accusoft®

Using PICTools to Decode DICOM Pixel Data

Miguel Restrepo
SDK Software Development Manager

DICOM stands for Digital Imaging and Communications in Medicine. It is the standard format used for the management and communication of medical imaging information and associated data. It enables an integrated environment of medical imaging services such as computing tomography (CT), magnetic resonance imaging (MRI), and others. DICOM actually includes protocols and definitions for image storage and compression, image visualization, image presentation and reports on patient data and results. This blog will focus on the image component of DICOM files, and specifically on how to access and decode DICOM pixel data using Accusoft's PICTools.

PICTools Medical

Accusoft offers a medical-specific version of PICTools called PICTools Medical. It supports lossy/lossless sequential JPEG, JPEG 2000, and JPEG-LS and can be used to encode or decode pixel data associated with these Transfer Syntax UIDs:

- 1.2.840.10008.1.2.4.50 Lossy JPEG 8-bit samples
- 1.2.840.10008.1.2.4.51 Lossy JPEG 12-bit samples
- 1.2.840.10008.1.2.4.70 Lossless JPEG 2-bit to 16-bit samples
- 1.2.840.10008.1.2.4.80 Lossless JPEG-LS 2-bit to 16-bit samples
- 1.2.840.10008.1.2.4.81 Lossy JPEG-LS 2-bit to 16-bit samples
- 1.2.840.10008.1.2.4.90 Lossless JPEG 2000 2-bit to 16-bit samples
- 1.2.840.10008.1.2.4.91 Lossy JPEG 2000 2-bit to 16-bit samples
- 1.2.840.10008.1.2.4.92 Lossless JPEG 2000 3D 2-bit to 16-bit samples
- 1.2.840.10008.1.2.4.93 Lossy/Lossless JPEG 2000 3D 2-bit to 16-bit samples

According to the PICTools Programmer's Guide, the approach taken by PICTools is to organize image compression/decompression technology into opcodes. Opcodes are basically modular cohesive libraries to enable said technology. In the case of PICTools Medical, here are the specific opcodes that are of interest:

- OP_D2S(10) - High-Speed Sequential JPEG Compression
- OP_S2D(11) - High-Speed Sequential JPEG Decompression
- OP_D2SE(12) - High-Speed Sequential JPEG and ePIC Compression
- OP_SE2D(18) - High-Speed Sequential JPEG and ePIC Decompression
- OP_D2SEPLUS(64) - Sequential JPEG Compression Plus High Gray Support
- OP_SE2DPLUS(65) - Sequential JPEG Decompression Plus High Gray Support
- OP_J2KPRGB(66) - JPEG 2000 Compression
- OP_J2KERGB(67) - JPEG 2000 Decompression
- OP_J2KP(68) - JPEG 2000 Compression Plus High Bit Depths
- OP_J2KE(69) - JPEG 2000 Decompression Plus High Bit Depths
- OP_J2KP3D(78) – JPEG 2000 Part 2 3D Compression
- OP_J2KE3D(79) – JPEG 2000 Part 2 3D Decompression
- OP_J2KTRANSCODE(28) - JPEG 2000 Transcoder
- OP_JPIPCLIENT(59)
- OP_JPIPSERVER(58)
- OP_JLSP(60) - Compress DIB or RAW to JPEG-LS
- OP_JLSE(61) - Expand JPEG-LS to DIB or Raw
- OP_LIP3(46) - Lossless Image Compression (Lossless JPEG)
- OP_LIE3(47) - Lossless Image Decompression (Lossless JPEG)
- OP_LIP3PLUS(62) - Lossless Image Compression Plus High Gray Support
- OP_LIE3PLUS(63) - Lossless Image Decompression Plus High Gray Support
- OP_D2FPLUS(84) - Standard File Format Creation Plus GIF/LZW Support
- OP_F2DPLUS(85) - Standard File Format Reading Plus GIF/LZW Support
- OP_HDPHOTOP(26) - HD Photo Compression
- OP_HDPHOTOE(27) - HD Photo Decompression

PICTools Medical also supports:

- ISO 14495-1 (Accusoft's JPEG LS)
- ISO 15444-1 (Accusoft's JPEG 2000)
- ISO 15444-2 (Accusoft's JPEG 2000 3D)
- ISO 15444-9 (Accusoft's JPIP)

For a more complete reference on DICOM Unique identifiers, please refer to the DICOM PS3.6 2019b Data Dictionary which includes Transfer Syntax UIDs used.

