OME-TIFF for digital pathology: sub-resolution support

Roger Leigh

May 2018

Reading and writing tiled pyramids
Big images: tiling and sub-resolution pyramids

36832×38432
Big images: tiling and sub-resolution pyramids
Big images: tiling and sub-resolution pyramids

36832×38432
Big images: tiling and sub-resolution pyramids
Big images: tiling and sub-resolution pyramids

36832×38432
Big images: tiling and sub-resolution pyramids

9208×9608

36832×38432
Big images: tiling and sub-resolution pyramids
Big images: tiling and sub-resolution pyramids

9208×9608

2302×2402

576×600

36832×38432
Big images: tiling and sub-resolution pyramids

9208×9608
2302×2402
576×600
144×150

36832×38432
Big images: tiling and sub-resolution pyramids

$3632 \times 38432$

$9208 \times 9608$

$2302 \times 2402$

$576 \times 600$

$144 \times 150$
TIFF and OME-TIFF sub-resolution support

Introduction

There have been several different proposals for images at different scales in the form of sub-resolutions (image “pyramids”) for TIFF and OME-TIFF in Bio-Formats and OME Files, which include:

- Storage of pyramid data in OME-TIFF (Melissa Linkert / Glencoe)
- Use of SubIFDs (Roger Leigh)
- TIFF/OME-TIFF extension to support pyramids (Damir Sudar)

This proposal will summarise the various possible approaches and their tradeoffs, including the practical implementations I have tested while evaluating them.

There are several strategies we could employ for sub-resolutions:

<table>
<thead>
<tr>
<th>A. Implicit order</th>
<th>B. SubIFD pointing to main IFDs</th>
<th>C. SubIFD point to separate IFDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0 (full)</td>
<td>R0 (full)</td>
<td>R0 (full)</td>
</tr>
</tbody>
</table>

Many thanks to Melissa Linkert, Damir Sudar and all other participants for their input.
Layout of a TIFF file

Image file directory
Layout of a TIFF file

Image file directory - 512×512, 16-bit
Layout of a TIFF file

- Image file directory
- Image file directory
- 512×512, 16-bit
Layout of a TIFF file

Image file directory

512×512, 16-bit

1024×1024 8-bit
Layout of a TIFF file

Image file directory

512×512, 16-bit

Image file directory

1024×1024 8-bit

Image file directory

1024×1024 float

Image file directory

512×512, 16-bit

Image file directory

8192×8192, 16-bit signed
Sub-resolutions in TIFF

(A) Implicit ordering

IFD 0

IFD 1

IFD 2

IFD 3

IFD 4

36832×38432

9208×9608

2302×2402

576×600

144×150
Sub-resolutions in TIFF

(B) Flattened SUBIFDS

IFD 0

IFD 1

IFD 2

IFD 3

IFD 4

36832×38432

SUBIFDS
Sub-resolutions in TIFF
(C) SUBIFDS

IFD 0 → 36832×38432 → SUBIFDS

IFD 1
IFD 2
IFD 3
IFD 4
Sub-resolutions in TIFF

(D) OME-XML with implicit resolution order

- IFD 0
- IFD 1
- IFD 2
- IFD 3
- IFD 4

36832×38432 resolutions=5
Sub-resolutions in TIFF

(E) OME-XML with explicit resolution order

TIFF
- IFD 0
- IFD 1
- IFD 2
- IFD 3
- IFD 4
- 36832×38432

OME-XML
- resolutions=5
- resolution1
- resolution2
- resolution3
- resolution4
## Sub-resolutions in TIFF

