OGC® DOCUMENT: 21-025

External identifier of this OGC® document: http://www.opengis.net/doc/PER/t17-D046

Open Geospatial Consortium

OGC TESTBED-17: CLOUD OPTIMIZED GEOTIFF SPECIFICATION ENGINEERING REPORT

ENGINEERING REPORT

PUBLISHED

Submission Date: 2021-11-19
Approval Date: 2021-12-17
Publication Date: 2022-02-08
Editor: Joan Maso

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Copyright notice

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

I ABSTRACT

Cloud Optimized GeoTIFF (COG) is a new approach in using existing standards to accelerate distribution and analysis of 2D regular grid coverage data on the web. COG combines the use of the TIFF format with data structured internally in tiles and low resolutions subfiles (also called overviews). The main subfile is georeferenced using GeoTIFF tags and the lower resolution subfiles inherit the same georeferencing. This organization allows for retrieving only the part of the data needed for presentation or analysis. This capability is possible not only in the file system but also over the web if the HTTP range header is supported by the servers.

This OGC Testbed 17 Engineering Report (ER) discusses the COG approach, describes how GeoTIFF is used for the lower resolution subfiles, and proposes a different path forward that integrates COG with the OGC Tile Matrix Set Standard (http://docs.opengeospatial.org/is/17-083r2/17-083r2.html). The ER includes a chapter that formalizes the draft COG specification with clear requirements.

One of the common use cases for COG is the provision of multispectral remote sensing data. The increase in spatial and spectral resolution combined with more accurate sensors that require more than 8 bits per pixel results in big files that can exceed the 4 Gbyte limit of the original TIFF format. Having an OGC standard formally specifying this approach would be useful. Therefore, this ER includes a chapter that formalizes a draft BigTIFF specification, defining clear requirements.

The objective is to be able to reference BigTIFF from the GeoTIFF and the COG standards.

II EXECUTIVE SUMMARY

There is a need for new approaches to drastically accelerate visualization and analysis on the World Wide Web. Some emerging formats are now reused in a way that improves internal organization of the file. This allows for retrieving only the part of the data needed for presentation or analysis. If this organization is combined with the HTTP range header in the GET operation, clients can request parts of the data over the Web without any server side APIs or any additional web services. In the case of geospatial data in a 2D regular grid coverage model, this strategy can be implemented through the COG. COG restricts the GeoTIFF format to an internal data structured based on tiles and low resolutions overviews.

This OGC Testbed 17 Engineering Report (ER) describes how common libraries and implementations work internally, and exposes issues in the current approach. The report detects two main problems in COG:

- The COG approach ignores the OGC Tile Matrix Set Standard (http://docs.opengeospatial.org/is/17-083r2/17-083r2.html) in the tile structure, forcing the same point of origin and extent for all overviews. In addition, it defines a overview schema that easily results in non-square pixels.

- Formally, COG is based in GeoTIFF that is explicitly dependent on TIFF version 6 and is limited to 4-Gbyte files. This limitation is mitigated by developers by adding support to BigTIFF, an emerging de-facto standard defined by the LibTIFF community.

The ER includes a chapter that formalizes an initial draft COG specification with clear requirements and modular structure. A draft version of the BigTIFF specification adapted to the OGC Standard for Modular Specifications is also proposed. GeoTIFF should extend its support to BigTIFF. Both draft proposals have been submitted to the GeoTIFF Standards Working Group for consideration by the OGC Membership.

A similar approach has been proposed for other formats, such as Zarr and Cloud Optimized Point Cloud (COPC) (https://copc.io/). Given the success of the approach, more work needs to be done to consider extending this practice to other formats and media types such as GeoJSON or NetCDF.

III KEYWORDS

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, COG, Cloud Optimized GeoTIFF

IV PREFACE

This document provides the initial bases for a potential OGC standard for COG. Currently the COG standard is specified in https://github.com/cogeotiff/cog-spec/blob/master/spec.md that briefly describes the format as well of the GDAL implementation. This ER contains text that is independent of implementations and more comprehensive but compatible with the GDAL implementation. This draft will be transferred to the OGC GeoTIFF Standards Working Group (SWG) as a starting point for a potential OGC COG standard. The continuation of this work can be followed in the public GitHub repository https://github.com/opengeospatial/CloudOptimizedGeoTIFF. This document also provides the initial basis for a potential OGC standard for BigTIFF. A small group (participated by Andrey Kiselev, Bob Friesenhahn, Chris Cox, Dan Smith, Frank Warmerdam, Gerben Vos, John Aldridge, Joris Van Damme, Leonard Rosenthol, Lynn Quam, Marco Schmidt, Phillip Crews, Rob van den Tillaart and Thomas J. Kacvinsky, among others) from the LibTIFF community (https://lists.osgeo.org/mailman/listinfo/tiff) has defined a modification of the original TIFF format called BigTIFF that modifies some headers to allow for 64-bit internal offsets. The approach is also described here http://bigtiff.org/ and https://www.awaresystems.be/imaging/tiff/bigtiff.html. This ER contains consolidated text that is compatible with the current implementations. This draft will be transferred to the OGC GeoTIFF Standards Working Group (SWG) as a starting point for a potential OGC COG standard. The continuation of this work will can be followed in the public GitHub repository https://github.com/opengeospatial/BigTIFF.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

V SECURITY CONSIDERATIONS

No security considerations have been made for this document

VI SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- Universitat Autonoma de Barcelona (CREAF)

VII SUBMITTERS

All questions regarding this document should be directed to the editor or the contributors:

NAME	ORGANIZATION	ROLE
Joan Masó	UAB-CREAF	Editor

1

SCOPE

1 SCOPE

This Engineering Report represents deliverable D046 of the OGC Testbed 17 initiative performed under the OGC Innovation Program. It describes the current usage of COG and an alternative path to georeferenced COG internal multiresolution structure aligned with Tile Matrix Sets.

The Engineering Report uses the TIFF standard v6 with no modification. However, the Engineering report proposes to also use BigTIFF to support more that 4 Gbyte file sizes.

This document aims to demonstrate the business value of COG for distributing remote sensing data over the web without forcing applications to completely download big files for visualization and analysis. The result is fast performance in visualization tools that can read remote file repositories immediately, saving time and storage space. The combined use of COG and easy to use remote sensing data catalogues such as SpatioTemporal Asset Catalogs (STAC, https://stacspec.org/), simplifies finding and accessing large volumes and long series of remote sensing products.

3 KEYWORDS

3 KEYWORDS

The following are keywords to be used by search engines and document catalogues. ogcdoc, OGC document, COG, Cloud Optimized GeoTIFF, tiles, overviews, bigTIFF, TIFF, coverage

4

SUBMITTING ORGANIZATION

4 SUBMITTING ORGANIZATION

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- Universitat Autonoma de Barcelona (CREAF)

2

TERMS, DEFINITIONS AND ABBREVIATED TERMS

2 TERMS, DEFINITIONS AND ABBREVIATED TERMS

This document uses the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

For the purposes of this document, the following additional terms and definitions apply.

2.1. Terms and definitions

2.1.1. **BigTIFF**

a file that uses modified TIFF headers to allow for internal 64 bit offset (adding support for TIFF files larger than 4 Gbytes)

2.1.2. **Cloud**

an on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, that are exposed in the web by cloud providers that do not require direct active management by the user

2.1.3. **Coverage**

feature that acts as a function to return values from its range for any direct position within its spatiotemporal domain [OGC Abstract Topic 6]

2.1.4. Cloud Optimized GeoTIFF

a regular GeoTIFF file, aimed at being hosted on a HTTP file server, with an internal organization that enables more efficient workflows on the cloud [https://www.cogeo.org/]

2.1.5. Geokey

an equivalent in function to a TIFF tag, but with a different storage mechanism defined by the GeoTIFF [GeoTIFF Format Specification 1.0]

2.1.6. GeoTIFF

Standard for storing georeference and geocoding information in a TIFF 6.0 compliant raster file. [GeoTIFF Format Specification 1.0]

2.1.7. Imagery

representation of phenomena as images produced electronically and/or optical techniques. [ISO 19101-2:2018, 3.14]

Note 1 to entry: In this document, it is assumed that the phenomena have been sensed or detected by one or more devices such as radar, cameras, photometers, and infra-red and multispectral scanners.

2.1.8. Overview

Image File Directory that contains a reduced resolution image

2.1.9. Range

a HTTP GET request type that lets clients ask for the portions of a web resource that they need

2.1.10. Raster

usually rectangular pattern of parallel scanning lines forming or corresponding to the display on a cathode ray tube [ISO 19123:2005, 4.1.30]

a continuous planar space in which pixel values are visually realized [GeoTIFF v1.0]

Note 1 to entry: A raster is a type of regular grid.

2.1.11. Regular grid

grid whose grid lines have a constant distance along each grid axis [OGC 09-146r8, Coverage Implementation Schema with Corrigendum]

2.1.12. Subfile

Image File Directory (a part of a TIFF file) that contains one raster image

2.1.13. Tag

a packet of numerical or ASCII values, which have a numerical “Tag” ID indicating their information content in a TIFF file [GeoTIFF Format Specification 1.0]

2.1.14. **Tile**

geometric shape with known properties that may or may not be the result of a tiling (tessellation) process. A tile consists of a single connected “piece” (topological disc) without “holes” or “lines” [OGC 19-014r3]

small rectangular representation of geographic data, often part of a set of such elements, covering a tiling scheme and sharing similar information content and graphical styling. A tile can be uniquely defined in a tile matrix by one integer index in each dimension. [OGC 17-083r3, fragment]

2.2. Abbreviated terms

COG	Cloud Optimized GeoTIFF
HTTP	Hypertext Transfer Protocol
IFD	Image File Directory
TIFF	Tagged Image File Format

5

INTRODUCTION

5 INTRODUCTION

A COG is a GeoTIFF file that uses Tiles and Reduced-Resolution Subfiles to organize the data for optimal retrieval of fragments at the required resolution. The goal is to have a format that can be hosted on a common HTTP web server, with an internal organization that enables more efficient workflows on the web. The use of COG better supports remote sensing imagery being stored in cloud data centers and offered as cloud services without any special configuration. Additionally, leveraging the ability of clients issuing HTTP GET range requests (IETF RFC7233) to ask for just the parts of a file they need is possible.

There are two characteristics of the TIFF format that are the key elements of the internal organization of COGs: Tiles and Reduced-Resolution Subfiles. The TIFF file is also georeferenced using GeoTIFF tags. COG can be considered a subset of what GeoTIFF and TIFF offers (see Figure 1).

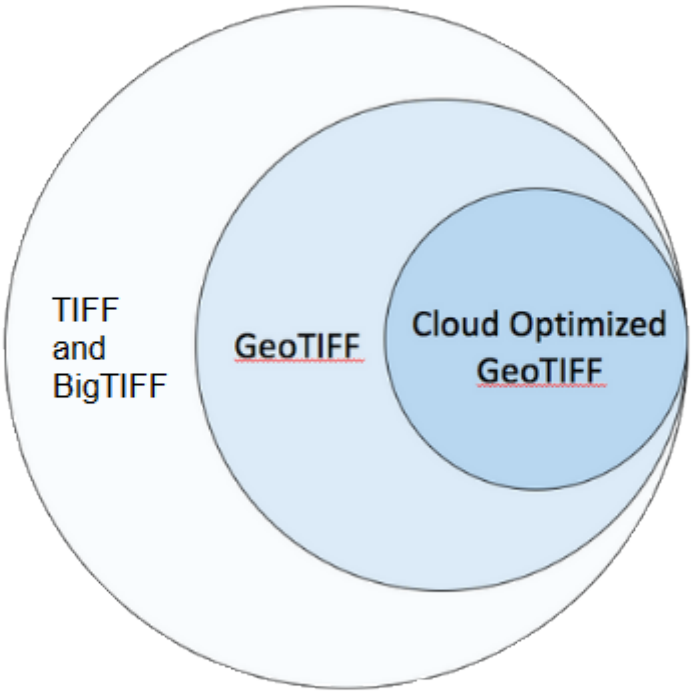

Figure 1 — Diagram to describe that a Cloud Optimized GeoTIFF as a subset of a GeoTIFF, which in turn is a subset of a TIFF (modified from the original https://www.eclipse.org/community/eclipse_newsletter/2018/december/geotrellis.php)

5.1. Tiles

As shown in Figure 2, typical raster images store data row by row. To avoid having a stream of bytes that is a too long, a TIFF image is divided into strips. The client must read most of the file to get the piece of the image it is interested in.

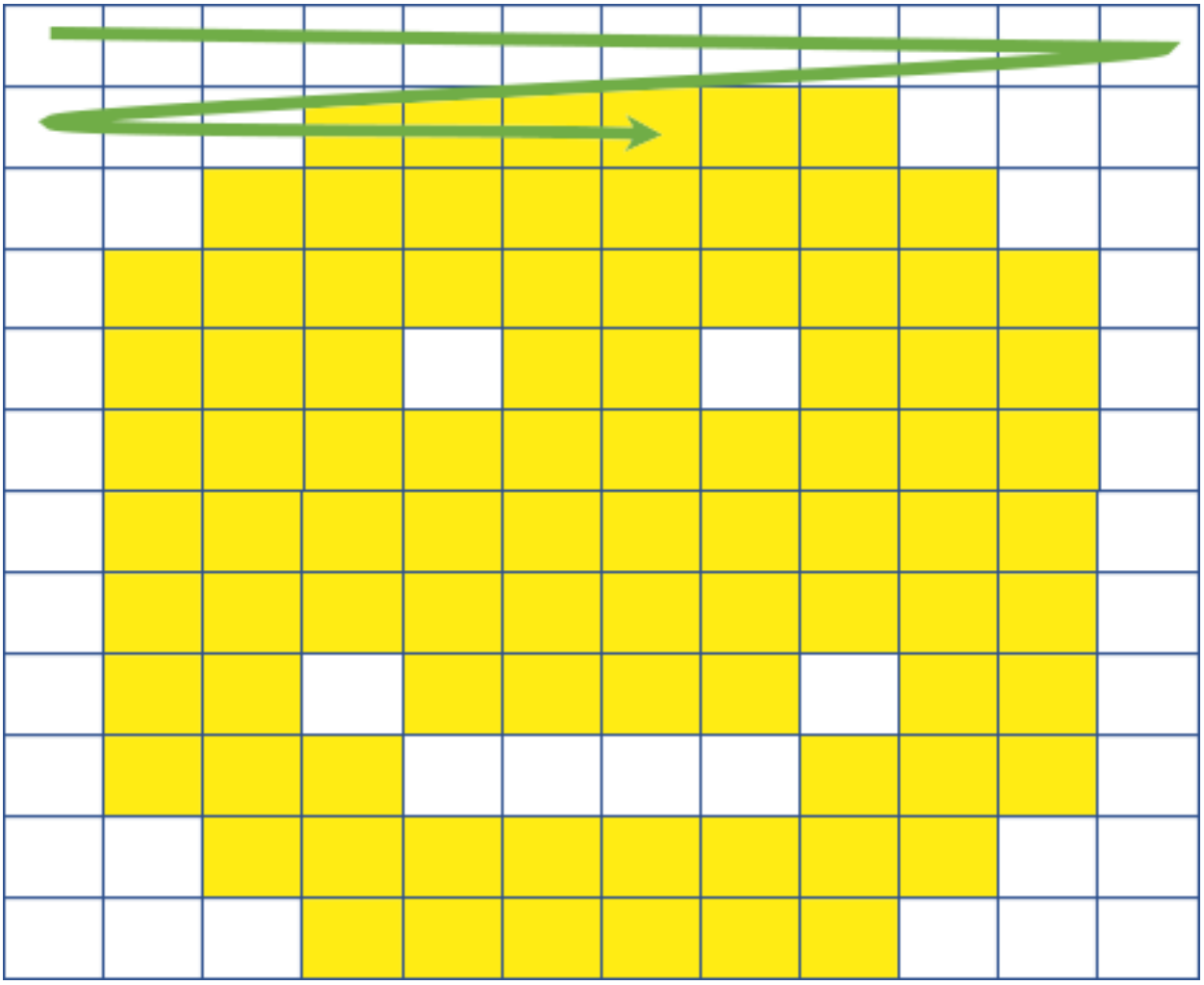

Figure 2 — Traditional raster image storage is row by row as indicated by the green path (from https://www.element84.com/blog/cloud-optimized-geotiff-vs-the-meta-raster-format)

