Package 'adimpro'

September 6, 2023

Version 0.9.6

Date 2023-09-05

Title Adaptive Smoothing of Digital Images

Author Karsten Tabelow <tabelow@wias-berlin.de>,
Joerg Polzehl <polzehl@wias-berlin.de>

Maintainer Karsten Tabelow <tabelow@wias-berlin.de>

Depends R (>= 3.2.0)

Imports awsMethods (>= 1.1-1), grDevices, methods

SystemRequirements Image Magick (for reading non PPM format), dcraw (for reading RAW images).

Description Implements tools for manipulation of digital images and the Propagation Separation approach by Polzehl and Spokoiny (2006) <DOI:10.1007/s00440-005-0464-1> for smoothing digital images, see Polzehl and Tabelow (2007) <DOI:10.18637/jss.v019.i01>.

License GPL (>= 2)

URL https://www.wias-berlin.de/software/imaging/

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-09-06 12:40:02 UTC

R topics documented:

`adimpro` *I/O Functions*

Description

Create image objects of class "adimpro" from arrays, RAW-format files and other image formats.

Usage

```
read.raw(filename, type="PPM",
         wb="CAMERA",cspace="Adobe",interp="Bilinear",maxrange=TRUE,
         rm.ppm=TRUE, compress=TRUE)
read.image(filename, compress=TRUE)
make.image(x,compress=TRUE, gammatype="None", whitep = "D65",
         cspace="Adobe", scale="Original",xmode="RGB")
```

Arguments

`filename`	file name
`x`	Array or matrix containing RGB or greyscale values in the range (0,1) or (0,65535).
`type`	option settings for dcraw. default "PPM". `type="png"` allows to read greyvalue png images as RAW-data (used as internal solution to store RAW information)
`wb`	white balance. default "CAMERA"
`cspace`	defines the output color space, default "sRGB" (sRGB D65), alternatives are "RAW" (Camera specific), "Adobe" (Adobe 1998 D65), "wGamut" (Wide Gamut D65), "kodak" (Kodak ProPhoto D65) and "XYZ", see manpages of dcraw.

interp	defines the interpolation method, default "Bilinear", Alternatives are "VNG", "AHD", "FourC" (Four color interpolation) and "Halfsize", see manpages of dcraw. "VNG" seems to provide the smallest spatial correlations.
maxrange	If TRUE increase range of values to maximum.
rm.ppm	remove intermediate tmp file? default TRUE
gammatype	character, determines the type of gamma correction within the image. "ITU" stands for ITU-R BT.709-3 as e.g. used by dcraw. Alternatives recognized within the package are "None", "sRGB" and "CIE" (CIE L*). Please specify if you know that your image is not gamma corrected using ITU-R BT.709-3.
whitep	White point in xyY space. Can be given as one of (character) c("A","B","C","E","D50","D55","D65", or as a two element numeric vector of chromatic xy coordinates. "D65" corresponds to the default white point of "sRGB" and "Adobe" RGB-spaces.
compress	logical, determines if image data are stored in raw-format.
scale	"Original" scales to (0,max(img$img)) if min(img$img)<0, otherwise keeps the original scale. "Maxcontrast" scales each channel to maximum contrast
xmode	xmode determines how to interpret the values in x if length(dim(x))==3. Implemented are xmode="RGB" (default) and xmode="HSI"

Details

If ImageMagick is available on the system, read.image reads any of the following image file formats: c("tif", "tiff", "pgm", "ppm", "png", "pnm", "gif", "jpg", "jpeg") converts it into a temporary "pgm" or "ppm" file. This file is removed after reading the image. If ImageMagick is not available only "pgm", "ppm" and "pnm" formats can be processed.

If dcraw is available on the system, read.raw reads many RAW formats. type sets options to dcraw: "PPM" sets "-4", "RAW" sets "-4 -d", "HALFSIZE" sets "-h", "INFO" sets "-i -v". wb indicates, which white balance should be used: "NONE", "AUTO", "CAMERA".

Functions read.raw(file,type="RAW") and read.image(file) provide identical results on pngmages. If the result is a color, greyvalued or RAW image depends on the contend of the comment associated with the png-image.

make.image converts an appropriate 2 or 3 dimensional array to an image object of class "adimpro".

Value

object of class "adimpro" containing the image. The object has the following components:

img	array containing the color values in the color space specified by value$type.
type	the color space.
depth	color depth, here "16bit".
dim	vector of length 2 containing the number of pixel in horizontal and vertival direction.
file	the argument file identifying the image.
cspace	the type of rgb space used, as specified by cspace.
interp	interpolation applied by dcraw, as specified by interp.

gamma	has a gamma correction been applied, here FALSE for read.raw and TRUE for read.image
gammatype	type of gamma correction read.image.
wb	type of white balance, as specified by wb.
compressed	image data are stored as raw-vector (TRUE) or array of integers (FALSE).

Note

The function read.raw requires dcraw to be installed.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

read.image

Examples

```
## Not run: demo(io)
```

---

adimpro.options *Set parameters for graphical display*

Description

On systems capabale of X11 the function sets the X11-type (preferably "Xlib"). It also sets a default size for graphical displays opended by functions from the package.

Usage

```
adimpro.options(xsize = NULL, ysize = NULL)
```

Arguments

xsize	display width in pt
ysize	display height in pt

Details

The function assignes the specified values to as a list to the variable name ".adimpro". This variable is, if it exists, evaluated by several other functions.

On some systems the default X11.options()$type "cairo" leads to significant slower image display. You may try to use X11.options(type=''Xlib'') instead. To automatically choose this option set the system environment variable R_X11type (setenv R_X11type Xlib or export R_X11type=Xlib) before loading the package.

Value

returns `invisible(NULL)`

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

`adjust.image` *Image Processing*

Description

Color space transformations, change of white balance and exposure, gamma correction and histogram equalization.

