Package ‘velox’

August 25, 2016

Type Package
Title Fast Raster Manipulation and Extraction
Version 0.1.0
Date 2016-08-24
Author Philipp Hunziker
Maintainer Philipp Hunziker <hunzikp@gmail.com>
Description C++ accelerated raster manipulation and extraction.
License GPL (>= 2)
Imports methods, Rcpp (>= 0.11.6), raster, rgeos, rgdal, sp
LinkingTo Rcpp
NeedsCompilation yes
LazyData TRUE
RoxygenNote 5.0.1
Repository CRAN
Date/Publication 2016-08-25 10:01:02

R topics documented:

velox ................................................................. 2
VeloxRaster-class ................................................... 3
VeloxRaster_aggregate ........................................... 4
VeloxRaster_as.matrix ........................................... 4
VeloxRaster_as.RasterLayer ..................................... 5
VeloxRaster_as.RasterStack ...................................... 5
VeloxRaster_col2im ............................................. 6
VeloxRaster_crop .................................................. 7
VeloxRaster_drop .................................................. 7
VeloxRaster_extract ............................................. 8
VeloxRaster_getCoordinates ..................................... 9
VeloxRaster_im2col ............................................... 9
VeloxRaster_meanFocal ......................................... 10
velox creates a VeloxRaster object.

Usage

velox(x, extent = NULL, res = NULL, crs = NULL)

Arguments

x
A RasterLayer, RasterStack, matrix, list of matrices, list of VeloxRaster objects, or character string pointing to a GDAL-readable file.

extent
An extent object or a numeric vector of length 4. Required if x is a matrix or list of matrices, ignored otherwise.

res
The x and y resolution of the raster as a numeric vector of length 2. Required if x is a matrix or list of matrices, ignored otherwise.

crs
Optional. A character string describing a projection and datum in the PROJ.4 format. Ignored if x is a Raster* object.

Details

Creates a VeloxRaster object. Note that VeloxRaster objects are Reference Class objects and thus mutable. Hence, the usual R copy on modify semantics do not apply.

Note that if x is a list of VeloxRasters, the extent and crs attributes are copied from the first list element.

Value

A VeloxRaster object.

Examples

## Create VeloxRaster from list of matrices
mat1 <- matrix(1:100, 10, 10)
mat2 <- matrix(100:1, 10, 10)
mat.ls <- list(mat1, mat2)
vl <- velox(mat.ls, extent=c(0,1,0,1), res=c(0.1,0.1), crs="+proj=longlat +datum=WGS84 +no_defs")
VeloxRaster-class

A Reference Class for velox rasters

Description

A Reference Class for velox rasters

Fields

- rasterbands  A list of matrices containing the raster data
- dim  Raster dimensions
- extent  Raster extent
- res  Raster resolution
- nbands  Number of raster bands
- crs  Coordinate reference system (Proj4 string)

Methods

aggregate(factor, aggttype = "sum") See veloxraster_aggregate.
as.matrix(band = 1) See veloxraster_as.matrix.
as.RasterLayer(band = 1) See veloxraster_asRasterLayer.
as.RasterStack() See veloxraster_asRasterStack.
col2im(mat, wrow, wcol, band, rowframe = 0, colframe = 0, rowstride = 1, colstride = 1)
  See veloxraster_col2im.
crop(x) See veloxraster_crop.
drop(bands) See veloxraster_drop.
extract(sp, fun) See veloxraster_extract.
getCoordinates() See veloxraster_getCoordinates.
im2col(wrow, wcol, band, padval = 0, rowframe = 0, colframe = 0, rowstride = 1, colstride = 1)
  See veloxraster_im2col.
meanFocal(weights, bands = 1) See veloxraster_meanFocal.
medianFocal(wrow, wcol, bands = 1) See veloxraster_medianFocal.
rasterize(spdf, field, band = 1, background = NULL) See veloxraster_rasterize.
sumFocal(weights, bands = 1) See veloxraster_sumFocal.
write(path, overwrite = FALSE) See veloxraster_write.

Examples

```r
## Make VeloxRaster objects using the 'velox' function
mat <- matrix(1:100, 10, 10)
vx <- velox(mat, extent=c(0,1,0,1), res=c(0.1,0.1), crs="+proj=longlat +datum=WGS84 +no_defs")
class(vx)
```
VeloxRaster_aggregate  Aggregate

Description
Aggregates a VeloxRaster object to a lower resolution.

Arguments

factor: A numeric vector of length 1 or 2 indicating the aggregation factor in the x and y dimensions. Must be positive integers > 1.

aggtype: A character string indicating the aggregation type. See Details.

Details

aggtype must be one of the following: "sum", "mean", "min", "max", "median".

Value
Void.