Instead, COG stores image data in tiles (a schema introduced in TIFF v6). With a tile matrix (commonly known as *tiling* in the COG community), only the tiles covering the area of interest need to be read by the client making data extraction and visualization faster (see Figure 3)

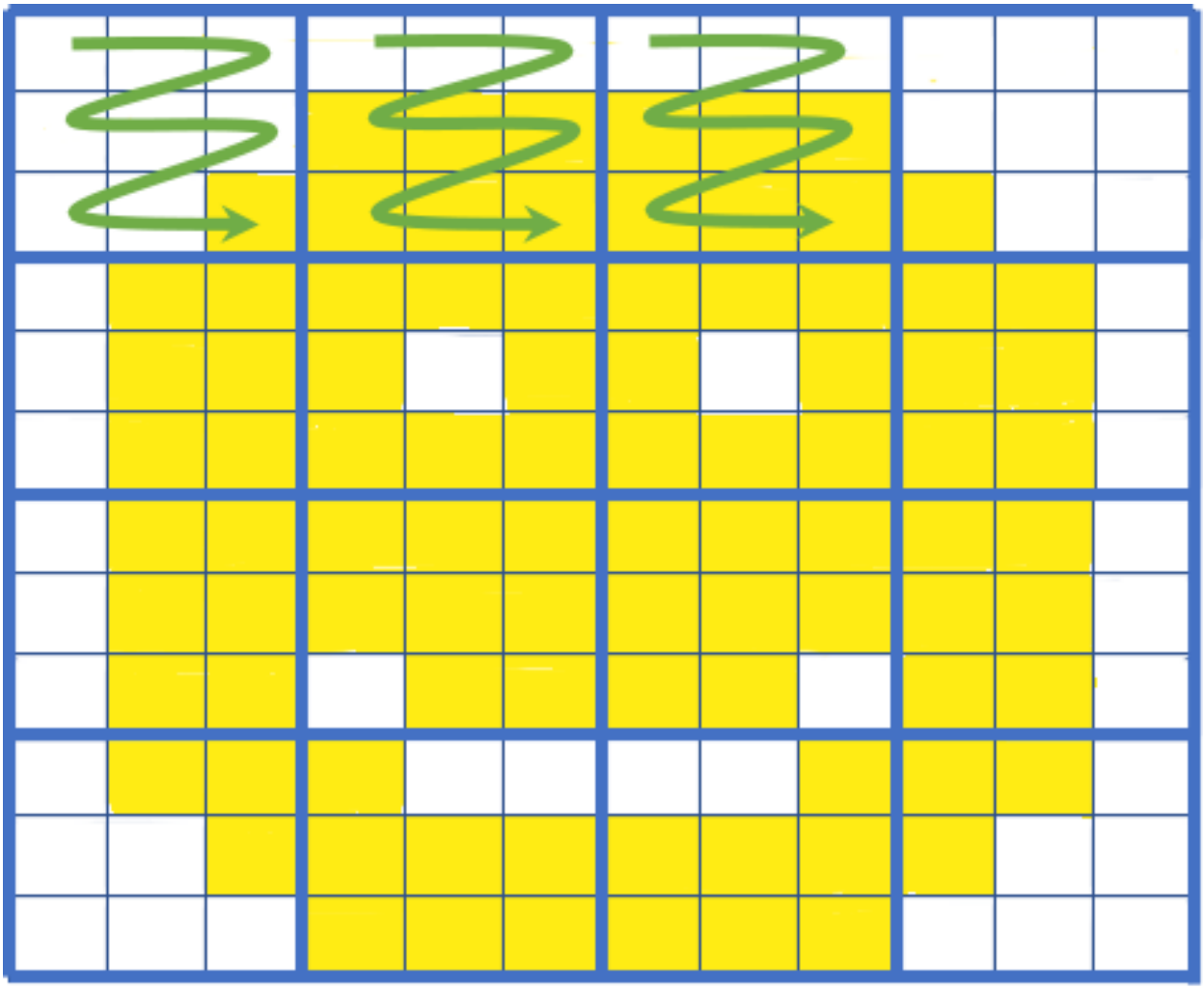

Figure 3 — COGs store images tile by tile instead of row by row (from https://www.element84.com/blog/cloud-optimized-geotiff-vs-the-meta-raster-format)

5.2. Reduced-Resolution Subfiles

Clients do not always need to show a full resolution image. Reduced-Resolution Subfiles (commonly known as *overviews* in the COG community) are down-sampled versions of the original image. They represent "zoomed out" versions of the image (see Figure 4).

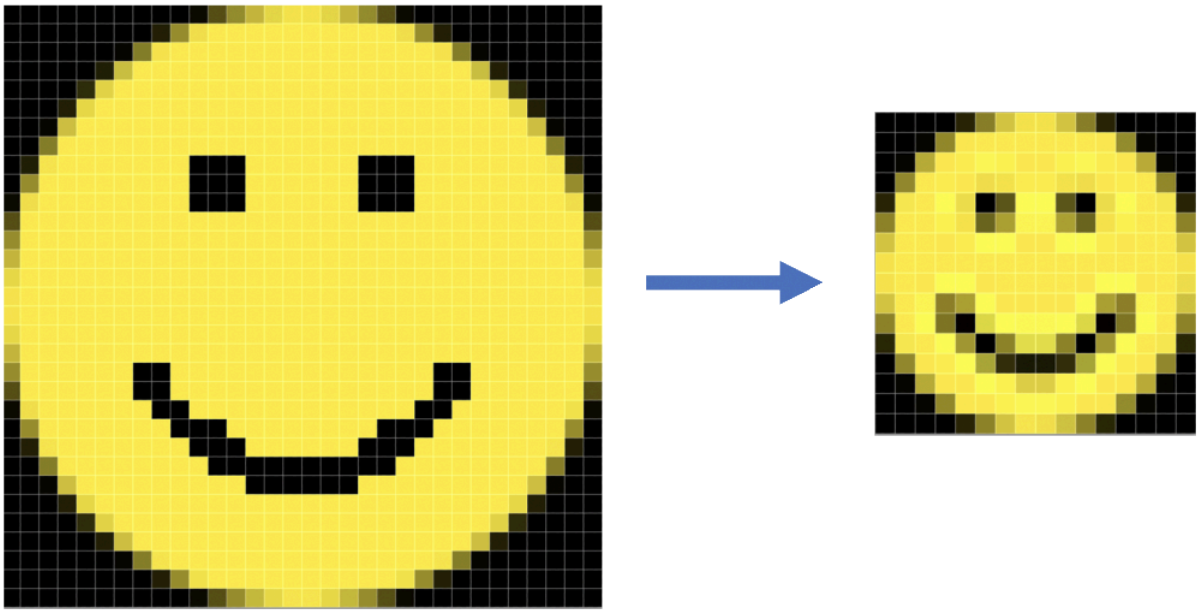

Figure 4 — Image reduction of resolution to generate overviews (from https://www.element84.com/blog/cloud-optimized-geotiff-vs-the-meta-raster-format)

Multiple overviews can be stored in a COG file to match multiple *zoom levels*. *Overviews* are stored as tiles just like the original image and they share the same georeference. So an application that supports zooming only needs to retrieve the tiles for the overview associated with the given *zoom level*.

The use of tiles is not new and is used in several circumstances, as shown in Table 1, which identifies use of tiles in servers implementing Web Feature Service (WFS), Web Map Service (WMS), Web Coverage Service (WCS), and Web Map Tile Service (WMTS) Standards. Servers can structure their internal data in tiles (that are invisible to the user) to produce faster responses or have rendered tiles directly that can be presented to the user. When data is transmitted to the client and rendered on the client side, COG is an ideal solution for gridded data.

Table 1 — Use of tiles in different services and approaches

SERVICE TYPE	FEATURE BASED DATA	GRIDDED DATA	SUMMARY
server-side rendering	WMS	WMS	easy to consume, does not require client processing
server-side rendering and client presentation	WMTS	WMTS	easy to consume, and allow for caching
data download	WFS	WCS	hardly suited for visualization (complex API, too much data)
client-side rendering	tiled feature data	COG	heavy client processing, much more rendering possibilities

(inspired by https://github.com/openlayers/openlayers/issues/10733)

To make the COG format an efficient data container available over the web, there is a need to be able to request a fragment of a file. Fortunately, HTTP 1.1 introduced support for range requests. *Range* requests enable a client to request only a portion of an HTTP message from a server. A server indicates support for range requests by returning the header Accept-Ranges: bytes. In the case of COGs, this enables the client to request the TIFF file header, the GeoTIFF tags, and the relevant individual tiles or tile ranges without downloading the entire file, as illustrated by Figure 5.

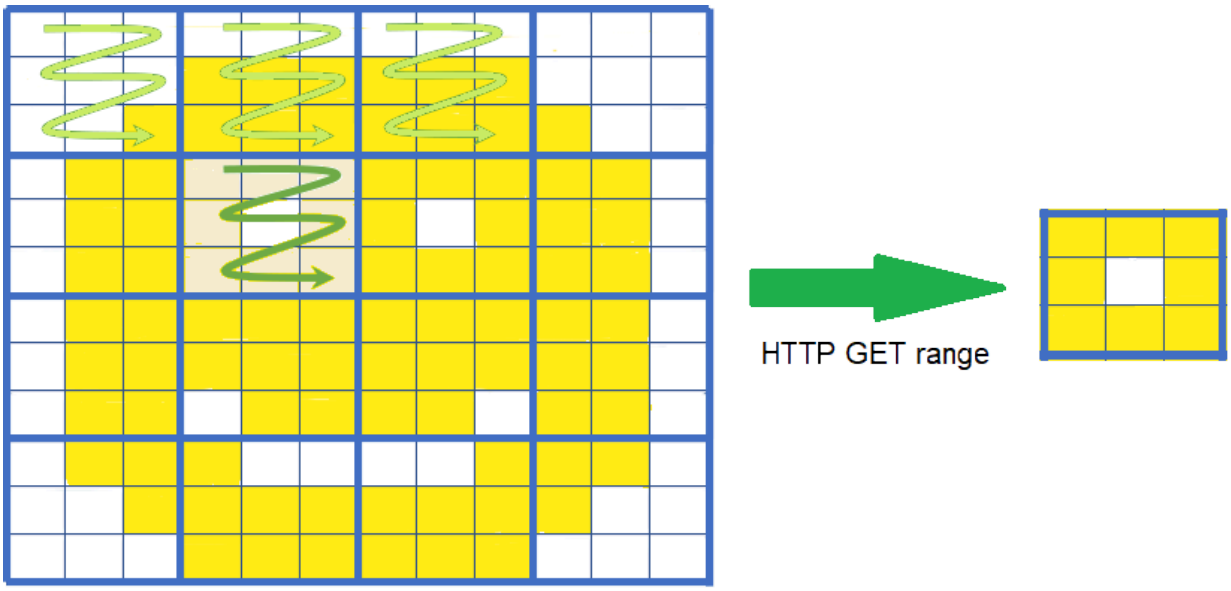

Figure 5 — Image transmission of only one tile

After this introduction, the ER is structured into the following sections:

Section 5 discusses the level of adoption of the COG format by clients and data providers

Section 6 discusses how COG is implemented in practice by providing examples

Section 7 proposes a set of requirements classes that could constitute the starting point for a future COG standard in the OGC Standards program

Section 8 discusses some issues found in the current approach for COG and proposes solutions

Section 9 proposes a set of requirements classes that could constitute the starting point for a future BigTIFF standard in the OGC Standards program

6

KEY FINDINGS

6 KEY FINDINGS

6.1. TIFF v6 and BigTIFF

GeoTIFF (and, by extension COG) is dependent on TIFF v6 which is limited to files sizes of less than 4 GBytes. However, implementations of GeoTIFF (and COG) overcome this limitation by adopting BigTIFF as an alternative to the TIFF v6 headers. There is currently a misalignment between the GeoTIFF Standard (that ignores BigTIFF) and its implementations (that implement BigTIFF) that could be solved by formalizing BigTIFF as an official standard and referencing BigTIFF in the GeoTIFF Standard.

6.2. Georeference in COG

From all of the tile matrices provided at different resolutions in a COG file, only the highest resolution is georeferenced by GeoTIFF tags. The georeference of the lower resolutions depends on this georeference. Clients have to deduce the georeference of the reduced resolution subfiles by computing a ratio between the number of columns of the highest resolution and the reduced resolution (and another ratio for the number of rows). Instead, COG could take advantage of the tile matrix set data structure defined in the OGC 17-083r2 OGC Two Dimensional Tile Matrix Set Standard (2d-TMS). COG could also include a Tile Matrix Set definition and tile indices as new GeoTIFF tags. More work and discussions are needed to assess the convenience of this alignment between OGC and 2d-TMS.

7

FUTURE WORK

7 FUTURE WORK

The work done to produce the list of requirements for COG as well as the list of requirements for BigTIFF will be transferred to the OGC GeoTIFF SWG for consideration as starting point for candidate Standards.

The strategy to divide a dataset into tiles or other smart organizations such as R-Trees, etc. could be applied to other file formats that could later be retrieved using the HTTP 1.1 range function. As an example, the header of a ZIP file could be retrieved and the information shown to the user in a web browser without the need to unzip the whole ZIP file. Future work could be done to determine if other file formats could get the same performance increase if organized conveniently and HTTP range is applied. For example, could we use a ZIP file to store vector tiles that can be unzipped at will when needed? The Testbed participants have already detected some implementation of HTTP range to read fragments of NetCDF v3, FlatGeoBuf (feature based format, https://github.com/flatgeobuf/flatgeobuf), Shapefiles, and others.

A good evaluation of netCDF-4 / HDF5 in cloud environment could be useful. The Testbed participants have seen some criticism against netCDF-4 in the Open Data Cube (ODC) community and it could be good to find out if there are intrinsic limitations in the netCDF-4 format or it is due to a limitation in the HDF5 library that can be overcome with more efficient code.