Usage

```
adjust.image(img, gammatype=NULL, cspace = NULL, whitep = NULL,
                  temp = NULL, black=0, exposure = 1, kind = "Bradford",
                  alg = 1, compress= TRUE)
```

Arguments

`img`	image object, class"adimpro".
`gammatype`	character, determines the type of gamma correction within the image. "ITU" stands for ITU-R BT.709-3 as e.g. used by `dcraw`. Alternatives recognized within the package are "None", "sRGB" and "CIE" (CIE L*). `NULL` keeps the actual setting. `gammatype="histogram"` forces histogram equalization based on the corresponding greyvalue image.
`cspace`	defines the output color space, default "sRGB" (sRGB D65), alternatives are "Adobe" (Adobe 1998 D65), "wGamut" (Wide Gamut D65), "kodak" (Kodak ProPhoto D65) "xyz", "yuv", "yiq" and "hsi". `NULL` keeps the actual setting.
`whitep`	White point in `xyY` space. Can be given as one of (character) `c("A","B","C","E","D50","D55","D65",` or as a two element numeric vector of chromatic `xy` coordinates. `"D65"` corresponds to the default white point of `"sRGB"` and `"Adobe"` RGB-spaces. `NULL` keeps the actual setting.
`temp`	Color temperature. Can be used to specify chromatic `xy` coordinates of the whitepoint. Only used if `is.null(whitep)`.
`black`	Adjustment for black color. Color values with luminance <= black will be assigned to black in RGB. Adjustment ist done in xyY space.
`exposure`	Multiplicative factor for all color channels (in xyz or rgb spaces). Applied in linear color space, i.e. if the image is gamma corrected the gamma correction is reversed first.
`kind`	Algorithm for chromatic adaptation. Alternatives are `"Bradford"`, `"VonKries"` and `"XYZscaling"`

alg	determines the approximation for the gamma correction. Select 1 for fastest computation and 3 for maximum accuracy, or 2 for a compromise.
compress	logical, determines if image data are stored in raw-format.

Details

This function adjusts color channels and applies gamma correction (if applicable).

If color.par$red or color.par$blue or color.par$brightness differ from 1.0 the corresponding channels are multiplied with the provided values. Saturated values are set to 1.

If img$gamma==FALSE, perform gamma correction with color.par$ga and color.par$bp. alg chooses between three different computing algorithms (approximations) with increasing computation time and precision (alg is 1,2, or 3).

Value

Adjusted image object of class "adimpro".

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

show.image, write.image

Examples

```
## Not run: demo(color)
```

---

awsimage *Propagation-Separation approach for smoothing of 2D images*

Description

This functions implement the Propagation-Separation approach (local constant and local polynomial model) for smoothing images. Function awsaniso uses anisotropic location weights. This is done by evaluating local gradient estimates obtained from the actual estimated color values.

Usage

```
awsimage(object, hmax=4, aws=TRUE, varmodel=NULL, ladjust=1.25,
         mask=NULL, xind = NULL, yind = NULL,
         wghts=c(1,1,1,1), scorr=TRUE,
         lkern="Plateau", plateau=NULL, homogen=TRUE, earlystop=TRUE,
         demo=FALSE, graph=FALSE,
         max.pixel=4.e2, clip = FALSE, compress=TRUE)
awspimage(object, hmax=12, aws=TRUE, degree=1, varmodel = NULL,
```

```
          ladjust=1.0, xind = NULL, yind = NULL,
          wghts=c(1,1,1,1), scorr= TRUE,
          lkern="Plateau", plateau=NULL, homogen=TRUE, earlystop=TRUE,
          demo=FALSE, graph=FALSE,
          max.pixel= 4.e2, clip = FALSE, compress=TRUE)
awsaniso(object, hmax = 4, g = 3, rho = 0, aws = TRUE, varmodel = NULL,
          ladjust = 1, xind = NULL, yind = NULL, wghts = c(1, 1, 1, 1),
          scorr = TRUE, lkern = "Triangle", demo = FALSE, graph = FALSE,
          satexp = 0.25, max.pixel = 400, clip = FALSE, compress = TRUE)
```

Arguments

object	Image object, class "adimpro", as from read.image, read.raw, or make.image.
hmax	Maximum bandwidth to use in the iteration procedure.
g	Bandwidth for anisotropic smoothing gradient estimates, preferably $g \geq 3$ for images with line type texture and small $g \approx 1$ for improving edges between homogeneous regions (function awsaniso only).
rho	Regularization parameter for anisotropic smoothing gradient estimates, preferably $\rho = 0$ for images with line type texture and large $\rho \approx 3$ for improving edges between homogeneous regions. (function awsaniso only)
aws	(logical). If TRUE the propagation - separation (PS) approach from Polzehl and Spokoiny (2006) is used. aws=FALSE turns off the statistical penalty resulting in a nonadaptive kernel estimate using a kernel with bandwidth hmax.
degree	Degree of the local polynomial model for awspimage. 0, 1, or 2 only.
varmodel	varmodel specifies how variances are to be estimated. This can be a homogeneous variance estimate (varmodel="None") assuming uncorrelated errors (both spatial and between channels). Alternatives are an adaptive homogeneous or linear (function of the mean) variance estimate that depends on estimated correlations and on residuals from the last iteration step. The default varmodel=NULL corresponds to varmodel == "Linear" if img$gamma==FALSE and varmodel == "Constant" otherwise.
ladjust	adjustment factor for lambda (>=1). Default values for lambda are selected for Gaussian distributions and default settings of parameters lkern and plateau. Skewed or heavy tailed distributions may require slightly larger values for lambda to meet the propagation condition. ladjust allows to increase lambda in such situations.
mask	logical array of the same size as the image or NULL (default). Smoothing is restricted to the smallest rectangle including all pixel where mask==TRUE and restricts computations to these pixel. This need not be a connected area (Typical usage: smooth all bright regions)! Only used if is.null(xind) && is.null(yind). Inactive if mask==NULL. Can only be used if varmodel="None".
xind, yind	Restrict smoothing to rectangular area defined by pixel indices xind,yind in x- and y-direction. Full range if NULL (default).
wghts	allows to weight the information from different (up to 4) color channels. The weights are used in the statistical penalty of the PS-procedure. Note that lambda-values are selected for wghts==c(1,1,1,1), please use parameter ladjust to set an appropriate value.