Examples
## Make VeloxRaster
mat <- matrix(1:100, 10, 10)
vx <- velox(mat, extent=c(0,1,0,1), res=c(0.1,0.1), crs="+proj=longlat +datum=WGS84 +no_defs")
## Aggregate
vx$aggregate(factor=c(2,2), aggtype='sum')

VeloxRaster_as.matrix  Cast a VeloxRaster band as a matrix

Description
as.matrix creates a matrix from a VeloxRaster band.

Arguments

band: Integer indicating the VeloxRaster band to be transformed.

Value
A matrix.
Examples

```r
## Make VeloxRaster
mat <- matrix(1:100, 10, 10)
vx <- velox(mat, extent=c(0,1,0,1), res=c(0.1,0.1), crs="+proj=longlat +datum=WGS84 +no_defs")
## Cast to matrix
vx.mat <- vx$as.matrix(band=1)
identical(mat, vx.mat)
```

---

### VeloxRaster_as.RasterLayer

*Cast a VeloxRaster band as a RasterLayer object*

**Description**

as.RasterLayer creates a RasterLayer object from a VeloxRaster band.

**Arguments**

- **band**
  - Integer indicating the VeloxRaster band to be transformed.

**Value**

A RasterLayer object.

**Examples**

```r
## Make VeloxRaster
mat <- matrix(1:100, 10, 10)
vx <- velox(mat, extent=c(0,1,0,1), res=c(0.1,0.1), crs="+proj=longlat +datum=WGS84 +no_defs")
## Cast to RasterLayer
library(raster)
rl <- vx$as.RasterLayer(band=1)
```

---

### VeloxRaster_as.RasterStack

*Cast a VeloxRaster as a RasterStack object*

**Description**

as.RasterStack creates a RasterStack object from a VeloxRaster.

**Value**

A RasterStack object.
Examples

```r
# Make VeloxRaster with two bands
mat1 <- matrix(1:100, 10, 10)
mat2 <- matrix(100:1, 10, 10)
vx <- velox(list(mat1, mat2), extent=c(0,1,0,1), res=c(0.1,0.1),
crs="+proj=longlat +datum=WGS84 +no_defs")
# Cast to RasterStack
library(raster)
rs <- vx$as.RasterStack()
```

---

**Description**

Assigns values to a VeloxRaster band from a matrix of flattened image patches. Patch frames, as specified by `rowframe` and `colframe`, are not assigned. This function is intended to be used with `mat` matrices constructed with the `im2col` function.

**Arguments**

- `mat`: The matrix of flattened image patches.
- `wrow`: Patch size in the y dimension.
- `wcol`: Patch size in the x dimension.
- `band`: The band to be assigned.
- `rowframe`: A non-negative integer specifying the size of the frame around the image patches in the y dimension.
- `colframe`: A non-negative integer specifying the size of the frame around the image patches in the x dimension.
- `rowstride`: A positive integer denoting the stride between extracted patches in the y dimension.
- `colstride`: A positive integer denoting the stride between extracted patches in the x dimension.

**Value**

Void.

**Examples**

```r
# Make VeloxRaster
mat <- matrix(1:100, 10, 10)
vx <- velox(mat, extent=c(0,1,0,1), res=c(0.1,0.1),
crs="+proj=longlat +datum=WGS84 +no_defs")
# Apply im2col
patch.mat <- vx$im2col(wrow=2, wcol=2, band=1, padval=0,
```
VeloxRaster_crop

Crop a VeloxRaster object

Description

crops a VeloxRaster object

Arguments

y An object from which an extent object can be extracted. Usually a Spatial* or Raster* object.

Details

Crops a VeloxRaster object to the extent of y. Note that currently the extent of y must overlap with the extent of x, otherwise an error is thrown.

Value

Void.

Examples

## Make VeloxRaster
mat <- matrix(1:100, 10, 10)
vx <- velox(mat, extent=c(0,1,0,1), res=c(0.1,0.1), crs="+proj=longlat +datum=WGS84 +no_defs")
## Crop
vx$crop(c(0.3,0.7,0.3,0.7))
vx$extent

VeloxRaster_drop

Delete a raster band from a VeloxRaster

Description

Delete a raster band from a VeloxRaster

Arguments

bands Numeric vector containing IDs of bands to be dropped.
Value

Void.

Examples

```r
## Make VeloxRaster with two bands
mat1 <- matrix(1:100, 10, 10)
mat2 <- matrix(100:1, 10, 10)
vx <- velox(list(mat1, mat2), extent=c(0,1,0,1), res=c(0.1,0.1),
     crs="+proj=longlat +datum=WGS84 +no_defs")
## Delete band 2
vx$drop(bands=2)
```

Description

Extracts the values of all cells whose centerpoint is in SpatialPolygons* object `sp` and applies R function `fun`.

Arguments

- `sp`: A SpatialPolygons* object.
- `fun`: An R function. See Details.

Details

`fun` must be an R function accepting a numeric vector as its sole input.

Value

A numeric matrix. One row per element in `sp`, one column per band in the VeloxRaster.

