Package 'nara'

May 28, 2026

Type Package

Title Native Raster Image Tools

Version 1.0.2

Maintainer Mike Cheng <mikefc@coolbutuseless.com>

Description Native rasters are a core R image format which use a compact color representation. This color representation closely aligns with graphics device internals meaning that these images can be rendered quickly. This package provides functions to quickly create, manipulate and composite native rasters.

URL https://github.com/coolbutuseless/nara, https://coolbutuseless.github.io/package/nara/index.html

BugReports https://github.com/coolbutuseless/nara/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

LinkingTo colorfast

Depends R (>= 4.1.0)

Imports colorfast

Suggests testthat (>= 3.0.0), jpeg, fastpng, png, knitr, rmarkdown, magick

Config/testthat/edition 3

Copyright The included 'deer' sprites are by Calciumtrice and licensed under CC BY 3.0. See 'inst/LICENSE-deer.txt' for more details. The included bitmap font 'spleen' was created by Frederic Cambus (c) 2018-2024. See 'inst/LICENSE-spleen.txt' for full LICENSE

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Mike Cheng [aut, cre, cph],
Julien Vernay [ctb] (Author of public domain code for thick polylines)

Repository CRAN

Date/Publication 2026-05-28 13:50:02 UTC

Contents

deer *Deer images - a list of native raster images*

Description

The 'deer' sprites are by Calciumtrice and licensed under CC BY 3.0. See 'inst/LICENSE-deer.txt' for more details. To view license information:

Usage

```
deer
```

Format

An object of class `list` of length 15.

Details

```
cat(readLines(system.file('LICENSE-deer.txt', package = 'nara')), sep = "\n")
```

is_nativeraster *Check if object is a native raster image*

Description

Check if object is a native raster image

Usage

```
is_nativeraster(x)
```

Arguments

x	object to check

Value

logical. TRUE if object is a native raster image

Examples

```
is_nativeraster(mtcars)

nr <- nr_new(100, 100)
is_nativeraster(nr)
```

magick_to_nr *Convert a 'magick' image to native raster image*

Description

Convert a 'magick' image to native raster image

Usage

```
magick_to_nr(im, dst = NULL)

nr_to_magick(nr)
```

Arguments

im	image from the magick package
dst	destination native raster image. If NULL (the default) a new native raster image will be created If a native raster image is supplied here, it must have the exact dimensions to match the matrix
nr	native raster image

Value

New native raster image

See Also

Other conversion functions: matrix_to_nr(), nr_to_raster(), nrs_to_gif(), nrs_to_mp4()

Examples

```
im <- magick::logo
nr <- magick_to_nr(im)
plot(nr)
```

matrix_to_nr *Convert a numeric matrix to native raster image*

Description

Convert a numeric matrix to native raster image

Usage

```
matrix_to_nr(mat, palette, fill = "transparent", min = 0, max = 1, dst = NULL)
```

Arguments

mat	integer matrix
palette	vector of colors. For an integer matrix, this palette must contain at least as many colors as the maximum integer value in mat. For a numeric matrix, any length palette is allowed, and the nearest corresponding color is chosen (no interpolation is done).
fill	Color to be used for values < 1 when input is an integer matrix. Default: 'transparent'.
min, max	assumed range for the numeric data. values from the palette will be interpolated using this range as the extents. An error will occur if a value lies outside this range. Default: (0, 1)
dst	destination native raster image. If NULL (the default) a new native raster image will be created If a native raster image is supplied here, it must have the exact dimensions to match the matrix

Value

native raster image

See Also

Other conversion functions: magick_to_nr(), nr_to_raster(), nrs_to_gif(), nrs_to_mp4()

Examples

```
# integer matrix
m <- matrix(1:12, 3, 4)
m
palette <- rainbow(12)
nr <- matrix_to_nr(m, palette)
plot(nr)


# numeric matrix
m <- matrix(runif(20 * 30), 20, 30)
palette <- c('red', 'blue', 'black', 'green', 'blue', 'yellow')
nr <- matrix_to_nr(m, palette)
plot(nr)
```

---

nrs_to_gif *Convert a list of native rasters to an animated gif*

Description

Convert a list of native rasters to an animated gif

Usage

```
nrs_to_gif(nr_list, filename, verbose = FALSE, framerate = 30, ...)
```

Arguments

nr_list	list of native raster image
filename	filename of gif
verbose	logical. default FALSE
framerate	frames per second
...	other arguments passed to magick::image_write_gif()

Value

None.