In the distribution of PICTools Medical, for each of the opcodes mentioned, a sample program that's ready to use is included. For example, the lossy sequential JPEG decoder opcode, OP_SE2DPLUS, is demonstrated with the sample se2dnw.exe (Win) or se2dplx (Linux).

Accessing Pixel Data from DICOM Files

It is important to consider that pixel data is not found at the start of DICOM files and most PICTools decoder samples only handle reads from the start of input files. Also, a single DICOM file can contain multiple images so getting the information to get to a specific image's pixels can be an elaborate process. With these points in mind, the main process to access pixel data from a DICOM file is as follows:

1. Determine the specifics of a given DICOM file. This step involves getting the tags and metadata included in the DICOM file. This can be accomplished with several tools, such as DICOMparser by Rubo Medical. For our example, you can also use Imagemagick, in particular the identify command like this:

   identify -verbose example.dcm

   The results from this command will be fairly complete and can be extensive, especially if the file contains multiple images. From the information results, find the specifics on the Transfer Syntax UID (followed by SOP Class UID if needed). For example, we can get:

   ```
   (0002), (0010)  UI        22  Transfer Syntax UID
   1.2.840.10008.1.2.4.50
   (0008), (0016)  UI        26  SOP Class UID
   1.2.840.10008.5.1.4.1.1.7
   ```

   From this information, we infer that the image data corresponds to a Lossy JPEG 8-bit as per the Transfer Syntax UID.

Also note that the image tags describe the pixel data. It is a good practice to analyze the rest of the data obtained from the tags in the DICOM file. In our example:

(0028), (0002) US 2 Samples per Pixel
1
(0028), (0004) CS 12 Photometric Interpretation
MONOCHROME1
(0028), (0006) US 2 Planar Configuration
20301
(0028), (0010) US 2 Rows
2040
(0028), (0011) US 2 Columns
2570
(0028), (0030) DS 14 Pixel Spacing
0.1136\0.1136
(0028), (0100) US 2 Bits Allocated
8
(0028), (0101) US 2 Bits Stored
8
(0028), (0102) US 2 High Bit
7
(0028), (0103) US 2 Pixel Representation
0

2. Next, we need to determine the starting point of where the pixel data starts and ends. This step can be tricky if the DICOM file contains multiple images, for example. We will not get into much detail on how to determine this, but in the case of our example, we get:

There were 1918 leading skipped bytes. Assume DICOM image.
There are 1157109 total bytes.

3. Once we have the starting point and the total number of bytes for the location of the pixels in the DICOM file, the actual image pixel information can be extracted. This can be accomplished with a simple program in languages such as C or Python that can handle binary data. A sample code snippet in Python is as follows:

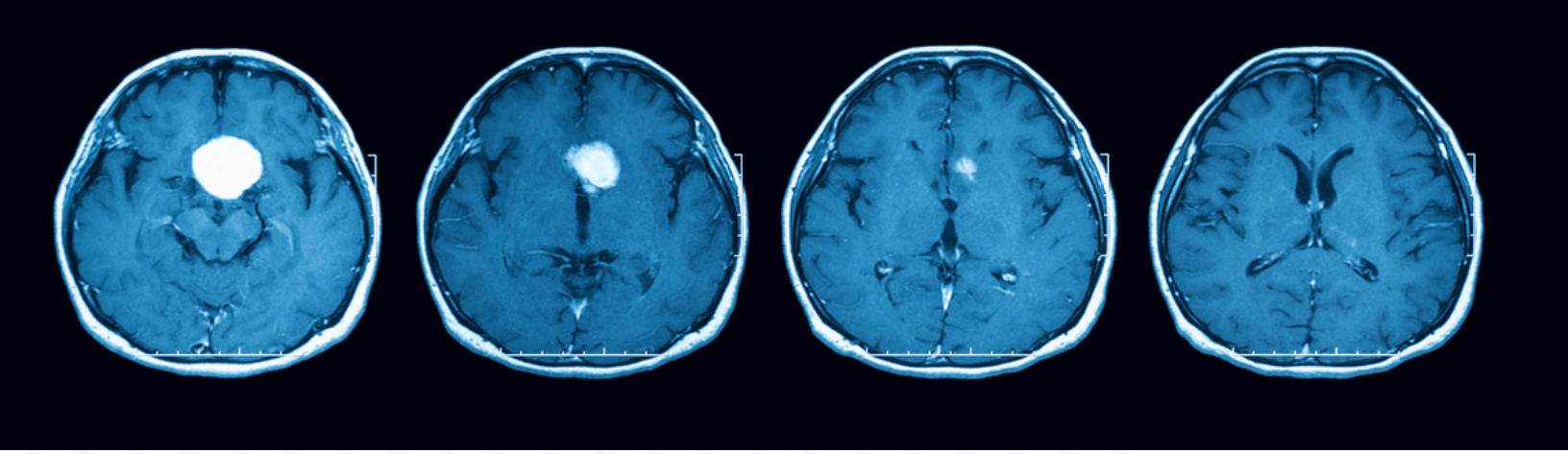
```
import os
def slice_file(input_path, offset, count, output_path):
    """ Partition a file into fixed-size files.

    Args:
        input_path (string): The path to the file to slice.
        offset (integer): The starting position relative to the beginning of the file.
        count (integer): The number of bytes to write.
        output_path (string): The path of the file to write.
```

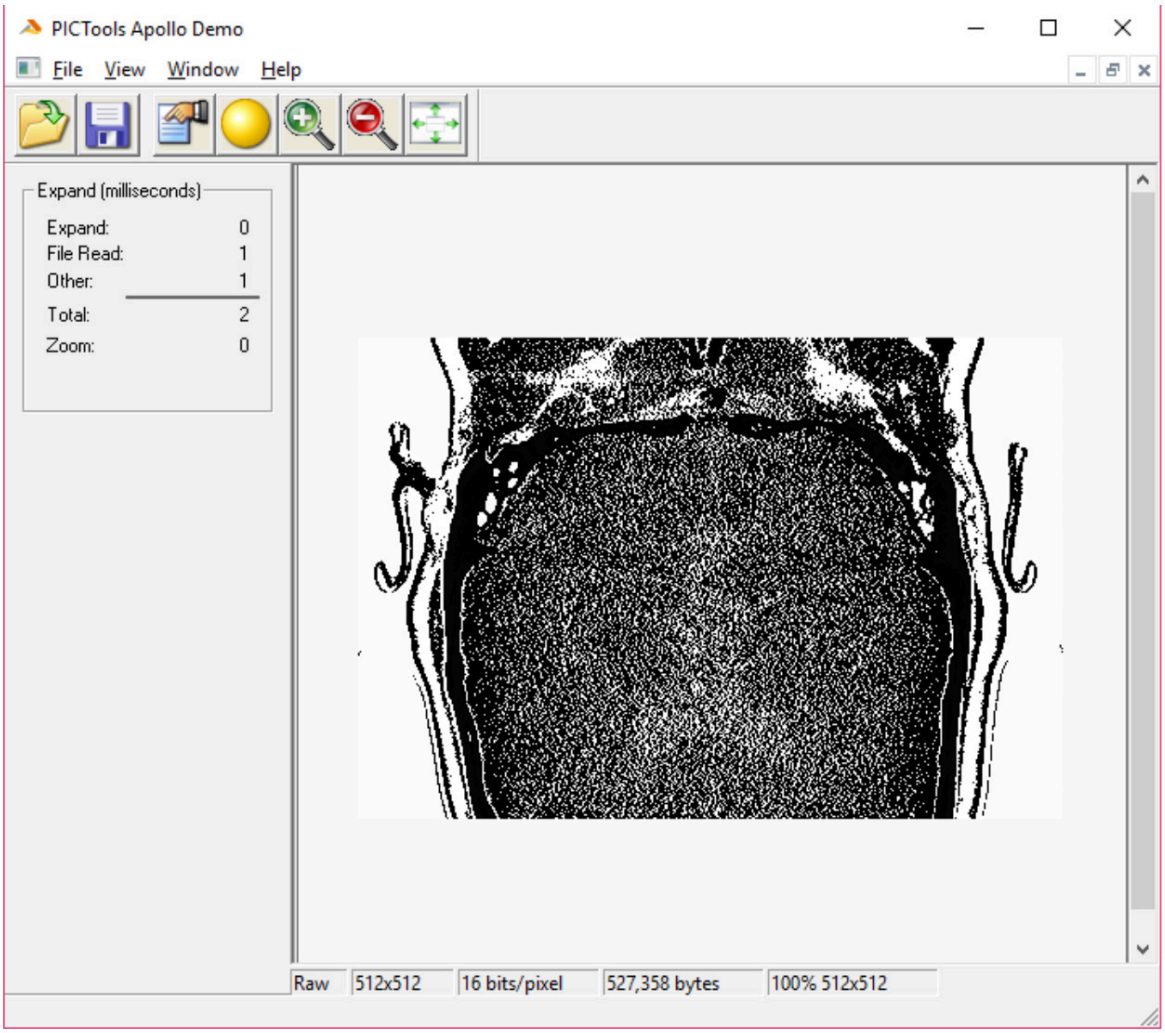
```
        """
        if offset <= 0:
            raise ValueError ('%s is expected to be a positive number. Value
    is %d.' % ('offset', offset))
    if count <= 0:
        raise ValueError ('%s is expected to be a positive number. Value
is %d.' % ('count', count))
    with open (input_path, 'rb') as s:
        s.seek (offset, os.SEEK_SET)
        with open (output_path, 'wb') as t:
            count = t.write (s.read(count))
            print ('Wrote %d bytes to %s' % (count, output_path))
```