8

OVERVIEW

8 OVERVIEW

Traditionally, high-resolution imagery requires big files that have to be entirely downloaded to the client to be analyzed or visualized. This can require considerable download time, thus preventing the creation of real-time applications. One of the most popular formats in the world is the TIFF format. The format was created by the Aldus Corporation for use in desktop publishing. Aldus released the last version of the TIFF specification in 1992 (v. 6.0), subsequently updated with an Adobe Systems copyright after the Adobe acquired Aldus in 1994. Several Aldus or Adobe technical notes have been published with minor extensions to the format, and several specifications have been based on TIFF 6.0, including TIFF/EP (ISO 12234-2), TIFF/IT (ISO 12639), TIFF-F (RFC 2306) and TIFF-FX (RFC 3949), GeoTIFF (OGC 19-008r4, v1.1), and BigTIFF.

TIFF is a flexible, adaptable file format for handling images and data within a single file by including the header tags (size, definition, image-data arrangement, applied image compression) that define the images. The ability to store image data in a lossless format makes a TIFF file a useful image archive. TIFF can be used to store grey scale, color, or RGB images as well as integer or floating point data making it ideal as a support for storing the rangeset of grid coverage data.

To improve TIFF performance over the web, COG relies on two characteristics of the TIFF v6 format, the georeference GeoTIFF keys and a relatively unused property of HTTP (GET Range). This way, COG allows for efficient streaming of imagery and grid coverage data in the web, enables fast data visualization and facilitates faster geospatial processing workflows. This particular type of TIFF has been recently used to set up large series of remote sensing images in repositories of cloud providers (e.g., Amazon Web Services) enabling cloud processing at lower traffic. In fact, COG-aware software can request just the portions of data that it needs, improving accessing time and bandwidth. This is why it is called “Cloud Optimized GeoTIFF.”

COG is based on the GeoTIFF standard and does not introduce new capabilities that are not already in TIFF v6. As such, legacy software should be able to read COG files with no additional modifications. However, the legacy software will not be able to take advantage of the streaming capabilities, but still can easily download the whole file and read it.

The amount of data available for geospatial analytics has increased considerably in recent years. Therefore, downloading the data into a single computer is often not feasible. Providing data in the COG format can help decrease how much data is downloaded and copied. This is because online software systems can stream the data applications do not need to keep their own copy of the data for efficient access. New online software can access the content efficiently, while old versions can download completely. This avoids the need to have multiple copies of the files: one for fast access and another for download purposes.

COG relies on two complementary approaches already available in the existing standards to achieve its goal:

- The first is the ability of GeoTIFF to store the raw pixels of the image organized in an efficient way; and

- The second is HTTP GET Range requests, that let web clients request just the portions of a file that they need.

Using the first approach COG organizes the GeoTIFF so the latter requests can easily select and get the parts of the file that are useful for processing.

8.1. Efficient organization of data in a TIFF file

The Tiling and Reduced-Resolution Subfiles (sometimes called Overviews) in the GeoTIFF format supports the necessary structure for COG files so that the HTTP GET Range queries can request just the part of the file that is relevant.

Reduced-Resolution Subfiles come into play when the client wants to render a quick image of the whole or a big part of the area represented in the file. Instead of downloading every pixel, the software can just request a smaller, already created, lower-resolution version. The structure of the COG file on an HTTP Range supporting web server enables client software to easily find just the part of the whole file that is needed.

Tiles come into play when some small area of the overall extent of the COG file needs to be processed or visualized. This could be part of a reduced-resolution subfile, or it could be at full resolution. Tile organization makes all the relevant bytes of an area (a tile) to be in the same part of the file, so the software can use HTTP GET Range request to get only the tiles it needs.

8.2. Relation to OGC Tile Set Standards

The combined use of tiles and resolution levels is not new in OGC Standards. In fact the OGC Two-dimensional Tile Matrix Set standard (and the older OGC WMTS 1.0) use exactly the same approach. However, the draft OGC API — Tiles specification and the older WMTS 1.0 Standard require either a service to be installed in the web server provided or thousands of pre-generated independent tiles to be created. None of this is necessary in the COG approach as most of the modern web services natively support HTTP range.

Improving the relationship between COG and 2DTMS can be beneficial, so this document includes an extra requirement class for COG that could support using a list of COGs to store a Tile Set based on Common Tile Matrix Sets.

9

IMPLEMENTATIONS OF COG

9 IMPLEMENTATIONS OF COG

This section offers some examples of the current implementations of the COG format. This is not an exhaustive list. Some more relevant examples of implementations may be missing.

9.1. How to support COG in browsers

COG is not supported natively in web browsers. However, EOX (https://eox.at) has developed a set of JavaScript files that are used by the COG explorer library https://github.com/geotiffjs/cog-explorer. The library adds COG support to any modern browser. This demo portal: https://geotiffjs.github.io/cog-explorer provides a live example of the use of this library.

To demonstrate the use of the COG explorer with any COG file, the participants tried to connect the COG explorer to the Spanish CNIG download center http://centrodedescargas.cnig.es/CentroDescargas/catalogo.do?Serie=LANDS#. In this site, the historical Landsat national mosaics are available as COG files. Unfortunately, the CNIG has decided to adopt a multipart file format as a response to an HTTP GET request. This approach has the advantage of suggesting a file name to save the file when it is downloaded. However, there are two disadvantages: it forces encoding the COG file in base64 (inflating the size of the transmission) and it does not support COG HTTP GET range requests, thereby making the random access over HTTP impossible. This also makes the CNIG website incompatible with the COG explorer. Instead, the participants decided to download one of the available files and copy it to one of the CREAF servers. Once copies in the web shared folder, a normal HTTP GET request with range was possible. The file was exposed in the following url: https://joanma.uab.cat/temp/cog/landsat5_1991.tif.

9.1.1. COG and CORS in browsers

The COG-explorer is only available as an HTTPS endpoint. The rules recently auto-imposed in web browsers prevent a page that comes from a HTTPS domain to read data from an HTTP service. To be compatible with COG-Explorer, servers have to expose COG files in a URL starting with https://. To meet this requirement, an Internet Information Service (IIS) in Windows 2012r2 as a web server was selected. IIS in Windows 2012r2 supports “HTTP range” by default. However, that is not enough due to other Cross-Origin Resource Sharing (CORS) restrictions imposed by browsers. To “authorize” the COG-explorer domain (geotiffjs.github.io) to read the COG file in another domain (joanma.uab.cat), adding these two headers in the IIS configuration was necessary.

```
Access-Control-Allow-Headers: range
Access-Control-Allow-Origin: *
```

This way, the client knows that the http://joanma.uab.cat authorizes any other domain to read data and check for range headers. Now, server data can be read by the COG-explorer application directly (see Figure 6).

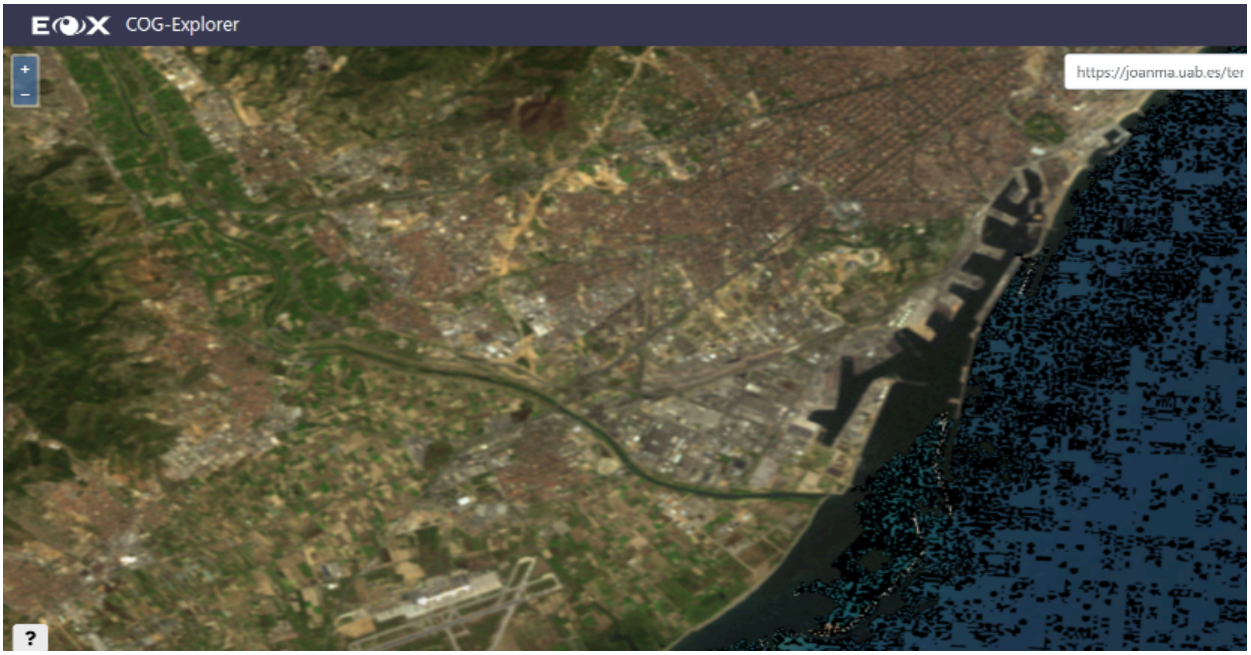

Figure 6 — Result of loading a COG file in the COG explorer that can be reproduced with https://geotiffjs.github.io/cog-explorer/#long=2.131&lat=41.347&zoom=8&scene=https://joanma.uab.es/temp/cog/landsat5_1991.tif

To demonstrate that library is able to request only the data belonging to the necessary tiles at the level of resolution needed to populate the screen without having to download the whole file, the Chrome *Developer Tools* can be used. The *Network* tab in the Chrome *Developer Tools* provides a list of the HTTP requests that are happening behind the scenes and reveals that the same COG file is accessed in several requests (see Figure 7).

landsat5_1991.tif	OPTIONS	200	preflight	Preflight	0 B
landsat5_1991.tif	GET	206	fetch	fetch.js:37	65.9 kB
wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap...	GET	200	jpeg	tileimage.js:371	22.1 kB
landsat5_1991.tif	OPTIONS	200	preflight	Preflight	0 B
wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap...	GET	200	jpeg	tileimage.js:371	1.4 kB
wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap...	GET	200	jpeg	tileimage.js:371	23.4 kB
wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap...	GET	200	jpeg	tileimage.js:371	8.2 kB
landsat5_1991.tif	GET	206	fetch	fetch.js:37	65.9 kB
wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap...	GET	200	jpeg	tileimage.js:371	18.0 kB
landsat5_1991.tif	OPTIONS	200	preflight	Preflight	0 B
landsat5_1991.tif	GET	206	fetch	fetch.js:37	197 kB
landsat5_1991.tif	OPTIONS	200	preflight	Preflight	0 B
landsat5_1991.tif	GET	206	fetch	fetch.js:37	65.9 kB
favicon.ico	GET	200	vnd.microsof...	Other	(disk cache)
landsat5_1991.tif	GET	206	fetch	fetch.js:37	65.9 kB
landsat5_1991.tif	GET	206	fetch	fetch.js:37	0 B
landsat5_1991.tif	GET	206	fetch	fetch.js:37	65.9 kB
landsat5_1991.tif	GET	206	fetch	fetch.js:37	197 kB
landsat5_1991.tif	GET	206	fetch	fetch.js:37	(disk cache)
landsat5_1991.tif	GET	206	fetch	fetch.js:37	263 kB
landsat5_1991.tif	GET	206	fetch	fetch.js:37	65.9 kB
landsat5_1991.tif	GET	206	fetch	fetch.js:37	65.9 kB
landsat5_1991.tif	GET	206	fetch	fetch.js:37	65.9 kB

Figure 7 — COG Explorer does several requests to the server

Some of the requests are HTTP HEAD requests that check if the connection will work. The HEAD request is also used to communicate the full size of the COG file (see Figure 8). This is necessary to know the valid range of a HTTP *range* header.

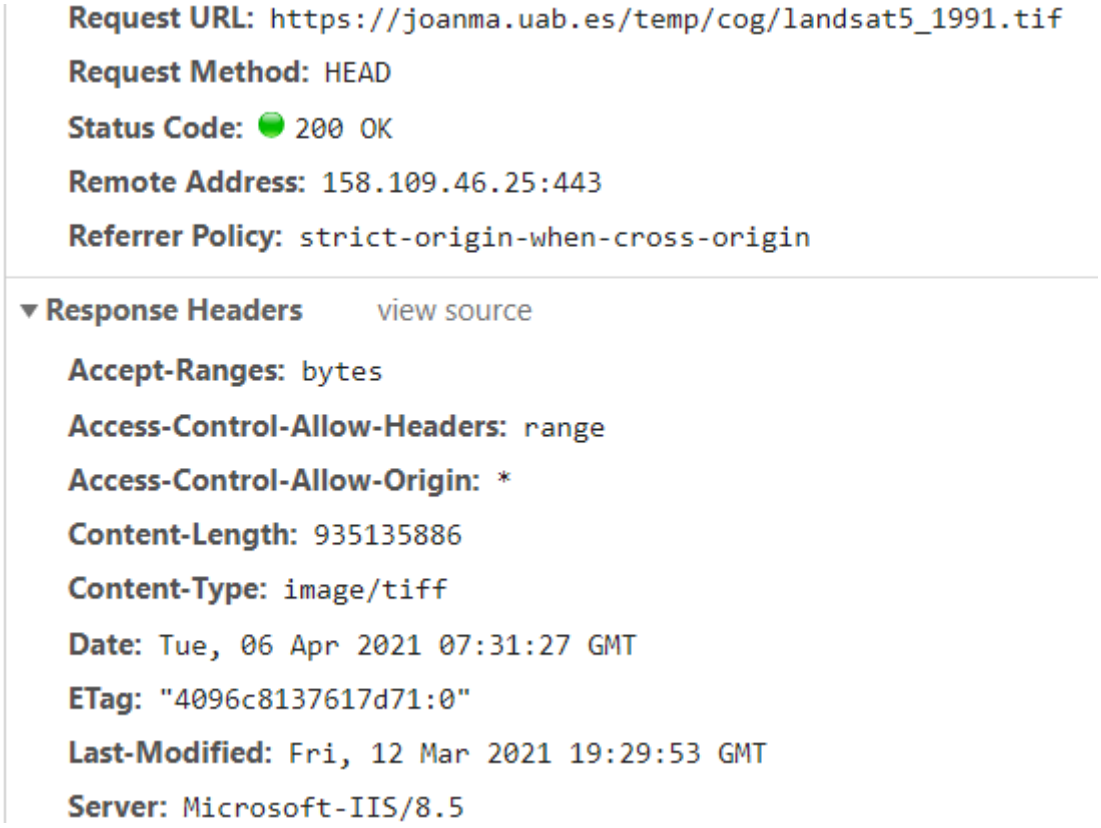

Figure 8 — COG Explorer HEAD request sample

Others requests are GET requests with the range header that limits the volume of data retrieved to a few kilobytes (see Figure 9).