See Also

Other conversion functions: magick_to_nr(), matrix_to_nr(), nr_to_raster(), nrs_to_mp4()

Examples

```
im <- magick::logo
nr1 <- magick_to_nr(im)
nr2 <- nr_copy(nr1)
nrs_to_gif(list(nr1, nr2), "nothing.gif")
```

---

nrs_to_mp4 *Convert a list of native rasters to an mp4 video file*

Description

Convert a list of native rasters to an mp4 video file

Usage

```
nrs_to_mp4(nr_list, filename, verbose = FALSE, ...)
```

Arguments

nr_list	list of native raster image
filename	mp4 filename
verbose	logical. default FALSE
...	other arguments passed to magick::image_write_video()

Value

None.

See Also

Other conversion functions: `magick_to_nr()`, `matrix_to_nr()`, `nr_to_raster()`, `nrs_to_gif()`

Examples

```
im <- magick::logo
nr1 <- magick_to_nr(im)
nr2 <- nr_copy(nr1)
nrs_to_mp4(list(nr1, nr2), "nothing.mp4")
```

---

nr_blit *Copy a native raster image into another at an arbitrary location, scale and rotation*

Description

Copy a whole image, or subset of an image, into the destination image and configure the angle and size of the copied image.

Usage

```
nr_blit(
  dst,
  src,
  x,
  y,
  xsrc = 0L,
  ysrc = 0L,
  w = -1L,
  h = -1L,
  hjust = 0.5,
  vjust = 0.5,
  angle = 0,
  scale = 1,
  use_alpha = TRUE
)
```

Arguments

`dst`, `src`	source and destination native raster images
`x`, `y`	Location in `dst` to place the `src` image. These values must be vectors of the same length. If the length is greater than 1, then the `src` will be pasted into `dst` at multiple locations.

xsrc, ysrc	Upper-left coordinates of the bounding box within the src image to copy. Default: (0, 0)
w, h	Width and height of the bounding box within the src to copy. If size is negative, then the actual width/height of the src is used. Default: (-1, -1)
hjust, vjust	specify horizontal and vertical justification of the handle on the src image to be placed at (x, y) within the dst image. Default (0, 0) is to use the top-left corner of the image as the handle. Use (0.5, 0.5) to centre the src image at the dst location (x, y)
angle	Rotation angle (clockwise) in radians. Default: 0. If this is a vector, then the src will be pasted into the dst at multiple angles.
scale	Zoom factor. Default: 1. IF this is a vector, then the src will be pasted into dst at multiple scales.
use_alpha	Use alpha channel when drawing? Logical. Default: TRUE

Details

This operation is vectorised such that a single src image can be pasted multiple times into the dst image with varying location, angle and scale.

Value

Invisibly return the supplied dst native raster image which was been modified in-place

See Also

Other blitting functions: nr_blit_multi()

Examples

```
nr <- nr_new(50, 50, 'grey80')
nr_blit(dst = nr, src = deer[[1]], x = 25, y = 25)
plot(nr)

nr <- nr_new(300, 200, 'grey80')
sq <- nr_new(20, 20, 'darkblue')
nr_blit(nr, src = sq, x = 100, y = 100, angle = pi/3, scale = 5)
plot(nr)

nr <- nr_new(800, 600, 'grey80')
sq <- fastpng::read_png(system.file("image/deer-1.png", package="nara"), type = 'nativeraster')
nr_blit(nr, src = sq, x = 300, y = 240, angle = pi/2, scale = 1)
plot(nr)
```

`nr_blit_multi` *Multiple blit operations in a single call*

Description

Multiple blit operations in a single call

Usage

```
nr_blit_multi(dst, src, config)
```

Arguments

`dst`	destination native raster
`src`	list of native raster images.

