Package 'pixmap'

January 29, 2021

Version 0.4-12

Date 2021-01-29

Title Bitmap Images / Pixel Maps

Imports methods, graphics, grDevices

Author Roger Bivand, Friedrich Leisch and Martin Maechler

Maintainer Friedrich Leisch <Friedrich.Leisch@R-project.org>

Description Functions for import, export, plotting and other manipulations of bitmapped images.

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-29 13:20:06 UTC

R topics documented:

`addlogo-methods` *Methods for Adding a Pixmap Logo to a Plot*

Description

This method allows the addition of a re-scaled pixmap to an existing plot, possibly as a logo, optionally preserving aspect. It may be used interactively with `locator`, and is positioned in the coordinate system of the plot region. Since the logo is displayed using `image`, it does not yet seem possible to use the function outside this region.

Usage

```
  addlogo(x, ...)
  ## S4 method for signature 'pixmap'
addlogo(x, px, py, asp = NULL)
```

Arguments

x	an object of class pixmap
px	a vector of two x coordinates, or a list with two named elements x and y, such as that returned by locator.
py	if px is not a list, a vector of two y coordinates
asp	if omitted or NULL (default), output respects both y coordinates, if a number greater than zero, aspect is preserved in proportion to the difference between x coordinates, multiplied by asp, and only the first y coordinate is respected.
...	potentially further arguments passed to and from methods.

Value

The same pixmap object with changed bounding box and cell resolution

Author(s)

Roger Bivand

Examples

```
x <- read.pnm(system.file("pictures/logo.ppm", package="pixmap")[1])
plot(x)
for (i in 1:7)
  addlogo(x, px=c(0, (101/77)*11), py=c((i-1)*11, i*11), asp=1)
```

---

channels-methods *Methods for Channel Manipulation*

Description

Functions for manipulation and extraction of colors from channel-based pixmap formats. Methods for generic addChannels adds the color channels of a colored pixmap and returns a grey version. Methods for generic getChannels return numeric matrices or arrays containing the specified channels.

Usage

```
  addChannels(object, coef = NULL)
  getChannels(object, colors = "all")
```

Arguments

object	Object of class "pixmap".
coef	Coefficients for the color channels, a numeric vercot with as many elements as there are color channels in the pixmap. The default for RGB is c(0.30,0.59,0.11), which makes a luminance-based conversion from color to grey.
colors	Character vector naming the color channels that shall be extracted. The default of "all" returns all channels simultaneously.

Author(s)

Friedrich Leisch

Examples

```
x <- pixmapRGB(rep(1:5, 3), nrow=4)
plot(x)
print(x)

getChannels(x)
getChannels(x, colors=c("red", "green"))

y = addChannels(x)
plot(y)
print(y)

## extract only the red channel
y = addChannels(x, coef=c(1,0,0))
plot(y)
```

---

pixmap *Pixmap Images*

Description

The family "pixmap" ("pixel maps") of classes provides methods for creating, plotting and converting bitmapped images in three different formats: RGB, grey and indexed pixmaps.

Usage

```
pixmap(data=NULL, nrow=dim(data)[1], ncol=dim(data)[2],
       bbox=NULL, bbcent=FALSE, cellres=NULL)
pixmapRGB(data, ...)
pixmapGrey(data, ...)
pixmapIndexed(data, col, ...)
```

Arguments

data	An optional data vector.
nrow	Vertical size of the image in pixels.
ncol	Horizontal size of the image in pixels.
bbox	Bounding box of the image, vector of length 4 of form c(x1,y1,x2,y2) with coordinates for the lower left corner and upper right corner.
bbcent	Logical, if TRUE the bounding box specifies the coordinates of the centers of the lower left and upper right pixels, default is the coordinates of the lower left and upper right corner of the image.
cellres	Numeric vector of length 1 or 2, specifies the resolution of pixels in horizontal and vertical direction. If only one value is given, resolution in both directions is identical.
col	Character vector of colors to use for indexed pictures, or a function like rainbow which can be used to create a palette. Colors set to NA are transparent; this can be used,e.g., for overlaying plots.
...	Additional arguments passed to pixmap().

Details

If the data argument is 2- or 3-dimensional, nrow and ncol default to the first two dimensions of data, such that pixmap does the expected when given a matrix or an array.

The arguments bbox, bbcent and cellres can be used to specify a coordinate system for the image. Note that together with nrow and ncol the coordinate system is overspecified, hence not all parameters must be specified, the rest is computed or set to sensible defaults.

For bbcent=FALSE we have cellres[1] = (bbox[3]-bbox[1])/ncol and cellres[2] = (bbox[4]-bbox[2])/nrow, for bbcent=TRUE we get cellres[1] = (bbox[3]-bbox[1])/(ncol-1) and cellres[2] = (bbox[4]-bbox[2])/(nrow-1)

The name pixmap was chosen because both image and bitmap are already used in R.