scorr	(logical). Specifies whether spatial correlation is to be estimated. Defaults to TRUE. Is set to FALSE if mask is not NULL.
lkern	Specifies the location kernel. Defaults to "Triangle", other choices are "Quadratic", "Cubic" and "Uniform". The use of "Triangle" corresponds to the Epanechnicov kernel nonparametric kernel regression.
plateau	Extension of the plateau in the statistical kernel. Can take values from (0,1), defaults to 0.25.
homogen	If TRUE the algorithm determines, in each design point i, a circle of maximum radius, such that the statistical penalty s_{ij} for all points j within the circle is less than the value specified in plateau. In subsequent iteration steps the statistical penalty for such points is set to zero. This is only used if plateau>0. This results in more stable intermediate estimates and in a smoother reconstruction. homogen=TRUE leads to increased memory requirements.
earlystop	If TRUE the algorithm determines, in each design point i, a circle of minimal radius, such that the circle includes all point j with positive weights w_{ij}. if this radius is considerably smaller than the actual bandwidth then the estimate in point i is fixed. This should considarably reduce computing time in case of large hmax.earlystop=TRUE slightly increases memory requirements.
demo	(logical). If demo=TRUE the function pauses after each iteration. Defaults to FALSE.
graph	(logical). If graph=TRUE intermediate results are illustrated after each iteration step. Defaults to FALSE.
max.pixel	Maximum dimension of images for display if graph=TRUE. If the true dimension is larger, the images are downscaled for display. See also show.image.
satexp	exponent used for scaling saturation in anisotropy visualization (function awsaniso only)
clip	(logical). If TRUE a clipping region is selected, see clip.image, using the information contained in xind or yind. If both are NULL a clipping region can be defined by left mouse clicks. The image object is reduced to the clipping region before smoothing.
compress	logical, determines if image data are stored in raw-format.

Details

The function implements the Propagation-Separation (PS) approach to nonparametric smoothing (formerly introduced as Adaptive Weights Smoothing) for varying coefficient likelihood (awsimage) and local polynomial (awspimage) models for greyscale and color images.

The distribution of grey (color) values is considered to be Gaussian. Noise can be colored.

The numerical complexity of the procedure is mainly determined by hmax. The number of iterations is 2*log(hmax)/log(1.25). Comlexity in each iteration step is Const*hakt*n with hakt being the actual bandwith in the iteration step and n the number of pixels. hmax determines the maximal possible variance reduction.

All other parameters of the approach only depend on the specified values for skern/lkern and are therefore set internally to meaningful default values.

For a detailed description of the procedure see references below.

The script used to control the values of parameter lambda is stored in directory inst/adjust.

Value

Object of class "adimpro"

img	Contains the reconstructed image.
ni	Contains the sum of weights, i.e. trace(W_i), in all grid points i.
ni0	Contains the maximum sum of weights for an nonadaptive kernel estimate with the same bandwidth.
hmax	Bandwidth used in the last iteration.
call	The arguments of the function call.
varcoef	Estimated coefficients in the variance model for the color channels, if varmodel is "Constant" or "Linear".
wghts	The weights used for the color channels.
scorr	Estimated spatial correlations for each channel, if scorr=TRUE
chcorr	Estimated correlations between color channels, if scorr=TRUE

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

References

Polzehl and Spokoiny (2006). Propagation-Separation Approach for Local Likelihood Estimation. Probability Theory and Related Fields. 3 (135) 335 - 362.

Polzehl and Spokoiny (2005). Structural adaptive smoothing adaptive smoothing by Propagation-Separation-methods. WIAS-Preprint No. 1068.

Polzehl, J. and Tabelow, K. (2007). Adaptive smoothing of digital images, Journal of Statistical Software 19 (1).

See Also

read.image, read.raw, make.image, show.image, clip.image

Examples

```
## Not run: demo(awsimage)
```

awsraw *Smoothing and demosaicing of RAW images*

Description

The function integrates smoothing and demosaicing of RAW image data.

Usage

```
awsraw(object, hmax = 4, aws = TRUE, wb = c(1, 1, 1), cspace = "Adobe",
       ladjust = 1, maxrange=TRUE, lkern = "Triangle", graph = FALSE,
       max.pixel = 400, compress = TRUE)
```

Arguments

object	an object of class adimpro containing the RAW image data. See read.raw for creating such objects.
hmax	maximal bandwidth to use in the smoothing algorithm.
aws	use adaptive weights if aws==TRUE.
wb	Vector containing factors for the three color chanels, allows to change the white balance.
cspace	Color space of the result,
ladjust	Factor for the critical value λ. Defaults to 1, smaller values increase sensitivity but may result in isolated noisy pixel. Values larger than 1 give smoother up to cartoon like results.
maxrange	If TRUE increase range of values to maximum.
lkern	Specifies the location kernel. Defaults to "Triangle", other choices are "Quadratic", "Cubic" and "Uniform". The use of "Triangle" corresponds to the Epanechnicov kernel nonparametric kernel regression.
graph	(logical). If graph=TRUE intermediate results are illustrated after each iteration step. Defaults to FALSE.
max.pixel	Maximum dimension of images for display if graph=TRUE. If the true dimension is larger, the images are downscaled for display. See also show.image.
compress	logical, determines if image data are stored in raw-format.

Details

Adaptive smoothing is performed on the original RAW data, restricting positive weights to pixel corresponding to the same color channel. Noise is assumed to have a variance depending linearly on the mean. Weights are determined by weigthed distances between color vectors. These color vectors are obtained by demosaicing that is applied to the smoothed RAW data after each iteration of the smoothing algorithm. The demosaicing algorithm is a 3D generalized median, see method="Median4" in function develop.raw.

Value

Object of class "adimpro"

img	Contains the reconstructed image.
ni	Contains the sum of weights, i.e. trace(W_i), in all grid points i.
ni0	Contains the maximum sum of weights for an nonadaptive kernel estimate with the same bandwidth.
hmax	Bandwidth used in the last iteration.
call	The arguments of the function call.
varcoef	Estimated coefficients in the linear variance model for the color channels.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

References

Polzehl, J. and Tabelow, K. (2007). Adaptive smoothing of digital images, Journal of Statistical Software 19 (1).

See Also

read.raw,awsimage, make.image, show.image, clip.image

Examples

```
## Not run: demo(raw)
```

---

clip.image *Create an image by clipping*

Description

The function allows to define a clipping region by arguments xind and yind or interactively by mouseclicks. A new image is created by cutting out the clipping region.

Usage

```
clip.image(img, xind = NULL, yind = NULL, compress=NULL, ...)
```

Arguments

img	Object of class "adimpro" as created by read.image, read.raw or make.image.
xind	xind defines the horizontal extension of the clipping region.
yind	yind defines the vertical extension of the clipping region.
compress	image data are stored as raw-vector (TRUE) or array of integers (FALSE). compress=NULL keeps the format used in img.
...	additional arguments to show.image can be passed here.

