Using Jpylyzer and Schematron for validating JPEG2000 files

William Palmer

SCAPE Information Day

British Library, UK, 14th July 2014

JPEG 2000

- Created in 2000
- Uses wavelet compression
- Not widely used (compared to JPEG)
- Software support immature
  - Implemented features and support varies between codecs
  - Two main contemporary codecs: OpenJPEG and Kakadu
- Supports lossless and lossy compression
- (Technically) can be used as both master and access copy
- Complex standard, uses “profiles” to determine which features are used

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

JPEG 2000 at The British Library

- In use for master images, and access images
- Profile developed specifically for the British Library
- Long term storage currently contains 8 million+ JP2 files

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

British Library JPEG 2000 Profile

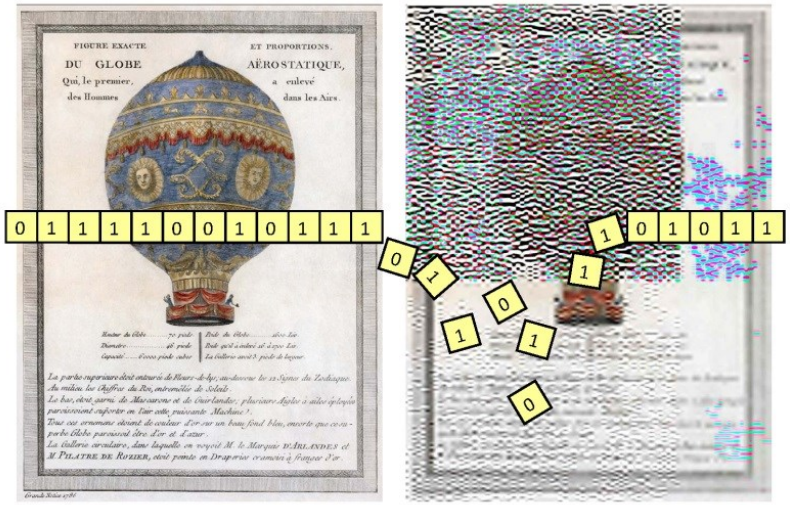
Parameter/Field	Value
Compression	Lossy
Number of components	3
Component Transform	Yes (irreversible)
Tile size	One tile for entire image
Wavelet filter	9-7 irreversible
Number of levels	Variable; 6 used for test image
Number of layers	Multiple
Progression order	RPCL
Codestream markers	Packet-length markers
Precincts	256x256, 256x256,128x128
Codeblock size	64x64
Coder Bypass	Yes
Compression ratios	-, 10, 8.7, 7, 5.2, 3.4, 2.1, 1.2, 0.6, 0.3, 0.15, 0.075

http://www.digitizationguidelines.gov/still-image/documents/Martin.pdf

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

If It's Complicated, then…

- How can we tell it's a valid JP2 (JPEG 2000) file?
- How can we tell if the profile matches the encoding parameters?
- We want to ensure our content is accessible and conforms to our policies



This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

Enter Jpylyzer

- JP2 validator and extractor (does not check image data)
- Command line tool written in Python
- Runs in Windows, Linux and MacOS X
- Checks if a JP2 file conforms to the specification
- Reports technical characteristics (feature extraction) including information about encoding settings (i.e. profile)
- XML output:

```
<?xml version="1.0" encoding="ASCII"?>
- <jpylyzer>
  - <toolInfo>
      <toolName>jpylyzer.exe</toolName>
      <toolVersion>1.6.3</toolVersion>
    </toolInfo>
  - <fileInfo>
      <fileName>WO1_ANJO_1847_09_15-0004.tif.0.jp2</fileName>
      <filePath>F:\eclipse-workspace\JP2Check\testfiles\images\op
      <fileSizeInBytes>32905342</fileSizeInBytes>
      <fileLastModified>Fri Jan 11 11:28:46 2013</fileLastModified>
    </fileInfo>
    <isValidJP2>True</isValidJP2>
    <tests/>
  - <properties>
      <signatureBox/>
    - <fileTypeBox>
        <br>jp2 </br>
        <minV>0</minV>
        <cL>jp2 </cL>
      </fileTypeBox>
```

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

Schematron

- Define the encoding profile in a Schematron file (XML) using “rules”:

```
<schema xmlns="http://purl.oclc.org/dsdl/schematron">
        <title>Schematron for checking Jpylyzer outputs</title>
        <pattern>
                <title>validity test</title>
                <rule context="jpylyzer/isValidJP2">
                        <assert test="text()='True'">JP2 must be valid</assert>
                </rule>
        </pattern>
        <pattern>
                <title>profile test</title>
                <rule context="cod/order">
                        <assert test="text()='RPCL'">progression order must fit profile</assert>
                </rule>
                <rule context="cod/transformation">
                        <assert test="text()='9-7 irreversible'">transformation must fit profile</assert>
                </rule>
                <rule context="cod/precincts">
                        <assert test="text()='yes'">must have precincts to fit profile</assert>
                </rule>
                <rule context="cod/precinctSize">
                        <assert test="text()='7'">number of precincts must fit profile</assert>
                </rule>
                <rule context="siz/numberOfTiles">
                        <assert test="text()='1'">number of tiles must fit profile</assert>
                </rule>
                <rule context="cod/levels">
```

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

Schematron

- XML transform against Jpylyzer output
- Look for “failed asserts” in output
- If zero “failed asserts” then settings match

- Easy to change the Schematron file to match your settings
- Get a yes/no answer
- Fast execution
- Support library written in Java to help: https://github.com/bl-dpt/jp2check

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

Summary

- Jpylyzer is a fast, efficient tool for extracting metadata
- In combination with a Schematron file that describes your institutional policy you can validate your JP2 images (but not image payload)
- Already in production use at a variety of institutions

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).