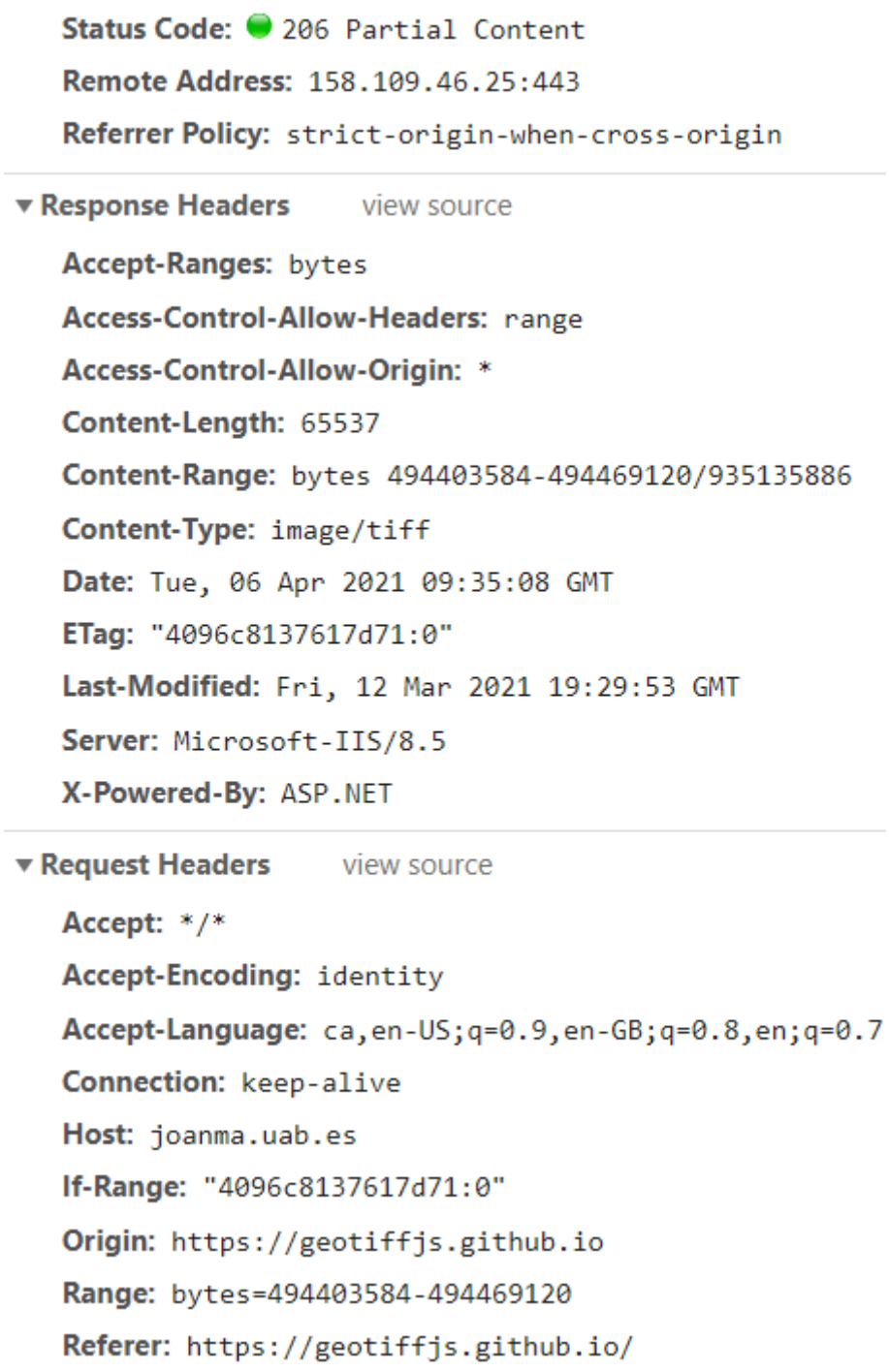

Figure 9 — COG Explorer GET request sample asking data in the 494403584-494469120 interval of bytes.

9.2. COG support in desktop solutions

Support for geospatial raster formats in many desktop solutions is driven by the GDAL libraries. There is a GDAL driver to read and write COG files. This driver was introduced in GDAL version 3.1 and now version 3.3 has the following high level characteristics:

- Support for TIFF or BigTIFF file;
- Create tiles (512 pixels by default) for imagery, mask, and overviews;
- Create lower resolution overviews until the maximum size of the smallest overview level is lower than 512 pixels;
- Compress the tiles if requested;
- Support pixel interleaving for multi-band dataset; and
- Create an optimized layout of TIFF sections to minimize the number of data requests needed by a reader doing random read access.

More about this driver is here: https://gdal.org/drivers/raster/cog.html

9.3. COG support in map services

The GeoServer community has developed an extension to support COG: https://docs.geoserver.org/latest/en/user/community/cog/cog.html

The COG plugin does not add new *stores*. Instead, the plugin adds COG support to existing *stores*. When configuring a GeoTIFF *store*, a new checkbox is available: "Cloud Optimized GeoTIFF (COG)." The plugin gives support for HTTP range and provides an optimized implementation for Amazon S3 buckets.

9.4. COG support in data cubes

The Open Data Cube (ODC) is an Open Source Geospatial Data Management and Analysis Software project that helps the user harness the power of Satellite data. At its core, the ODC is a set of Python libraries and a PostgreSQL database that helps the user work with geospatial raster data. In the past, in order to accelerate access to long time series, the Open Data Cube had a double-step process for adding remote sensing satellite data in the system. After downloading the data, the files were first indexed and latter transformed and ingested as netCDF tiles. Indexing is a process of including a reference to a downloaded file in a database. Even if reading the indexed version of the files is possible, depending on the format, such an

index could not be optimal. Ingesting is a process of taking the indexed files and transforming those files into a set of netCDFs files, each one representing a big tile. Ingesting requires time and requires disk space as it creates a duplicate of the data but results in an increase in data retrieval performance. If original data is downloaded as COG, there is no need for ingesting the data since the indexed version is performing optimally. That is why the new version of ODC replaces by a direct indexing of COG data files, avoiding the need for data ingestion and transformation. The documentations says that the limitation of netCDF reading to a single thread means that reading from COG files on disk could be faster than ingested netCDF in some situations. The single thread limitation is originated by the library used and not by the file format.

9.5. COG data adoption

At the time of writing this ER, a two-speed level of adoption was observed. Some cloud-based data repositories are duplicates of the official space agencies' distribution systems created by some Earth observation companies. This is showing that Earth observation industry sees a business opportunity in simplifying access to remote sensing data using COG. These repositories have catalogues based on SpatioTemporal Asset Catalog (STAC) that are exposing the remote sensing data transformed to COG format. These are two examples:

- Accessing Landsat 8 Collection 2 Level 2 data with the Planetary Computer STAC API (https://planetarycomputer.microsoft.com/dataset/landsat-8-c2-l2#Example-Notebook)
- Sentinel-2 Cloud-Optimized GeoTIFFs (https://registry.opendata.aws/sentinel-2-l2a-cogs/)

On the other hand, there are efforts from the remote sensing agencies themselves to migrate to COG format. A significant example is the USGS, which is making the whole Landsat archive available directly in COG (https://prd-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/atoms/files/LSDS-1388-Landsat-Cloud-Optimized-GeoTIFF_DFCB-v2.0.pdf) as well as other products such as the USGS Digital Elevation Model (https://www.usgs.gov/news/usgs-digital-elevation-models-dem-switching-new-distribution-format). NASA also considers COG an appropriate format to stimulate participation in a hackathon with the objective of reusing remote sensing data (https://earthdata.nasa.gov/collaborate/cloud-optimized-geotiffs).

9.6. Criticism to the approach

One of the main impediments to COG adoption is the lack of support in browsers. This means that libraries are needed to make COG readable in browsers and it will be desirable to have native support to maps in web browsers (see Figure 10 and Figure 11)

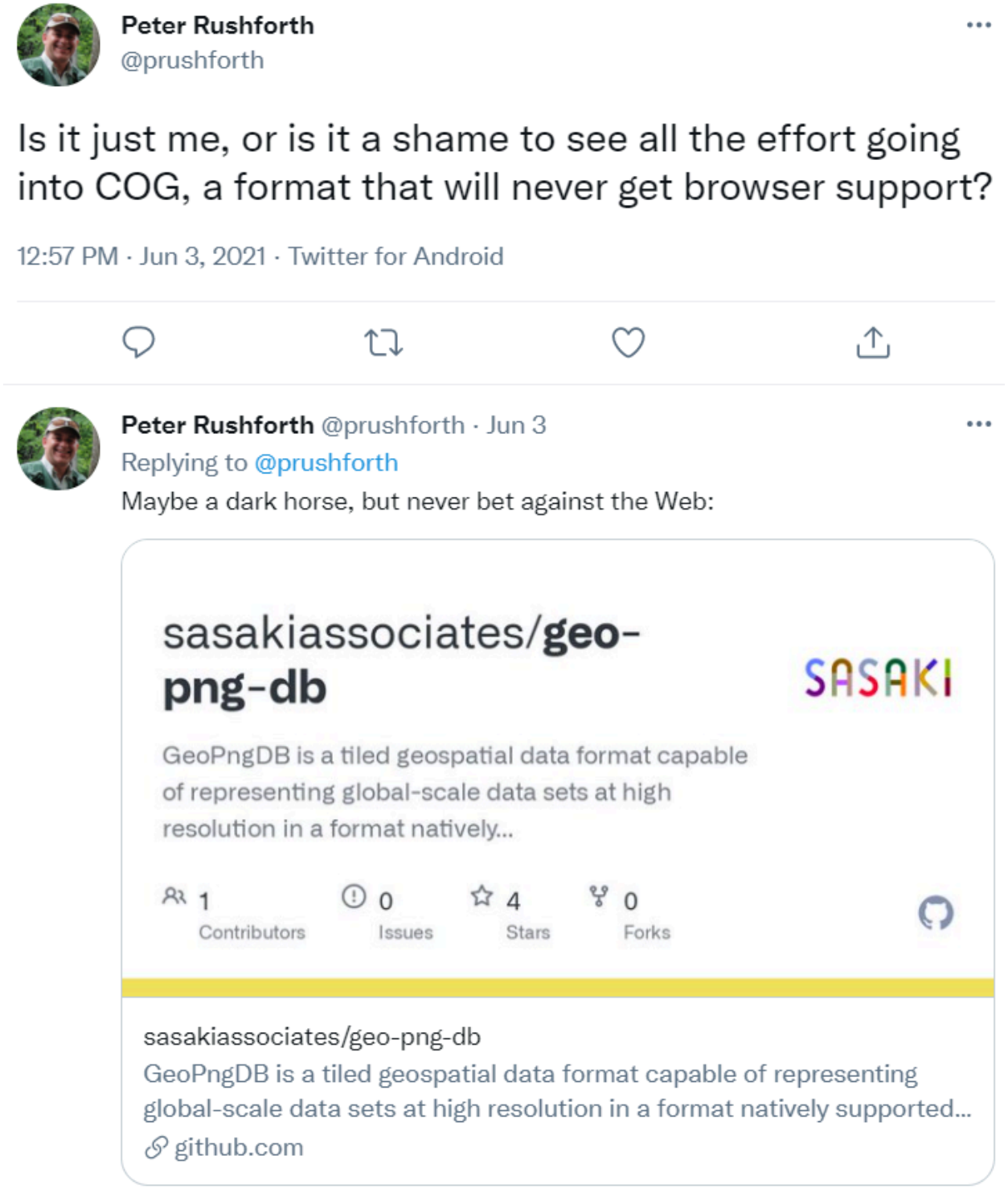

Figure 10 — Twitter about COG not being natively supported by browsers (Part 1)

Figure 11 — Twitter about COG not being natively supported by browsers (Part 2)

For example, a WMS service that uses COG data in the backend could overcome this limitation by providing on-the-fly translation from COG tiles to PNG or JPEG formats.

Other criticisms derive from the TIFF original design. For example, COG cannot easily be used for multidimensional data beyond 2D as the TIFF internal structure is limited to one or more 2D data subfiles.

9.7. Alternatives to COG

There are other formats that are alternatives to COG.

- Meta Raster Format (MRF) is an image and data storage format designed for fast access to imagery within a georeferenced tile pyramid at discrete resolutions. (https://github.com/nasa-gibs/mrf/blob/master/spec/mrf_spec.md)
- GeoPngDB is a tiled geospatial data format capable of representing global-scale data sets at high resolution in a format natively supported by web browsers. (https://github.com/sasakiassociates/geo-png-db)
- The Cloud Raster Format (CRF) is an Esri-created raster format that is optimized for writing and reading large files in a distributed processing and storage environment.
- GeoPackage Tiled Gridded Coverages is a way to store tiles in a GeoPackage. The tiles are stored as TIFF files. In this case the TIFF should not have internal

tiles but they are independent TIFF files each one representing a single tile. This way you can extract the desired tile (one TIFF tile) using a SQLite queries. This approach does not support HTTP GET Range to request a set of bytes in the GeoPackage, but nothing prevents setting up a web service that uses GeoPackage in the backend to provide access to the tiles, one by one.

- netCDF-4 / HDF5 are formats for scientific data. Data can be structured in several tables that represent 2-dimensional arrays or in cubes of n-dimensional arrays. This tables or cubes could be internally arranged in tiles and overviews. Some metadata will be needed to make the structure of tables discoverable. With the right libraries, netCDF-4 / HDF5 could be accessed over the network using HTTP range.

It is worth mention that optimized organizations for the Cloud are also being defined for other types of coverages. For example, the Cloud Optimized Point Cloud (COPC) (https://copc.io/) is a LAZ 1.4 file that stores a point cloud organized in a clustered octree. COPC contains a Variable Length Record (VLR) that describes the octree organization of data that are stored in standard LAZ 1.4 chunks. It was inspired by the Entwine Point Tile format (https://entwine.io/entwine-point-tile.html) but instead of using some JSON files, the data is stored in binary structures in the LAZ format. A reader does not know about COPC can still read the LAZ without taking advantage of the octree organization.

10

COG IN PRACTICE

10 COG IN PRACTICE

In order to be able to define a COG standard or best practice, the community needs to pay attention to the existing documentations and implementations.

10.1. Current specification

The COG website at https://www.cogeo.org/ provides some help on how COG works and directs to the current specification in GitHub: https://github.com/cogeotiff/cog-spec/blob/master/spec.md. The specification is surprisingly short and consists of one page text and a description of a reference implementation in the GDAL library. The document is contributed by Jeff Albrecht, Kyle Barron, and Vincent Sarago.

10.2. Common organization of a COG file

In a COG file created by the GDAL library using libtiff >= 4.0.11, the information for a COG file results in a TIFF file (or a BigTIFF file) organized in sections following the sequence below.

1. TIFF/BigTIFF header/signature and pointer to first IFD (Image File Directory)
2. Internal GDAL information called “ghost area”
3. IFD of the full resolution image, followed by GeoTIFF tags values, excluding the TileOffsets and TileByteCounts arrays
4. IFD of the mask of the full resolution image, if present, followed by GeoTIFF tags values, excluding the TileOffsets and TileByteCounts arrays
5. IFD of the first (largest in dimensions) overview level of the full resolution image, if present
6. IFD of the following overviews (repeated)
7. IFD of the last (smallest) overview level, if present
8. IFD of the first (largest in dimensions) overview level of the mask, if present
9. IFD of the following mask overviews (repeated)
10. IFD of the last (smallest) overview level of the mask, if present
11. TileOffsets and TileByteCounts arrays of the above IFDs

12. tile data of the smallest overview, if present (with each tile followed by the corresponding tile of mask data, if present), with leader and trailer bytes

13. tile data of the following overviews (repeated)

14. tile data of the largest overview, if present (interleaved with mask data if present)

15. tile data of the full resolution image, if present (interleaved with corresponding mask data if present)

Note that the IFD or full-resolution image and the overviews and the actual tile data are in opposite order. The full-resolution data is at the end to the file while the smallest overview is at the beginning of the file (after the IFDs). Please note that only the full resolution image contains GeoTIFF. The georeference of the overviews need to be derived from the relation between number of columns and rows between the full resolution image and the overviews.