### Cost/benefit trade-offs between strategies

<table>
<thead>
<tr>
<th>Strategy</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubIFDs usage</td>
<td>None</td>
<td>Simple</td>
<td>Full</td>
<td>Optional‡</td>
<td>Optional‡</td>
</tr>
<tr>
<td>z reduction</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>libtiff compatibility</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes‡</td>
<td>Yes</td>
</tr>
<tr>
<td>Photoshop compatibility</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sub-resolution access without SubIFDs</td>
<td>Yes</td>
<td>Yes§</td>
<td>No</td>
<td>Optional‡</td>
<td>Optional‡</td>
</tr>
<tr>
<td>Reading portability</td>
<td>High</td>
<td>Mid</td>
<td>Low†</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Writing portability</td>
<td>High</td>
<td>Mid</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Implementation complexity</td>
<td>Low</td>
<td>Mid</td>
<td>Mid</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Model changes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Simple</td>
<td>Complex</td>
</tr>
<tr>
<td>Multi-series OME-TIFF</td>
<td>No*</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes*‡</td>
<td>Yes</td>
</tr>
<tr>
<td>Multi-file OME-TIFF</td>
<td>No*</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes*‡</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- Maximise format compatibility, including forward compatibility for existing software
- Use only baseline TIFF and standard extensions
- Avoid costly breaking model and API changes
- Provide additional functionality within these constraints
Strategy C chosen: baseline TIFF with tiling and SUBIFDS extensions

- Should be fully backward and forward compatible
- Bio-Formats required some bugfixes due to historic implementation choices
- Will handle all dimensions of OME-XML (z, t, c, Modulo) transparently
- Making things forward compatible is very hard!
Technical changes

- **Added** `CoreMetadataList` to replace `List<CoreMetadata>`
- **Added** `SubResolutionFormatReader` to replace `FormatReader`
- **All sub-resolution and TIFF formats extend** `SubResolutionFormatReader`
- **Fixed** `SUBIFDS` usage in `TiffParser` and `MinimalTIFFReader`
- **Added** `SUBIFDS` usage to `OMETiffReader`
Where is the stuff

**Source code**  east integration branch of bioformats.git at https://github.com/openmicroscopy/bioformats/tree/east

**Builds**  east-ci Jenkins CI at https://web-proxy.openmicroscopy.org/east-ci/job/BIOFORMATS-merge/
Viewing: Command-line tools

- `libtiff` `tiffinfo`
- **Bio-Formats** `showinf` `-noflat`
Read Leica SCN and OME-TIFF with the Bio-Formats plugin.
https://web-proxy.openmicroscopy.org/east-web/webclient/

Username: demo
Password: demo
Future work

- Bio-Formats
  - OME-TIFF sub-resolution writing (in progress)
  - bfconvert
  - Documentation and example code
  - Sample OME-TIFF images
Future work

- **Bio-Formats**
  - OME-TIFF sub-resolution writing (in progress)
  - `bfconvert`
  - Documentation and example code
  - Sample OME-TIFF images
  - A Bio-Formats release with OME-TIFF and `bfconvert` sub-resolution support
Future work

- Bio-Formats
  - OME-TIFF sub-resolution writing (in progress)
  - bfconvert
  - Documentation and example code
  - Sample OME-TIFF images
  - A Bio-Formats release with OME-TIFF and bfconvert sub-resolution support

- OME Files
  - C++ support for reading and writing sub-resolutions
Future work

- Bio-Formats
  - OME-TIFF sub-resolution writing (in progress)
  - bfconvert
  - Documentation and example code
  - Sample OME-TIFF images
  - A Bio-Formats release with OME-TIFF and bfconvert sub-resolution support

- OME Files
  - C++ support for reading and writing sub-resolutions

- Additional improvements
  - ImageJ plugin resolution level selection
  - Default resolution flattening to off? Or remove entirely?
  - Support for sub-resolutions in HDF5 and other containers
  - Support for additional metadata including label images
Future work

- Bio-Formats
  - OME-TIFF sub-resolution writing (in progress)
  - bfconvert
  - Documentation and example code
  - Sample OME-TIFF images
  - A Bio-Formats release with OME-TIFF and bfconvert sub-resolution support

- OME Files
  - C++ support for reading and writing sub-resolutions

- Additional improvements
  - ImageJ plugin resolution level selection
  - Default resolution flattening to off? Or remove entirely?
  - Support for sub-resolutions in HDF5 and other containers
  - Support for additional metadata including label images

- What you **you** need from us for your work, and when?
  - What should we prioritise?
I would like to thank everyone involved in the design, testing and implementation for their contribution to this work:

- Sébastien Besson
- David Gault
- Melissa Linkert
- Josh Moore
- Damir Sudar
- Jason Swedlow

This work was funded by the BBSRC and MULTIMOT.