`config`	data.frame of configuration information for each blit. Columns in this data.frame match the arguments to `nr_blit()`. Requires columns:
	`idx` The index of the image in the `src` list i.e. `src[[idx]]`
	`x,y` location in `dst` to place `src[[idx]]`
	Optional columns which take a default value if not present:
	`xsrc,ysrc` start coordinates within src. Default: (0, 0)
	`w,h` size within `src`. Default: use whole width/height
	`hjust,vjust` Default: (0.5, 0.5)
	`angle` Default: 0
	`scale` Zoom factor. Default: 1
	`use_alpha` Default: true
	`render` Should this given operation be rendered? Default: true

Value

Invisibly return the supplied `dst` native raster image which was been modified in-place

See Also

Other blitting functions: `nr_blit()`

Examples

```
nr <- nr_new(90, 90, 'grey60')

config <- data.frame(
  idx = c(1, 2, 3, 4),
  x = c(10, 10, 40, 40) + 15,
  y = c(10, 40, 40, 10) + 15,
  xsrc = 0L,
  ysrc = 0L,
```

```
  w = -1L,
  h = -1L,
  hjust = 0.5,
  vjust = 0.5,
  angle = c(0, 0, 0, pi/4),
  scale = c(0.5, 1, 1, 1),
  use_alpha = TRUE,
  render = TRUE
)

nr_blit_multi(dst = nr, src = deer, config = config)
plot(nr)
```

`nr_circle` *Draw circles on a native raster image*

Description

Draw circles on a native raster image

Usage

```
nr_circle(nr, x, y, r, fill = "black", color = NA, use_alpha = TRUE)
```

Arguments

`nr`	native raster image
`x, y`	coordinates of centre of circle. [vector]
`r`	radius [vector]
`fill`	interior fill color [vector]
`color`	outline color. Default: NA. [vector]
`use_alpha`	Use alpha channel when drawing? Logical. Default: TRUE

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other drawing functions: `nr_polygon()`, `nr_polyline()`, `nr_rect()`, `nr_tri_mesh()`

Examples

```
set.seed(1)
w <- 200
h <- 150
nr <- nr_new(w, h, 'black')

N <- 10
coords <- expand.grid(x = seq(2, w, length.out = N),
                      y = seq(2, h, length.out = N))

cols <- sample(grDevices::terrain.colors(nrow(coords)))

nr_circle(
  nr,
  x = coords$x,
  y = coords$y,
  r = w/N/1.3,
  fill = cols
)
plot(nr)
```

`nr_color_replace` *Replace colors in a native raster image*

Description

Replace colors in a native raster image

Usage

```
nr_color_replace(nr, old, new)
```

Arguments

`nr`	native raster image
`old`	Vector of old colors
`new`	Vector of replacement colors

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other color manipulation functions: `nr_desaturate()`, `nr_dither()`, `nr_threshold()`

Examples

```
nr <- nr_new(10, 10, 'hotpink')
nr_color_replace(nr, 'hotpink', 'grey80')
plot(nr)
```

---

nr_copy *Create a new native raster image and copy the dimensions and contents from an existing image*

Description

Create a new native raster image and copy the dimensions and contents from an existing image

Usage

```
nr_copy(nr)
```

Arguments

nr	native raster image

Value

New native raster image

See Also

Other image creation functions: nr_new()

Examples

```
nr1 <- nr_new(200, 200, 'hotpink')
nr2 <- nr_copy(nr1)
plot(nr2)
```

nr_copy_into *Copy the contents of one native raster image into another.*

Description

The source and destination native raster images must have the same dimensions.

Usage

```
nr_copy_into(dst, src)
```

Arguments

src, dst	Source and destination native raster images

Details

If the native raster images are of different sizes or alpha blending is required, use the nr_blit() function.