10.3. Validating and extracting GeoTIFF metadata from COG files

There is an online tool to validate if a COG is well formed based on the GDAL library. If COG becomes an OGC standard in the future, it could be the basis for testing implementations: http://cog-validate.radiant.earth/html

This online tool has a user interface and can also be invoked as an API with the url as a single query parameter. In the latter case, the response is a JSON document containing several details extracted from the file including the bounding box, the tile count for each overview and band, the Well Known Text (WKT) CRS information, the geotransform matrix, and the offset to each overview.


```
{
 "status": "success",
 "gdal_info": {
  "files": [ "/vsicurl/https://joanma.uab.es/temp/cog/landsat5_1991.tif" ],
  "cornerCoordinates": {
   "upperRight": [ 1132533.893, 4883529.188 ],
   "lowerLeft": [ -74126.107, 3870639.188 ],
   "lowerRight": [ 1132533.893, 3870639.188 ],
   "upperLeft": [ -74126.107, 4883529.188 ],
   "center": [ 529203.893, 4377084.188 ]
  },
  "wgs84Extent": {
   "type": "Polygon",
   "coordinates": [
    [
     [ -10.1459972, 43.8809094 ],
     [ -9.273279, 34.8161184 ],
     [ 3.9075608, 34.7816184 ],
     [ 4.8665274, 43.8333089 ],
     [ -10.1459972, 43.8809094 ]
    ]
   ]
```

```
},
"description": "https://joanma.uab.es/temp/cog/landsat5_1991.tif",
"driverShortName": "GTiff",
"driverLongName": "GeoTIFF",
"bands": [
 {
  "mask": {
   "overviews": [
    {
     "size": [ 20111, 16882 ]
    },
    {
     "size": [ 10056, 8441 ]
    },
    {
     "size": [ 5028, 4221 ]
    },
    {
     "size": [ 2514, 2111 ]
    },
    {
     "size": [ 1257, 1056 ]
    },
    {
     "size": [ 629, 528 ]
    },
    {
     "size": [ 315, 264 ]
    },
    {
     "size": [ 158, 132 ]
    },
    {
     "size": [ 79, 66 ]
    }
   ],
   "flags": [ "PER_DATASET", "ALPHA" ]
  },
  "band": 1,
  "colorInterpretation": "Red",
  "overviews": [
   {
    "size": [ 20111, 16882 ]
   },
   {
    "size": [ 10056, 8441 ]
   },
   {
    "size": [ 5028, 4221 ]
   },
   {
    "size": [ 2514, 2111 ]
   },
   {
    "size": [ 1257, 1056 ]
   },
   {
    "size": [ 629, 528 ]
   },
   {
    "size": [ 315, 264 ]
   },
   {
```

```
      "size": [ 158, 132 ]
     },
     {
      "size": [ 79, 66 ]
     }
    ],
    "type": "Byte",
    "block": [ 256, 256 ],
    "metadata": {

    }
   },
   {
    "mask": {
     "overviews": [
      {
       "size": [ 20111, 16882 ]
      },
      {
       "size": [ 10056, 8441 ]
      },
      {
       "size": [ 5028, 4221 ]
      },
      {
       "size": [ 2514, 2111 ]
      },
      {
       "size": [ 1257, 1056 ]
      },
      {
       "size": [ 629, 528 ]
      },
      {
       "size": [ 315, 264 ]
      },
      {
       "size": [ 158, 132 ]
      },
      {
       "size": [ 79, 66 ]
      }
     ],
     "flags": [ "PER_DATASET", "ALPHA" ]
    },
    "band": 2,
    "colorInterpretation": "Green",
    "overviews": [
     {
      "size": [ 20111, 16882 ]
     },
     {
      "size": [ 10056, 8441 ]
     },
     {
      "size": [ 5028, 4221 ]
     },
     {
      "size": [ 2514, 2111 ]
     },
     {
      "size": [ 1257, 1056 ]
     },
```

```
    {
     "size": [ 629, 528 ]
    },
    {
     "size": [ 315, 264 ]
    },
    {
     "size": [ 158, 132 ]
    },
    {
     "size": [ 79, 66 ]
    }
   ],
   "type": "Byte",
   "block": [ 256, 256 ],
   "metadata": {

   }
  },
  {
   "mask": {
    "overviews": [
     {
      "size": [ 20111, 16882 ]
     },
     {
      "size": [ 10056, 8441 ]
     },
     {
      "size": [ 5028, 4221 ]
     },
     {
      "size": [ 2514, 2111 ]
     },
     {
      "size": [ 1257, 1056 ]
     },
     {
      "size": [ 629, 528 ]
     },
     {
      "size": [ 315, 264 ]
     },
     {
      "size": [ 158, 132 ]
     },
     {
      "size": [ 79, 66 ]
     }
    ],
    "flags": [ "PER_DATASET", "ALPHA" ]
   },
   "band": 3,
   "colorInterpretation": "Blue",
   "overviews": [
    {
     "size": [ 20111, 16882 ]
    },
    {
     "size": [ 10056, 8441 ]
    },
    {
     "size": [ 5028, 4221 ]
```

```
      },
      {
       "size": [ 2514, 2111 ]
      },
      {
       "size": [ 1257, 1056 ]
      },
      {
       "size": [ 629, 528 ]
      },
      {
       "size": [ 315, 264 ]
      },
      {
       "size": [ 158, 132 ]
      },
      {
       "size": [ 79, 66 ]
      }
     ],
     "type": "Byte",
     "block": [ 256, 256 ],
     "metadata": {

     }
    },
    {
     "band": 4,
     "colorInterpretation": "Alpha",
     "overviews": [
      {
       "size": [ 20111, 16882 ]
      },
      {
       "size": [ 10056, 8441 ]
      },
      {
       "size": [ 5028, 4221 ]
      },
      {
       "size": [ 2514, 2111 ]
      },
      {
       "size": [ 1257, 1056 ]
      },
      {
       "size": [ 629, 528 ]
      },
      {
       "size": [ 315, 264 ]
      },
      {
       "size": [ 158, 132 ]
      },
      {
       "size": [ 79, 66 ]
      }
     ],
     "type": "Byte",
     "block": [ 256, 256 ],
     "metadata": {

     }
```

```
    }
   ],
   "coordinateSystem": {
    "wkt": "PROJCS[\"WGS 84 / UTM zone 30N\",\n    GEOGCS[\"WGS 84\",\n
        DATUM[\"WGS_1984\",\n            SPHEROID[\"WGS 84\",6378137,298.
257223563,\n                AUTHORITY[\"EPSG\",\"7030\"]],\n
    AUTHORITY[\"EPSG\",\"6326\"]],\n        PRIMEM[\"Greenwich\",0,\n
           AUTHORITY[\"EPSG\",\"8901\"]],\n        UNIT[\"degree\",0.
0174532925199433,\n            AUTHORITY[\"EPSG\",\"9122\"]],\n
 AUTHORITY[\"EPSG\",\"4326\"]],\n    PROJECTION[\"Transverse_Mercator\"],\n
    PARAMETER[\"latitude_of_origin\",0],\n    PARAMETER[\"central_meridian\",-
3],\n    PARAMETER[\"scale_factor\",0.9996],\n    PARAMETER[\"false_easting
\",500000],\n    PARAMETER[\"false_northing\",0],\n    UNIT[\"metre\",1,\n
         AUTHORITY[\"EPSG\",\"9001\"]],\n    AXIS[\"Easting\",EAST],\n
 AXIS[\"Northing\",NORTH],\n    AUTHORITY[\"EPSG\",\"32630\"]]"
   },
   "geoTransform": [ -74126.10725287361, 30.0, 0.0, 4883529.1883060075, 0.0, -
30.0 ],
   "metadata": {
    "": {
     "TIFFTAG_DATETIME": "Tue Feb  2 12:16:23 2021\n",
     "BIGTIFF": "YES",
     "BLOCKYSIZE": "256",
     "PHOTOMETRIC": "RGB",
     "COMPRESS": "LZW",
     "AREA_OR_POINT": "Area",
     "INTERLEAVE": "PIXEL",
     "BLOCKXSIZE": "256",
     "TILED": "NO",
     "ALPHA": "NON-PREMULTIPLIED",
     "TIFFTAG_SOFTWARE": "BlueMarble Geographics GeoCore"
    },
    "IMAGE_STRUCTURE": {
     "INTERLEAVE": "PIXEL",
     "COMPRESSION": "JPEG"
    }
   },
   "size": [ 40222, 33763 ]
  },
  "details": {
   "ifd_offsets": {
    "overview_8": 784204,
    "overview_6": 783216,
    "overview_7": 783750,
    "overview_4": 780596,
    "overview_5": 782426,
    "overview_2": 753256,
    "overview_3": 774766,
    "overview_0": 335356,
    "overview_1": 669442,
    "main": 16
   },
   "data_offsets": {
    "overview_8": 784593,
    "overview_6": 823612,
    "overview_7": 794429,
    "overview_4": 1165093,
    "overview_5": 905154,
    "overview_2": 5677864,
    "overview_3": 2093193,
    "overview_0": 73529801,
    "overview_1": 19658902,
    "main": 274599054
```

```
  }
 }
}
```

Another way to extract the internal metadata of a COG file is to use the EXIF tool: https://metacpan.org/pod/release/EXIFTOOL/Image-ExifTool-5.67/exiftool.

The syntax as follows generates a report of the file metadata.

```
exiftool
c:\COG\pnt_landsat5_2006_mosaico_islas_canarias_b321_80porciento_b543_
20porciento_hu28_8bits_COG_90.tif -a -U -g1
```

Adding `-U` in the command line results in descriptions instead of tag names:

```
---- ExifTool ----
ExifTool Version Number         : 12.25
---- System ----
File Name                       : pnt_landsat5_2006_mosaico_islas_canarias_
b321_80porciento_b543_20porciento_hu28_8bits_COG_90.tif
Directory                       : D:/docs/Recerca/COG
File Size                       : 13 MiB
File Modification Date/Time     : 2021:05:02 10:00:26+02:00
File Access Date/Time           : 2021:05:07 23:03:06+02:00
File Creation Date/Time         : 2021:05:02 10:00:24+02:00
File Permissions                : -rw-rw-rw-
---- File ----
File Type                       : BTF
File Type Extension             : btf
MIME Type                       : image/x-tiff-big
---- IFD0 ----
Image Width                     : 15829
Image Height                    : 6520
Bits Per Sample                 : 8 8 8
Compression                     : JPEG
Photometric Interpretation      : RGB
Samples Per Pixel               : 3
X Resolution                    : 72
Y Resolution                    : 72
Planar Configuration            : Chunky
Resolution Unit                 : inches
Software                        : Adobe Photoshop CC 2018 (Windows)
Modify Date                     : 2020:06:25 10:30:36
Tile Width                      : 256
Tile Length                     : 256
Tile Offsets                    : (Binary data 13380 bytes, use -b option to
 extract)
Tile Byte Counts                : (Binary data 6873 bytes, use -b option to
 extract)
Sample Format                   : Unsigned; Unsigned; Unsigned
JPEG Tables                     : (Binary data 73 bytes, use -b option to
 extract)
Pixel Scale                     : 30 30 0
Model Tie Point                 : 0 0 0 187333.999630584 3255440 0
Geo Tiff Directory              : (Binary data 106 bytes, use -b option to
 extract)
Geo Tiff Ascii Params           : (Binary data 29 bytes, use -b option to
 extract)
---- ICC-header ----
Profile CMM Type                : Adobe Systems Inc.
Profile Version                 : 2.1.0
Profile Class                   : Display Device Profile
Color Space Data                : RGB
```

```
Profile Connection Space        : XYZ
Profile Date Time               : 1999:06:03 00:00:00
Profile File Signature          : acsp
Primary Platform                : Apple Computer Inc.
CMM Flags                       : Not Embedded, Independent
Device Manufacturer             : none
Device Model                    :
Device Attributes               : Reflective, Glossy, Positive, Color
Rendering Intent                : Perceptual
Connection Space Illuminant     : 0.9642 1 0.82491
Profile Creator                 : Adobe Systems Inc.
Profile ID                      : 0
---- ICC_Profile ----
Profile Copyright               : Copyright 1999 Adobe Systems Incorporated
Profile Description             : Adobe RGB (1998)
Media White Point               : 0.95045 1 1.08905
Media Black Point               : 0 0 0
Red Tone Reproduction Curve     : (Binary data 14 bytes, use -b option to
 extract)
Green Tone Reproduction Curve   : (Binary data 14 bytes, use -b option to
 extract)
Blue Tone Reproduction Curve    : (Binary data 14 bytes, use -b option to
 extract)
Red Matrix Column               : 0.60974 0.31111 0.01947
Green Matrix Column             : 0.20528 0.62567 0.06087
Blue Matrix Column              : 0.14919 0.06322 0.74457
---- IFD1 ----
Subfile Type                    : Reduced-resolution image
Image Width                     : 7915
Image Height                    : 3260
Bits Per Sample                 : 8 8 8
Compression                     : JPEG
Photometric Interpretation      : RGB
Samples Per Pixel               : 3
Planar Configuration            : Chunky
Tile Width                      : 128
Tile Length                     : 128
Tile Offsets                    : (Binary data 12895 bytes, use -b option to
 extract)
Tile Byte Counts                : (Binary data 6789 bytes, use -b option to
 extract)
Sample Format                   : Unsigned; Unsigned; Unsigned
JPEG Tables                     : (Binary data 73 bytes, use -b option to
 extract)
---- IFD2 ----
Subfile Type                    : Reduced-resolution image
Image Width                     : 3958
Image Height                    : 1630
Bits Per Sample                 : 8 8 8
Compression                     : JPEG
Photometric Interpretation      : RGB
Samples Per Pixel               : 3
Planar Configuration            : Chunky
Tile Width                      : 128
Tile Length                     : 128
Tile Offsets                    : (Binary data 2880 bytes, use -b option to
 extract)
Tile Byte Counts                : (Binary data 1710 bytes, use -b option to
 extract)
Sample Format                   : Unsigned; Unsigned; Unsigned
JPEG Tables                     : (Binary data 73 bytes, use -b option to
 extract)
---- IFD3 ----
```

```
Subfile Type                    : Reduced-resolution image
Image Width                     : 1979
Image Height                    : 815
Bits Per Sample                 : 8 8 8
Compression                     : JPEG
Photometric Interpretation      : RGB
Samples Per Pixel               : 3
Planar Configuration            : Chunky
Tile Width                      : 128
Tile Length                     : 128
Tile Offsets                    : (Binary data 783 bytes, use -b option to
 extract)
Tile Byte Counts                : (Binary data 483 bytes, use -b option to
 extract)
Sample Format                   : Unsigned; Unsigned; Unsigned
JPEG Tables                     : (Binary data 73 bytes, use -b option to
 extract)
---- IFD4 ----
Subfile Type                    : Reduced-resolution image
Image Width                     : 990
Image Height                    : 408
Bits Per Sample                 : 8 8 8
Compression                     : JPEG
Photometric Interpretation      : RGB
Samples Per Pixel               : 3
Planar Configuration            : Chunky
Tile Width                      : 128
Tile Length                     : 128
Tile Offsets                    : (Binary data 222 bytes, use -b option to
 extract)
Tile Byte Counts                : (Binary data 139 bytes, use -b option to
 extract)
Sample Format                   : Unsigned; Unsigned; Unsigned
JPEG Tables                     : (Binary data 73 bytes, use -b option to
 extract)
---- IFD5 ----
Subfile Type                    : Reduced-resolution image
Image Width                     : 495
Image Height                    : 204
Bits Per Sample                 : 8 8 8
Compression                     : JPEG
Photometric Interpretation      : RGB
Samples Per Pixel               : 3
Planar Configuration            : Chunky
Tile Width                      : 128
Tile Length                     : 128
Tile Offsets                    : (Binary data 47 bytes, use -b option to
 extract)
Tile Byte Counts                : (Binary data 38 bytes, use -b option to
 extract)
Sample Format                   : Unsigned; Unsigned; Unsigned
JPEG Tables                     : (Binary data 73 bytes, use -b option to
 extract)
---- IFD6 ----
Subfile Type                    : Reduced-resolution image
Image Width                     : 248
Image Height                    : 102
Bits Per Sample                 : 8 8 8
Compression                     : JPEG
Photometric Interpretation      : RGB
Samples Per Pixel               : 3
Planar Configuration            : Chunky
Tile Width                      : 128
```

```
Tile Length                     : 128
Tile Offsets                    : 72539 76095
Tile Byte Counts                : 3556 3300
Sample Format                   : Unsigned; Unsigned; Unsigned
JPEG Tables                     : (Binary data 73 bytes, use -b option to
 extract)
---- IFD7 ----
Subfile Type                    : Reduced-resolution image
Image Width                     : 124
Image Height                    : 51
Bits Per Sample                 : 8 8 8
Compression                     : JPEG
Photometric Interpretation      : RGB
Samples Per Pixel               : 3
Planar Configuration            : Chunky
Tile Width                      : 128
Tile Length                     : 128
Tile Offsets                    : 69071
Tile Byte Counts                : 3468
Sample Format                   : Unsigned; Unsigned; Unsigned
JPEG Tables                     : (Binary data 73 bytes, use -b option to
 extract)
---- IFD8 ----
Subfile Type                    : Reduced-resolution image
Image Width                     : 62
Image Height                    : 26
Bits Per Sample                 : 8 8 8
Compression                     : JPEG
Photometric Interpretation      : RGB
Samples Per Pixel               : 3
Planar Configuration            : Chunky
Tile Width                      : 128
Tile Length                     : 128
Tile Offsets                    : 67168
Tile Byte Counts                : 1903
Sample Format                   : Unsigned; Unsigned; Unsigned
JPEG Tables                     : (Binary data 73 bytes, use -b option to
 extract)
---- IFD9 ----
Subfile Type                    : Reduced-resolution image
Image Width                     : 31
Image Height                    : 13
Bits Per Sample                 : 8 8 8
Compression                     : JPEG
Photometric Interpretation      : RGB
Samples Per Pixel               : 3
Planar Configuration            : Chunky
Tile Width                      : 128
Tile Length                     : 128
Tile Offsets                    : 65171
Tile Byte Counts                : 1997
Sample Format                   : Unsigned; Unsigned; Unsigned
JPEG Tables                     : (Binary data 73 bytes, use -b option to
 extract)
---- Composite ----
Image Size                      : 15829x6520
Megapixels                      : 103.2
```

