

Package: rustycogs (via r-universe)

September 26, 2026

Title TIFF Virtualization and Decoding via Rust

Version 0.2.0.9004

Description Extract byte-range chunk references from cloud-hosted TIFF and COG files without reading pixel data. Uses the Rust 'async-tiff' and 'object_store' crates for async I/O across S3, GCS, Azure, HTTP, and local storage. Returns a data frame of tile offsets and metadata suitable for constructing Kerchunk or Zarr virtual stores.

License MIT + file LICENSE

URL <https://github.com/hypertidy/rustycogs>

BugReports <https://github.com/hypertidy/rustycogs/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

SystemRequirements Cargo (Rust's package manager), rustc

Depends R (>= 4.1.0)

Imports jsonlite

Suggests arrow, testthat (>= 3.0.0)

Config/rextendr/version 0.5.0

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libclang-dev

Repository <https://hypertidy.r-universe.dev>

Date/Publication 2026-08-20 02:10:36 UTC

RemoteUrl <https://github.com/hypertidy/rustycogs>

RemoteRef HEAD

RemoteSha 56f9537eb9829ec896ed04ad249484beee0e8b9f

Contents

refs_tiles	2
tiff_ifd_info	3
tiff_refs	4
tiff_tile	5
tiff_tiles	6
tile_to_array	7
tiles_to_matrix	8
Index	9

refs_tiles	<i>references as tiles</i>
------------	----------------------------

Description

Add extent information xmin,xmax,ymin,ymax to refs table

Usage

refs_tiles(x)

Arguments

x file path or url

Value

dataframe with xmin,xmax,ymin,ymax add to the tiffs_refs output

See Also

tiles_to_matrix

Examples

```
vfile <- system.file("extdata/volcano-cog.tif", package = "rustycogs", mustWork = TRUE)
refs <- refs_tiles(vfile)
dim(refs)
str(refs)
```

tiff_ifd_info	<i>Summarise IFD-level metadata from TIFF/COG files.</i>
---------------	--

Description

Returns one row per IFD per file — no tile explosion. Useful for understanding file structure before deciding which tiles to fetch.

Returns one row per IFD per file with no tile explosion. Use this to understand file structure — how many IFDs (overviews), tile grid dimensions, dtype, CRS, geotransform — before deciding which tiles to fetch with [tiff_refs\(\)](#) or [tiff_read_tiles\(\)](#).

Usage

```
tiff_ifd_info(paths, region = NULL, anon = FALSE, concurrency = 16L)
```

```
tiff_ifd_info(paths, region = NULL, anon = FALSE, concurrency = 16L)
```

Arguments

paths	Character vector of file paths or URLs.
region	Optional AWS region string (e.g. "ap-southeast-2").
anon	Logical. Use anonymous/unsigned requests. Default FALSE.
concurrency	Integer. Maximum concurrent file scans. Default 16L.

Value

A data.frame with one row per IFD containing image dimensions, tile grid size, dtype, compression, photometric interpretation, predictor, planar configuration, CRS, nodata, and geotransform.

A data frame with one row per IFD and columns: path, ifd, is_tiled, image_w, image_h, tile_w, tile_h, n_tiles_x, n_tiles_y, dtype, compression, bits_per_sample, samples_per_pixel, photometric, predictor, planar_configuration, crs_epsg, gdal_nodata, scale_x, scale_y, origin_x, origin_y.

See Also

[tiff_refs\(\)](#) for the full per-tile data frame

Examples

```
## Not run:
tiff_ifd_info("scene.tif")
tiff_ifd_info("s3://my-bucket/data.tif", anon = TRUE)

## End(Not run)
```

tiff_refs

*Extract tile byte-range references from TIFF/COG files.***Description**

Scans one or more TIFF or COG files and returns a data frame with one row per tile per IFD, containing the byte-range metadata needed to construct Kerchunk or Zarr virtual stores.

Usage

```
tiff_refs(paths, region = NULL, anon = FALSE, concurrency = 16L)
```

```
tiff_refs(paths, region = NULL, anon = FALSE, concurrency = 16L)
```

Arguments

paths	Character vector of file paths or URLs. Supported schemes: s3://, gs://, az://, http://, https://, or plain local paths.
region	Optional AWS region string (e.g. "ap-southeast-2"). Ignored for non-S3 URLs.
anon	Logical. Use anonymous/unsigned requests. Default FALSE.
concurrency	Integer. Maximum number of files scanned concurrently. Default 16L.

Value

A data.frame with columns path, ifd, tile_col, tile_row, offset, length, image_w, image_h, tile_w, tile_h, valid_w, valid_h, dtype, compression, bits_per_sample, samples_per_pixel, crs_epsg, gdal_nodata, planar_configuration, scale_x, scale_y, origin_x, origin_y.

A data frame with columns: path, ifd, tile_col, tile_row, offset, length, image_w, image_h, tile_w, tile_h, dtype, compression, bits_per_sample, samples_per_pixel, crs_epsg, gdal_nodata, planar_configuration, scale_x, scale_y, origin_x, origin_y.