Value

Invisibly return the supplied 'dst' native raster image which was been modified in-place

Examples

```
nr1 <- nr_new(200, 100, 'hotpink')
nr2 <- nr_new(200, 100, 'green')
nr_copy_into(nr1, nr2)
plot(nr1)
```

nr_crop *Crop a section out of a native raster image into a new image*

Description

Crop a section out of a native raster image into a new image

Usage

```
nr_crop(nr, x, y, w, h)

nr_crop2(nr, loc)
```

Arguments

nr	native raster image
x, y, w, h	dimensions of cropped section
loc	dimensions of cropped section. A vector of 4 values i.e. c(x, y, w, h)

Value

New native raster image

Examples

```
nr <- deer[[1]]
dim(nr)
plot(nr)

nr2 <- nr_crop(nr, 16, 0, 16, 16)
dim(nr2)
plot(nr2)
```

---

nr_desaturate *Move image colors to gray*

Description

Move image colors to gray

Usage

```
nr_desaturate(nr, factor = 1)
```

Arguments

nr	native raster image
factor	desaturation factor. Default: 1 (fully desaturate)

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other color manipulation functions: nr_color_replace(), nr_dither(), nr_threshold()

Examples

```
plot(deer[[1]])
nr <- deer[[1]] |> nr_copy()

nr_desaturate(nr)
plot(nr)
```

`nr_dither` *Dither to binary image*

Description

Dither to binary image

Usage

```
nr_dither(nr, value = 0.5, algo = "fs")
```

Arguments

`nr`	native raster image
`value`	Threshold value. Default: 0.5 (valid range [0, 1])
`algo`	Dithering algorithm. 'fs' (floyd-steinberg) or 'atkinson'. Default: 'fs'

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other color manipulation functions: `nr_color_replace()`, `nr_desaturate()`, `nr_threshold()`

Examples

```
nr <- nr_copy(deer[[1]])
plot(nr)

nr_dither(nr, 0.99)
plot(nr)
```

nr_fill *Fill a native raster image with the given color*

Description

Fill a native raster image with the given color

Usage

```
nr_fill(nr, color)
```

Arguments

nr	native raster image
color	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb

Value

Invisibly return the supplied native raster image which was been modified in-place

Examples

```
nr <- nr_new(400, 300, 'hotpink')
nr_fill(nr, 'blue')
plot(nr)
```

nr_fliph *Flip a native raster image horizontally*

Description

Flip a native raster image horizontally

Usage

```
nr_fliph(nr)
```

Arguments

nr	native raster image

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other transformation functions: nr_flipv(), nr_rotate(), nr_transpose()

Examples

```
nr <- nr_new(400, 200, 'white')
nr_rect(nr, 0, 0, 30, 15)
plot(nr)
nr_fliph(nr)
plot(nr)
```

---

nr_flipv *Flip a native raster image vertically*

Description

Flip a native raster image vertically

Usage

```
nr_flipv(nr)
```

Arguments

nr	native raster image

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other transformation functions: nr_fliph(), nr_rotate(), nr_transpose()

Examples

```
nr <- deer[[1]] |> nr_copy()
plot(nr)
nr_flipv(nr)
plot(nr)
```

nr_line *Draw lines on a native raster image*

Description

Uses Bresenham's algorithm to draw lines. No antialiasing.

Usage

```
nr_line(nr, x1, y1, x2, y2, color = "black", linewidth = 1, use_alpha = TRUE)
```

Arguments

nr	native raster image
x1, y1, x2, y2	Vectors of coordinates of endpoints of line
color	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb
linewidth	Line linewidth. Default: 1. If linewidth = 1 then a naive version of Bresenham is used to draw the points. If linewidth is greater than 1, then the line is convert to a triangle strip and rendered as polygons.
use_alpha	Use alpha channel when drawing? Logical. Default: TRUE

Value

Invisibly return the supplied native raster image which was been modified in-place

Examples

```
w <- 200
h <- 150
nr <- nr_new(w, h, 'grey80')

N <- 40
nr_line(
  nr,
  x1 = seq(0, 2*w, length.out =  N), y1 = 0,
  x2 = 0, y2 = seq(0, 2 * h, length.out = N),
  color = rainbow(N)
)

plot(nr)
```

`nr_mask_begin` *Start/end region of masked drawing*

Description

These functions are used to globally set a mask which determines where pixels are affected by drawing functions.

