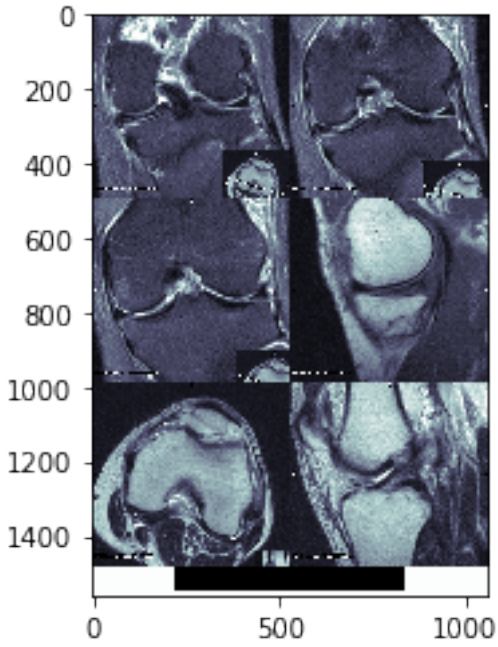
dicom_aim_markup_reader

October 1, 2018

In [28]:
```
# authors : Guillaume Lemaitre <g.lemaitre58@gmail.com>
# authors: Thomas Tsai <thomas@nchc.org.tw>
# license : MIT
# dicom standard https://dicom.innolitics.com/ciods
import matplotlib.pyplot as plt
import pydicom
from pydicom.data import get_testdata_files

print(__doc__)

# FIXME: add a full-sized MR image in the testing data
#filename = get_testdata_files('MR_small.dcm')[0]
#filename="/home/thomas/tmp/dicom/24759123/20010101/OT999999/20057"
#filename="/home/thomas/tmp/dicom/24759123/20010101/MR2/19665"
#filename="/home/thomas/work_house/jupyter/myenv/7-260.dcm"
filename = "test.dcm"
ds = pydicom.dcmread(filename)

# get the pixel information into a numpy array
dicom_img_data = ds.pixel_array
plt.imshow(ds.pixel_array, cmap=plt.cm.bone)

#print('The image has {} x {} voxels'.format(data.shape[0], data.shape[1]))
#data_downsampling = data[::8, ::8]
print('The downsampled image has {} x {} voxels'.format(
    data_downsampling.shape[0], data_downsampling.shape[1]))

# copy the data back to the original data set
ds.PixelData = data_downsampling.tobytes()
# update the information regarding the shape of the data array
ds.Rows, ds.Columns = data_downsampling.shape

# print the image information given in the dataset
print('The information of the data set after downsampling: \n')
print(ds)
print("SOP Instance UID")
print(ds[0x08,0x18].value)
```

```
Automatically created module for IPython interactive environment
The downsampled image has 64 x 64 voxels
The information of the data set after downsampling:

(0008, 0008) Image Type                          CS: ['DERIVED', 'SECONDARY']
(0008, 0012) Instance Creation Date              DA: '20030402'
(0008, 0013) Instance Creation Time              TM: '095251.969'
(0008, 0014) Instance Creator UID                UI: 1.3.6.1.4.1.5962.3
(0008, 0016) SOP Class UID                       UI: Secondary Capture Image Storage
(0008, 0018) SOP Instance UID                    UI: 1.3.6.1.4.1.5962.1.1.0.0.0.1194732126.1303
(0008, 0020) Study Date                          DA: '20010101'
(0008, 0023) Content Date                        DA: '20010101'
(0008, 0030) Study Time                          TM: '000000'
(0008, 0033) Content Time                        TM: '005530.969'
(0008, 0050) Accession Number                    SH: '2'
(0008, 0060) Modality                            CS: 'OT'
(0008, 0064) Conversion Type                     CS: 'WSD'
(0008, 0070) Manufacturer                        LO: ''
(0008, 0090) Referring Physician's Name          PN: ''
(0008, 0201) Timezone Offset From UTC            SH: '+0000'
(0008, 1030) Study Description                   LO: 'Lower Extremity^Knee'
(0008, 103e) Series Description                  LO: 'MONTAGE'
(0008, 1090) Manufacturer's Model Name           LO: ''
(0008, 2111) Derivation Description              ST: 'DRS:DOE, HARRY    24759123  1/01/01  3
(0010, 0010) Patient's Name                      PN: 'Doe^Harry'
(0010, 0020) Patient ID                          LO: '24759123'
(0010, 0030) Patient's Birth Date                DA: ''
(0010, 0040) Patient's Sex                       CS: 'M'
(0010, 1010) Patient's Age                       AS: '054Y'
(0012, 0062) Patient Identity Removed            CS: 'YES'
(0012, 0063) De-identification Method            LO: 'dcanon; Burned in text blacked out'
(0018, 1010) Secondary Capture Device ID         LO: '3_92'
(0018, 1012) Date of Secondary Capture           DA: '20010101'
(0018, 1014) Time of Secondary Capture           TM: '005530.969'
(0018, 1016) Secondary Capture Device Manufactur LO: 'DR Systems, Inc.'