Details

If both xind==NULL and yind==NULL, show.image is called and the clipping region can be set by two left mouse clicks to opposite corners.

Value

An object of class "adimpro".

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

read.image, read.raw, make.image, awsimage, awspimage

Examples

```
## Not run: demo(manipulate)
```

---

colorspace *Color Space Conversion*

Description

Color space conversion functions for RGB, YUV, YIQ, XYZ, and HSI.

Usage

```
rgb2grey(obj, compress=TRUE)
rgb2hsi(obj)
hsi2rgb(obj, cspace = "Adobe", compress=TRUE)
rgb2yuv(obj)
yuv2rgb(obj, cspace = "Adobe", compress=TRUE)
rgb2yiq(obj)
yiq2rgb(obj, cspace = "Adobe", compress=TRUE)
rgb2xyz(obj)
xyz2rgb(obj, cspace = "Adobe", black= 0, exposure=1, compress=TRUE)
```

Arguments

obj	an object of class "adimpro", obj$type should coincide with the first three letters of the function name, that is obj$img containes color values in the corresponding color space.
cspace	Target color space, one of c("sRGB","Adobe","wGamut","kodak","xyz"). For function xyz2rgb also c("yuv","yiq") are implemented.

exposure	Multiplicative factor for all color channels (in xyz or rgb spaces). Applied in linear color space, i.e. if the image is gamma corrected the gamma correction is reversed first.
black	Adjustment for black color. Color values with luminance <= black will be assigned to black in RGB.
compress	logical, determines if image data are returned in raw-format.

Details

The functions convert an image `obj` from one color space into another.

`rgb2grey` converts from RGB to GREYSCALE.

`rgb2hsi` and `hsi2rgb` convert from RGB to HSI and vice versa.

`rgb2yuv` and `yuv2rgb` convert from RGB to YUV and vice versa.

`rgb2yiq` and `yiq2rgb` convert from RGB to YIQ and vice versa.

`rgb2xyz` and `xyz2rgb` convert from RGB to CIE XYZ and vice versa.

Conversion to `XYZ`, `YIQ`, `YUV` and `HSI` involves an inverse gamma correction if required.

Value

an object of class "adimpro", with `value$type` specifying the color space (last three letters of the function name or 'greyscale' for `rgb2grey`) and `value$img` containing the color values.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

References

Gonzalez, R.C., and Woods, R.E. (2002). Digital Image Processing. Prentice Hall.

Polzehl, J., and Tabelow, K. (2007). Adaptive smoothing of digital images, Journal of Statistical Software 19 (1).

Examples

```
## Not run: demo(color)
```

combine *Pixelwise operations on a pair of images*

Description

The function allows to perform pixelwise operations, specified by a supplied function, on a pair of images.

Usage

```
combine(img1, img2, fun = "+", rescale = TRUE, compress = TRUE, gammatype = "None",
        whitep = "D65", cspace = "Adobe", xmode = "RGB", ...)
```

Arguments

img1	image, object of class adimpro
img2	image, object of class adimpro, need to have the same dimension as img1
fun	A function or primitive of two (or more) arguments specifying the operation. The first argument corresponds to grey/color-values in img1, the second to img2. Auxiliary parameters can passed throug "..."
rescale	logical: if TRUE the resulting image is rescaled to fit into the range of possible grey/color-values, if FALSE values outside the range are truncated.
compress	logical, determines if image data are stored in raw-format.
gammatype	character, determines the type of gamma correction within the image. "ITU" stands for ITU-R BT.709-3 as e.g. used by dcraw. Alternatives recognized within the package are "None", "sRGB" and "CIE" (CIE L*). Please specify if you know that your image is not gamma corrected using ITU-R BT.709-3.
whitep	White point in xyY space. Can be given as one of (character) c("A","B","C","E","D50","D55","D65", or as a two element numeric vector of chromatic xy coordinates. "D65" corresponds to the default white point of "sRGB" and "Adobe" RGB-spaces.
cspace	defines the output color space, default "sRGB" (sRGB D65), alternatives are "RAW" (Camera specific), "Adobe" (Adobe 1998 D65), "wGamut" (Wide Gamut D65), "kodak" (Kodak ProPhoto D65) and "XYZ", see manpages of dcraw.
xmode	xmode determines how to interpret the values in x if length(dim(x))==3. Implemented are xmode="RGB" (default) and xmode="HSI"
...	additional parameters for function fun

Details

There are two mayor applications for this function. First it allows to add noise to an image by first creating an image that contains the noise and then adding this image using fun="+". Second it offers a way to replace parts of an image, see examples.

Value

object of class "adimpro" containing the image. The object has the following components:

`img`	array containing the color values in the color space specified by `value$type`.
`type`	the color space.
`depth`	color depth, here "16bit".
`dim`	vector of length 2 containing the number of pixel in horizontal and vertival direction.
`file`	the argument `file` identifying the image.
`cspace`	the type of rgb space used, as specified by `cspace`.
`interp`	interpolation applied by dcraw, as specified by `interp`.
`gamma`	has a gamma correction been applied, here FALSE for `read.raw` and TRUE for `read.image`
`gammatype`	type of gamma correction `read.image`.
`wb`	type of white balance, as specified by `wb`.
`compressed`	image data are stored as raw-vector (TRUE) or array of integers (FALSE).

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

`make.image`

Examples

```
## Not run: demo(combine)
```

---

develop.raw *Create a color image from RAW image data.*

Description

The function generates a color image from RAW image data.

Usage

```
develop.raw(object, method = "BILINEAR", wb = c(1, 1, 1), maxrange= TRUE, compress = TRUE)
```

Arguments

object	An object of class "adimpro" containing RAW image data, e.g. created by function read.raw. Such an object is characterized by object$type=="RAW".
method	Method to fill missing color values. Currently implemented are method="HALF" (reduce image size by factor of 2), method="FULL" (use color from neighbor within (2x2) Bayer mask)), method="BILINEAR" (bilinear interpolation), method="Median4" and method="Median16", the latter two being based on L1-MM over shifted Bayer masks. method="Median16" delivers much smoother results, but is considerably slower then the other methods.
wb	numerical vector of length 3 containing multiplicative factors for the three color channels.
maxrange	If TRUE increase range of values to maximum.
compress	logical, determines if image data are stored as raw.

Details

RAW image data usually contain only one color value at each pixel, with colors arranged in a so called Bayer mask. Converting RAW images into color images requires to fill the missing entries in the color channels.

