneticFieldStrength: 3
 NumberOfPhaseEncodingSteps: 223
           EchoTrainLength: 229
           PercentSampling: 100
    PercentPhaseFieldOfView: 100
            PixelBandwidth: 590
        DeviceSerialNumber: '45407'
          SoftwareVersions: ''
              ProtocolName: 'T2_tse_vfl_wave_1mm_R9_blip'
          TransmitCoilName: 'Body'
         AcquisitionMatrix: [4×1 uint16]
InPlanePhaseEncodingDirection: 'ROW'
                 FlipAngle: 120
     VariableFlipAngleFlag: 'Y'
                       SAR: 0
                      dBdt: 0
           PatientPosition: 'HFS'
          StudyInstanceUID: '1.3.12.2.1107.5.2.19.45407.30000018102520545318300000028'
         SeriesInstanceUID: '1.3.12.2.1107.5.2.19.45407.2018102617182732149201999.0.0.0'
                   StudyID: '1'
              SeriesNumber: 21
         AcquisitionNumber: 1
            InstanceNumber: 76
      ImagePositionPatient: [3×1 double]
    ImageOrientationPatient: [6×1 double]
        FrameOfReferenceUID: '1.3.12.2.1107.5.2.19.45407.1.20181026163228673.0.0.4934'
 PositionReferenceIndicator: ''
             SliceLocation: -18.7376
```

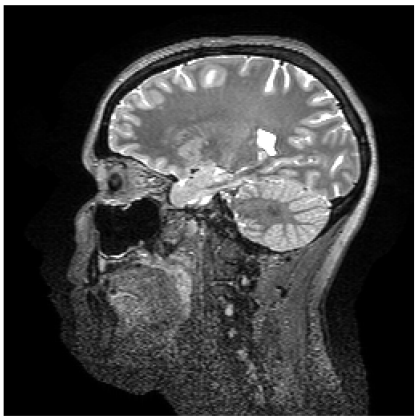
```
                    ImageComments: 'T2_tse_vfl_wave_1mm_R9_blip_RR←↲LW_181026_2/19900101/18.10.26-16:32:12-DST-
                  SamplesPerPixel: 1
           PhotometricInterpretation: 'MONOCHROME2'
                             Rows: 256
                          Columns: 256
                       PixelSpacing: [2×1 double]
                     BitsAllocated: 16
                        BitsStored: 16
                           HighBit: 15
                PixelRepresentation: 0
              SmallestImagePixelValue: 0
               LargestImagePixelValue: 648
                         WindowCenter: 210
                          WindowWidth: 496
    PerformedProcedureStepStartDate: '19390825'
    PerformedProcedureStepStartTime: '163228.586000'
           PerformedProcedureStepID: 'MR19390825163228'
PerformedProcedureStepDescription: 'FOLLOWYELLOWBRICKROAD'
```

```
imgBrain76 = dicomread("T2_76.IMA");     % Read in single image from 3D volume
```

Similar commands also exist for NifTI and Analyze 7.5 files.

Once we've read in the medical image with the appropriate command, we can use the tools we've already used in this course. Use `imshow` to display `imgBrain76`:

```
imshow(uint8(imgBrain76))
```



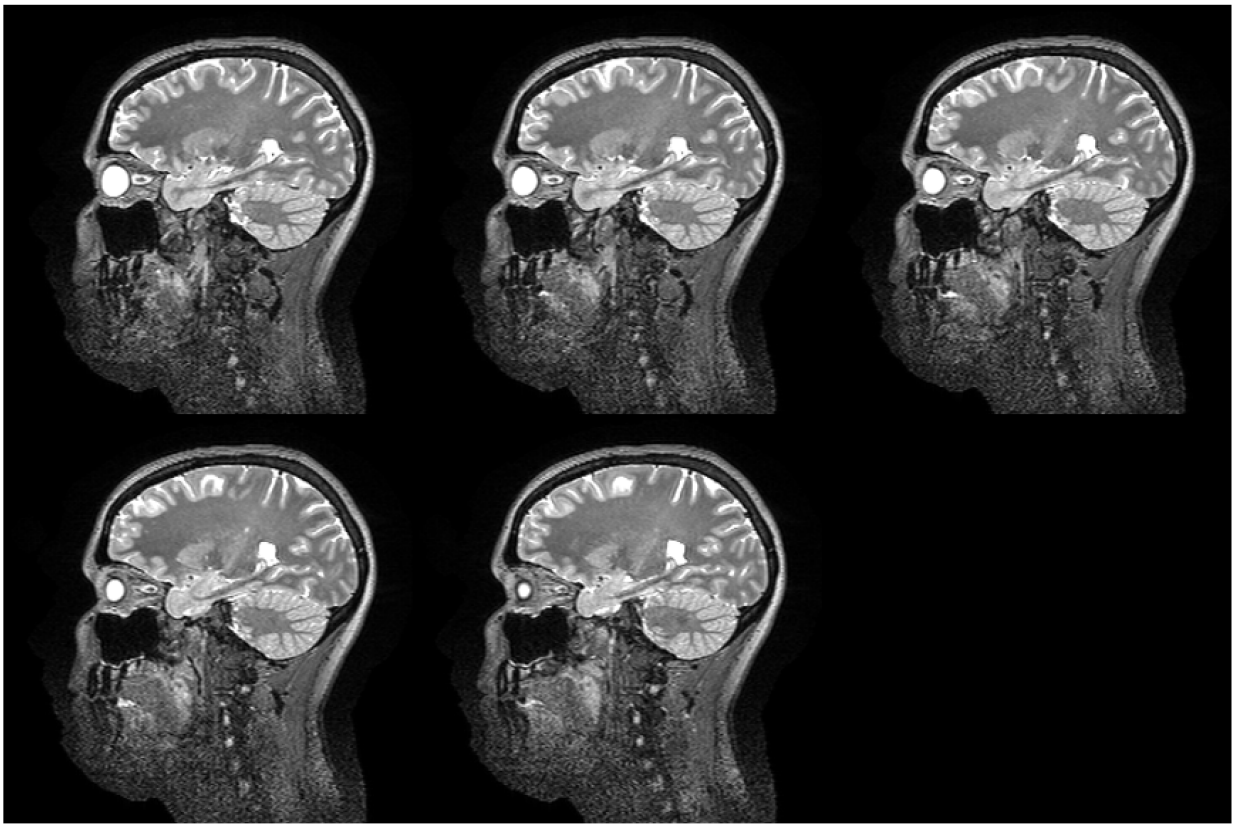
If you're having trouble seeing the image, you may want to review Viewing Meta-Data and non-uint8 Images.

3D Images as a Series of Images

Volumes are a series of images organized over x, y, and z spatial planes. One of the most intuitive ways to visualize this is to look at a series of 2D images taken at regular points over the third spatial plane.

Read in `T2_71.IMA` through `T2_75.IMA` and display them to get an idea of what the volume looks like in a section of the Y-Z (left to right, or "sagittal") plane.

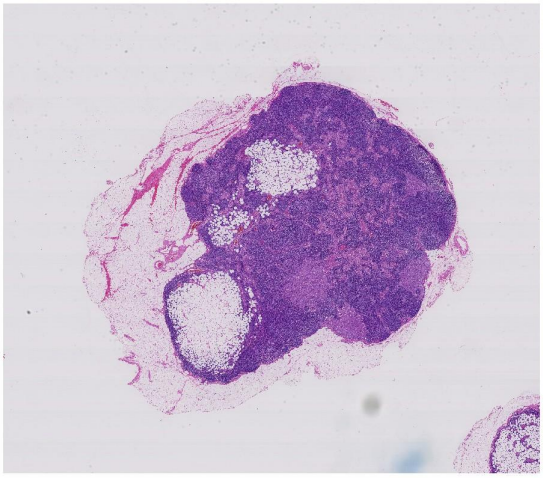
```
imgBrain71 = dicomread("T2_71.IMA");
imgBrain72 = dicomread("T2_72.IMA");
imgBrain73 = dicomread("T2_73.IMA");
imgBrain74 = dicomread("T2_74.IMA");
imgBrain75 = dicomread("T2_75.IMA");

montage({uint8(imgBrain71), uint8(imgBrain72), uint8(imgBrain73),
uint8(imgBrain74), uint8(imgBrain75)})
```



Viewing volumes by looking at a series of images in one plane, or slices, is very popular with radiologists.

3D Images as Volumes

While viewing volumes as series of 2D images makes it easy to visualize 3D objects on the 2D surface of your screen, it also has drawbacks. Consider this image of a single slice of a stained tumor:



We can see the size of the removed tumor at a single cross-section, but usually we would want to see the whole tumor before it's removed in order to make treatment decisions.

Creating a Volume Object

Creating a volume allows us to see images on our screen with an approximation of the depth we see in the real world. It also allows us to calculate useful parameters such as volume and surface area.

As with our `/'T2 MRI Scan'` directory, 2D slices of a volume will usually be saved with sequential names. This enables MATLAB to assemble the images from a folder in the correct order to create a volume.

Before you can create the volume, you first need to set the path to the `'T2 MRI Scan'` folder provided with the course files:

```
mrDir = uigetdir
```

```
mrDir =
'D:\Mina\Document\Coursera\Image Processing for Engineering and Science\Data\MathWorks Images\T2 MRI Scan'
```

Now you are ready to load the volume:

```
vol = dicomreadVolume(fullfile(mrDir));
```

Removing extra dimensions

You might notice something strange about our new, 3D volume:

```
size(vol)
```

```
ans = 1×4
   256   256     1   192
```

Why does our 3D volume have 4 dimensions? And why does our 3rd dimension have a length of 1? Because our images, like many medical images, are grayscale, the "color" dimension of our volume is empty, leading to a "singleton" dimension. Let's remove it and check the size again:

```
vol = squeeze(vol);
size(vol)
```

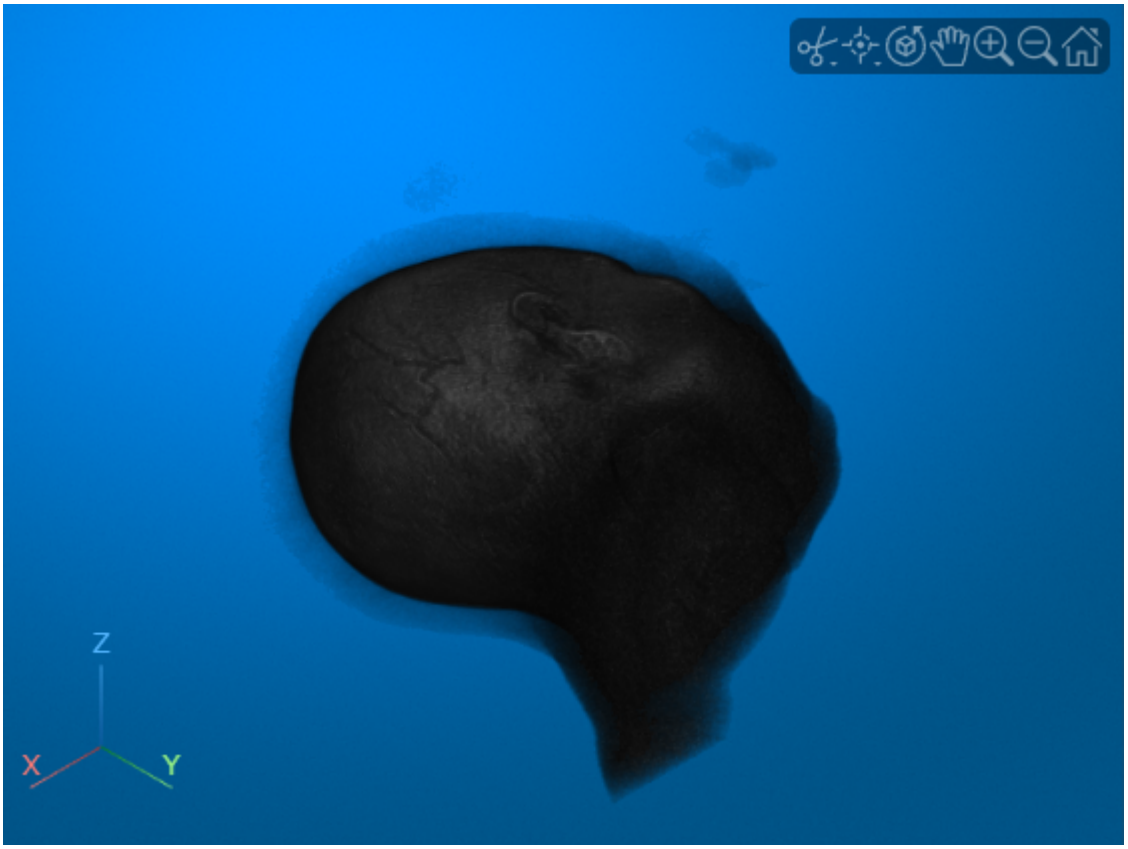
```
ans = 1×3
   256   256   192
```

That makes a lot more sense!

Viewing a Volume Object

Now we're finally ready to see our volume:

```
volshow(vol);
```



Now that is an impressive head. Let's check out some individual slices:

```
figure
sliceViewer(vol)
```

```
ans =
  sliceViewer with properties:

            SliceDirection: [0 0 1]
               SliceNumber: 97
                    Parent: [1×1 Panel]
                  Colormap: [256×3 double]
              DisplayRange: [0 740]
              ScaleFactors: [1 1 1]
    DisplayRangeInteraction: 'on'
```

You can scroll through slices from different planes using the `SliceDirection` parameter:

```
sliceViewer(vol,"SliceDirection","X")
```

```
ans =
  sliceViewer with properties:

             SliceDirection: 'X'
                SliceNumber: 97
                     Parent: [1×1 Panel]
                   Colormap: [256×3 double]
               DisplayRange: [0 740]
               ScaleFactors: [1 1 1]
    DisplayRangeInteraction: 'on'
```

Keep in mind, if your imported image isn't oriented correctly, the slice view direction might not correspond to the plane you expect. Try viewing the volume in different slice planes:

```

```

Once you have a volume object, you can view it, perform manual calculations, or import it from the workspace into the Volume Segmenter app.