So for example, if we incorporate the code snippet into a Python program called slice.py **that takes the offset and number of bytes to produce an independent JPEG file with the pixel data:**

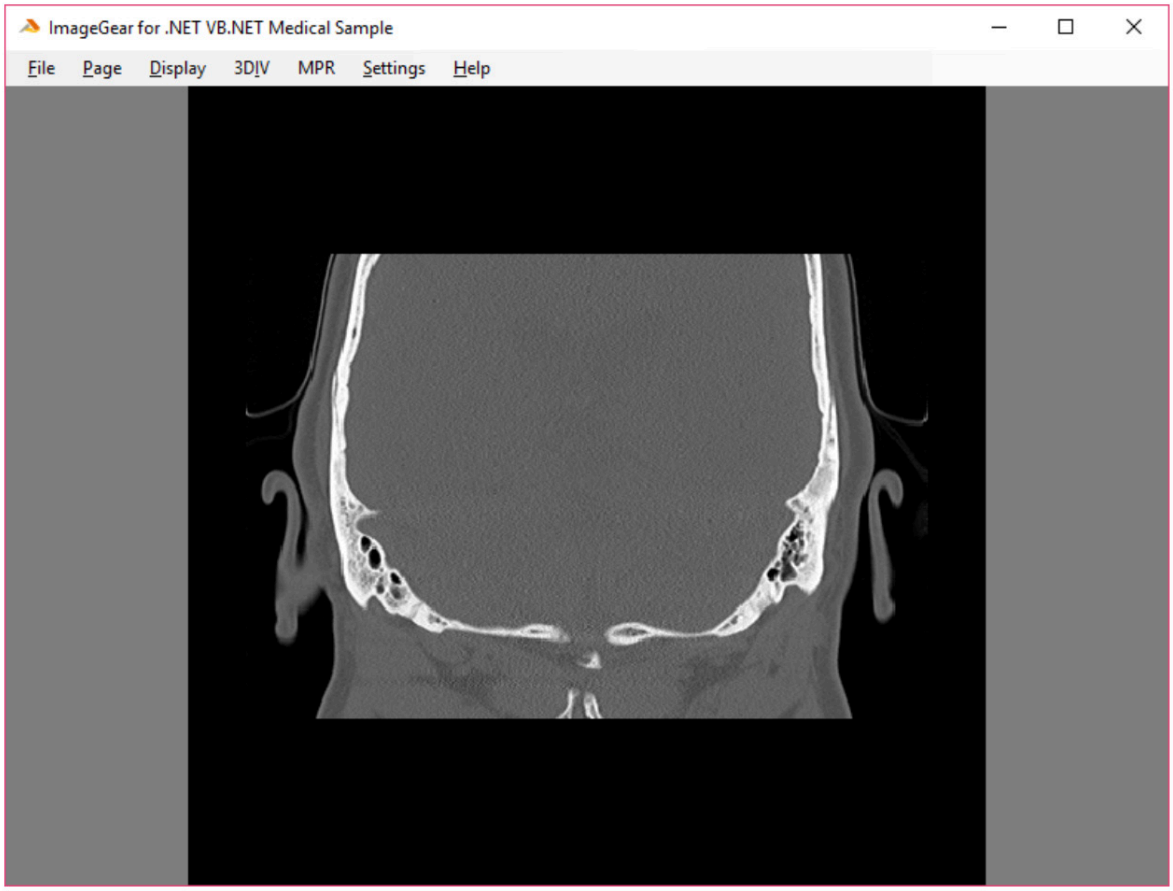
```
python slice.py -f example.dcm -s 1918 1157109
example.jpg
```



