

Package ‘imagerExtra’

July 3, 2018

Type Package

Title Extra Image Processing Library Based on 'imager'

Version 1.1.1

Date 2018-07-03

Maintainer Shota Ochi <shotaochi1990@gmail.com>

Description Providing several advanced functions for image processing based on the package 'imager'.

License GPL (>= 3)

Depends R (>= 2.10.0), imager (>= 0.40.2)

Imports Rcpp (>= 0.12.14), dtt, magrittr

Suggests testthat (>= 2.0.0), knitr, rmarkdown

URL <https://github.com/ShotaOchi/imagerExtra>,
<https://shotaochi.github.io/>

BugReports <https://github.com/ShotaOchi/imagerExtra/issues>

LinkingTo Rcpp

LazyData true

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation yes

Author Shota Ochi [aut, cre],
Guoshen Yu [ctb, cph],
Guillermo Sapiro [ctb, cph],
Catalina Sbert [ctb, cph],
Image Processing On Line [cph]

Repository CRAN

Date/Publication 2018-07-02 22:00:24 UTC

R topics documented:

BalanceSimplest	2
DenoiseDCT	3
dogs	4
EqualizePiecewise	4
GetHue	5
Grayscale	6
imagerExtra	6
RestoreHue	7
SPE	7
ThresholdTriclass	8

Index	10
--------------	-----------

BalanceSimplest	<i>Balance color of image by Simplest Color Balance</i>
-----------------	---

Description

Balance color of image by Simplest Color Balance

Usage

```
BalanceSimplest(im, sleft, sright, range = c(0, 255))
```

Arguments

im	a grayscale image of class cimg
sleft	left saturation percentage. sleft can be specified by numeric or string, e.g. 1 and "1%". note that sleft is a percentile.
sright	right saturation percentage. sright can be specified by numeric or string. note that sright is a percentile.
range	this function assumes that the range of pixel values of of input image is [0,255] by default. you may prefer [0,1].

Value

a grayscale image of class cimg

Author(s)

Shota Ochi

References

Nicolas Limare, Jose-Luis Lisani, Jean-Michel Morel, Ana Belen Petro, and Catalina Sbert, Simplest Color Balance, Image Processing On Line, 1 (2011), pp. 297-315. <https://doi.org/10.5201/ipol.2011.11mps-scb>

Examples

```
dev.new()
par(mfcol = c(1,2))
boats_g <- grayscale(boats)
plot(boats_g, main = "Original")
BalanceSimplest(boats_g, 1, 1) %>% plot(., main = "Simplest Color Balance")
```

DenoiseDCT	<i>denoise image by DCT denoising</i>
------------	---------------------------------------

Description

denoise image by DCT denoising

Usage

```
DenoiseDCT(im, sdn, flag_dct16x16 = FALSE)
```

Arguments

im	a grayscale image of class cimg
sdn	standard deviation of Gaussian white noise
flag_dct16x16	flag_dct16x16 determines the size of patches. if TRUE, the size of patches is 16x16. if FALSE, the size of patches is 8x8.

Value

a grayscale image of class cimg

Author(s)

Shota Ochi

References

Guoshen Yu, and Guillermo Sapiro, DCT Image Denoising: a Simple and Effective Image Denoising Algorithm, Image Processing On Line, 1 (2011), pp. 292-296. <https://doi.org/10.5201/ipol.2011.js-dct>

Examples

```
dev.new()
par(mfcol = c(1,2))
boats_g <- grayscale(boats)
boats_noisy <- imnoise(dim = dim(boats_g), sd = 0.05) + boats_g
plot(boats_noisy, main = "Noisy Boats")
DenoiseDCT(boats_g, 0.05) %>% plot(., main = "Denoised Boats")
```

 dogs

Photograph of a dog from GAHAG

Description

This photograph was downloaded from <http://gahag.net/img/201603/03s/gahag-0062116383-1.jpg>. Its size was reduced by half to speed up loading and save space.

Usage

dogs

Format

an image of class cimg

Source

<http://gahag.net/img/201603/03s/gahag-0062116383-1.jpg>

 EqualizePiecewise

Enhance contrast of image by Piecewise Affine Equalization

Description

Enhance contrast of image by Piecewise Affine Equalization

Usage

```
EqualizePiecewise(im, N, smax = 255, smin = 0, range = c(0, 255))
```

Arguments

im	a grayscale image of class cimg
N	number of subintervals of partition. N controls how the input gray levels will be mapped in the output image. if N is large, Piecewise Affine Equalization and Histogram Equalization are very similar.
smax	maximum value of slopes. if smax is small, contrast enhancement is suppressed.
smin	minimum value of slopes. if smin is large, contrast enhancement is propelled, and saturations occur excessively.
range	range of the pixel values of image. this function assumes that the range of pixel values of of an input image is [0,255] by default. you may prefer [0,1]. if you change range, you should change smax. one example is this (smax = range[2] - range[1]).

