Including TIFF tags in the MetaDataDictionary

Release 0.00

Richard Beare

April 30, 2010

Richard.Beare@med.monash.edu.au

Abstract

Tiff images are widely used and domain and instrument specific information is often included in some tags. There isn't any standard way of making use of this information in ITK at present. This article introduces some minor additions to the ITK tiff reader to include the content of some potentially useful tags in the image MetaDataDictionary, thus enabling domain specific information to be extracted from the tags within an ITK application.

Contents

1 Introduction

Tiff is a widely used image format that is frequently extended to new domains by including information in fields in the tiff directory. The directory in tiff files is equivalent to the header in other formats, but needn't be at the start of the file, and multi image tiff file will have multiple directories.

Most standard image information describing image pixel layout on disk, image width, image height, colour encoding and so on is encoded in "standard" tiff fields and is already used by the ITK image reader. How-

ever additional information that may be useful to some applications can be stored in other tiff fields. This article describes how a modified tiff reader can make these fields available to an ITK application via the MetaDataDictionary.

2 Domain specific extensions

An example of domain specific extensions to tiff is the Open Microscopy Environment (OME) tiff format – http://www.ome-xml.org/wiki/OmeTiff. This extension encodes a large range of microscopy related meta-data in XML in the tiff *ImageDescription* field.

Correct interpretation of the image by an application may require access to information in this field. In many cases a customized reader may be more appropriate, but this is not always feasible in the early days of using a new format. The extensions described in this article provide an intermediate solution.

3 TIFF fields

The fields likely to be relevant to an application include:

- ARTIST
- COPYRIGHT
- DATETIME
- DOCUMENTNAME
- HOSTCOMPUTER
- IMAGEDESCRIPTION
- INKNAMES
- MAKE
- MODEL
- PAGENAME
- SOFTWARE
- TARGETPRINTER
- XMLPACKET

The *tiffinfo* tool distributed with libtiff provides a dump of these fields that is useful when interpreting a new tiff variant. The code in this article makes the same information available to an ITK application.

4 TIFFImageIO modifications

Modifications to TIFFImageIO have been made and the files are included in the source package. The two files, *itkTIFFImageIO.h* and *itkTIFFImageIO.cxx* are replacements for existing versions in *InsightToolkit/-Code/IO*. The examples following assume that ITK has been built using these files.

The modifications simply encode the field names and contents in the MetaDataDictionary.

5 Accessing the fields via the MetaDataDictionary

Standard mechanisms can be used to access the contents of the MetaDataDictionary, as illustrated below. This example is from *check.cxx*, included in this package.

```
int main(int argc, char * argv[])
{

  if( argc != 2 )
    {
    std::cerr << "usage: " << argv[0] << " intput " << std::endl;
    std::cerr << " input: the input image" << std::endl;
    // std::cerr << "  : " << std::endl;
    exit(1);
    }

  const int dim = 2;

  typedef unsigned char PType;
  typedef itk::Image< PType, dim > IType;


  typedef itk::ImageFileReader< IType > ReaderType;
  ReaderType::Pointer reader = ReaderType::New();
  reader->SetFileName( argv[1] );
  reader->Update();

  typedef itk::MetaDataDictionary   DictionaryType;
  typedef itk::MetaDataObject< std::string > MetaDataStringType;

  const  DictionaryType & dictionary = reader->GetImageIO()->GetMetaDataDictionary();
  DictionaryType::ConstIterator itr = dictionary.Begin();
  DictionaryType::ConstIterator end = dictionary.End();
  while( itr != end )
    {
    itk::MetaDataObjectBase::Pointer  entry = itr->second;

    MetaDataStringType::Pointer entryvalue =
      dynamic_cast<MetaDataStringType *>( entry.GetPointer() );
    if( entryvalue )
      {
      std::string tagkey   = itr->first;
      std::cout << "MetaDataDict " << tagkey << ": " << entryvalue->GetMetaDataObjectValue() << std::en
```

```
      }

    ++itr;
    }

  return 0;
}
```

6 Sample output

Apply check to an OME tiff included in the package provides the following output (line breaks included for clarity):

```
> check images/single-channel.ome.tif

MetaDataDict TIFFTAG_IMAGEDESCRIPTION: <?xml version="1.0"
encoding="UTF-8"?><!-- Warning: this comment is an OME-XML metadata
block, which contains crucial dimensional parameters and other
important metadata. Please edit cautiously (if at all), and back up
the original data before doing so. For more information, see the
OME-TIFF web site: http://loci.wisc.edu/ome/ome-tiff.html. --><OME
xmlns="http://www.openmicroscopy.org/XMLschemas/OME/FC/ome.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openmicroscopy.org/XMLschemas/OME/FC/ome.xsd
http://www.openmicroscopy.org/XMLschemas/OME/FC/ome.xsd"><Image
ID="openmicroscopy.org:Image:1" Name="single-channel"
DefaultPixels="openmicroscopy.org:Pixels:1-1"><CreationDate>2007-11-08T14:52:24</CreationDate><Pixels
ID="openmicroscopy.org:Pixels:1-1" DimensionOrder="XYZCT"
PixelType="Uint8" BigEndian="true" SizeX="439" SizeY="167" SizeZ="1"
SizeC="1" SizeT="1"><TiffData/></Pixels></Image></OME>
```

7 Conclusions and further work

The changes described in this article only provide access to the standard fields in which data is stored as text. It is also possible for vendors to include proprietary tiff fields, and future versions may be able to include this information in the MetaDataDictionary.

However the current version should allow useful, domain specific, interpretation of data without major changes to ITK infrastructure and thus be useful in early project development. In particular, information such as voxel size, scaling parameters and anything else that might be relevant to correct image interpretation and encoded as strings in tiff tags may be accessed in an ITK application via relatively simple string parsing code.

References

[1] L. Ibanez and W. Schroeder. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-10-6, http://www.itk.org/ItkSoftwareGuide.pdf, 2003.