Support for Streaming the JPEG2000 File Format

Release 0.00

Kishore Mosaliganti[1], Luis Ibanez[2], and Sean Megason[1]

August 8, 2010

[1]Department of Systems Biology, Harvard Medical School
[2]Kitware Inc., Albany, NY

Abstract

This paper describes our contribution of two new classes to the Insight Toolkit (ITK) community. This includes a base and derived ImageIO class for the JPEG2000 file formats based on the openjpeg-v2 libraries. In the current release of ITK, support for JPEG2000 is absent. Our current implementation of the JPEG2000 supports 2D images (8-bit, 16-bit and RGB) and tiling. The latest code is available in the NAMIC Sandbox SVN repository under the project JPEG2000ImageIO. We include 2D example code and tests in this submission. This code is likely to be integrated into ITKv4 very soon.

Latest version available at the Insight Journal [http://hdl.handle.net/1926/1]

Contents

1 Introduction

JPEG 2000 is an image compression standard and coding system based on wavelet transforms. It was created by the Joint Photographic Experts Group committee in 2000 and is an improvement over their original discrete cosine transform-based JPEG standard (created in 1992)

While there is a modest increase in compression performance of JPEG 2000 compared to JPEG, the main advantage offered by JPEG 2000 is the significant flexibility of the codestream. The codestream obtained after compression of an image with JPEG 2000 is scalable in nature, meaning that it can be decoded in a number of ways; for instance, by truncating the codestream at any point, one may obtain a representation of the image at a lower resolution, or signal-to-noise ratio. By ordering the codestream in various ways, applications can achieve significant performance increases. However, as a consequence of this flexibility, JPEG 2000 requires encoders/decoders that are complex and computationally demanding. Another difference, in comparison with JPEG, is in terms of visual artifacts: JPEG 2000 produces ringing artifacts, manifested as blur and rings near edges in the image, while JPEG produces ringing artifacts and 'blocking' artifacts, due to its 8×8 blocks.

The image is split into so-called tiles, rectangular regions of the image that are transformed and encoded separately. Tiles can be any size, and it is also possible to consider the whole image as one single tile. Once the size is chosen, all the tiles will have the same size (except optionally those on the right and bottom borders). Dividing the image into tiles is advantageous in that the decoder will need less memory to decode the image and it can opt to decode only selected tiles to achieve a partial decoding of the image. The disadvantage of this approach is that the quality of the picture decreases due to a lower peak signal-to-noise ratio.

2 Implementation

Our current implementation consists of two classes: `itkJpeg2000ImageIOFactory` `itkJpeg2000ImageIO`. These classes inherit directly from base classes in ITK `ObjectFactoryBase` and `StreamingImageIOBase`. These classes depend on encoder/decoders in the openjpeg v2 library which is linked with separately. Currently, the following features have been enabled in $2D$ images:

- 8-bit, 16-bit and RGB images
- Image tiling with reading of a subset of tiles encompassing a region-of-interest. Images can also be written out with user-specified tile sizes.

3 Future Work

Future work in this project consists of adding support for $2D+t$ files, compression schemes and heirarchical images that can support out-of-core registration procedures on large images. As per the ITKv4 migration plan, this code is likely to be integrated very soon and hence users may first check if this is the case.

4 Software Requirements and Usage

Current development is hosted in the NAMIC Sandbox http://svn.na-mic.org/NAMICSandBox/trunk/JPEG2000ImageIC
The latest code can be directly downloaded from there.

The following components are needed for these classes to run correctly:

- Insight Toolkit 3.2 with `BUILD_REVIEW` turned `ON`
- CMake 2.8
- openjpeg

An openjpeg v1 library is carried by ITK in the Utilities directory. The GDCM library already uses JPEG2000 in order to manage the compression in DICOM images. However, this version of openjpeg is old and has to be turned off at various locations in the CMake files. For the convenience of the user, we are also submitting a patch `patch.txt` to apply on the ITK source code. To apply the patch:

```
cd  Insight
patch -p0 < patch.txt
```

The latest openjpeg library is available for download at: http://www.openjpeg.org/. Openjpeg libraries can first be compiled using CMake with `BUILD_SHARED_LIBRARIES` set to `ON`. While compiling the current set of classes, the CMake variables `OPENJPEG_INCLUDE_DIR` and `OPENJPEG_LIBRARY` need to be set appropriately within the source and binary directories of openjpeg. Openjpeg has a dashboard at http://my.cdash.org/index.php?project=OPENJPEG.

Additionally, for running ctest, the variable `CLIF2007_DATA_ROOT` needs to be set to the folder containing the test images. Using ctest, all the tests can be executed and submitted to the dashboard here: http://www.cdash.org/CDash/index.php?project=Insight.

5 Acknowledgements

The authors would like to acknowledge the contribution of Mathieu Malaterre during the NAMIC Project Week, Boston, 2010.

References