Some extra sentences helps us to extract the GeoTiff entries

```
exiftool c:\COG\pnt_landsat5_2006_mosaico_islas_canarias_b321_80porciento_b543_
20porciento_hu28_8bits_COG_90.tif -b -GeoTiffAsciiParams
```

```
exiftool c:\COG\pnt_landsat5_2006_mosaico_islas_canarias_b321_80porciento_b543_
20porciento_hu28_8bits_COG_90.tif -b -GeoTiffDirectory
```

Resulting in:

```
WGS 84 / UTM zone 28N|WGS 84|

1 1 0 7 1024 0 1 1 1025 0 1 1 1026 34737 22 0 2049 34737 7 22 2054 0 1 9102
 3072 0 1 32628 3076 0 1 9001II
```

10.4. What is the georeference of the overviews?

In a COG file, there is only one set of GeoTIFF keys for each full-resolution image or band. That means that the georeference applies to the full-resolution subfile leaving the georeference of the reduced-resolution subfiles to interpretation. In the existing COG documentation, there is no information on how to infer the georeference of the subfiles. However, the implementation of the GDAL COG library provides guidance, as follows. The full-resolution subfile and the reduced-resolution subfiles share exactly the same geospatial extent. If the number of pixels of the full-resolution image and the overviews are carefully selected, this generates non-rounded pixel sizes in the lower resolution. If the number of columns is even and the number of rows is odd in the full-resolution image and the relations between the full-resolution and the overview is 2, this results in a non-square pixels in the lower resolution. Non square pixels are not necessary a bad thing but they are uncommon and not well-supported in some implementations.

In order to know about georeference used in the GDAL library for the reduced resolution subfiles, use the following instruction: `gdalinfo my.tif -oo OVERVIEW_LEVEL={ol}`, where `{ol}` refers to the reduced resolution subfile starting, and being 0 the largest in dimension overview (omitting the `–oo OVERVIEW_LEVEL` parameter gives us the georeference of the full resolution image). These are the results of an arbitrary example over the Canary Islands downloaded from the Spanish CNIG website: pnt_landsat5_2006_mosaico_islas_canarias_b321_80porciento_b543_20porciento_hu28_8bits_COG_90.tif

Table 2 — Resolutions in the COG canary islands example

OL	IMAGE WIDTH	IMAGE HEIGHT	PIXEL WIDTH	PIXEL HEIGHT	LOWER LEFT CORNER	UPPER RIGHT CORNER
-	15829	6520	30	-30	187334.0, 3059840.0	662204.0, 3255440.0
0	7915	3260	59.99621	-60	187334.0, 3059840.0	662204.0, 3255440.0
1	3958	1630	119.97726	-120.0	187334.0, 3059840.0	662204.0, 3255440.0
2	1979	815	239.95452	-240.0	187334.0, 3059840.0	662204.0, 3255440.0
3	990	408	479.66667	-479.41176	187334.0, 3059840.0	662204.0, 3255440.0

OL	IMAGE WIDTH	IMAGE HEIGHT	PIXEL WIDTH	PIXEL HEIGHT	LOWER LEFT CORNER	UPPER RIGHT CORNER
4	495	204	959.33333	-958.82352	187334.0, 3059840.0	662204.0, 3255440.0
5	248	102	1914.79839	-1917.64706	187334.0, 3059840.0	662204.0, 3255440.0
6	124	51	3829.59677	-3835.29412	187334.0, 3059840.0	662204.0, 3255440.0
7	62	26	7659.19355	-7523.07692	187334.0, 3059840.0	662204.0, 3255440.0
8	31	13	15318.38710	-15046.15385	187334.0, 3059840.0	662204.0, 3255440.0

The assumption that overviews have the exact same bounding box results in image artifacts. With the current convention, it is very difficult for a COG to match with a common TileMatrixSet that gives priority to maintaining square pixels. For example, a square image with a very particular extent configuration that has number of columns and rows based on a power of 2 can fit with a TileMatrixSet that use scale denominators based on powers of 2. In this particular case, a COG file could be used as a container for tiles following the 2D-TMS standard.

10.5. How to define a COG standard based on conformance classes

An initial examination of the COG specification concluded that there are two standardization targets for the requirements defining COG: COG creation libraries and COG HTTP servers. This should result in two requirement classes, one for the TIFF specification (tiling+overviews) and one for the HTTP range services. Additional discussion identified the need to separate the first class into two classes. A more granular consideration of the requirements concluded that some COG files could be used in analytical tools that primarily require the maximum resolution level so they does not need “overviews.” That is why, in the end, three conformance classes were defined.

However, there is nothing in the three conformance classes that has anything particularly related to GeoTIFF, so a fourth conformance class was defined. This class deals with the current approach to georeference based on the GeoTIFF keys that does the georeference based on a common bounding box. This approach suggests that an alternative fifth conformance class could be defined in the future to deal with georeference using TileMatrixSet indices instead allowing for overviews with slightly different bounding boxes. However, this last requirement class requires the acceptance by the community that there is agreement on how important the use case it represents is (convergence with common TileMatrixSets).

11

COG REQUIREMENT CLASSES

11 COG REQUIREMENT CLASSES

The following sections describe a proposal for a set of requirements grouped in the above described conformance classes that will be proposed to the OGC GeoTIFF SWG as a starting point for a formal COG standard in the OGC.

11.1. GeoTIFF format requirements

A COG file is a TIFF file or a BigTIFF file that uses two main organization techniques available in the TIFF version 6.0 specification: Tiling and Reduced-Resolution Subfiles (sometimes called *Overviews*). The tiled data can also be compressed for more efficient passage online. In addition, a COG file uses the GeoTIFF conventions for storing the relevant geospatial metadata.

11.1.1. Requirement Class GeoTIFF Tiles

Requirements Class	
http://www.opengis.net/spec/COG/1.0/req/req-class-geotiff-format	
Target type	TIFF Encoder
Dependency	http://www.opengis.net/spec/GeoTIFF/1.1/req/Core
Dependency	https://www.adobe.io/content/dam/udp/en/open/standards/tiff/TIFF6.pdf
Dependency	bigtiff.org

11.1.1.1. Basic format

A COG file is a TIFF or a BigTiFF file.

Requirement 1	/req/req-class-geotiff-format/use-geotiff
A	If the resulting COG file is bigger than 4 Gb, the COG file SHALL follow the Big TIFF standard that modifies the TIFF version 6.0 standard

Recommendation 1	/rec/rec-class-geotiff-format/use-geotiff
A	If the resulting COG file is smaller than 4Gb a COG file SHOULD follow the TIFF version 6.0 standard

11.1.1.2. Tiles

In the context for a TIFF file, **Tiling** is a strategy for dividing the content in the TIFF file differently than using the classical *Strips*. In the *Strips* approach the data are organized into sequences of lines (rows) while **tiling** creates a number of internal *tiles* stored in the actual image. In **Tiling**, the information related to a particular bounding box is easier to extract as it is closer in the file than in the *strips* approach. Actually, with *strips*, all data from the previous rows need to be read to get to the following rows. With **tiling**, much quicker access to a certain area is possible so that the portion of the file that needs to be read can be adjusted to better match the application needs.

Requirement 2	/req/req-class-geotiff-format/tiling
A	All Image File Directories (IFD) in a COG file SHALL be organized into **tiles** (instead of strips) as described in section 15 of the TIFF version 6.0 specification

NOTE: The concept Image File Directories (IFD) becomes important when the COG has also *overviews*. See the next requirements class for more details.

Tiles as defined in the TIFF version 6.0 specification can be mapped to the ones defined in the OGC Two Dimensional Tile Matrix Set Standard (2S-TMS). For example in 2D-TMS, the TIFF 6.0 forces all tiles to be of the same size. This is possible with the introduction of the concept of padding: if necessary extra blank rows or columns added to the right-most and bottom-most tile of the to make the tile the same shape as other tiles. However, the naming of the TIFF tags used version 6.0 and the property names used in the 2D-TMS differ. The following table provides a mapping between the two standards.

Table 3 — Mapping between the OGC the 2D-TMS standard and the TIFF version 6.0, section 15

OGC 2D-TMS	TIFF V. 6.0	DEFINITION
TileWidth	TileWidth	The tile width in pixels; The number of columns in each tile
TileHeight	TileLength	The tile height in pixels; The number of rows in each tile
MatrixWidth	TilesAcross	Number of tiles in width direction

OGC 2D-TMS	TIFF V. 6.0	DEFINITION
MatrixHeight	TilesDown	Number of files in the height direction

Please note that the TIFF 6.0 specification imposes the requirement that *TileWidth and TileLength must be a multiple of 16*. The specification argues that this restriction improves performance in some graphics environments and enhances compatibility with compression schemes such as JPEG. There is no such restriction in the OGC 2D-TMS.

11.1.1.3. Compression

Compression of the bytes is a general best practice for enabling software to quickly access data, in particular over a network. The combination of compression with the HTTP GET Range requests maximizes efficiency. HTTP GET Range is standardized in another requirements class.

Recommendation 2	/rec/rec-class-geotiff/rec-compression
A	In a COG file, tiled data should be compressed efficiently

11.1.2. Requirement Class GeoTIFF Overviews

Requirements Class	
http://www.opengis.net/spec/COG/1.0/req/req-class-geotiff-overviews	
Target type	TIFF encoder
Dependency	http://www.opengis.net/spec/COG/1.0/req/req-class-geotiff-format

11.1.2.1. Requirement overviews

Reduced-Resolution Subfiles (a.k.a *Overviews*) are down sampled versions of the same image included in the same TIFF file. This means that an overview is a *zoomed out* version from the original image. It has less detail but is also smaller. For visualization purposes or for analytical processes that do not require full resolution, a COG can provide *overviews* that match different scale denominators or cell sizes required by clients. *Reduced-Resolution Subfiles* increase the size of the file but also increase performance.

NOTE 1: The COG description uses the concept of *overviews*. Actually, there is nothing called *overviews* in the TIFF version 6.0 specification. Instead, the TIFF version 6.0 describes how a

TIFF file can be composed of one or more image(S), each one stored in an Image File Directories (IFD). Some IFDs may contain reduced-resolution subfiles. This is the reason why this document uses the expression *Reduced-Resolution Subfile* instead of *overview*.

There may be more than one IFD in a TIFF file and each IFD indicates the offset to where the next IFD sits in the file (or 0 if no other IFD is available). Each IFD defines a subfile. The SubFileType entry specifies three basic types of subfiles, two important types as follows. Type 1 means “full-resolution image data” and type 2 means “reduced-resolution image data of the full resolution one.”

Requirement 3	/req/req-class-geotiff-overviews/overviews
A	A COG file SHALL have at least one IFD with SubfileType 1 (full-resolution image data).
B	An IFD with SubfileType 1 SHALL have an offset to the next IFD that will point to an IFD with SubfileType 2 (reduced-resolution image data), or 0 if there is no reduced-resolution image.
C	This IFD with SubfileType 2 SHALL have an Image Width an Image Height inferior than the previous IFD.
D	If other reduced resolution images are needed, a IFD with SubfileType 2 SHALL have an offset to the next IFD that will point to an IFD with Subfile Type 2 (reduced-resolution image data).
E	This additional IFD with SubfileType 2 SHALL have an Image Width an Image Height inferior than the previous IFD.

NOTE 1: The presence of reduced-resolution subfiles in a COG file is optional. Only the COG files that conform to this conformance class are forced to have reduced-resolution subfiles.

NOTE 2: There can be more than one full-resolution image data in the file. In this case, each full-resolution image data will be followed by its reduced-resolution subfiles.

11.1.3. Requirement Class GeoTIFF Keys

The GeoTIFF standard defines a mechanism to add GeoTIFF keys to a TIFF file. The main purpose of these keys is to add a geospatial reference to the data stored in the TIFF file. In a COG, GeoTIFF keys are a fundamental part of the specification as most of the COG files are geospatially referenced TIFFs.