4. At this point, it is possible to visualize the image with any of several possible methods, such as the display command in Imagemagick, or with the Apollo image visualizing sample program in the Win 32-bit distribution of PICTools Medical:



In this example image (from a sinus CT scan), the pixel data is 16-bpp grayscale and Apollo does not handle medical grayscale transforms to 8-bpp. Opening the DICOM file with Accusoft's ImageGear for .NET (IGNET) will account for Window Center, Window Width, and Photometric Interpretation:



5. The main purpose of this process is to decode the pixel information. JPEG files are a possible final format to produce, but if the need is to produce a bitmap object or file instead, for a lossy JPEG file, we can further do:

    sep2dlx example.jpg example.bmp

Some Extra Notes:

Sometimes the pixel data in a DICOM file may come as uncompressed, raw image data. This is detected in the DICOM tags in this way:

```
...
(0002), (0010)  UI    20  Transfer Syntax UID
1.2.840.10008.1.2.1
...
(0028), (0002)  US     2  Samples per Pixel          1
(0028), (0004)  CS    12  Photometric Interpretation
MONOCHROME2
(0028), (0010)  US     2  Rows                      512
(0028), (0011)  US     2  Columns                   512
(0028), (0030)  DS    18  Pixel Spacing
0.468000\0.468000
(0028), (0100)  US     2  Bits Allocated            16
(0028), (0101)  US     2  Bits Stored               16
(0028), (0102)  US     2  High Bit                  15
(0028), (0103)  US     2  Pixel Representation       1
(0028), (1050)  DS     4  Window Center             350
(0028), (1051)  DS     4  Window Width
2700
(0028), (1052)  DS     2  Rescale Intercept         0
(0028), (1053)  DS     2  Rescale Slope            1
...
(7fe0), (0010)  OW 524288  Variable Pixel Data
```

Transfer syntax 1.2.840.10008.1.2.1 is for “Explicit VR Little Endian” and indicates that pixel data is uncompressed.

In this situation, it is possible to extract the pixel data as ‘raw’ like this:

```
python  slice.py  -f  example2.dcm  -s  3070  524288  example2.raw
```

To produce a bitmap file in this case, OP_ZOOM2(88) opcode can help us. It can also downsample 16-bpp to 8-bpp and save to a bitmap:

```
zoom2nnw - raw16:512:512:1024:0  -nbc8 example2.raw example2.bmp
```

References:

- PICTools Medical distribution by Accusoft
- DICOM standard and reference - https://www.dicomstandard.org/current/
- Imagemagick - open source image utilities (multiplatform) - https://imagemagick.org/
- DICOMparser free utility - http://www.rubomedical.com/dicom_parser/index.html
- PICTools Programmer's Reference guide - included as a searchable PDF in the PICTools distributions by Accusoft. https://help.accusoft.com/PICTools/ProgrammersReference/webframe.html
- Rod McMullen, Sr. Software Engineer for SDK at Accusoft - personal communications