Usage

```
nr_mask_begin(nr, mask)

nr_mask_end(nr)
```

Arguments

`nr`	native raster image
`mask`	native raster image to use as mask. Must be the same size as `nr`.

Details

Drawing operations between `nr_mask_begin()` and `nr_mask_end()` calls will only affect pixels where the `mask` is not transparent.

Value

Invisibly return the original native raster image which has been modified in-place

Examples

```
mask <- deer[[1]] |> nr_resize(600, 400)
plot(mask)
nr <- nr_new_from(mask)

nr_mask_begin(nr, mask = mask)
nr_circle(nr, 300, 200, 200, fill = 'blue', color = 'black')
nr_circle(nr, 350, 200, 100, fill = 'darkgreen', color = 'black')
nr_mask_end(nr)

plot(nr)
```

nr_new *Create a native raster image*

Description

A native raster image in R looks like an integer matrix, but is interpreted differently by graphics devices:

Usage

```
nr_new(width, height, fill = "white")

nr_new_from(nr, fill = "white")
```

Arguments

width, height	Image dimensions in pixels
fill	Background fill color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb
nr	native raster image to use as the template for the size of the new image in nr_new_from()

Details

- The data should be treated as RGBA pixels in row-major ordering
- Each 32-bit integer should be interpreted as 4-bytes - one for each of the R, G, B and A color channels

Value

native raster image

See Also

Other image creation functions: nr_copy()

Examples

```
nr <- nr_new(400, 300, 'hotpink')
plot(nr)
```

nr_point *Draw points on a native raster image*

Description

Draw points on a native raster image

Usage

```
nr_point(nr, x, y, color = "black", use_alpha = TRUE)
```

Arguments

nr	native raster image
x, y	Vectors of point coordinates
color	Vector of colors
use_alpha	Use alpha channel when drawing? Logical. Default: TRUE

Value

Invisibly return the supplied native raster image which was been modified in-place

Examples

```
w <- 200
h <- 150
nr <- nr_new(w, h, 'grey80')

coords <- expand.grid(x = seq(w) - 1, y = seq(h) - 1)
cols   <- sample(rainbow(nrow(coords)))
nr_point(nr, x = coords$x, y = coords$y, color = cols)
plot(nr)
```

nr_polygon *Draw multiple polygons on a native raster image*

Description

Draw multiple polygons on a native raster image

Usage

```
nr_polygon(
  nr,
  x,
  y,
  id = NULL,
  fill = "black",
  color = NA,
  linewidth = 1,
  mitre_limit = linewidth,
  use_alpha = TRUE
)
```

Arguments

nr	native raster image
x, y	Vectors of point coordinates
id	integer vector used to separate coordinates into multiple polygons. Consecutive runs of the same id value belong to the same polygon. If NULL (the default) then all coordinates are assumed to be vertices of a single polygon.
fill	fill color
color	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb
linewidth	Line linewidth. Default: 1. If linewidth = 1 then a naive version of Bresenham is used to draw the points. If linewidth is greater than 1, then the line is convert to a triangle strip and rendered as polygons.
mitre_limit	Limit the size of the mitre when two lines meet at an acute angle and linewidth is greater than 1. Default: same as line linewidth which mostly looks OK.
use_alpha	Use alpha channel when drawing? Logical. Default: TRUE

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other drawing functions: nr_circle(), nr_polyline(), nr_rect(), nr_tri_mesh()

Examples

```
w <- 200
h <- 150
nr <- nr_new(w, h, 'grey80')

N  <- 20
theta <- seq(0, 2 * pi, length.out = N)
xs <- w/2 + 50 * cos(theta)
```

```
ys <- h/2 + 50 * sin(theta)

nr_polygon(nr, xs, ys, fill = 'grey50', col = 'black')
nr_point(nr, xs, ys, col = 'red')
plot(nr)
```