Requirements Class	
http://www.opengis.net/spec/COG/1.0/req/req-class-geotiff-keys	
Target type	GeoTIFF Encoder

Dependency	http://www.opengis.net/spec/COG/1.0/req/req-class-geotiff-format

11.1.3.1. Requirement GeoTIFF

A COG uses GeoTIFF to document the metadata about the TIFF file.

Requirement 4	/req/req-class-geotiff-format/basic-metadata-format
A	A COG file SHALL include and encode geospatial metadata in the GeoTIFF format following the GeoTIFF standard

11.1.3.2. Requirement Georeference Keys

There is a geometrical relation between the reduced-resolution subfiles and the corresponding full-resolution subfile. Only full-resolution subfiles are required to have GeoTIFF keys.

Requirement 5	/req/req-class-geotiff-keys/georeference
A	In a COG file, the IFD Subfiles Type 1 SHALL contain georeference data using ModelTiepointTag, ModelPixelScaleTag, and GeoKeyDirectoryTag tags.

Clients have two options to interpret the georeference of the reduced-resolution subfiles:

- use common tile matrix set information in the IFD, in the full-resolution subfile that describes all the related reduced-resolution subfiles; or
- use geometrical relations between the reduced-resolution subfiles and the corresponding full-resolution subfile.

11.1.3.3. Requirement Tile Matrix Set

In some circumstances, defining a COG that follows a Tile Matrix Set as defined in the 2DTMS standard is possible. In this case, the georeference of the tiles can be clearly defined by linking to an external Tile Matrix Set definition, the Tile Matrix identifiers corresponding to each IFD (that provides the PixelScale of each resolution) and the TileRow and TileCol or the first tile in each IFD (that provides the position of this tile in the space).

Requirement 6	/req/req-class-geotiff-keys/tms
A	In a COG file that follows a Tile Matrix Set, the IFD Subfiles Type 1 SHALL contain the lowest scale denominator Tile Matrix.

B	In a COG file that follows a Tile Matrix Set, the first IFD Subfile Type 2 related to IFD Subfiles Type 1 SHALL contain the immediately upper scale denominator.
C	In a COG file that follows a Tile Matrix Set, the following IFD Subfile Type 2 related to IFD Subfile Type 2 SHALL contain the immediately upper scale denominator and so on.

In the case the IFDs contain a TileMatrixSet, the following requirement specifies how to reference it. This capability allows for a rapid integration of COG files in clients that are already capable of managing Tile Matrix Sets and may facilitate services to use a set of COG files to store TileSets.

Requirement 7	/req/req-class-geotiff-keys/tms-keys
A	In a COG file that follows a TileMatrixSet, the GeoTIFF Key 5136 (TMSReferenceKey) ASCII type key SHALL contain a pointer to the text ASCII key that contains an external reference to the TileMatrixSet definition.
B	In a COG file that follows a TileMatrixSet, the GeoTIFF Key 5137 (TileMatrix LowestIdKey) ASCII type key SHALL contain a pointer to the text ASCII key that contains the identifier of the lowest scale denominator of the TileMatrix present in the COG file.
C	In a COG file that follows a TileMatrixSet, the GeoTIFF Key 5138 (TileMatrix HighestIdKey) ASCII type key SHALL a pointer to the text ASCII key that contains the identifier of the highest scale denominator of the TileMatrix present in the COG file
D	In a COG file that follows a TileMatrixSet, the GeoTIFF Key 5139 (TMSLimits2dKey) LONG Type key SHALL point to an array of LONG numbers in the list defined by the TIFF tag for GeoTIFF long values (GeoLongParams Tag) that are interpreted as a sequence of pairs of LONG numbers specifying an ordered list of lowest TileRow and TileCol indices starting by the highest scale denominator and ending in the lowest scale denominator of the Tile Matrix present in the COG file. The number of members of the array SHALL be the number of TileMatrix between the highest and the lowest ones (including both the highest and the lowest) multiplied by 2. If the key TMSLimits2dKey is missing, all values in the array are assumed to be zero (default value).
E	In a COG file that follows a TileMatrixSet, the georeference information specified using ModelTiepointTag, ModelPixelScaleTag in the full-resolution subfile SHALL be consistent with the georeference of the tile corresponding with the lowest scale denominator of the TileMatrix and first pair of TileCol and TileRow specified in the TMSLimits2dKey GeoKey.

NOTE 1: In a COG file that follows a TileMatrixSet, the georeference information specified using ModelTiepointTag, ModelPixelScaleTag in the full-resolution subfile is redundant to the Tile Matrix Set and ONLY affects the full-resolution subfile. The georeference of the IFD Subfile Type 2 is ONLY determined by the information about the Tile Matrix Set. This

approach relaxes the need for having all Subfiles starting in the same point of origin that applies when no Tile Matrix Set is indicated (see next requirement).

NOTE 2: The start of the georeference (4th and 5th numbers in ModelTiepointTag) can be the PointOfOrigin of the Tile Matrix only if the first tile of the Tile Matrix is present in the COG file (see considerations about TileMatrixSetLimits). In general, the two values are expected to be different.

NOTE 3: This requirement was not in the original COG specification before it was brought to the OGC.

11.1.3.4. Requirement No Tile Matrix Set

In the case of a COG file that does not report a link to any Tile Matrix Set (the keys TMSReferenceKey, TileMatrixLowestIdKey, TileMatrixHighestIdKey, and TMSLimits2dKey are missing), all linked reduced-resolution subfiles (a.k.a. IFDs SubfileType 2) that are linked to a SubfileType 1 IFD share the set of georeference keys from the full resolution IFD. In practice, this means that they have a common CRS and extent information and differ only by their resolution. This means that in absence of any information about a common tile matrix set, there is the assumption that all related subfiles share the same point of origin for the top left corner or the image.

Requirement 8	/req/req-class-geotiff-keys/no-tms-keys
A	In a COG file that does not follow a TileMatrixSet (the GeoTIFF Key 5136 (TMSReferenceKey) is absent), all reduced-resolution subfiles have the same point of origin for the top left corner of the image than the full resolution subfile.

In this case (not report a link to any Tile Matrix Set), the pixel size in the first dimension of a reduced-resolution subfile can be calculated by multiplying the full-resolution subfile pixel size by the ImageWidth ratio between the full-resolution and the reduced-resolution. The pixel size in the second dimension of a reduced-resolution subfile can be calculated by multiplying the full-resolution subfile pixel size by the ImageHeight ratio between the full-resolution and the reduced-resolution.

For the IFD reduced-resolution subfiles linked to IFD0, IFDn_PixelScaleX and IFDn_PixelScaleY can be calculated like this:

```
IFDn_PixelScaleX=IFD0_PixelScaleX*IFD0_ImageWidth/IFDn_ImageWidth
IFDn_PixelScaleY=IFD0_PixelScaleY*IFD0_ImageHeight/IFDn_ImageHeight
```

For a georeferenced file with a projected CRS that is fully defined in the EPSG database, there is no need to define the base Geographic CRS, geodetic datum, etc. in the TIFF file. In these cases, the following keys in the IFD0 are sufficient;

```
ImageWidth = 15829
ImageHeight = 6520
ModelTiepointTag = (0 0 0 187334 3255440 0)
```

```
ModelPixelScaleTag = (30 30 0)
GeoKeyDirectoryTag:
-    GTModelTypeGeoKey = 1 (ModelTypeProjected 2D)
-    GTRasterTypeGeoKey = 1 (RasterPixelIsArea)
-    ProjectedCRSGeoKey = 32628 (Projected 2D CRS WGS 84 / UTM zone 28N)
```

This example corresponds to a Landsat5 mosaic over the Canary Islands with a resolution of 30 meters per pixel. There is a grid intersection line in the image at pixel location (0, 0)

The 2 first numbers in ModelTiepointTag correspond to the projected coordinate reference system easting/northing of (X:187334, Y:3255440) (the 4th and 5th numbers in ModelTiepointTag) in the EPSG:32628 (WGS 84 / UTM zone 28N).

The georeference of the reduced-resolution subfiles (a.k.a. IFDs SubfileType 2) that are linked to this IFD0 are supposed to be the same except that `ModelPixelScaleTag` represents meters per pixel (pixel size). In this example, IFD0 has `ImageWidth=20111` and `ImageHeigth=16882` while IFD1 has `ImageWidth=10056` and `ImageHeigth=8441`, so `IFD1_PixelScaleX` and `IFD1_PixelScaleY` can be calculated as: 30*20111/10056=59.997 and 30*16882/8441=60. As we can see in the example, non-TileMatrixSet based tiles can easily result in non-square-pixels; a situation that may complicate presenting or combining the data with other data that has square pixels.

11.2. HTTP range support requirements

HTTP Version 1.1 introduced a range header in the GET requests that supports requesting only a fragment of a resource. If the server advertises with an “Accept-Ranges: bytes” header in its response, the server is telling the client that bytes of data can be requested in parts in separated requests. The client can request just the bytes that it needs from the server at any time. In a web environment, this is very useful for serving things such as video as clients do not need to download the entire file to begin playing it. In the COG use case, it is useful to get only the tiles needed at a particular time. A client showing a COG file on the screen can request the resolutions needed and only the tiles needed to cover the screen. Once the user pans or zooms, other GET range requests will query for other resolutions and tiles.

11.2.1. Requirement Class HTTP range

Requirements Class	
http://www.opengis.net/spec/COG/1.0/req/req-class-http-range	
Target type	Web server
Dependency	https://datatracker.ietf.org/doc/html/rfc7233

Dependency	http://www.opengis.net/spec/GeoTIFF/1.1/req/Core
Dependency	https://www.adobe.io/content/dam/udp/en/open/standards/tiff/TIFF6.pdf
Dependency	bigtiff.org

11.2.1.1. Requirement range

Requirement 9	/req/req-class-http-range/use-geotiff
A	A HTTP or HTTPS server serving COG files SHALL support HTTP ranges
B	A HTTP or HTTPS server serving COG files shall use byte ranges.

The use of range in web browsers using HTTPS is restricted by the Cross-Origin Resource Sharing (CORS) rules (read more about CORS here: https://support.streamroot.io/hc/en-us/articles/115003168773-Range-request-basics). This is why the servers must explicitly declare that they allow the "range" header from a client that is not in the same domain. Since this is a very common use case in geospatial data sharing, the decision was to make this a requirement.

Requirement 10	/req/req-class-http-range/use-geotiff
A	A HTTPS server serving COG files shall advertise that allows access control for ranges by adding this line in the headers Access-Control-Allow-Headers: range

NOTE: Support for range in HTTPS is necessary but not sufficient. A server that wants to authorize a client application from another domain (under the CORS rules) should also use the Access-Control-Allow-Origin: header to indicate what domains are authorized to read the COG file (or use * to indicate that all domains are authorized). In general, services are willing to allow other clients to read the data from other domains since many want to comply with the geospatial data sharing principles. However, this is a voluntary decision of the server implementers.

12

ISSUES WITH COG, TIFF AND GEOTIFF

12 ISSUES WITH COG, TIFF AND GEOTIFF

12.1. How manage TIFF tags?.

TIFF tags are defined in a way that is extensible. TIFF v.6 defines a set of baseline tags.

> An organization might wish to store information meaningful to only that organization in a TIFF file. Tags numbered 32768 or higher, sometimes called private tags, are reserved for that purpose. Upon request, the TIFF administrator (send email to devsup-person@adobe.com) will allocate and register one or more private tags for an organization. This is to avoid possible conflicts with other organizations.
>
> — TIFF v. 6 standard

However, there is no evidence that Adobe had published any list of private tags. There is a list of private tags maintained by https://www.awaresystems.be/imaging/tiff/tifftags/private.html that include the GeoTIFF tags as well as some of the ones introduced by ISO12234-2 for the photographic industry. This situation complicates the extensibility and affects the capacity to define new tags for the new versions of GeoTIFF.

12.2. Need for a LONG type in the GeoTIFF standard

GeoTIFF Keys are available for SHORT and for FLOAT numbers but there is no key to define long numbers. To be able to define a COG subfile as related to some common TileMatrixSets the ranges of long indices need to be defined. The indices cannot be a SHORT because tilecol and tilerow can easily be larger than 65K if the scale denominator is small. Creating a GeoLongParamsTag and associating a TIFF tag number 34738 to it is proposed. There is no evidence that this number has been used for other purposes. If this extension to GeoTIFF is accepted and included, it is possible to define a GeoTIFF Key to indicate the TileMatrixSetLimits covered by a COG file.

12.3. Media type for COG

Whether a specific media type for COG is needed or if image/tiff is sufficient given that fact that an classic TIFF and a BigTIFF can be identified by the version in the first bytes of the header is not clear (this was discussed in the past and recorded here: https://github.com/opengeospatial/geotiff/issues/34).

Many possibilities have been proposed but no agreement has been reached so far.

- image/tiff; application=geotiff, profile=COG
- image/tiff; application=geotiff; profile=cloud-optimized (proposed in https://github.com/radiantearth/stac-spec/blob/master/best-practices.md#common-media-types-in-stac)
- image/tiff; application=geotiff; cloud-optimized=true
- image/tiff; application=geotiff; cog=true
- image/cogeo+tiff
- image/cog+geotiff

The issue continues the discussions with elaborated media type proposals to make explicit or negotiate the characteristics of a COG file:

- image/tiff; application=geotiff; tiled=true; tile-size=256; overviews=true; compress=DEFLATE;

The specification for the tiff media type in the IANA https://www.iana.org/assignments/media-types/image/tiff only specifies a single optional parameter "application." Adding a new official parameter such as "profile" will require a modification to the IANA registration.

Considering that a COG is a particular case of GeoTIFF, specifying that the application is COG could be enough. Writing the full name instead of the acronym makes visible the dependency with GeoTIFF.

- image/tiff; application=cloud-optimized-geotiff

However, this approach will prevent applications looking for image/tiff; application=geotiff to detect the media type as compatible to them.

Recently, a new W3C standard proposal named "Negotiation by profile" (https://www.w3.org/TR/dx-prof-conneg/) adds completely different possibility. Instead of extending media types, they are used in combination with the profile concept. The "Content-Profile" new entry in the headers of requests can be added to indicate the name of a profile in addition to the current header Content-Type. This allows us to respect the originally standardized media type indicated in the GeoTIFF standard. Please note that the value of the profile name should be a URI. Having values as full URIs avoids the need for a centralized registry of parameter values. Applying this suggestion, a header of a HTTP response containing a COG file could look like this:

```
Content-Type: image/tiff; application=geotiff;
Content-Profile: http://www.ogc.org/profile/image/tiff/cog
```

12.4. BigTIFF

All versions of TIFF define a format that uses 32-bit offsets and, as such, the file size is limited to 4 GBytes. This limit has been sufficient for many years. Today, however, there is a need for a good multi-purpose open image file format that can handle huge images, or very large arrays of bands, beyond the 4 GByte boundary.

BigTIFF is a variant that closely resembles TIFF. However BigTIFF redefines a couple of internal data structures to support 64-bit offsets and this way extends support for very large files.

The BigTIFF specification is available on the Aware Systems website: https://www.awaresystems.be/imaging/tiff/bigtiff.html. Aware Systems is managed by Joris Van Damme. In the register of data formats maintained by the Library of the Congress (USA), the BigTIFF format is included and in the "licensing and patents" section it says that: "The BigTIFF enhancements in Aperio's libtiff were made by Ole Eichhorn (formerly of Aperio Technologies, acquired by Leica Biosystems in 2012)." In the words of the bigtiff.org web page: "are donated to the public domain, in gratitude to Sam Leffler, Silicon Graphics, Joris Van Damme, AWare Systems, Frank Warmerdam, Andrey Kisley, Mike Welles, and all who have worked on libtiff over the years to provide such a great tool. The aperio libtiff changes were published on an 'as is' basis and neither Ole Eichhorn nor Aperio Technologies make any warranty as to their fitness for any intended use." (from: https://www.loc.gov/preservation/digital/formats/fdd/fdd000328.shtml).

