Package ‘magickGUI’

September 11, 2023

Type Package

Title GUI Tools for Interactive Image Processing with 'magick'

Version 1.3.1

Maintainer Shota Ochi <shotaochi1990@gmail.com>

Description Enables us to use the functions of the package 'magick' interactively.

License GPL-3

Depends R (>= 3.1.2), magick (>= 2.2)

Imports tcltk

Suggests testthat (>= 2.0.0), knitr, rmarkdown

URL https://github.com/ShotaOchi/magickGUI

BugReports https://github.com/ShotaOchi/magickGUI/issues

NeedsCompilation no

SystemRequirements ImageMagick (>= 6.9.5.4)

RoxygenNote 7.2.3

Encoding UTF-8

Author Shota Ochi [aut, cre]

Repository CRAN

Date/Publication 2023-09-11 13:50:02 UTC

R topics documented:

interactive_annotate 2
interactive_blur 3
interactive_canny 5
interactive_charcoal 6
interactive_composite 7
interactive_crop 8
interactive_despeckle 9
interactive_emboss 10

interactive_fill 11
interactive_implode 12
interactive_modulate 13
interactive_motion_blur 14
interactive_oilpaint 15
interactive_quantize 16
interactive_reducenoise 17
interactive_shade 18
interactive_threshold 19
magickGUI 20

Index **21**

interactive_annotate *interactive annotation*

Description

Using image_annotate of 'magick' interactively. location, degrees, size, weight, and kerning are parameters of image_annotate. See reference manual of 'magick' for detail.

Usage

```
interactive_annotate(
  image,
  text,
  gravity = "northwest",
  font = "",
  style = "normal",
  decoration = NULL,
  color = NULL,
  strokecolor = NULL,
  boxcolor = NULL,
  range_max_size = 1000,
  range_max_weight = 850,
  range_max_kerning = 300,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image`	a magick image object
`text`	character vector of length equal to 'image' or length 1
`gravity`	string with gravity value from gravity_types.
`font`	string with font family such as "sans", "mono", "serif", "Times", "Helvetica", "Trebuchet", "Georgia", "Palatino" or "Comic Sans".

`style` value of style_types for example "italic"

`decoration` value of decoration_types for example "underline"

`color` a valid color string such as "navyblue" or "#000080". Use "none" for transparency.

`strokecolor` a color string adds a stroke (border around the text)

`boxcolor` a color string for background color that annotation text is rendered on.

`range_max_size` define maximum of size in slider. must be positive.

`range_max_weight`
define maximum of weight in slider. must be positive.

`range_max_kerning`
define maximum of kerning in slider. must be positive.

`resolution` resolution of slider

`return_param` If return_param is TRUE, returns a list of values of location, degrees, size, weight, and kerning. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or a list of values of location, degrees, size, weight, and kerning

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_annotate(wizard, "hello")
}
```

`interactive_blur` *interactive blurring*

Description

Using image_blur of 'magick' interactively. radius and sigma are parameters of image_blur. See reference manual of 'magick' for detail.

Usage

```
interactive_blur(
  image,
  range_max_radius = 5,
  range_max_sigma = 5,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image` a magick image object

`range_max_radius`
define maximum in slider of radius. must be positive.

`range_max_sigma`
define maximum in slider of sigma. must be positive.

`resolution` resolution of slider

`return_param` If return_param is TRUE, returns values of radius and sigma. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or values of radius and sigma

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_blur(wizard)
}
```

`interactive_canny` *interactive canny edge detection*

Description

Using image_canny of 'magick' interactively. radius, sigma, lower%, and upper% are parameters of image_canny. See reference manual of 'magick' for detail.

Usage

```
interactive_canny(
  image,
  range_max_radius = 30,
  range_max_sigma = 2,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image`	a magick image object
`range_max_radius`	define maximum in slider of radius. must be positive.
`range_max_sigma`	define maximum in slider of sigma. must be positive.
`resolution`	resolution of slider
`return_param`	If return_param is TRUE, returns values of radius, sigma, lower%, and upper% represented in the format of 'magick'. If return_param is FALSE, returns a magick image object.
`scale`	geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or values of radius, sigma, lower%, and upper% represented in the format of 'magick'

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_canny(wizard)
}
```

---

interactive_charcoal *interactive charcoal filtering*

Description

Using image_charcoal of 'magick' interactively. radius and sigma are parameters of image_charcoal. See reference manual of 'magick' for detail.

Usage

```
interactive_charcoal(
  image,
  range_max_radius = 5,
  range_max_sigma = 5,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image`	a magick image object
`range_max_radius`	define maximum in slider of radius. must be positive.
`range_max_sigma`	define maximum in slider of sigma. must be positive.
`resolution`	resolution of slider
`return_param`	If return_param is TRUE, returns values of radius and sigma. If return_param is FALSE, returns a magick image object.
`scale`	geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or values of radius and sigma

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_charcoal(wizard)
}
```

`interactive_composite` *interactive image compositing*

Description

Using image_composite of ’magick’ interactively. offset is a parameter of image_composite. see reference manual of ’magick’ for detail.

