

# 基于 Java 的 DICOM 文件格式转换与信息提取

李 伟, 李仕红, 韩中保

(盐城卫生职业技术学院, 江苏盐城 224006)

[中图分类号] TH772+.2    [文献标识码] B    [文章编号] 1002-2376 (2014) 04-0015-02

**〔摘 要〕** 在普通计算机上对 DICOM 格式文件图像信息的显示及相关临床信息的读取至今仍然是一个问题, 我们研究了使用较为流行 Java 语言对这种标准医学影像文件进行处理的一些方法。包括对原始的图像进行有损和无损的转换, 以及对存储在文件中的其它临床数据进行提取、修改。

**〔关键词〕** Java; DICOM; 图像处理; JPEG

## 1 简介

DICOM 作为医学数字成像和通信的标准, 详细定义了医学影像文件及其相关信息的组成格式和交换方法。利用这个标准, 人们可以在影像设备上建立一个接口来完成影像数据的输入/输出工作。然而 DICOM 格式是针对医疗设备特有的格式, 普通的个人计算机对其支持非常有限。基于此, 我们探索了利用 Java 语言来对 DICOM 格式文件的像素及其他信息的进行获取和修改的方法

## 2 工具

由于 Java 具有良好的可移植性, 我们选择用它作为对 DICOM 原始文件进行处理编程语言。对于 Java 来说, 目前已经有较为成熟的插件可供使用, 其中也包括一些比较好的开源项目如 Dcm4che 等, 他们大都符合 DICOM 的标准。利用现成的插件可以显著地减少我们在时间、精力上的投入, 而且使程序更具健壮性和可移植性。

我们选择使用的是 DICOM Image I/O Plug-in, 这是一个适用于 Java 语言的 DICOM 格式文件的处理插件。它提供了 DICOM 到 Java 之间的一个接口。换句话说, DICOM Image I/O Plug-in 扩充了 JAVA 的 API, 使得 Java 几乎能像处理普通图像那样去处理 DICOM 文件。DICOM Image I/O Plug-in 插件不但可以读取并修改本地 DICOM 文件 (或文件集), 也支持通过网络访问远程服务器或数据库中的文件和数据。

## 3 实现

### 3.1 获取单个 DICOM 文件的图像信息

我们可以直接将一个 DICOM 格式文件 (此处为 image.dcm) 创建为 Java 中的 BufferedImage 对象:

```
BufferedImage image = getIO.read (
    new File (" image.dcm"));
```

在后续的代码中我们可以像操纵普通的图像对象那样来对 DICOM 文件进行直接操作, 如我们可以通过 getWidth (), getHeight () 方法获得图像的高度、宽度, 以及复制指定区域 (感兴趣区) 的数据进行再处理等等。更进一步地, 我们可以从创建的 BufferedImage 对象所实现的接口获取相关图像的原始信息, 因而可以在像素级别上对图像进行处理。由于很多 DICOM 文件的灰度空间为 12 位甚至是 16 位, 如果格式转换后再进行图像处理, 那么图像的失真更大。所以较好的办法就是先处理再进行格式的转换。把图像创建为 BufferedImage 可以将像素的损失最小化。

### 3.2 读取 DICOM 文件, 并抽取相关临床信息

DICOM 文件中包含了大量的临床信息, 这对疾病的诊断、疗效的评估都相当重要, 因而对这些数据获取的重要性也是显而易见的。

在从一个 DICOM 标准格式文件中读取属性这前我们需要创建一个 DICOM 元数据 (Metadata) 对象用于存储在文件中的所有属性, 此外还需要一个迭代器 (Iterator) 来遍历取得的结果。

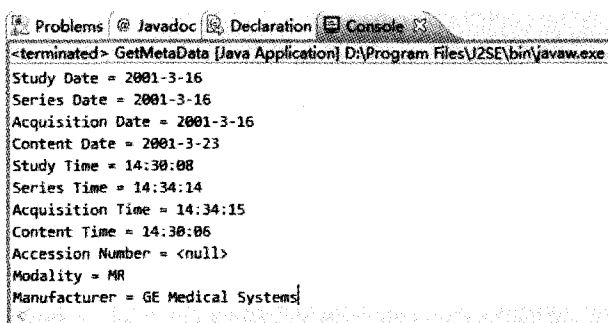
//创建一个 DicomMetadata (Dicom 元数据) 对象

```
DicomMetadata dmdata =
reader.getDicomMetadata ();
//获得元数据中病人 ID 信息, 并保存到一个
```

String 对象中

```
String patient_id =  
dmd.getAttributeString (Tag.PatientID);
```

