DicomImport

Release 1.00

Rashindra Manniesing[1]

October 15, 2009

[1]Biomedical Imaging Group Rotterdam (BIGR)
Erasmus MC - University Medical Center Rotterdam,
the Netherlands
r.manniesing@erasmusmc.nl

Abstract

This document describes a new class `itk::itkDicomDirectoryToImage`, which, given a directory consisting of raw dicom files, converts all found series to volumetric data with consistent filenaming based on the dicom headers. In addition many internal checks concerning the dicom files are performed and reported, and the user has the possibility to add some simple criteria to match on the series description. The main advantage of the provided class is the ease of processing and converting very large dicom databases in a consistent way.

Contents

1 Introduction **1**

2 Available Class **2**

3 Use and Functionality **2**

1 Introduction

Dealing with hunderds to thousands of patient data is certainly no trivial task. It requires some thinking, but mainly a proper organization to make full benefit of the available information. To aid in in the automated analysis at the beginning of the pipeline we introduce a small class to convert raw dicom files into volumetric data, ready for further image analysis.

In this document we describe the ITK class `itk::itkDicomDirectoryToImage` which has been inspired by the MevisLab [3] module called dicomimport and basically provides the same functionality as this module (in Section 3 we give a full description of the class's functionality). MevisLab is an advanced development environment for medical image processing, GUI development and visualization. Using this class in combination with the recently provided IO class for reading MevisLab's Dicom/TIFF image format [4] establishes an even smoother interchange usage of these excellent image processing environments.

Acknowledgments

We are very grateful to Marcel Koek for his feedback on the usage of earlier versions of this class. This work was supported by Economic Affairs innovation grant ISO44070 Automatic Diagnostic Vascular Analysis of CTA Examinations (ADVANCE).

2 Available Class

The following class is provided:

`itkDicomDirectoryToImage.h/cxx`

It has not been derived from any other ITK class and essentially groups together a number of functionalities. The class has been developed using ITK 3.10.0, GDCM 2.0.10 [2] and some functions of the Boost Library version 1.37.0 [1].

The following Boost headers are required to use this class: `dynamic_bitset` (to implement the criteria functionality), `filesystem` (which provides the functionality of reading and writing (sub)directories and files) and `progress` (not essential, but handy to report the total conversion time).

3 Use and Functionality

The use of the class is straightforward:

```
itk::DicomDirectoryToImage import(inputdir);
import.SetVerbose(true);
import.SetOutputDirectory(outputdir);
import.SetMatchOnSeries(``NecK AND cT'');
import.SetMinimumSlices(10);
import.SetAnonymize(false);
import.SetUseAccessionNumberInNaming(true);
import.SetFileNameExtension(``mhd'');
const bool success = import.ProcessDirectory();
```

The input directory should contain the raw dicom files, possibly containing sub directories. The criteria to search for does a match on the extracted series description of a particular serie (dicomtag

<0x008,0x103e>called “Series Description”). Simple AND/OR operators are allowed which are interpreted in reading direction, case insensitive and no parenthess are taken into account. The member function SetMinimumSlices gives the possibility to only convert those dicom data which match the minimal number of slices. The member function SetAnonymize removes the following tags if true from the output file:

<0x008,0x0080>	Institution Name
<0x008,0x0081>	Institution Address
<0x010,0x0010>	Patient’s Name
<0x010,0x0030>	Patient’s Birth Date
<0x010,0x0040>	Patient’s Sex
<0x010,0x1010>	Patient’s Age
<0x010,0x1040>	Patient’s Address
<0x010,0x21c0>	Pregnancy Status

For each valid serie that has been found, the program always write one dcm file, and depending on the (ITK supported) extension that is given one or two image file(s). The proposed file naming is as follows:

`patientname_patientid_acquisitiondate_accessionnumber_convolutionkernel.<ext>`

obviously, leaving out the patient’s name if anonymization is required. If a serie results in a filename that already exists in the output directory, then the current filename is post-fixed with increasing 4-digits numbers before writing. Therefore, no files in the output directoy will be overwritten (unless, of course, the very unusual situation occurs that more than 10k series exist of one particular patient acquired on one day! Perhaps). Inclusion of the accession number (dicom tag <0x0008,0x0050>, “Accession Number”) in the file naming is optional. During the conversion of one serie, a simple check is performed on the correspondence of the dimensions of each slice to the dimensions of the first slice, next to validity checks on reading the dicom tags. Upon errors, these are simply reported to screen, and the program tries to continue the conversion of subsequent series.

Internally, this class uses `itk::GDCMSeriesFileNames` to collect all files of one certain serie. When converting the serie to a volumetric data, linear intensity scaling and pixel type conversion (if required) are applied and rescale values are modified accordingly in the output dicom file. Intensity scaling concerns the dicom tags “Rescale Intercept” <0x0028,1052>and “Rescale Slope” <0x0028,1053>.

Currently, the new class provides conversion of series to 3D data only. Thus extension to nD, and more advanced interpretation of the matching criteria on the series description are appreciated.

References

[1] Boost c++ libraries. URL http://www.boost.org/. 2

[2] GDCM, release 2.0. http://apps.sourceforge.net/mediawiki/gdcm/index.php?title=GDCM_Release_2. 2

[3] MevisLab, software for medical image processing and visualization. http://www.mevislab.de. 1

[4] Rashindra Manniesing. Read and write support for mevislab dicom/tiff format, 2008. URL http://hdl.handle.net/1926/1514. 1