CMPT 371 – DICOM REPORT

Virtual Reality Medical Imaging Software with Luxsonic Technologies Inc.

JANUARY 31, 2017

WHAT IS DICOM?

When we refer to DICOM, we are talking about the Digital Imaging and Communication in Medicine standard. It is a universal standard created to help in transmitting medical images and information between professionals. Such professionals include, but are not limited to radiologists, cardiologists, neurologists, dermatologists, dentists, and veterinarians. The devices they use that create DICOM files include:

- Magnetic Resonance Imaging (MRI)
- Ultrasound
- X-Ray
- Microscopy
- Endoscopy
- Angiography
- Image Printers
- And several different record and reporting systems

Having a DICOM standard helps to enable various medical imaging devices from different manufacturers be able to integrate with each other. For example, a patient complaining of chest pain might require both an x-ray and an ultrasound to determine the problem. The standard itself is comprised of 20 different parts describing the data dictionary, object definitions, network communications, and general architecture and syntax of reports. The entire standard can be found here.

Like other image files, DICOM files are comprised of two main components: a header and pixel data. What is important to us is the header, which contains information like patient name, date, details about the patient, and image details (such as dimensions and bits per pixel). Everything in a DICOM file can be viewed as an object. All objects in DICOM files follow the standard of Information Object Definitions (IODs). IODs are data abstractions of real world objects, which are defined by their attributes. Essentially, this means that DICOMs are just a series of abstract classes made up of different attributes. The attributes consist of things like patient name, study date, physician name, and so forth. These attributes are represented by one of 27 different possible data types. To make it easier for medical professionals to understand, the DICOM standard refers to these data types as 'value representations'. Each data type (or Value Representation, if you prefer) is represented by two capital letters. For example, DT is the data type for Date Time. Each of the 27 data types is unique, and ranges from very specific sequences of strings and characters to simple values like numbers.

Since DICOM is a standard, all this information is stored as tags with values. Each attribute has a 4-byte tag that is divided into two parts: a group tag and an element tag. Therefore, a single tag for some attribute consists of eight characters. For example, the tag `0100 0010` represents the attribute 'Patient's Name' that has a data type of PN (which means Person

Name). To avoid future confusion, neither the element tag nor the group tag are specific to data types. The tag `0008 0090` represents the attribute 'Referring Physician's Name' with a data type of PN. Notice that both the group tags and element tags do not match. How the tags are broken up are not representative of how we as computer scientists would like to view them. Some tags are considered mandatory to have for each DICOM file and are thus tagged more on a priority basis than on a class basis. The dictionary of what every tag represents can be found in Part 5 of the DICOM standards.

The DICOM file format consists of several key features:

- A short preamble, which explains what is in the file
- DIMC character prefix (signifies that the file is a DICOM file)
- File Meta header (which includes information on transfer syntax which is based on how the machine which created the file packaged it)
- Data set containing the data values with their attributes (which is what we want!)
- Pixel data contains information on the image or images.

Different machines and manufacturers can save DICOMs slightly differently. This difference is important for DICOM viewers so that they can be able to read and interpret the files from a wide range of manufacturers.

How Do We Deal With DICOM?

While integrating DICOM creation and decoding within our program may be a challenge, there fortunately several options available for us. One option would be to use a well-known DICOM viewer to decipher our chosen DICOM files. Once done, we can acquire the images as JPEGs or PNGs and easily download them into our program. Such DICOM file viewers include Philips, RadiAnt, MicroDicom, OsiriX, Sante, ezDICOM, and several more. All of these DICOM viewers are free to download and use. While this may seem like the simplest way to integrate into our program, we must consider some of the legal restrictions that surround developing programs for commercial purposes that integrate with others. Using a third party viewer would be appropriate if we decided to continue down the route of having what is essentially a virtual reality of the desktop.

To avoid any legal problems or reliance on outside programs, we would ideally want to download and decipher DICOM files ourselves. To accomplish this, several libraries would have to be provided for us. For starters, we would need access to a library of tags and attributes; otherwise, we would have to spend a tremendous amount of time hard coding several hundred lines of code. Fortunately, this library is readily accessible to everyone (and in C#, no less!). Unfortunately, the libraries that are available for use are only partially complete, meaning that we will be unable to decipher a few hundred types of attributes. While this could be problematic, the attributes omitted from the libraries online are those that are more obscure

and not commonly used. So how do we use the library to extract the image we need? In order to achieve this, the library must be used to determine where the pixel data is in the file. Once extracted, the following information must be checked and further extracted:

- The number of images stored in the file: Single frame or multi-frame
- Number of bits per pixel, big Euclidean or small Euclidian
- Compression type (JPEG, RLE)
- Other than established DICOM viewers.
- Photometric interpretation of each image (Grayscale, RGB)

There are currently resources available online in C# that allow for the extraction and view of various DICOM images. Using these within our program could violate legal restrictions on commercial use of the code. Based on the code that is available, we can get a picture of what a decoder should look like when implemented. In order to implement it, considerable amount of time would have to be used in order to