`nr_polyline` *Draw a polyline on a native raster image*

Description

Draw a polyline on a native raster image

Usage

```
nr_polyline(
  nr,
  x,
  y,
  color = "black",
  linewidth = 1,
  mitre_limit = linewidth,
  close = FALSE,
  use_alpha = TRUE
)
```

Arguments

`nr`	native raster image
`x`, `y`	Vectors of point coordinates
`color`	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form `#rrggbbaa`, `#rrggbb`, `#rgba` or `#rgb`
`linewidth`	Line linewidth. Default: 1. If `linewidth = 1` then a naive version of Bresenham is used to draw the points. If linewidth is greater than 1, then the line is convert to a triangle strip and rendered as polygons.
`mitre_limit`	Limit the size of the mitre when two lines meet at an acute angle and linewidth is greater than 1. Default: same as line linewidth which mostly looks OK.
`close`	Should the polyline be closed? I.e. should a line be drawn between the last point and the first point? Default: FALSE
`use_alpha`	Use alpha channel when drawing? Logical. Default: TRUE

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other drawing functions: nr_circle(), nr_polygon(), nr_rect(), nr_tri_mesh()

Examples

```
w <- 200
h <- 150
nr <- nr_new(w, h, 'grey80')

N  <- 20
theta <- seq(0, 2 * pi, length.out = N)
xs <- w/2 + 50 * cos(theta)
ys <- h/2 + 50 * sin(theta)

nr_polyline(nr, xs, ys)
nr_point(nr, xs, ys, col = 'red')
plot(nr)
```

---

nr_rect *Draw rectangles on a native raster image*

Description

Draw rectangles on a native raster image

Usage

```
nr_rect(
  nr,
  x,
  y,
  w,
  h,
  fill = "black",
  color = NA,
  hjust = 0,
  vjust = 0,
  linewidth = 1,
  use_alpha = TRUE
)
```

Arguments

nr	native raster image
x, y	coordinates of lower left corner of rectangle. [vector]
w, h	width and height of rectangle. [vector]
fill	interior fill color [vector]

color	outline color. Default: NA. [vector]
hjust, vjust	specify horizontal and vertical justification of the handle on the src image to be placed at (x, y) within the dst image. Default (0, 0) is to use the top-left corner of the image as the handle. Use (0.5, 0.5) to centre the src image at the dst location (x, y)
linewidth	Line linewidth. Default: 1. If linewidth = 1 then a naive version of Bresenham is used to draw the points. If linewidth is greater than 1, then the line is convert to a triangle strip and rendered as polygons.
use_alpha	Use alpha channel when drawing? Logical. Default: TRUE

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other drawing functions: nr_circle(), nr_polygon(), nr_polyline(), nr_tri_mesh()

Examples

```
set.seed(1)
w <- 200
h <- 150
nr <- nr_new(w, h, 'grey80')

N <- 10
coords <- expand.grid(x = seq(2, w, length.out = N),
                      y = seq(2, h, length.out = N))

cols <- sample(grDevices::heat.colors(nrow(coords)))

nr_rect(
  nr,
  x = coords$x,
  y = coords$y,
  w = w/N - 1,
  h = h/N - 1,
  fill = cols
)
plot(nr)
```

---

nr_resize *Resize a native raster by specifying the output dimensions*

Description

Resize a native raster by specifying the output dimensions

Usage

```
nr_resize(nr, width, height, algo = "nn")
```

Arguments

nr	native raster image
width, height	dimensions for output image
algo	'nn' for nearest neighbor (the default), or 'bilinear' for bilinear interpolation.