Value

object of class "adimpro" containing the image.

Note

The function requires dcraw to be installed.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

read.image

Examples

```
## Not run: demo(io)
## Not run: demo(raw)
```

edges *Image Processing*

Description

Edge detection using Laplacian, Sobel, or Robert Cross filter.

Usage

```
edges(img, type = "Laplacian", ltype=1, abs=FALSE)
```

Arguments

img	an object of class "adimpro".
type	type of edges detection filter. "Laplacian" (default), "Sobel" , or "Robertcross".
ltype	type of laplacian filter. 1,2,3, or 4
abs	take absolute values of results. This has only an effect for tyoe="Laplacian"

Details

This function applies the Laplacian, Sobel, or Robert Cross filter to the input image img. The filter is applied to each color channel separately. ltype determines the different matrices for Laplacian filter used in the literature. ltype == 1 will use:

```
conv <- matrix(c(-1,-1,-1,-1,-1, -1,-1,-1,-1,-1, -1,-1,24,-1,-1, -1,-1,-1,-1,-1, -1,-1,-1,-1,-1),5,5)
```

ltype == 2 will use:

```
conv <- matrix(c(0,-1,0,-1,4,-1,0,-1,0), 3, 3)
```

ltype == 3 will use:

```
conv <- matrix(c(-1,-1,-1,-1,8,-1,-1,-1,-1), 3, 3)
```

ltype == 4 (default) will use:

```
conv <- matrix(c(1,-2,1,-2,4,-2,1,-2,1), 3, 3)
```

Value

Array containing the values for the edge detector in each pixel and color channel.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

References

Gonzalez, R.C., and Woods, R.E. (2002). Digital Image Processing. Prentice Hall.

Examples

```
## Not run: demo(manipulate)
## Not run: demo(awspimage)
```

`extract.image` *Extract image data from adimpro object*

Description

Extract image data from adimpro object

Usage

```
extract.image(object)
```

Arguments

`object`	adimpro object

Value

array or matrix of image data

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

`make.image`

Examples

```
## Not run: demo(io)
## Not run: demo(manipulate)
```

`extract.info` *Extract EXIF information and additional characteristics from an object of class "adimpro".*

Description

Extract EXIF information and additional characteristics from an object of class "adimpro".

Usage

```
extract.info(object, what = "Bayer")
```

Arguments

`object`	an object of class "adimpro" or a character string that was written as a comment by functions `write.image` or `write.raw`.
`what`	A character string specifying which information is to be extracted. Currently implemented are `"Bayer"` (Bayer mask), `"Daymulti"` (Daylight multiplyers), `"Cammulti"` (camera multipliers), `"Camera"` (camera model), `"Isize"` (Image size), `"Osize"` (image output size), `"File"` (source filename), `"Interpolation"`(Interpolation used to convert RAW to RGB), `"Gammatype"` (type of gamma correction), `"WhiteBalance"` (white balance), `"WhitePoint"` (white point), `"Type"` (image type), `"xind"` (horizontal subindex used by function `clip.image`) and `"yind"` (vertical subindex used by function `clip.image`).

Details

This function is mainly used to access and manipulate information when reading or developing images. A more comprehensive overview is provided by function `summary`.

Value

A character string or integer vector depending on the argument `"what"`.

Note

ImageMagick has to be installed on the system to write "tif", "tiff", "png", "gif", "jpg" or "jpeg" files.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

`read.raw`, `read.image`, `develop.raw`, `summary`

extract.ni *extract information about effective size of neighborhoods*

Description

The function allows to extract information about the effective size of neighborhoods used in each pixel from objects generated by awsimage or awspimage. The result is converted into a greyscale image.

Usage

```
extract.ni(object, gammatype = "ITU", compress = TRUE)
```

Arguments

object	object returned from awsimage or awspimage.
gammatype	character, determines the type of gamma correction within the image. "ITU" stands for ITU-R BT.709-3 as e.g. used by dcraw. Alternatives recognized within the package are "None", "sRGB" and "CIE" (CIE L*).
compress	logical, if TRUE the returned image will be compressed.

Value

an object of class "adimpro".

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

awsimage, awspimage, show.image, write.image

imganiso2D *create an image that visualizes anisotropy*

Description

The function creates an object of class adimpro that visualizes anisotropy information using the HSI color space for main direction of anisotropy (H), maximum eigenvalue^satexp (S) and log(excentricity) (I).

Usage

```
imganiso2D(x, satexp = 0.25, g=3, rho=0)
```

Arguments

x	a field of 2D tensors (Dimension c(3,n1,n2)) or an object of class "adimpro".
satexp	exponent for maximum eigenvalue in saturation channel. Determines the contrast in this channel.
g	Bandwidth for anisotropic smoothing gradient estimates, preferably $g \geq 3$ for images with line type texture and small $g \approx 1$ for improving edges between homogeneous regions.
rho	Regularization parameter for anisotropic smoothing gradient estimates, preferably $\rho = 0$ for images with line type texture and large $\rho \approx 3 * variance$ for improving edges between homogeneous regions.

Value

an object (image) of class adimpro.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

References

Polzehl, J. and Tabelow, K. (2007). Adaptive smoothing of digital images, Journal of Statistical Software 19 (1).

See Also

awsaniso,read.image, read.raw, make.image, show.image, clip.image

Examples

```
## Not run: demo(awsimage)
## Not run: demo(manipulate)
```

---

mask.create *Create a mask for use within function awsimage*

Description

Select part of image according to greyscale (or color) value.

Usage

```
mask.create(img,
            range1 = c(0, 1), range2 = c(0, 1), range3 = c(0, 1),
            locate = TRUE)
```

Arguments

img	Image object of class "adimpro", usually the result returned from read.image, read.raw, or make.image.
range1	Range of gray values or values in the first color channel. mask is set to FALSE for all pixel with values outside this range. Defaults to c(0,1).
range2	Range of values in the second color channel. mask is set to FALSE for all pixel with values outside this range. Defaults to c(0,1).
range3	Range of values in the third color channel. mask is set to FALSE for all pixel with values outside this range. Defaults to c(0,1).
locate	(logical). If TRUE (default), the image is displayed and two opposite corners of a rectangular region can be selected using the mouse (call of locator(2)).

Details

Function to create a mask. The returned object mask can be used to restrict computations in function awsimage to a region characterized by mask==TRUE.