Author(s)

Friedrich Leisch

See Also

pixmap-class, read.pnm

Examples

```
## A simple example
x <- pixmapIndexed(rep(1:8, 9), nrow=6, col=terrain.colors(8))
plot(x)

## The same with different colors, and passing the function instead of
## a color vector
x <- pixmapIndexed(rep(1:8, 9), nrow=6, col=rainbow)
plot(x)
```

```
plot(x, asp=.5, axes=TRUE)

## Read data from a file
x <- read.pnm(system.file("pictures/logo.ppm", package="pixmap")[1])
plot(x)

## Another example that math can be beautiful
x <- seq(-3,3,length=100)
z1 <- outer(x,x,function(x,y) abs(sin(x)*sin(y)))
z2 <- outer(x,x,function(x,y) abs(sin(2*x)*sin(y)))
z3 <- outer(x,x,function(x,y) abs(sin(x)*sin(2*y)))

## Notice that we specify a bounding box to get the correct
## coordinates on the axes. z1, z2 and z3 are used as red,
## green and blue channel, respectively.
z <- pixmapRGB(c(z1,z2,z3), 100, 100, bbox=c(-1,-1,1,1))
plot(z, axes=TRUE)

## look at a grey version
plot(as(z, "pixmapGrey"))

## subsetting works as expected
plot(z[1:20,])
plot(z[,1:40])
plot(z[1:20,10:40])

## overlay different images using transparency
## base image as before
x <- pixmapIndexed(rep(1:8, 9), nrow=6, col=terrain.colors(8))
plot(x)
## make a mask of vertical bars
mask <- array(0,dim=c(6,12))
mask[,seq(1,12,3)] <- 1
## plot this mask over existing image with transparent and black color
plot(pixmapIndexed(mask,col=c("NA","#000000")),add=TRUE)
```

`pixmap-class` *Class Family "pixmap".*

Description

The family `"pixmap"` ("pixel maps") of classes provides methods for creating, plotting and converting bitmapped images in currently three different formats: RGB (`"pixmapRGB"`), grey (`"pixmapGrey"`)and indexed pixmaps (`"pixmapIndexed"`).

Objects from the Class

Objects can be created by calls of the form `new("pixmap",...)` or using the creator functions `pixmap` (similar for all child classes of name (`"pixmapXXX"`).

Slots

`size`: Object of class `"integer"` and length 2 (number of rows and columns).

`cellres`: Object of class `"numeric"` and length 2 specifying the cell resolution of each pixel in user coordinates.

`bbox`: Object of class `"numeric"` and length 4, the coordinates of the bounding box (x bottom, y bottom, x top, y top).

`channels`: A character vector naming the channel slots of the object (`NULL` for indexed pixmaps).

`red`, `green`, `blue`: Only for class `"pixmapRGB"` with matrices specifying the red, green and blue channel of the picture.

`grey`: Only for class `"pixmapGrey"`, a matrix specifying the grey intensity (0=black, 1=white) of the picture.

`col`: Only for class `"pixmapGrey"`, a character vector with a map of color names.

`index`: Only for class `"pixmapIndexed"`, an integer matrix with codes from the color map.

Details

Class `"pixmap"` specifies the basic geometry of a picture: the size in pixels together with information for an optional coordinate system, see `pixmap` for details.

Grey and indexed pixmaps are besically matrices (contained in the `grey` or `index` slot, respectively). The element `[1,1]` corresponds to the upper left corner as usual. For grey pixmaps the elements must be between 0 (black) and 1 (white). Indexed pixmaps have integer elements, each giving the index number corresponding to the palette specified in slot `"col"`. Colors are given using the usual R color strings (either names like `"red"` or hex values like `"#FF0000"`). Alternatively, a function to create a color palette can be specified, see `rainbow` or `heat.colors` for examples.

RGB pixmaps have three matrices for each of the three color channels. Elements of the matrices must be between 0 (=color off) and 1 (=color at maximum intensity).

Methods for coercion between all formats are available.

Class `"pixmapChannels"` is a helper parent class currently containing classes `"pixmapRGB"` and `"pixmapGrey"`.

Author(s)

Friedrich Leisch

See Also

`pixmap`

pnm *Read/Write Portable Anymap Images*

Description

Reading and writing of bitmap images in PBM (black/white), PGM (grey) and PPM (color) format.

Usage

```
read.pnm(file, ...)
write.pnm(object, file= NULL, forceplain = FALSE, type = NULL, maxval = 255)
```

Arguments

file	name of the pnm file (general connections do not work at the moment).
...	further arguments passed to pixmap (like bbox).
object	an object of class "pixmap".
forceplain	logical; if true, an ASCII pnm file is written. Default is to write a binary (raw) file.
type	one of "pbm", "pgm" or "ppm". Default is to use "pgm" for grey images and "ppm" for color images.
maxval	the maximum color-component value; the default is a colour depth of 8 bits, i.e., the integer 255.

Details

read.pnm reads a pnm file and loads the image into an object of class pixmap.

write.pnm writes an object of class pixmap to a pnm file, the type argument controls wheter the written image file is a black-and-white bitmap (pbm), grey (pgm) or color (ppm).

plot.pnm plots a pnm object using the command image. The only difference is that the element [1,1] of pnmobj is plotted as the upper left corner (plain image would plot [1,1] as the lower left corner.

Value

read.pnm returns an object of class pixmapRGB for color pixmaps (ppm), and an object of class pixmapGrey for pbm and pgm. Note that the *type* of file as determined by the first two bytes according to pnm standards is important, *not the extension* of the file. In fact, the file name extension is completely ignored.

Author(s)

Roger Bivand and Friedrich Leisch

See Also

pixmap

Examples

```
x <- read.pnm(system.file("pictures/logo.ppm", package="pixmap")[1])
plot(x)
print(x)

x <- read.pnm(system.file("pictures/logo.pgm", package="pixmap")[1])
plot(x)

x <- read.pnm(system.file("pictures/logo.pbm", package="pixmap")[1])
plot(x)
```

Index