Value

New native raster image

See Also

Other resizing functions: nr_scale()

Examples

```
stretched <- deer[[1]] |>
    nr_copy() |>
    nr_resize(100, 40, algo = 'nn')
plot(stretched)
```

---

nr_rotate *Rotate a native raster image by 90,180,270 degrees*

Description

Rotate a native raster image by 90,180,270 degrees

Usage

```
nr_rotate(nr, angle)
```

Arguments

nr	native raster image
angle	one of 0,90,180,270

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other transformation functions: nr_fliph(), nr_flipv(), nr_transpose()

Examples

```
nr <- nr_new(20, 10, 'hotpink')
dim(nr)
nr_rotate(nr, 90)
dim(nr)
```

`nr_scale` *Resize a native raster image using a scale factor*

Description

Resize a native raster image using a scale factor

Usage

```
nr_scale(nr, scale, algo = "nn")
```

Arguments

`nr`	native raster image
`scale`	scale factor
`algo`	'nn' for nearest neighbor (the default), or 'bilinear' for bilinear interpolation.

Value

New native raster image

See Also

Other resizing functions: `nr_resize()`

Examples

```
big <- deer[[1]] |> nr_scale(2)
plot(big)
```

nr_text_mono *Draw text on a native raster image using the built-in monospaced bitmapped font.*

Description

The only font currently available is 'spleen' - a monospace bitmap font from: https://github.com/fcambus/spleen

Usage

```
nr_text_mono(
  nr,
  x,
  y,
  str,
  color = "black",
  fontsize = 8L,
  hjust = 0.5,
  vjust = 0.5,
  use_alpha = TRUE
)
```

Arguments

nr	native raster image
x, y	coordinates of lower-left corner of text
str	character string
color	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb
fontsize	height of font in pizels. Only valid values are 8, 12 and 16. Default: 8.
hjust, vjust	Justification of text relative to its (x,y) location. Default (0.5, 0.5) to centre the text at (x, y)
use_alpha	Use alpha channel when drawing? Logical. Default: TRUE

Details

The 'spleen' font is licensed under BSD and the license is included in this package as "LICENSE-spleen.txt". To view LICENSE:

```
cat(readLines(system.file('LICENSE-spleen.txt', package = 'nara')), sep = "\n")
```

Value

Invisibly return the supplied native raster image which was been modified in-place

Examples

```
w <- 150
h <- 30
nr <- nr_new(w, h, 'grey80')

nr_text_mono(nr, x = w/2, y = h/2, str = "Hello RStats!", fontsize = 16)
plot(nr)

nr <- nr_new(w, h, 'grey80')
nr_text_mono(nr, x = 0, y = 0, str = "Hello RStats!", hjust = 0, vjust = 1, fontsize = 16)
plot(nr)
```

---

nr_threshold *Threshold to binary image*

Description

Threshold to binary image

Usage

```
nr_threshold(nr, value = 0.5)
```

Arguments

nr	native raster image
value	Threshold value. Default: 0.5 (valid range [0, 1])

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other color manipulation functions: nr_color_replace(), nr_desaturate(), nr_dither()

Examples

```
nr <- nr_copy(deer[[1]])
plot(nr)

nr_threshold(nr, 0.9)
plot(nr)
```

nr_to_raster *Convert native raster images to/from other R objects*

Description

Convert native raster images to/from other R objects

Usage

```
nr_to_raster(nr)

raster_to_nr(ras, dst = NULL)

nr_to_array(nr)

array_to_nr(arr, dst = NULL)
```

Arguments

nr	native raster image
ras	standard R raster i.e. a character matrix of hex color values
dst	destination native raster image. If NULL (the default) a new
arr	3d numeric array representing R,G,B,A values with dimensions [nrow, ncol, 4] or [nrow, ncol, 3]. Each value is in range [0,1].

Value

raster, array or native raster image

See Also

Other conversion functions: magick_to_nr(), matrix_to_nr(), nrs_to_gif(), nrs_to_mp4()

Examples

```
nr <- nr_new(12, 8, 'hotpink')
nr_to_raster(nr)

nr_to_array(nr)
```

nr_transpose *Transpose*

Description

Transpose

Usage

```
nr_transpose(nr)
```

Arguments

nr	native raster image

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other transformation functions: nr_fliph(), nr_flipv(), nr_rotate()

Examples

```
nr <- nr_new(20, 10, 'hotpink')
dim(nr)
nr_transpose(nr)
dim(nr)
```

nr_tri_mesh *Draw multiple triangles from mesh data*

Description

Mesh data is a similar format to the mesh3d class of data defined in the rgl package.