Value

logical matrix of image dimension.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

References

Polzehl, J., and Tabelow, K. (2007). Adaptive smoothing of digital images, Journal of Statistical Software 19 (1).

See Also

awsimage

---

plot.adimpro *I/O functions*

Description

Visualize image data.

Usage

```
## S3 method for class 'adimpro'
plot(x, new=FALSE, gammatype = NULL, cspace = NULL,
                 whitep = NULL, temp = NULL, black = 0, exposure = 1,...)
```

Arguments

x	image object of class "adimpro"
new	should new X11() be opened? default FALSE
gammatype	character, determines the type of gamma correction within the image. "ITU" stands for ITU-R BT.709-3 as e.g. used by dcraw. Alternatives recognized within the package are "None", "sRGB" and "CIE" (CIE L*). NULL keeps the actual setting. gammatype="histogram" forces histogram equalization based on the corresponding greyvalue image.
cspace	defines the output color space, default "sRGB" (sRGB D65), alternatives are "Adobe" (Adobe 1998 D65), "wGamut" (Wide Gamut D65), "kodak" (Kodak ProPhoto D65) "xyz", "yuv", "yiq" and "hsi". NULL keeps the actual setting.
whitep	White point in xyY space. Can be given as one of (character) c("A","B","C","E","D50","D55","D65", or as a two element numeric vector of chromatic xy coordinates. "D65" corresponds to the default white point of "sRGB" and "Adobe" RGB-spaces. NULL keeps the actual setting.
temp	Color temperature. Can be used to specify chromatic xy coordinates of the whitepoint. Only used if is.null(whitep).
black	Adjustment for black color. Color values with luminance <= black will be assigned to black in RGB. Adjustment ist done in xyY space.
exposure	Multiplicative factor for all color channels (in xyz or rgb spaces). Applied in linear color space, i.e. if the image is gamma corrected the gamma correction is reversed first.
...	not used

Details

This functions shows information on the image. This includes histograms of color values in each channel of the specified (x$type) color space, a thumbnail (in "sRGB" with gamma correction gammatype if specified and gammatype="ITU" elsewhere), some information on the image and, if x was produced by awsimage, an image illustrating the local adaptation.

Value

nothing is returned.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

summary.adimpro, adjust.image

Examples

```
## Not run: demo(color)
```

`rimage` *Slighty enhanced image function.*

Description

The function builds upon function from package graphics, but allows to change some of it's defaults. These changes can be made through `rimage.options` for subsequent calls of `rimage` or by direct specification in `....`

Usage

```
rimage(x = seq(0, 1, length.out = nrow(z)), y = seq(0, 1, length.out = ncol(z)), z, ...)
rimage.options(...)
```

Arguments

`x, y`	locations of grid lines at which the values in 'z' are measured. These must be finite, non-missing and in (strictly) ascending order. By default, equally spaced values from 0 to 1 are used. If 'x' is a 'list', its components 'x$x' and 'x$y' are used for 'x' and 'y', respectively. If the list has component 'z' this is used for 'z'. (Same as for function image)
`z`	a numeric or logical matrix containing the values to be plotted ('NA's are allowed). Note that 'x' can be used instead of 'z' for convenience. (Same as for function image)
`...`	The following arguments can be supplied to both `rimage` and `rimage.options`: • zquantiles - numeric(2): quantiles of image intensity values to be usd to determine ,zlim as `zlim <- quantile(z, zquantiles)`. not used if zlim is supplied directly. default `c(0,1)`. • up - color for intensity values larger than `zlim[2]`, default "white". • low - color for intensity values smaller than `zlim[1]`, default "black". • NAcolor - color for intensity NA values, default 0 (transparent). • col - color scheme for values in the range of `zlim`, default `grey(0:255/255)`. • asp - aspect ratio, default TRUE. • xlab - name for x-axis, default "x" • ylab - name for y-axis, default "y" • xaxt - axis type for x-axis, default "s", see par. • yaxt - axis type for y-axis, default "s", see par. • bty - type of box to draw, default "o", see par. • swapx - swap x axis, default FALSE • swapy - swap y axis, default FALSE Additionally all parameters that can be passed to function `image` via `...` can be used with function `rimage`.

Details

This function exists just for convenience to be used if sequences of images are to be plotted using the same settings / style. Function `rimage.options` uses an hidden object `.rimage` within an evironment `.adimproOpts` in the space of package adimpro to store the options.

Value

Both functions return `invisible(NULL)`.

Author(s)

Joerg Polzehl <polzehl@wias-berlin.de>

See Also

`image`, `par`

---

rotate.image *Image Processing*

Description

Rotate an image by 0, 90, 180 or 270 degrees.

Usage

```
rotate.image(img, angle = 90, compress=NULL)
```

Arguments

`img`	image data, that is an object of class "adimpro".
`angle`	0, 90, 180 or 270 degrees
`compress`	store result as compressed image if `compress=TRUE`. `compress=NULL` uses the same format as the original object.

Details

The function rotates the image `img` by 0, 90, 180 or 270 degrees. Any other value for `degree` will cause an exception. The returned object contains an additional component `rotate` determining the rotation.

Value

An object of class "adimpro" containing the rotated image.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

show.image

Examples

```
## Not run: demo(manipulate)
```

---

segment *Segmentation by AWS*

Description

The function allows to segment an image into two or three level sets.

Usage

```
segment(object, level=0.5, delta = 0, thresh = 3, fov = NULL, channel = 0,
        hmax = 4, aws = TRUE, varmodel = NULL, ladjust = 1.25, xind = NULL,
        yind = NULL, wghts = c(0.299, 0.587, 0.114, 0), scorr = TRUE,
        lkern = "Triangle", plateau = NULL, homogen = TRUE,
        earlystop = TRUE, demo = FALSE, select = FALSE, sext = 1.4,
        connected = FALSE, graph = FALSE, max.pixel = 400, compress = TRUE)
```

Arguments

object	Image object, class "adimpro", as from read.image, read.raw, or make.image.
level	center of gray/color-values of the second segment, will not be used if select=TRUE. May be specified such that either level-delta and level+delta are within the interval (0,1) or such that they are within the interval (0,65535) (2 Byte integers).
delta	half width of gray/color-values of the second segment, nay be increased if select=TRUE. May be specified such that either level-delta and level+delta are within the interval (0,1) or such that they are within the interval (0,65535) (2 Byte integers).
thresh	Critical value for final assignment to segment 1 or 3 , should be specified as a quantile of the standard Gaussian distribution.
fov	size of field of view in pixel
channel	specifies which information to use for segmentation. 0: use grey valued image obtained from color images, 1-3: use the specified color channel.
hmax	Maximum bandwidth to use in the iteration procedure.
aws	(logical). If TRUE the propagation - separation (PS) approach from Polzehl and Spokoiny (2006) is used. aws=FALSE turns off the statistical penalty resulting in a nonadaptive kernel estimate using a kernel with bandwidth hmax.

