

Package ‘tiff’

February 20, 2015

Version 0.1-5

Title Read and write TIFF images

Author Simon Urbanek <Simon.Urbaneck@r-project.org>

Maintainer Simon Urbanek <Simon.Urbaneck@r-project.org>

Depends R (>= 2.9.0)

Description This package provides an easy and simple way to read, write and display bitmap images stored in the TIFF format. It can read and write both files and in-memory raw vectors.

License GPL-2 | GPL-3

SystemRequirements tiff and jpeg libraries

URL <http://www.rforge.net/tiff/>

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-09-04 07:27:36

R topics documented:

readTIFF	1
writeTIFF	3
Index	6

readTIFF	<i>Read a bitmap image stored in the TIFF format</i>
----------	--

Description

Reads an image from a TIFF file/content into a raster array.

Usage

```
readTIFF(source, native = FALSE, all = FALSE, convert = FALSE,  
info = FALSE, indexed = FALSE, as.is = FALSE)
```

Arguments

source	Either name of the file to read from or a raw vector representing the TIFF file content.
native	determines the image representation - if FALSE (the default) then the result is an array, if TRUE then the result is a native raster representation (suitable for plotting).
all	TIFF files can contain more than one image, if all = TRUE then all images are returned in a list, otherwise only the first image is returned.
convert	first convert the image into 8-bit RGBA samples and then to an array, see below for details.
info	if set to TRUE then the resulting image(s) will also contain information from TIFF tags as attributes
indexed	if set to TRUE then indexed images will be returned in the indexed form, i.e., as a matrix of integer indices referencing into a color map which is returned in the "color.map" attribute. This flag cannot be combined with convert or native and has no effect on images that are not indexed.
as.is	attempt to return original values without re-scaling where possible

Details

Most common files decompress into RGB (3 channels), RGBA (4 channels), Grayscale (1 channel) or GA (2 channels). Note that G and GA images cannot be directly used in `rasterImage` unless `native` is set to TRUE because `rasterImage` requires RGB or RGBA format (`nativeRaster` is always 8-bit RGBA).

TIFF images can have a wide range of internal representations, but only the most common in image processing are directly supported (8-bit, 16-bit integer and 32-bit float samples). Other formats (color maps, sub-8-bit images, etc.) are only supported via `convert=TRUE` which uses the built-in facilities of the TIFF library to convert the image into RGBA format with 8-bit samples (i.e. total of 32-bit per pixel) and then store the relevant components from there into real arrays. This is the same path as used by `native=TRUE` and so differs only in the output value. Note that conversion may result in different values than direct access as it is intended mainly for viewing and not computation.

Value

If `native` is FALSE then an array of the dimensions height x width x channels. If there is only one channel the result is a matrix. The values are reals between 0 and 1 (except for 32-bit floating point sample storage which are unscaled reals, and for indexed and `as.is=TRUE` which are integers). If `native` is TRUE then an object of the class `nativeRaster` is returned instead. The latter cannot be easily computed on but is the most efficient way to draw using `rasterImage`.

If `all` is TRUE then the result is a list of the above with one or more elements..

Note

Some non-standard formats such as 12-bit TIFFs are partially supported (there is no standard for packing order for TIFFs beyond 8-bit so we assume big-endian packing similar to the default fill order and only support single channel or indexed).

The `as.is=TRUE` option is experimental, cannot be used with `native` or `convert` and only works for integer storage TIFFs.

See Also

[rasterImage](#), [writeTIFF](#)

Examples

```
# read a sample file (R logo)
img <- readTIFF(system.file("img", "Rlogo.tiff", package="tiff"))

# read it also in native format
img.n <- readTIFF(system.file("img", "Rlogo.tiff", package="tiff"), native=TRUE)

# and also in converted
img.c <- readTIFF(system.file("img", "Rlogo.tiff", package="tiff"), convert=TRUE)

# if your R supports it, we'll plot it
if (exists("rasterImage")) { # can plot only in R 2.11.0 and higher
  plot(1:2, type='n')

  if (names(dev.cur()) == "windows") {
    # windows device doesn't support semi-transparency so we'll need
    # to flatten the image
    transparent <- img[, ,4] == 0
    img <- as.raster(img[, ,1:3])
    img[transparent] <- NA

    # interpolate must be FALSE on Windows, otherwise R will
    # try to interpolate transparency and fail
    rasterImage(img, 1.2, 1.27, 1.8, 1.73, interpolate=FALSE)

  } else {
    # any reasonable device will be fine using alpha
    rasterImage(img, 1.2, 1.27, 1.8, 1.73)
    rasterImage(img.n, 1.5, 1.5, 1.9, 1.8)
  }
}
```

writeTIFF

Write one or more bitmap images in TIFF format

Description

Writes images into a TIFF file or a raw vector representing such.

Usage

```
writeTIFF(what, where, bits.per.sample = 8L,
          compression = c("LZW", "none", "PackBits", "RLE", "JPEG", "deflate"),
          reduce = TRUE)
```

Arguments

what	either an image or a list of images. An image is a real matrix or array of three dimensions, or an object of the class "nativeRaster".
where	file name or a raw vector
bits.per.sample	number of bits per sample (numeric scalar). Supported values in this version are 8, 16, and 32.
compression	desired compression algorithm (string). Optionally, it can be specified as a numeric value corresponding to the compression TIFF tag, but it needs to be also supported by the underlying TIFF library
reduce	if TRUE then writeTIFF will attempt to reduce the number of planes in native rasters by analyzing the image to choose one of RGBA, RGB, GA or G formats, whichever uses the least planes without any loss. Otherwise the image is always saved with four planes (RGBA).

Details

By default writeTIFF uses the same number of planes as there are planes in the input image. For native images it is always four unless reduce = TRUE is set (see above). Consequently, color maps are not used. The output always uses contiguous planar configuration (baseline TIFF). The output is tagged with a photometric tag of either RGB (3 or 4 planes) or zero-is-black (1 or 2 planes). If what is a list then the TIFF output will be a directory of the corresponding number of images (in TIFF speak - not to be confused with file directories).

Value

If where is a raw vector then the value is the raw vector containing the TIFF contents, otherwise a scalar integer specifying the number of images written in the file.

Author(s)

Simon Urbanek

See Also

[readTIFF](#)

Examples

```
img <- readTIFF(system.file("img", "Rlogo.tiff", package="tiff"))
# write without the alpha channel
tiff <- writeTIFF(img[,-4], raw(0))
# read as native
```

writeTIFF

5

```
i2 <- readTIFF(tiff, native=TRUE)
# write reduced - should be the same as tiff
t2 <- writeTIFF(i2, raw(0), reduce=TRUE)
```

Index

*Topic **IO**

readTIFF, [1](#)

writeTIFF, [3](#)

rasterImage, [2](#), [3](#)

readTIFF, [1](#), [4](#)

writeTIFF, [3](#), [3](#)