Package ‘imbibe’

November 9, 2022

Version 0.1.1

Date 2022-11-04

Title A Pipe-Friendly Image Calculator

Maintainer Jon Clayden `<code@clayden.org>`

Imports Rcpp, RNifti, magrittr

LinkingTo Rcpp, RNifti

Suggests mmand, tinytest, covr

Description Provides a set of fast, chainable image-processing operations
which are applicable to images of two, three or four dimensions,
particularly medical images.

License BSD_3_clause + file LICENCE

URL `https://github.com/jonclayden/imbibe`

BugReports `https://github.com/jonclayden/imbibe/issues`

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation yes

Author Jon Clayden [aut, cre] (<`https://orcid.org/0000-0002-6608-0619`>),
Chris Rorden [aut] (<`https://orcid.org/0000-0002-7554-6142`>),
John Muschelli [ctb] (<`https://orcid.org/0000-0001-6469-1750`>),
Exstrom Laboratories LLC [cph]

Repository CRAN

Date/Publication 2022-11-09 21:50:16 UTC

R topics documented:

---

add — *Basic binary operations*

Description

Basic binary operations

Usage

```
add(image, arg)

subtract(image, arg)

multiply(image, arg)

divide(image, arg)

remainder(image, arg)

mask(image, arg)

maximum(image, arg)

minimum(image, arg)
```

Arguments

`image`	An image object or pipeline.
`arg`	Numeric or image argument.

Value

An updated pipeline.

`dilate` *Mathematical morphology and filtering operations*

Description

Mathematical morphology and filtering operations

Usage

```
dilate(image, kernel = NULL, ..., max = FALSE, nonzero = TRUE)

dilateall(image, kernel = NULL, ...)

erode(image, kernel = NULL, ..., min = FALSE)

filter_median(image, kernel = NULL, ...)

filter_mean(image, kernel = NULL, ..., norm = TRUE)

smooth_gauss(image, sigma)

subsample(image, offset = FALSE)
```

Arguments

`image`	An image object or pipeline.
`kernel`	A suitable kernel function (see `kernels`). If `NULL`, the most recently set kernel in the pipeline is used, if any, otherwise the default kernel (`kernel_3d`).
`...`	Additional arguments to the kernel function, if any.
`max`	Logical value: if `TRUE`, maximum filtering is used for dilation; otherwise mean filtering is used. Mean filtering is always used by `dilateall`.
`nonzero`	Logical value: if `TRUE`, the default, dilation is only applied to nonzero pixels/voxels. Otherwise it is applied everywhere (and maximum filtering is always used).
`min`	Logical value: if `TRUE`, minimum filtering is used for erosion; otherwise nonzero voxels overlapping with the kernel are simply zeroed.
`norm`	Logical value indicating whether the mean filter will be normalised or not.
`sigma`	Numeric value giving the standard deviation of the Gaussian smoothing kernel.
`offset`	Logical value indicating whether subsampled pixels should be offset from the original locations or not.

Value

An updated pipeline.

`dim_mean` *Dimensionality reduction operations*

Description

Dimensionality reduction operations

Usage

```
dim_mean(image, dim = 4L)

dim_sd(image, dim = 4L)

dim_max(image, dim = 4L)

dim_whichmax(image, dim = 4L)

dim_min(image, dim = 4L)

dim_median(image, dim = 4L)

dim_quantile(image, dim = 4L, prob)

dim_AR1(image, dim = 4L)
```

Arguments

`image`	An image object or pipeline.
`dim`	Integer value between 1 and 4, giving the dimension to apply the reduction along.
`prob`	For `drt_quantile`, the quantile probability to extract (analogously to `quantile`).

Value

An updated pipeline.

`expect_pipeline_result` *Expectation for testing pipeline output*

Description

This function provides an expectation for use with the "tinytest" package, which runs the pipeline specified in its first argument and compares the result to its second.

Usage

```
expect_pipeline_result(current, target, precision = "double", ...)
```

Arguments

`current`	The pipeline to run, which should have class `"imbibe"`.
`target`	The target value to compare against, a numeric array of some kind, which will be converted to a `"niftiImage"` object.
`precision`	A string specifying the working precision. Passed to `run`.
`...`	Further arguments to `expect_equal`.

Value

A `"tinytest"` object.

`exponent` *Basic unary operations*

Description

Basic unary operations

Usage

```
exponent(image)

logarithm(image)

sine(image)

cosine(image)

tangent(image)

arcsine(image)

arccosine(image)

arctangent(image)

square(image)

squareroot(image)

reciprocal(image)
```

```
absolute(image)

binarise(image, invert = FALSE)

binarize(image, invert = FALSE)
```

Arguments

`image`	An image object or pipeline.
`invert`	Logical value: if `TRUE`, binarising will also perform logical inversion so that only zeroes in the original image will be nonzero; if `FALSE`, the default, the usual sense is used, in which zeroes remain as they are, and everything else is converted to 1.

Value

An updated pipeline.

`imbibe` *Create an operation pipeline*

Description

Create an operation pipeline

Usage

```
imbibe(image)

## S3 method for class 'imbibe'
asNifti(x, ...)

## S3 method for class 'imbibe'
as.array(x, ...)

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

Arguments

`image`	An image object or existing pipeline.
`x`	An `"imbibe"` object.
`...`	Additional arguments to methods.

kernel_3d *Mathematical morphology kernels*

Description

Mathematical morphology kernels

Usage

```
kernel_3d(image)

kernel_2d(image)

kernel_box(image, width, voxels = FALSE)

kernel_gauss(image, sigma)

kernel_sphere(image, radius)

kernel_file(image, file)
```

Arguments

`image`	An image object or pipeline.
`width`	The width of the kernel in appropriate units. If `voxels` is `FALSE` a value can be specified for each of the three dimensions; otherwise only a single value should be given and the kernel will be isotropic.
`voxels`	Logical value: if `TRUE`, the `width` is given in pixels/voxels and must be an odd integer; otherwise, the units are millimetres and can take any value.
`sigma`	Numeric value giving the standard deviation of a Gaussian kernel, in millimetres.
`radius`	Numeric value giving the radius of a sphere kernel, in millimetres.
`file`	Name of a NIfTI file containing the kernel.

Value

An updated pipeline.

run *Run a pipeline and return an image result*

Description

Run a pipeline and return an image result

Usage

```
run(pipe, precision = getOption("imbibe.precision", "double"))
```

Arguments

`pipe`	An operation pipeline.
`precision`	The internal precision used for calculations. May be `"double"`, `"float"` or `"single"`; the latter two are equivalent.

Value

An image

Examples

```
im <- RNifti::readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))
pipe <- im %>% threshold_below(500) %>% binarise()
run(pipe)
```

threshold *Image thresholding*

Description

Image thresholding

Usage

```
threshold(
  image,
  value,
  reference = c("none", "image", "nonzero"),
  above = FALSE
)

threshold_below(image, value, reference = c("none", "image", "nonzero"))

threshold_above(image, value, reference = c("none", "image", "nonzero"))
```

Arguments

`image`	An image object or pipeline.
`value`	Numeric threshold value.
`reference`	String indicating what the `value` should be referenced against, if anything. If `"none"`, the default, the `value` is taken literally. If `"image"`, it is interpreted as a proportion of the "robust range" of the current image's intensities. If `"nonzero"` it is interpreted as a proportion of the "robust range" of the nonzero pixel intensities.
`above`	Logical value: if `TRUE` the operation zeroes values above the threshold; otherwise it zeroes values below it. The `threshold_below` and `threshold_above` function variants set argument implicitly.

Value

An updated pipeline.

Index