varmodel — varmodel specifies how variances are to be estimated. This can be a homogeneous variance estimate (varmodel="None") assuming uncorrelated errors (both spatial and between channels). Alternatives are an adaptive homogeneous or linear (function of the mean) variance estimate that depends on estimated correlations and on residuals from the last iteration step. The default varmodel=NULL corresponds to varmodel == "Linear" if img$gamma==FALSE and varmodel == "Constant" otherwise.

ladjust — adjustment factor for lambda (>=1). Default values for lambda are selected for Gaussian distributions. Skewed or heavy tailed distributions may require slightly larger values for lambda to meet the propagation condition. ladjust allows to increase lambda in such situations.

xind, yind — Restrict smoothing to rectangular area defined by pixel indices xind,yind in x- and y-direction. Full range if NULL (default).

wghts — allows to weight the information from different (up to 4) color channels. The weights are used in the statistical penalty of the PS-procedure.

scorr — (logical). Specifies whether spatial correlation is to be estimated. Defaults to TRUE. Is set to FALSE if mask is not NULL.

lkern — Specifies the location kernel. Defaults to "Triangle", other choices are "Quadratic", "Cubic" and "Uniform". The use of "Triangle" corresponds to the Epanechnicov kernel nonparametric kernel regression.

plateau — Extension of the plateau in the statistical kernel. Can take values from (0,1), defaults to 0.25.

homogen — If TRUE the algorithm determines, in each design point i, a circle of maximum radius, such that the statistical penalty s_{ij} for all points j within the circle is less than the value specified in plateau. In subsequent iteration steps the statistical penalty for such points is set to zero. This is only used if plateau>0. This results in more stable intermediate estimates and in a smoother reconstruction. homogen=TRUE leads to increased memory requirements.

earlystop — If TRUE the algorithm determines, in each design point i, a circle of minimal radius, such that the circle includes all point j with positive weights w_{ij}. if this radius is considerably smaller than the actual bandwidth then the estimate in point i is fixed. This should considarably reduce computing time in case of large hmax.earlystop=TRUE slightly increases memory requirements.

demo — (logical). If demo=TRUE the function pauses after each iteration. Defaults to FALSE.

select — if TRUE a homogeneous rectangular region can be specified interactively. A value of level is the generated as the median of values within the selected region.

sext — if select==TRUE the value of delta is increased by sext times the standard deviation (estimated by IQR) of the values in the selected region.

connected — if TRUE the set of pixel within the same segment connected to the specified pixel is extracted.

graph — (logical). If graph=TRUE intermediate results are illustrated after each iteration step. Defaults to FALSE.

max.pixel	Maximum dimension of images for display if graph=TRUE. If the true dimension is larger, the images are downscaled for display. See also show.image.
compress	logical, determines if image data are stored in raw-format.

Details

The image is segmented into three parts by performing multiscale tests of the hypotheses H1 value >= level - delta and H2 value <= level + delta. Pixel where the first hypotesis is rejected are classified as -1 (segment 1) while rejection of H2 results in classification 1 (segment 3). Pixel where neither H1 or H2 are rejected ar assigned to a value 0 (segment 2). Critical values for the tests are adjusted for smoothness at the different scales inspected in the iteration process using results from multiscale testing, see e.g. Duembgen and Spokoiny (2001). Critical values also depend on the size of the region of interest specified in parameter fov.

Within segment 2 structural adaptive smoothing is performed while if a pair of pixel belongs to segment 1 or segment 3 the corresponding weight will be nonadaptive.

If connected==TRUE pixel in segment 2 0 are reassigned to a value 2 if they belong to a maximal connected subset of segment2 that contains the center of the specified homogeneous set.

Value

Object of class "adimpro" with

img	containing a greyvalued image with 3 or 4 levels corresponding to the identified segments.

and additional list elements

hsegm	containing the maximal bandwidth used
level	the value of parameter level used
delta	the value of parameter delta used
thresh	the value of parameter thresh used

Note

This function is still experimental and may be changes considerably in future.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

References

Duembgen, L. and Spokoiny, V. (2001). Multiscale testing of qualitative hypoteses. Ann. Stat. 29, 124–152.

Polzehl, J. and Spokoiny, V. (2006). Propagation-Separation Approach for Local Likelihood Estimation. Probability Theory and Related Fields. 3 (135) 335 - 362.

See Also

read.image, read.raw, make.image, show.image, clip.image

Examples

```
## Not run: demo(segment)
```

---

show.image *I/O functions*

Description

Display an image on the screen.

Usage

```
show.image(img, max.x = 1000, max.y = 1000, gammatype = "ITU",
           whitep = NULL, temp = NULL, cspace = "sRGB", black=0, exposure = 1,
           channel=NULL, new = FALSE,  ...)
```

Arguments

img	image data, an object of class "adimpro".
max.x	maximum value of pixels in x dimension to be displayed.
max.y	maximum value of pixels in y dimension to be displayed.
gammatype	character, determines the type of gamma correction within the image. "ITU" stands for ITU-R BT.709-3 as e.g. used by dcraw. Alternatives recognized within the package are "None", "sRGB" and "CIE" (CIE L*). gammatype="histogram" forces histogram equalization based on the corresponding greyvalue image.
cspace	defines the output color space, default "sRGB" (sRGB D65), alternatives are "Adobe" (Adobe 1998 D65), "wGamut" (Wide Gamut D65), "kodak" (Kodak ProPhoto D65) "xyz", "yuv", "yiq" and "hsi". NULL keeps the actual setting. If color.space%in%c("hsi","yuv","yiq","xyz") the individual channels are rescaled to provide maximum contrast. Information from the three channels is coded as "red", "green" and "blue" providing a miscolored image in these cases. Individual channels can be displayed as greyvalue images by specifying the channel argument. color.space="greyvalue" provides a greyvalue image.
whitep	White point in xyY space. Can be given as one of (character) c("A","B","C","E","D50","D55","D65", or as a two element numeric vector of chromatic xy coordinates. "D65" corresponds to the default white point of "sRGB" and "Adobe" RGB-spaces. NULL keeps the actual setting.
temp	Color temperature. Can be used to specify chromatic xy coordinates of the whitepoint. Only used if is.null(whitep).

black	Adjustment for black color. Color values with luminance <= black will be assigned to black in RGB. Adjustment ist done in xyY space.
exposure	Multiplicative factor for all color channels (in xyz or rgb spaces). Applied in linear color space, i.e. if the image is gamma corrected the gamma correction is reversed first.
channel	allows to select a color channel (1: red, 2: green, 3: blue in case of "rgb") for display.
new	should new X11() be opened? default FALSE
...	additional arguments to image can be passed here.