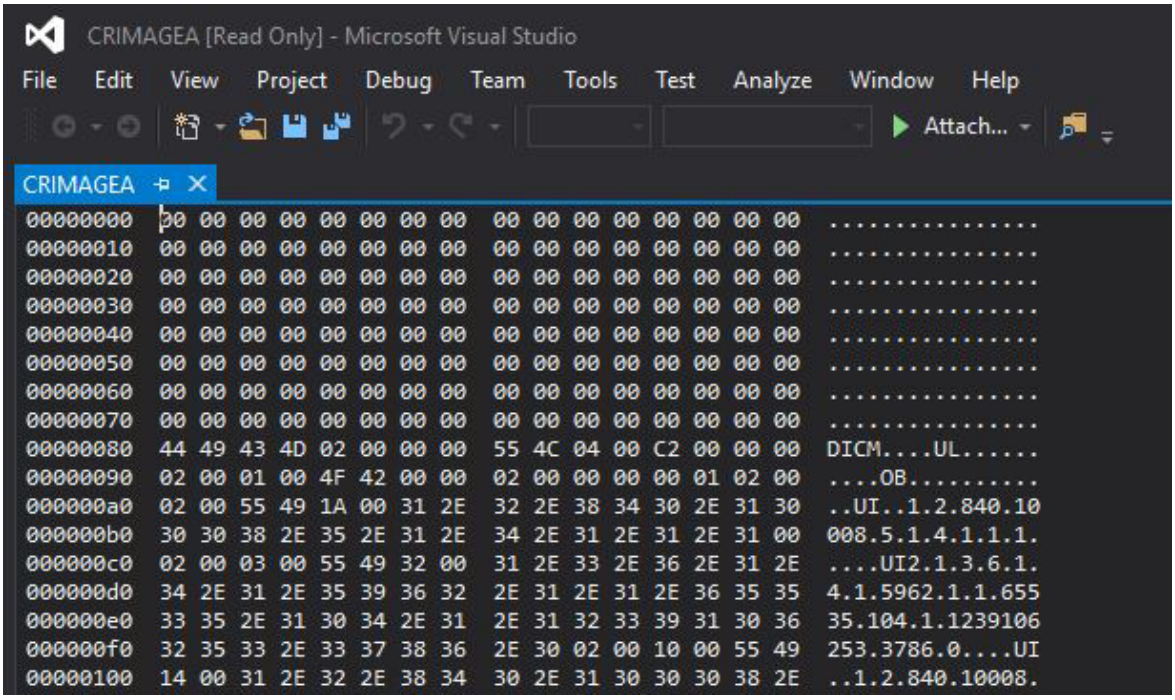

Figure 1. Screenshot of what a DICOM file looks like in Microsoft Visual Studio. File CRIMAGEA was obtained online through an open source library of DICOMs. This only represents a few out of a couple hundred lines of information.

create a functioning program that could decode and store DICOM images. Such an endeavor would be a project in and of itself, likely not possible for the limited time given to complete the project.

Microsoft Visual Studio is one of the few programs that is able to open and look at DICOM files (as shown in Figure 1). Attempting to do so in programs that are strictly ASCII based like

Notepad, WordPad, or Notepad++, or other forms of text editors will result in a mess of unreadable garbage. Notice in Figure 1 that in the file CRIMAGEA, the left hand column is made up eight characters (the group and element tags). Under the DICOM standard, all DICOM files must list the tags used in ascending order. The first tag is 0000 0000 and the last tag is 0000 0100 (remember that characters are used as well).

While the ability to read each DICOM file helps us to understand its structure, it is not overly useful if we are unable to actually display and show the DICOM image. Fortunately, many programs are able to open up DICOM images for viewing. Such programs include GIMP, Adobe Photoshop, and some versions of paint. This ability is restricted to DICOM files that have no extension. These programs are unable to open DICOMs that have the '.dcm' extension. Opening the CRIMAGEA file in GIPM yields Figure 2. As can be seen, the file we've been looking at is a pixel spacing test image.

Figure 2. Screenshot of what a DICOM file looks like in GIMP.

The DICOM file extensions vary depending on the manufacturer of the medical device that created the file. Most commonly, DICOM files will have no extensions at all (as seen with our CRIMAGEA example in Figure 1) or have a ".dmc" extension. They can also be stored as a series of compressed JPEG files. Most DICOM viewers available for use are able to open any type of DICOM file,

with several exceptions regarding compressed JPEG files. For our project, we will focus on being able to open files with no extensions.

SUMMARY

DICOMs are a useful standard that helps medical proffesionals share information in a way that is easily accessable and easily integerated. While the file format may be useful for medical proffessionals to store the neccesary information required for patient image examinations, it is unfortunately difficult to decipher and package. Due to the amount of time it would require to create something capable of decoding a DICOM image (or even integrating an DICOM viewer into our program) it might not be a feasable task for the current length of the project without an appropriate library.

RESOURCES

Amarnath S, Mahesh Reddy S, Bangalore (April 2009). DICOM Image Viewer. Retreived from https://www.codeproject.com/kb/graphics/dicomimageviewer.aspx

Burdescu D. D., Caldare A., Georgescu E., Ion A., Stanescu L. (2006). Extracting, Storing And Viewing The Data From Dicom Files. Retrieved from: https://www.medetel.eu/download/2006/parallel_sessions/presentation/0405/Stanescu.pdf

Osirix (2017) DICOM Image Library. Retreived from http://www.osirix-viewer.com/resources/dicom-image-library/

Vasishtha V. (April 2016). Converting Dicom Files to Standard Image Format (.jpeg, .png, etc.) in MVC Using C#. Retrieved from https://www.codeproject.com/Articles/1042400/Converting-Dicom-Files-to-Standard-Image-Format-jp