In an email conversation (June, 16th 2021, 15:05), Joris Van Damme declared that "I don't mind people building on from my BigTIFF page, and I think you [OGC] should. You [OGC] can quote me on this, this is my official stance." "As long as you don't flat out copy and paste, and I much like to see things move forward."

The OGC staff and participants in the Testbed 17 discussed the possibility to look for consensus in the community and adopt BigTIFF as a Standard and no legal barriers were found. This way the OGC will be able to maintain the specification and reference it from other work such as future versions of the GeoTIFF Standard. The GeoTIFF SWG has accepted the work item and started to explore this possibility.

The OGC GeoTIFF SWG should add *credits* or a similar section in section to the draft standard to credit Aware Systems for the material that served as starting point, and more generally to all past contributors of the BigTIFF design (possibly pointing to the initial discussion thread at https://www.asmail.be/msg0055453930.html).

12.4.1. GeoTIFF does not explicitly adopt BigTIFF

Version 1.1 of the GeoTIFF Standard requires the use of TIFF v.6. In practice this excludes the use of BigTIFF and prevents the validation of GeoTIFFs that are larger than 4 GBytes as a valid GeoTIFF. This limitation is ignored by the GDAL library and can be ignored by other implementations too. Future versions for GeoTIFF should edit the TIFF v.6 requirements and also allow for BigTIFF.

13

BIGTIFF REQUIREMENT CLASSES

13 BIGTIFF REQUIREMENT CLASSES

The following subsections present an overview of the BigTIFF format and a proposal for a set of requirements that could constitute the starting point to prepare a BigTIFF standard in the GeoTIFF SWG.

13.1. Overview

13.1.1. What is BigTIFF

The TIFF file format uses 32-bit offsets and, as such, is limited to a 4 GByte file size. This has been quite sufficient for many years. Today however, there is a need for a good multi-purpose open image file format that can handle huge images, or very large collections of images, that exceed the 4 GByte boundary.

BigTIFF is backward compatible with the original TIFF and extends TIFF. The benefits of BigTIFF closely resembling TIFF are huge. For instance, existing TIFF libraries can quite easily extend their support for TIFF to also include this new variant. This results in a very short list of requirements. All the much-appreciated properties of the TIFF format that have been around and have been extended for more than a decade are inherited. All properly known tags are being reused, all supported bit depths and datatypes remain valid. The arbitrary number of 'extra channels,' the tiling and striping schemes, the multitude of compression schemes, and the private tag scheme, that made TIFF very useful in pre-press as well as for storing scientific data, and many other applications, all remain intact. Yet, by changing the offset bit depth, BigTIFF files are no longer restrained by the 4-GByte limitation from which classic TIFF suffers. Since the BigTIFF format is based on the TIFF format, the reader is encouraged to familiarize with the current version of the TIFF specification.

13.1.2. Origin and compatibility

This draft specification describes a new variant of TIFF, called BigTIFF, that closely resembles TIFF, but uses 64-bit offsets instead. The results of the BigTIFF discussion resulted in an original specification exposed in the Aware Systems website: https://www.awaresystems.be/imaging/tiff/bigtiff.html. Aware Systems is managed by Joris Van Damme. The proposed draft specification documented in this ER began with Joris Van Damme's initial work. Another source of information on the BigTIFF format is the BigTIFF design page at the LibTiff site http://bigtiff.org/.

There are at least two independent implementations of the original BigTIFF specification. (AWare Systems AsTiff, and LibTiff since version 4.0). The draft specification does not contradict

the original bigTIFF specification. As such, implementations BigTIFF remain compatible with this draft specification.

13.1.3. Changes to the original version

The editors of this document prefer to use a clearer naming for the data types.

Table 4 — Changes in data types.

ORIGINAL	ADOPTED	DESCRIPTION
Word	uint16	unsigned short integer (2 bytes)
Unsigned long	uint32	unsigned long integer (4 bytes)
Signed 8Byte	int64	signed long long integer (8 bytes)
Unsigned 8Byte	uint64	unsigned long long integer (8 bytes)

13.2. BigTIFF requirements

The BigTIFF specification provides a solution for a TIFF file that requires a file size larger than 4 Gigabytes. In principle the BigTIFF specification will also support smaller files. However, files that do not require file size bigger than 4 Gigabytes or not using the new data types, should use TIFF version 6 specification instead. This is in order to ensure backwards compatibility with old implementations not supporting BigTIFF.

A BigTIFF file (and a TIFF) is defined to be a sequence of 8-bit bytes, where the bytes are numbered from 0 to N. In a BigTIFF file, the largest possible TIFF file is 2^{64} bytes in length.

13.2.1. Requirement Class Core

Requirements Class Core	
http://www.opengis.net/spec/BigTIFF/1.0/req/req-class-core	
Target type	BigTIFF Encoder
Dependency	https://www.adobe.io/content/dam/udp/en/open/standards/tiff/TIFF6.pdf

13.2.1.1. Basic format

Permission 1	**/per/core/big-offsets**
A	A BigTIFF implementation MAY ignore the introduction and the first 2 subsections in Section 2 of the TIFF version 6 specification describing the *Image File Header* and the *Image File directory* and use a *Big Image File Header* and a *Big Image File directory* instead. When some part of the *Image File directory* subsection is still relevant for this standards it is explicitly referenced

Requirement 11	/req/req-class-core/big-offsets
A	A BigTIFF SHALL begin with a 16 byte *big image file header* (that replaces the TIFF v6 8 byte *image file header*)
B	A *big image file header* SHALL point to the first *big image file directory* (BIFD). A *big image file directory* contains information about the image, as well as pointers to the actual image data in a similar way as an *image file directory* does in TIFF version 6

Requirement 12	/req/req-class-core/bifh
A	A *big image file header* SHALL follow the structure presented in Table 5

The proposed BigTIFF structures

Table 5 — BigTIFF file header (normative)

OFFSET	DATATYPE	VALUE
0	**UINT16**	Byte order indication (use the values stated in the TIFF version 6 Image File Header Bytes 0-1)
2	**UINT16**	Version number (always 43)
4	**UINT16**	Size of offset in bytes Always 8 in BigTIFF (it could be 16 in a future version for superbig TIFFs)
6	**UINT16**	Reserved. Populate with 0
8	**UINT64**	Offset (in bytes) to first IFD

(The [annex-classic-tiff-basic-struct-informative] contains the TIFF v.6 classic structures for comparison purposes).

While the classic TIFF *image file header* points to the first *Image File Directory* (IFD), the BigTIFF *big image file header* points to the first *Big Image File Directory* (BIFD). In that sense IFD and BIFD are equivalent. Both IFD and BIFD can be located anywhere in the file. Every 'page' in a multi-page TIFF, classic TIFF, or BigTIFF, is represented by exactly one IFD and one BIFD respectively.

Requirement 13	/req/req-class-core/bifd
A	There SHALL be at least ONE *big image file directory* in a TIFF file and each BIFD must have at least one entry.
B	A *big image file directory* SHALL follow the structure of *Big Directory Entries* presented in Table 6
C	The *big directory entries* in a BIFD SHALL be sorted by code (in the same way than classic TIFF does)
D	Each *big directory entry* SHALL follow the structure presented in Table 7
E	The tag *value* SHALL be written inside the tag structure, in the *big directory entry*, if its size is smaller than or equal to 8 bytes. Otherwise, it's written elsewhere, and pointed to by the offset written as the value
F	In case of multibyte values (e.g., a SHORT, LONG, etc), the byte order (II or MM) in the TIFF header SHALL govern the order of the bytes
G	The reader SHALL check the type to verify that it contains an expected value. TIFF allows for more than one valid type for some fields. For example, Image Width and ImageLength are usually specified as having type SHORT, but images with more than 64K rows or columns should use the LONG field type. TIFF readers SHALL accept BYTE, SHORT, LONG, or LONG8 values for any unsigned integer field. This allows a single procedure to retrieve any integer value, and makes reading more robust, and saves disk space in some situations

A *Big Image File Directory* (BIFD) consists of an 8-byte count of the number of directory entries (i.e., the number of fields), followed by a sequence of 20-byte field entries, followed by an 8-byte offset of the next BIFD (or 0 if no other BIFD is required).

Every *Big Directory Entry* takes up exactly 20 bytes in BigTIFF, and the complete BIFD looks like this:

Table 6 — BigTIFF Image File Directory (BIFD)

OFFSET	DATATYPE	VALUE
0	uint64	Number of big directory entries in BIFD (B)
8+0*20	Big directory entry structure (see Table 7)	First tag data

OFFSET	DATATYPE	VALUE
8+1*20	Big directory entry structure (see Table 7)	Second tag data
...		
8+(B-1)*20	Big directory entry structure (see Table 7)	Last tag data
8+B*20	uint64	Offset to next BIFD, if there is a next BIFD or 0 otherwise

A *TIFF field* is a logical entity consisting of TIFF tag and its value. This logical concept is implements as a *Directory Entry* in the TIFF v.6 and it is implemented in here as a *Big Directory Entry*, plus the actual value if it doesn't fit into the value/offset part, the last 8 bytes of the BIFD Entry.

Table 7 — BigTIFF Tag structure

OFFSET	DATATYPE	VALUE
0	uint16	Tag identifying code [1]
2	uint16	Datatype of tag data. See Table 8
4	uint64	Number of values (it is not the total number of bytes but the number of values)
12	x * Tag data or uint64 offset	Tag data or offset to tag data

[1] See a list of codes in the TIFF version 6 specification. The GeoTIFF standard adds some extra TIFF tag identifiers.

Each TIFF field has an associated Count. This means that all fields are actually one-dimensional arrays, even though most fields contain only a single value.

The same rule for value 'inlining' the tag data applies to both classic TIFF and BigTIFF, only the threshold size differs. Values above the threshold are pointed by an offset and written elsewhere.

The following table summarizes the Tag datatypes. The "Types" description in the TIFF v.6 specification applies also for BigTIFF. In BigTIFF, some new types have been defined (see the last part of Table 8).

Table 8 — BigTIFF Datatype Tag values

VALUE	NAME	DESCRIPTION	SOURCE
1	BYTE (uint8)	8-bit unsigned integer	TIFF specification

VALUE	NAME	DESCRIPTION	SOURCE
2	ASCII	8-bit byte that contains a 7-bit ASCII code; the last byte SHALL be a NUL character (binary zero)	TIFF specification
3	SHORT (uint16)	16-bit (2-byte) unsigned integer	TIFF specification
4	LONG (uint32)	32-bit (4-byte) unsigned integer	TIFF specification
5	RATIONAL	Two LONGs: the first represents the numerator of a fraction; the second, the denominator	TIFF specification
6	SBYTE	An 8-bit signed (twos-complement) integer	TIFF version 6
7	UNDEFINED	An 8-bit byte that may contain anything, depending on the definition of the field [1]	TIFF version 6
8	SSHORT (int16)	A 16-bit (2-byte) signed (twos-complement) integer	TIFF version 6
9	SLONG (int32)	A 32-bit (4-byte) signed (twos-complement) integer.	TIFF version 6
10	SRATIONAL	Two SLONGs: the first represents the numerator of a fraction, the second the denominator	TIFF version 6
11	FLOAT	Single precision (4-byte) IEEE format	TIFF version 6
12	DOUBLE	Double precision (8-byte) IEEE format	TIFF version 6
13	IFD	A uint32 IFD offset	TIFF PageMaker version 6
16	LONG8 (uint64)	64-bit (8-byte) unsigned integer	BigTIFF
17	SLONG8 (int64)	64-bit (8-byte) signed integer	BigTIFF
18	IFD8	A uint64 IFD offset	BigTIFF

[a] It can be used to store a complicated data structure in a single private field, The Count will be the number of bytes required to hold the data structure.

Permission 2	/per/core/long8
A	StripOffsets, StripByteCounts, TileOffsets, and TileByteCounts tags are allowed to have the datatype LONG8 in BigTIFF when necessary. LONG, and SHORT are still valid
B	Tags that point to other IFDs, like e.g. the SubIFDs tag (see TIFF PageMaker version 6), are allowed to have the datatype IFD8. The datatypes IFD (and the not recommendable LONG) are still valid

A

ANNEX A (INFORMATIVE) REVISION HISTORY

A ANNEX A (INFORMATIVE) REVISION HISTORY

DATE	RELEASE	AUTHOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
April 18, 2021	.1	Joan Maso	all	initial version
April 16, 2021	.2	Joan Maso	all	start working on the COG standard draft
June 7, 2021	.3	Joan Maso	all	main sections of the ER have content
June 27, 2021	.4	Joan Maso	9	start working on BigTIFF standard draft
July 7, 2021	.5	Joan Maso	6	proposal for TMS in COG clarified
September 5, 2021	.6	Joan Maso	7,9	Standard draft requirements are now section of the ER
October 6, 2021	.7	Joan Maso	all	ER text finalized pending final review
October 17, 2021	.8	Imma Serra	all	Grammar fixes

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Cloud Optimized GeoTIFF. An imagery format for cloud-native geospatial processing. In Internet: https://www.cogeo.org/, last accessed 2021/11/02

[2] GDAL COG Cloud Optimized GeoTIFF generator. In Internet: https://gdal.org/drivers/raster/cog.html last accessed 2021/11/02

[3] R. Fielding, Y. Lafon and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Range Requests (IETF RFC 7233). In Internet: https://datatracker.ietf.org/doc/html/rfc7233

[4] Joris Van Damme. The BigTIFF File Format. In Internet: https://www.awaresystems.be/imaging/tiff/bigtiff.html

ISO 12234-2 Photography — Electronic still picture imaging — Removable memory Part 2: Image data format — TIFF/EP

NOTE: ISO 12234-2 is based on Adobe Developers Association, TIFF Revision 6.0 Final, June 3, 1992, Adobe Systems Incorporated, https://www.adobe.io/open/standards/TIFF.html. However seem to be more focused on photography

[5] TIFF Revision 6.0 Final — June 3, 1992. Adobe Developers Association. In Internet: https://www.adobe.io/open/standards/TIFF.html

[6] Durbin, C., Quinn, P. and Shum, D., 2020. Task 51-Cloud-Optimized Format Study. In Internet: https://ntrs.nasa.gov/api/citations/20200001178/downloads/20200001178.pdf?attachment=true

[7] Richardson, M., Kearns, E. and O'Neil, J., 2020. Data dissemination best practices and challenges identified through NOAA's Big Data Project (No. EGU2020-12386). Copernicus Meetings.

[8] Hanson, M. and Leith, A., 2020, December. Sentinel-2 Cloud-Optimized GeoTIFF Public Dataset. In AGU Fall Meeting Abstracts (Vol. 2020, pp. IN042-0002).

[9] Roberts, N., Pieschke, R. and Lemig, K., 2019, December. Landsat in the cloud: Improving access and usability of the USGS Landsat record. In AGU Fall Meeting Abstracts (Vol. 2019, pp. IN22A-01).