Usage

```
interactive_composite(
  image,
  composite_image,
  operator = "atop",
  compose_args = "",
  resolution = 1,
  return_param = FALSE,
  scale
)
```

Arguments

`image` a magick image object

`composite_image`
composition image

`operator` string with a composite operator

`compose_args` additional arguments needed for some composite operations

`resolution` resolution of slider

`return_param` If return_param is TRUE, returns values of offset. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

magick a image object or values of offset

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_composite(wizard, rose)
}
```

interactive_crop *interactive cropping*

Description

Using image_crop of 'magick' interactively. geometry is a parameter of image_crop. See reference manual of 'magick' for detail.

Usage

```
interactive_crop(image, color = "white", return_param = FALSE, scale)
```

Arguments

`image`	a magick image object
`color`	color of background. a valid color string such as "navyblue" or "#000080". "none" is not allowed.
`return_param`	If return_param is TRUE, returns a value of geometry. If return_param is FALSE, returns a magick image object.
`scale`	geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or a value of geometry.

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_crop(wizard)
}
```

interactive_despeckle *interactive despeckling*

Description

Using image_despeckle of 'magick' interactively. times is a parameter of image_despeckle. See reference manual of 'magick' for detail.

Usage

```
interactive_despeckle(
  image,
  range_max = 50,
  resolution = 1,
  return_param = FALSE,
  scale
)
```

Arguments

`image` a magick image object

`range_max` define maximum in slider. must be positive.

`resolution` resolution of slider

`return_param` If return_param is TRUE, returns value of times. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or value of times

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_despeckle(wizard)
}
```

---

interactive_emboss *interactive embossing*

Description

Using image_emboss of 'magick' interactively. radius and sigma are parameters of image_emboss. See reference manual of 'magick' for detail.

Usage

```
interactive_emboss(
  image,
  range_max_radius = 5,
  range_max_sigma = 5,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image`	a magick image object
`range_max_radius`	define maximum in slider of radius. must be positive.
`range_max_sigma`	define maximum in slider of sigma. must be positive.
`resolution`	resolution of slider
`return_param`	If return_param is TRUE, returns values of radius and sigma. If return_param is FALSE, returns a magick image object.
`scale`	geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or values of radius and sigma

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_emboss(wizard)
}
```

interactive_fill *interactive filling*

Description

Using image_fill of 'magick' interactively. point and fuzz are parameters of image_fill. See reference manual of 'magick' for detail.

Usage

```
interactive_fill(
  image,
  color,
  refcolor = NULL,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image` a magick image object

`color` a valid color string such as "navyblue" or "#000080". Use "none" for transparency.

`refcolor` if set, fuzz color distance will be measured against this color, not the color of the starting point. Any color (within fuzz color distance of the given refcolor), connected to starting point will be replaced with the color. If the pixel at the starting point does not itself match the given refcolor (according to fuzz) then no action will be taken.

`resolution` resolution of slider of fuzz

`return_param` If return_param is TRUE, returns a list values of point and fuzz. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or a list of values of point and fuzz

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_fill(wizard, "black")
}
```

interactive_implode *interactive imploding*

Description

Using image_implode of 'magick' interactively. factor is a parameter of image_implode. See reference manual of 'magick' for detail.

Usage

```
interactive_implode(
  image,
  range_max = 1,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image` a magick image object

`range_max` define maximum in slider. must be positive.

`resolution` resolution of slider

`return_param` If return_param is TRUE, returns value of factor. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or value of factor

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_implode(wizard)
}
```

interactive_modulate *interactive modulating*

Description

Using image_modulate of 'magick' interactively. brightness and saturation and hue are parameters of image_modulate. See reference manual of 'magick' for detail.

Usage

```
interactive_modulate(
  image,
  range_max_brightness = 200,
  range_max_saturation = 200,
  range_max_hue = 200,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image` a magick image object

`range_max_brightness`
define maximum in slider of brightness. must be positive.

`range_max_saturation`
define maximum in slider of saturation. must be positive.

`range_max_hue` define maximum in slider of hue. must be positive.

`resolution` resolution of slider

`return_param` If return_param is TRUE, returns values of brightness and saturation and hue. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or values of brightness, saturation, and hue

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_modulate(wizard)
}
```

interactive_motion_blur

interactive motion blurring

Description

Using image_motion_blur of 'magick' interactively. radius and sigma and angle are parameters of image_motion_blur. See reference manual of 'magick' for detail.

Usage

```
interactive_motion_blur(
  image,
  range_max_radius = 100,
  range_max_sigma = 100,
  range_max_angle = 360,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image` a magick image object

`range_max_radius` define maximum in slider of radius. must be positive.

`range_max_sigma` define maximum in slider of sigma. must be positive.