Usage

```
nr_tri_mesh(nr, vertices, indices, color, tris = "all", use_alpha = TRUE)

nr_tri_coords(nr, coords, color, tris = "all", use_alpha = TRUE)
```

Arguments

nr	native raster image
vertices	Wide matrix of vertex coordinates where the first two rows are x and y coordinates. Extra coordinates are ignored.
indices	3xN wide integer matrix of index information. Each column holds the 3 indices which indicate the column index of the triangle vertices defined in the vertices argument.
color	color specification. Single color or one color per tri
tris	Which triangles should be drawn? Valid options: 'all' (default), 'ccw', 'cw'. The options 'ccw' and 'cw' limit plotting to those triangles where the order of the vertices are defined in a counter-clockwise or clockwise manner respectively. Note: when considering orientation or triangles remember that the y-axis is defined vertically *down* the screen.
use_alpha	Use alpha channel when drawing? Logical. Default: TRUE
coords	Wide numeric matrix of direct coordinate data for each triangle where the first two rows are x and y coordinates (extra rows are ignored) Each group of 3 columns defines the (x,y) coordinates for one triangle.

Value

Invisibly return the supplied native raster image which was been modified in-place

See Also

Other drawing functions: nr_circle(), nr_polygon(), nr_polyline(), nr_rect()

Examples

```
#' Using direct coordinates
set.seed(1)
w <- 100
h <-  80
nr <- nr_new(w, h)

n_tri <- 10
xs <- runif(n_tri * 3, 0, w - 1)
ys <- runif(n_tri * 3, 0, h - 1)
coords <- rbind(xs, ys)
cols <- rainbow(n_tri)

nr_tri_coords(nr, coords, cols, tris = 'all')
plot(nr)

# Using a standard mesh structure
# i.e. a matrix of vertices, and a matrix of indices
nr <- nr_new(w, h)

# Matrix of coordinates
```

```
xs <- rep(seq(0, w - 1, length.out = 10), 10)
ys <- rep(seq(0, h - 1, length.out = 10), each = 10)
vertices <- rbind(xs, ys)
nr_point(nr, xs, ys)

plot(nr)

# Matrix of indices indicating which vertices make up each triangle
indices <- matrix(sample(length(xs), 3 * n_tri), nrow = 3)

nr_tri_mesh(nr, vertices, indices, cols, tris = 'all')
plot(nr)
```

`plot.nativeRaster` *Plot a native raster image (after first clearning the device)*

Description

Plot a native raster image (after first clearning the device)

Usage

```
## S3 method for class 'nativeRaster'
plot(x, y, ...)
```

Arguments

`x`	native raster image
`y`	ignored
`...`	other arguments passed to `grid::grid.raster()`

Value

Invisibly return the supplied native raster image

Examples

```
nr <- nr_new(200, 100, 'hotpink')
plot(nr)
```

print.nativeRaster *Print method*

Description

Print method

Usage

```
## S3 method for class 'nativeRaster'
print(x, ...)
```

Arguments

x	native raster image
...	ignored

Value

None

Examples

```
nr <- nr_new(500, 400)
print(nr)
```

tileset *A list of nativeraster tiles used for constructing a side-scrolling game.*

Description

The tiles in this list are of a common base size and can be conbined to construct various scenes.

Usage

```
tileset
```

Format

An object of class `list` of length 44.

tileset_config	*A data.frame of configuration information which can be used with* nr_blit_multi() *and the* tileset *data.*

Description

Use this configuration wtih nr_blit_multi()

Usage

```
tileset_config
```

Format

An object of class spec_tbl_df (inherits from tbl_df, tbl, data.frame) with 51 rows and 6 columns.

Index