Value

a grayscale image of class cimg

Author(s)

Shota Ochi

References

Jose-Luis Lisani, Ana-Belen Petro, and Catalina Sbert, Color and Contrast Enhancement by Controlled Piecewise Affine Histogram Equalization, Image Processing On Line, 2 (2012), pp. 243-265.
<https://doi.org/10.5201/ipol.2012.lps-pae>

Examples

```
dev.new()  
par(mfcol = c(1,2))  
boats_g <- grayscale(boats)  
plot(boats_g, main = "Original")  
EqualizePiecewise(boats_g, 10) %>% plot(., main = "Piecewise Affine Equalization")
```

GetHue

store hue of color image

Description

store hue of color image

Usage

```
GetHue(imcol)
```

Arguments

imcol a color image of class cimg

Value

a color image of class cimg

Author(s)

Shota Ochi

Examples

```
GetHue(boats)
```

Grayscale *compute average of RGB channels*

Description

compute average of RGB channels

Usage

```
Grayscale(imcol)
```

Arguments

imcol a color image of class cimg

Value

a grayscale image of class cimg

Author(s)

Shota Ochi

Examples

```
Grayscale(boats) %>% plot
```

imagerExtra *imagerExtra: Extra Image Processing Library Based on Imager*

Description

imagerExtra is built on imager. imager by Simon Simon Barthelme provides an interface with CImg that is a C++ library for image processing. imager makes functions of CImg accessible from R and adds many utilities for accessing and working with image data from R. imagerExtra provides several advanced functions based on imager.

RestoreHue	<i>restore hue of color image</i>
------------	-----------------------------------

Description

restore hue of color image

Usage

```
RestoreHue(im, hueim)
```

Arguments

im	a grayscale image of class cimg
hueim	a color image of class cimg

Value

a color image of class cimg

Author(s)

Shota Ochi

Examples

```
g <- Grayscale(boats)
hue <- GetHue(boats)
layout(matrix(1:2, 1, 2))
plot(g, main = "Original")
RestoreHue(g, hue) %>% plot(main="Resotred")
```

SPE	<i>Correct inhomogeneous background of image by solving Screened Poisson Equation</i>
-----	---

Description

Correct inhomogeneous background of image by solving Screened Poisson Equation

Usage

```
SPE(im, lamda, s = 0.1, range = c(0, 255))
```

Arguments

im	a grayscale image of class cimg
lamda	this function corrects inhomogeneous background while preserving image details. lamda controls the trade-off. when lamda is too large, this function acts as an edge detector.
s	saturation percentage. this function uses <code>BalanceSimplest</code> . s is used as both sleft and sright. that's why s can not be over 50%.
range	this function assumes that the range of pixel values of of an input image is [0,255] by default. you may prefer [0,1].

Value

a grayscale image of class cimg

Author(s)

Shota Ochi

References

Jean-Michel Morel, Ana-Belen Petro, and Catalina Sbert, Screened Poisson Equation for Image Contrast Enhancement, Image Processing On Line, 4 (2014), pp. 16-29. <https://doi.org/10.5201/ipol.2014.84>

Examples

```
dev.new()
par(mfcol = c(1,2))
boats_g <- grayscale(boats)
plot(boats_g, main = "Original")
SPE(boats_g, 0.1) %>% plot(main = "Screened Poisson Equation")
```

ThresholdTriclass *Iterative Triclass Thresholding*

Description

compute threshold value by Iterative Triclass Threshold Technique

Usage

```
ThresholdTriclass(im, stopval = 0.1, repeatnum, intervalnumber = 1000,
  returnvalue = FALSE)
```


Arguments

<code>im</code>	a grayscale image of class <code>cimg</code>
<code>stopval</code>	value to determine whether stop iteration of triclass thresholding or not. Note that if <code>repeat</code> is set, <code>stop</code> is ignored.
<code>repeatnum</code>	number of repetition of triclass thresholding
<code>intervalnumber</code>	interval number of histogram
<code>returnvalue</code>	if <code>returnvalue</code> is <code>TRUE</code> , <code>ThresholdTriclass</code> returns threshold value. if <code>FALSE</code> , <code>ThresholdTriclass</code> returns <code>pixset</code> .

Value

`pixset` or threshold value

Author(s)

Shota Ochi

References

Cai HM, Yang Z, Cao XH, Xia WM, Xu XY (2014). A New Iterative Triclass Thresholding Technique in Image Segmentation. IEEE TRANSACTIONS ON IMAGE PROCESSING.

Examples

```
g <- grayscale(boats)
layout(matrix(1:4, 2, 2))
plot(boats, main = "Original")
plot(g, main = "Grayscale")
threshold(g) %>% plot(main = "Otsu")
ThresholdTriclass(g) %>% plot(main = "Triclass")
```

Index

*Topic **datasets**

dogs, [4](#)

BalanceSimplest, [2](#), [8](#)

DenoiseDCT, [3](#)

dogs, [4](#)

EqualizePiecewise, [4](#)

GetHue, [5](#)

Grayscale, [6](#)

imagerExtra, [6](#)

imagerExtra-package (imagerExtra), [6](#)

RestoreHue, [7](#)

SPE, [7](#)

ThresholdTriclass, [8](#)