`range_max_angle` define maximum in slider of angle. must be positive.

`resolution` resolution of slider

`return_param` If return_param is TRUE, returns values of radius and sigma and angle. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or values of radius, sigma, and angle

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_motion_blur(wizard)
}
```

`interactive_oilpaint` *interactive oil painting*

Description

Using image_oilpaint of 'magick' interactively. radius is a parameter of image_oilpaint. See reference manual of 'magick' for detail.

Usage

```
interactive_oilpaint(
  image,
  range_max = 10,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image` a magick image object

`range_max` define maximum in slider. must be positive.

`resolution` resolution of slider

`return_param` If return_param is TRUE, returns value of radius. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or value of radius

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_oilpaint(wizard)
}
```

interactive_quantize *interactive quantization*

Description

Using image_quantize of 'magick' interactively. max is a parameter of image_quantize. See reference manual of 'magick' for detail.

Usage

```
interactive_quantize(
  image,
  colorspace = "rgb",
  dither = NULL,
  treedepth = NULL,
  range_max = 256,
  resolution = 1,
  return_param = FALSE,
  scale
)
```

Arguments

`image`	a magick image object
`colorspace`	specify colorspace. for example, "rgb", "gray", or "cmyk".
`dither`	apply Floyd/Steinberg error diffusion to the image
`treedepth`	depth of the quantization color classification tree
`range_max`	define maximum in slider. must be positive.
`resolution`	resolution of slider
`return_param`	If return_param is TRUE, returns value of max. If return_param is FALSE, returns a magick image object.
`scale`	geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or value of max

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_quantize(wizard)
}
```

---

interactive_reducenoise *interactive denoising*

Description

Using image_reducenoise of 'magick' interactively. radius is a parameter of image_reducenoise. See reference manual of 'magick' for detail.

Usage

```
interactive_reducenoise(
  image,
  range_max = 30,
  resolution = 1,
  return_param = FALSE,
  scale
)
```

Arguments

`image`	a magick image object
`range_max`	define maximum in slider. must be positive.
`resolution`	resolution of slider
`return_param`	If return_param is TRUE, returns value of radius. If return_param is FALSE, returns a magick image object.
`scale`	geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or value of radius

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_reducenoise(wizard)
}
```

interactive_shade *interactive shading*

Description

Using image_shade of 'magick' interactively. azimuth and elevation are parameters of image_shade. See reference manual of 'magick' for detail.

Usage

```
interactive_shade(
  image,
  color = FALSE,
  range_max_azimuth,
  range_min_azimuth,
  range_max_elevation,
  range_min_elevation,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

image	a magick image object
color	Set to true to shade the red, green, and blue components of the image
range_max_azimuth	define maximum in slider of azimuth
range_min_azimuth	define maximum in slider of azimuth

`range_max_elevation`
define maximum in slider of elevation

`range_min_elevation`
define maximum in slider of elevation

`resolution` resolution of slider

`return_param` If return_param is TRUE, returns values of azimuth and elevation. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or values of azimuth and elevation

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_shade(wizard)
}
```

`interactive_threshold` *interactive thresholding*

Description

Using image_threshold of 'magick' interactively. threshold is a parameter of image_threshold. See reference manual of 'magick' for detail.

Usage

```
interactive_threshold(
  image,
  type = c("black", "white"),
  channel = NULL,
  resolution = 0.1,
  return_param = FALSE,
  scale
)
```

Arguments

`image` a magick image object

`type` type of thresholding, either one of lat, black or white

`channel` a value specifying which channel(s) to set

`resolution` resolution of slider

`return_param` If return_param is TRUE, returns threshold value. If return_param is FALSE, returns a magick image object.

`scale` geometry to be passed to image_scale function of magick package. image is scaled just for preview and result image is not scaled if scale is given.

Value

a magick image object or threshold value

Author(s)

Shota Ochi

Examples

```
if (interactive())
{
  interactive_threshold(wizard)
}
```

`magickGUI` *magickGUI: GUI tools for interactive image processing with 'magick'*

Description

magickGUI enables us to use the functions of the package 'magick' interactively.

Author(s)

Maintainer: Shota Ochi <`shotaochi1990@gmail.com`>

See Also

Useful links:

- `https://github.com/ShotaOchi/magickGUI`
- Report bugs at `https://github.com/ShotaOchi/magickGUI/issues`

Index

interactive_annotate, 2
interactive_blur, 3
interactive_canny, 5
interactive_charcoal, 6
interactive_composite, 7
interactive_crop, 8
interactive_despeckle, 9
interactive_emboss, 10
interactive_fill, 11
interactive_implode, 12
interactive_modulate, 13
interactive_motion_blur, 14
interactive_oilpaint, 15
interactive_quantize, 16
interactive_reducenoise, 17
interactive_shade, 18
interactive_threshold, 19

magickGUI, 20
magickGUI-package *(*magickGUI*)*, 20