See Also

[tiff_ifd_info\(\)](#) for a compact one-row-per-IFD summary, [tiff_read_tiles\(\)](#) to fetch pixel data

Examples

```
## Not run:
refs <- tiff_refs("s3://my-bucket/data.tif", anon = TRUE)
refs <- tiff_refs(c("file1.tif", "file2.tif"), concurrency = 4L)

## End(Not run)
```

tiff_tile	<i>Fetch and decode a single tile from a TIFF/COG file.</i>
-----------	---

Description

Fetch and decode a single tile from a TIFF/COG file.

Fetch and decode a single tile from a TIFF/COG file

Usage

```
tiff_tile(path, ifd_index = 0L, col, row, region = NULL, anon = FALSE)
```

```
tiff_tile(path, ifd_index = 0L, col, row, region = NULL, anon = FALSE)
```

Arguments

path	File path or URL to the TIFF.
ifd_index	IFD index (0-based). 0 is the full-resolution image.
col	Tile column index (0-based).
row	Tile row index (0-based).
region	Optional AWS region string.
anon	Logical. Use anonymous requests. Default FALSE.

Value

A named list with data, dim, dtype.

A named list with:

- data: numeric vector of decoded pixel values (row-major)
- dim: integer vector c(height, width, bands)
- dtype: character dtype string (e.g. "<f4", "<u2")

See Also

[tile_to_array\(\)](#) to reshape the result, [tiff_tiles\(\)](#) for batches

Examples

```
## Not run:
tile <- tiff_tile("scene.tif", col = 0L, row = 0L)
m <- tile_to_array(tile)

## End(Not run)
```

tiff_tiles

*Fetch and decode multiple tiles from a TIFF/COG file.***Description**

Uses `ifd.fetch_tiles()` for a single vectorized range request.

Fetches a batch of tiles from a single file in one vectorized request. More efficient than calling `tiff_tile()` in a loop. For multi-file batches use `tiff_read_tiles()`.

Usage

```
tiff_tiles(path, ifd_index = 0L, cols, rows, region = NULL, anon = FALSE)
```

```
tiff_tiles(path, ifd_index = 0L, cols, rows, region = NULL, anon = FALSE)
```

Arguments

<code>path</code>	File path or URL to the TIFF.
<code>ifd_index</code>	IFD index (0-based). Default 0L.
<code>cols</code>	Integer vector of tile column indices (0-based).
<code>rows</code>	Integer vector of tile row indices (0-based). Must be the same length as <code>cols</code> .
<code>region</code>	Optional AWS region string.
<code>anon</code>	Logical. Use anonymous requests. Default FALSE.

Value

A list of tile result lists, each with data, dim, dtype.

A list of tile results, each with data, dim, and dtype components as described in `tiff_tile()`.

See Also

`tile_to_array()` to reshape individual results, `tiff_read_tiles()` for multi-file batches

Examples

```
## Not run:
tiles <- tiff_tiles("scene.tif", cols = 0:3, rows = rep(0L, 4))
arrays <- lapply(tiles, tile_to_array)

## End(Not run)
```

tile_to_array	<i>Convert a tile result to a matrix or array</i>
---------------	---

Description

Reshapes the flat numeric vector returned by `tiff_tile()`, each element of `tiff_tiles()`, or each element of the data column from `tiff_read_tiles()` into a matrix (single band) or 3D array (multi-band).

Usage

```
tile_to_array(tile)
```

Arguments

tile	A tile result list from <code>tiff_tile()</code> , <code>tiff_tiles()</code> , or an element of the data column from <code>tiff_read_tiles()</code> .
------	---

Details

The matrix is filled row-by-row (`byrow = TRUE`), consistent with the row-major (C) order returned by `async-tiff` and expected by `graphics::rasterImage()` and `ximage()`. Note that `as.vector(m)` on the result does **not** recover the original input order; use `as.vector(t(m))` for a round-trip.

Value

A matrix for single-band tiles, or a 3D array with dimensions `[height, width, bands]` for multi-band tiles.

Examples

```
## Not run:
tile <- tiff_tile("scene.tif", col = 0L, row = 0L)
m <- tile_to_array(tile)
dim(m) # c(256, 256) for a single-band 256x256 tile
ximage::ximage(m)

## End(Not run)
```

tiles_to_matrix	<i>Turn already-read data into a full matrix</i>
-----------------	--

Description

Turn already-read data into a full matrix

Usage

```
tiles_to_matrix(refs)
```

Arguments

refs data frame, output of tiff_read_tiles

Value

matrix, data shaped from raw read into 'raster order'

See Also

refs_tiles

Examples

```
vfile <- system.file("extdata/volcano-cog.tif", package = "rustycogs", mustWork = TRUE)
refs <- refs_tiles(vfile)
mat <- refs |> subset(ifd == 4) |>
  tiff_read_tiles() |> tiles_to_matrix()
```

Index

`graphics::rasterImage()`, [7](#)

`refs_tiles`, [2](#)

`tiff_ifd_info`, [3](#)

`tiff_ifd_info()`, [4](#)

`tiff_read_tiles()`, [3](#), [4](#), [6](#), [7](#)

`tiff_refs`, [4](#)

`tiff_refs()`, [3](#)

`tiff_tile`, [5](#)

`tiff_tile()`, [6](#), [7](#)

`tiff_tiles`, [6](#)

`tiff_tiles()`, [5](#), [7](#)

`tile_to_array`, [7](#)

`tile_to_array()`, [5](#), [6](#)

`tiles_to_matrix`, [8](#)