Details

This function displayes greyscale and color images on the screen. If the actual dimension of the image exceeds max.x or max.y the image is shrinked by displaying only part of the pixels (every second/third/... such that the resulting dimension is smaller than max.x or max.y)

Value

Nothing is returned

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

read.image, write.image, adjust.image

Examples

```
## Not run: demo(io)
```

---

shrink.image *Image Processing*

Description

Shrink resolution of an image

Usage

```
shrink.image(img, method = "median",
             xt = img$dim[1], yt = img$dim[2], ratio = TRUE,
             compress=TRUE)
```

Arguments

img	image data, an object of class "adimpro".
method	method to be used to shrink the image. "median" (default), "mean", or "nearest". "median" is supposed to give best results. For a considerably faster result use "nearest".
xt	target x-dimension
yt	target y-dimension
ratio	logical. preserve x-y ratio? default: TRUE
compress	logical, determines if image data are stored in raw-format.

Details

This function shrinkes the resolution of the image such that the x-y dimension of the resulting image is smaller than the original one. xt and yt give the target dimension of the image. If ratio == TRUE (default) the ratio between x- and y-dimension is preserved.

method "nearest" selects a pixel nearest to the new coordinates. method "mean" defines the color of a pixel as the mean of all pixel identified with the new coordinate. method "median" set the color of a pixel as color of an L1-generalized median of all pixel identified with the new coordinate.

Value

shrinked image

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

Examples

```
## Not run: demo(manipulate)
```

---

summary.adimpro *I/O functions*

Description

'summary' method for class '"adimpro"'.

Usage

```
## S3 method for class 'adimpro'
summary(object, ...)
```

Arguments

object	an object of class adimpro, usually, a result of a call to read.raw, or read.image.
...	further arguments passed to or from other methods.

Details

The method tries to print information on the image, like image dimension, color space, value range, etc.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

Examples

```
## Not run: demo(io)
## Not run: demo(manipulate)
```

`write.image` *I/O Functions*

Description

Write an image file.

Usage

```
write.image(img, file = "tmp.ppm", max.x=NULL, max.y=NULL, depth = NULL, gammatype="ITU",
        whitep = NULL, temp = NULL, cspace = NULL, black = 0, exposure = 1)
```

Arguments

`img`	image data, an object of class "adimpro".
`file`	file name, the extension determines the image file format.
`max.x`	maximum value of pixels in x dimension.
`max.y`	maximum value of pixels in y dimension.
`depth`	color depth, either 8 or 16 (bit)! The value is reset to 8 if the specified image file format does not allow for 16 Bit encoding.
`gammatype`	character, determines the type of gamma correction within the image. "ITU" stands for ITU-R BT.709-3 as e.g. used by `dcraw`. Alternatives recognized within the package are "None", "sRGB" and "CIE" (CIE L*). `NULL` keeps the actual setting. `gammatype="histogram"` forces histogram equalization based on the corresponding greyvalue image.
`cspace`	defines the output color space, default is `img$cspace` can be set to "sRGB", "Adobe" (Adobe 1998), "wGamut" (Wide Gamut), "kodak" (Kodak ProPhoto) "xyz", "yuv", "yiq" and "hsi"
`whitep`	White point in `xyY` space. Can be given as one of (character) `c("A","B","C","E","D50","D55","D65",` or as a two element numeric vector of chromatic `xy` coordinates. `"D65"` corresponds to the default white point of `"sRGB"` and `"Adobe"` RGB-spaces. `NULL` keeps the actual setting.

temp	Color temperature. Can be used to specify chromatic xy coordinates of the whitepoint. Only used if is.null(whitep).
black	Adjustment for black color. Color values with luminance <= black will be assigned to black in RGB. Adjustment ist done in xyY space.
exposure	Multiplicative factor for all color channels (in xyz or rgb spaces). Applied in linear color space, i.e. if the image is gamma corrected the gamma correction is reversed first.

Details

This function writes the image data in img to the file file. Color depth depth is used for writing, but if image has a component "depth", this argument will be ignored. Note: Not all target formats support 16bit coding. The target format is determined from the file extension, and should be one of the many that ImageMagick supports.

Note that write image by default applies a gamma correction with gammatype="ITU". This provides a good standard for display on a screen. For printing cspace="Adobe" should be prefered. Images that are intended for further editing should preferably be saved in an image format that allows for 16Bit depth (tiff, png) using either cspace="Adobe", cspace="wGamut" or cspace="kodak" and preferably no gamma correction to prevent from additional loss of information.

Value

Nothing is returned.

Note

ImageMagick has to be installed on the system to write "tif", "tiff", "png", "gif", "jpg" or "jpeg" files.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

read.image, adjust.image

Examples

```
## Not run: demo(io)
```

write.raw *Write image RAW data as greyvalue png image*

Description

Image RAW data is saved as a 16-Bit greyvalue png image. EXIF information contained in the original RAW image is stored as a comment.

Usage

```
write.raw(img, filename = "tmp.png")
```

Arguments

img	object of class "adimpro" containing image RAW data (img$type=="RAW".)
filename	Name of the resulting png-image. If filename does not include an extension ".png" the extension ".png" is added.

Details

EXIF information contained in the original RAW image as well as other available information in object img are added as a comment to the resulting png-image. This comment is evaluated when the image is read by functions read.raw or read.image

Value

Nothing is returned.

Note

ImageMagick has to be installed on the system to write "png" files.

Author(s)

Karsten Tabelow <tabelow@wias-berlin.de> and Joerg Polzehl <polzehl@wias-berlin.de>

See Also

read.raw, develop.raw

Examples

```
## Not run: demo(raw)
```

Index