在本例中, 我们仅用一个 String 对象来显示其 PatientID 信息, 其它属性都可以利用上述方法来获得, 在此不一一列举。我们在具体的实现中获得的部分数据如下所示:



The screenshot shows a Java application window titled 'GetMetaData [Java Application]'. The console output displays the following DICOM metadata:

```
<terminated> GetMetaData [Java Application] D:\Program Files\J2SE\bin\javaw.exe  
Study Date = 2001-3-16  
Series Date = 2001-3-16  
Acquisition Date = 2001-3-23  
Content Date = 2001-3-23  
Study Time = 14:30:08  
Series Time = 14:34:14  
Acquisition Time = 14:34:15  
Content Time = 14:30:06  
Accession Number = <null>  
Modality = MR  
Manufacturer = GE Medical Systems
```

DICOM 标准包含了大量的字典, DICOM Image I/O Plug-in 支持其中的绝大部分内容, 满足要求。

### 3.3 将信息写入到 DICOM 文件

除了读取信息外, 也可以把指定的属性值写入到文件中, 过程与读取是类似的, 现仅列出关键的写入部分代码:

```
//设置指定的 PatientID (12345) 属性  
dmd.setAttribute (Tag.PatientID, 12345);  
//写入到一个 DICOM 文件  
File file = new File (" image.dcm");  
writer.setOutput ( new FileOutputStream  
(file));  
writer.write (dmd, new IIOMImage (image, null,  
null), null);
```

在上述程序中, 为了写入 DICOM 属性, 首先创建一个 DicomMetadata () 对象, 然后利用 setAttribute () 方法来重新设置病人的 PatientID 属性为 12345, 对于不同的数据元素所使用的语句略有差别, 具体可参考文档。在接下来的代码中, 我们将修改后的内容重新写入到一个新的标准 DICOM 格式文件。

### 3.4 对文件进行格式转换

对于影像诊断来说, 文件的像素信息是最为重要的部分。DICOM 文件格式显然不利于随时随地的查阅、参考以及同行之间的交流。更进一步地, 有时需要对图像进行一些简单的处理, 如锐化、对比度的调节等等。这时就有必要把 DICOM 文件中的图像信息提取出来, 保存为一种比较易于访问的

格式。DICOM Image I/O Plug-in 支持将 DICOM 文件保存为当前最为流行的 JPEG 格式。此外, 如果对图像的质量有比较高的要求的话, 我们可以考虑使用 JPEG2000 格式, 在此处我们仅举一个简单的例子来说明一下写入的过程:

```
// 应用灰度变换  
BufferedImage bi = dmd.apply Grayscale Trans-  
formations (bi_stored, i);  
BufferedImage bi_out = new BufferedImage  
(bi.getWidth (),  
bi.getHeight (), BufferedImage.TYPE_3BYTE  
_BGR);  
bi_out.createGraphics ().drawImage (bi, 0,  
0, null);
```

// 保存图像

```
File f_out = new File (" i" + i + ".jpg");  
f_out.delete ();  
ImageIO.write (bi_out, " jpeg", f_out);
```

如前所述, DICOM 标准建议的图像灰度为 12 以上, 即大于 4096 个量级, 这需要专业的医用显示器才能完整地显示。但很多时候由于条件的限制我们所用到的大多都是 8 位的一般民用显示器, 而且超过 8 位 (256 级) 灰度人眼就已经不太容易分辨出来。

## 4 结论

医学图像的处理是比较一个热门的话题。然而如前文所述, 由于医学图像文件灰度的特殊性, 其转换为常见图像格式时必然会带来信息的丢失。如果能在原始数据直接进行高级的图像处理, 那么程序会有更加广泛的应用, 这也是我们下一步要努力的目标。

### [参考文献]

- [1] DICOM 官方文档, <http://medical.nema.org/>
- [2] Cay S. Horstmann, Gary Cornell 著, 叶乃文等译, Java 核心技术 (第 8 版). 机械工业出版社 - 2008
- [3] DICOM Image I/O Plug-in 官方文档, <http://imaging.apertyx.fr/dicom>
- [4] Java Image I/O API Guide, <http://www.oracle.com>
- [5] Oleg S. Panykh 著, Digital Imaging and Communications in Medicine. Springer - Verlag Berlin and Heidelberg - 2008
- [6] Wilhelm Burger 等著, 黄华译, 数字图像处理 - Java 语言算法描述. 清华大学出版社 - 2010
- [7] 梁云, 宋鸿陟等. DICOM 医学图像数据转换算法研究, 现代计算机 (专业版), 2008 - 04: 42 - 44