(0018, 1018) Secondary Capture Device Manufactur LO: 'Dominator'
(0018, 1019) Secondary Capture Device Software V LO: '6.1B64D16'
(0018, 1020) Software Version(s)                 LO: 'syngo MR 2002B 4VA21A'
(0020, 000d) Study Instance UID                  UI: 1.3.6.1.4.1.5962.1.1.0.0.0.1194732126.1303
(0020, 000e) Series Instance UID                 UI: 1.3.6.1.4.1.5962.1.1.0.0.0.1194732126.1303
(0020, 0010) Study ID                            SH: '2'
(0020, 0011) Series Number                       IS: '999999'
(0020, 0013) Instance Number                     IS: '0'
(0020, 0020) Patient Orientation                 CS: ''
(0028, 0002) Samples per Pixel                   US: 1
(0028, 0004) Photometric Interpretation          CS: 'MONOCHROME2'
(0028, 0008) Number of Frames                    IS: '1'
(0028, 0010) Rows                                US: 64
```

```
(0028, 0011) Columns                                US: 64
(0028, 0100) Bits Allocated                         US: 8
(0028, 0101) Bits Stored                            US: 8
(0028, 0102) High Bit                               US: 7
(0028, 0103) Pixel Representation                   US: 0
(0028, 0301) Burned In Annotation                   CS: 'NO'
(0028, 1050) Window Center                          DS: "128"
(0028, 1051) Window Width                           DS: "256"
(0028, 2110) Lossy Image Compression                CS: '01'
(0028, 2112) Lossy Image Compression Ratio          DS: "4"
(7fe0, 0010) Pixel Data                             OB: Array of 8192 bytes
SOP Instance UID
1.3.6.1.4.1.5962.1.1.0.0.0.1194732126.13032.0.112
```




```
In [21]: import xmltodict
         import json
         with open('test.aim.xml') as fd:
             doc = xmltodict.parse(fd.read())


         imgid = doc['ImageAnnotationCollection']['imageAnnotations']['ImageAnnotation']['marku

         print("the reference image id:", imgid)

         #imageAnnotations
```

```
        #imagingObservationEntityCollection
        #markupEntityCollection
        data = doc['ImageAnnotationCollection']['imageAnnotations']['ImageAnnotation']['marku

        #print(data)
        #print(json.dumps(data))
        polydata=[]
        for p in data:
            index = (p['coordinateIndex']['@value'])
            x = (p['x']['@value'])
            y = (p['y']['@value'])
            polydata.append([index, x, y])

        print(polydata)
```

```
the reference image id: 1.3.6.1.4.1.5962.1.1.0.0.0.1194732126.13032.0.112
[['0', '830.910400390625', '778.837463378906'], ['1', '821.610900878906', '771.86279296875'],
```

```
In [32]: import numpy as np
         import cv2
         from matplotlib import pyplot as plt

         #  512x512  RGB
         #img = np.zeros((1024, 1024, 3), np.uint8)
         img = dicom_img_data
         #  (200, 200, 200)
         #img.fill(200)

         #  5 px
         #cv2.line(img, (0, 0), (255, 255), (0, 0, 255), 5)

         #
         #pts = np.array([[170, 55], [165, 75], [220, 80], [200, 60]], np.int32)
         pts_data = []
         #print(polydata)
         for poly_x_y in polydata:
             index, x, y = poly_x_y
             newx = int(float(x))
             newy = int(float(y))
             pts_data.append([newx, newy])
         print(pts_data)
         pts = np.array(pts_data, np.int32)

         #  (, 1, 2)
         pts = pts.reshape((-1, 1, 2))

         #
```

```
        cv2.polylines(img, [pts], True, (255, 255, 255), 10)

        #
        #cv2.imshow('My Image', img)



        #
        #cv2.waitKey(0)
        #cv2.destroyAllWindows()

        #show_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        show_img = img
        plt.imshow(show_img, cmap=plt.cm.bone)
        plt.show()
```

```
[[830, 778], [821, 771], [814, 762], [807, 755], [805, 748], [805, 736], [805, 727], [807, 720]
```