Examples

```r
## Make VeloxRaster with two bands
set.seed(0)
mat1 <- matrix(rnorm(100), 10, 10)
mat2 <- matrix(rnorm(100), 10, 10)
vx <- velox(list(mat1, mat2), extent=c(0,1,0,1), res=c(0.1,0.1),
     crs="+proj=longlat +datum=WGS84 +no_defs")
## Make SpatialPolygons
library(sp)
library(rgeos)
coord <- cbind(0.5, 0.5)
spoint <- SpatialPoints(coords=coord)
spols <- gBuffer(spgeom=spoint, width=0.5)
```
### VeloxRaster_getCoordinates

*Get coordinates*

#### Description

Returns a matrix containing the x-y coordinates of all cell center points of a VeloxRaster.

#### Value

A numeric matrix.

#### Examples

```r
## Extract
vx$extract(sp=spolsL fun=mean)
```

### VeloxRaster_im2col

*im2col*

#### Description

Creates a matrix of flattened image patches from a VeloxRaster band. Order is left-to-right, top-to-bottom. Note that if any(c(rowframe, colframe)>0), the image patches are (partially) overlapping.

#### Arguments

- `wrow`: Patch size in the y dimension.
- `wcol`: Patch size in the x dimension.
- `band`: The band to be flattened.
- `padval`: A padding value.
- `rowframe`: A non-negative integer specifying the size of the frame around the image patches in the y dimension.
- `colframe`: A non-negative integer specifying the size of the frame around the image patches in the x dimension.
- `rowstride`: A positive integer denoting the stride between extracted patches in the y dimension. I.e. only every `rowstride`th patch is extracted.
- `colstride`: A positive integer denoting the stride between extracted patches in the x dimension. I.e. only every `colstride`th patch is extracted.
A numeric matrix with \((w\text{row}+2\times\text{rowframe})\times(w\text{col}+2\times\text{colframe})\) columns.

### Examples

```r
## Make VeloxRaster
mat <- matrix(1:100, 10, 10)  
vx <- velox(mat, extent=c(0,1,0,1), res=c(0.1,0.1), crs="+proj=longlat +datum=WGS84 +no_defs")
## Apply im2col
patch.mat <- vx$im2col(wrow=2, wcol=2, band=1, padval=0,  
                        rowframe=1, colframe=1, rowstride=2, colstride=2)
dim(patch.mat)
```

### Description

Applies a mean filter with weights matrix \(weights\) to a VeloxRaster.

### Arguments

- **weights**: A numeric matrix of weights. Both dimensions must be uneven.
- **bands**: Numeric vector indicating bands where filter is applied.

### Details

Padding is currently not implemented.

### Value

Void.

### Examples

```r
## Make VeloxRaster with two bands
mat1 <- matrix(1:100, 10, 10)  
mat2 <- matrix(100:1, 10, 10)  
vx <- velox(list(mat1, mat2), extent=c(0,1,0,1), res=c(0.1,0.1),  
            crs="+proj=longlat +datum=WGS84 +no_defs")
## Mean focal
weights <- matrix(1, 5, 5)  
vx$meanFocal(weights=weights, bands=c(1,2))
```
VeloxRaster_medianFocal

Median focal

Description

Applies a median filter of dimension \( wcol \times wrow \) to a VeloxRaster.

Arguments

- \( wrow \): y dimension of filter. Must be uneven integer.
- \( wcol \): x dimension of filter. Must be uneven integer.
- \( bands \): Numeric vector indicating bands where filter is applied.

Details

Padding is currently not implemented.

Value

Void.

Examples

```r
## Make VeloxRaster with two bands
mat1 <- matrix(1:100, 10, 10)
mat2 <- matrix(100:1, 10, 10)
vx <- velox(list(mat1, mat2), extent=c(0,1,0,1), res=c(0.1,0.1),
           crs="+proj=longlat +datum=WGS84 +no_defs")
## Median focal
vx$medianFocal(wrow=5, wcol=5, bands=c(1,2))
```

VeloxRaster_rasterize  Rasterize Polygons

Description

Rasterizes a SpatialPolygonsDataFrame, i.e. assigns the values in the field column of the SPDF to the raster cells intersecting with the respective SpatialPolygon.
Arguments

spdf A SpatialPolygonsDataFrame object.
field A character string corresponding to the name of a numeric column in spdf.
band A positive integer denoting the ID of the band where the rasterized values are written.
background Optional. A numeric value assigned to all background cells.

Details

Note that rasterization is performed sequentially. Hence, cells being contained by multiple polygons are assigned the value of the last polygon in the spdf object.

Value

Void.

Examples

```r
## Make VeloxRaster
mat <- matrix(0, 10, 10)
vx <- velox(mat, extent=c(0,1,0,1), res=c(0.1,0.1), crs="+proj=longlat +datum=WGS84 +no_defs")
## Make SpatialPolygonsDataFrame
library(sp)
library(rgeos)
coord <- cbind(.0, .5)
spoint <- SpatialPoints(coords=coord)
spols <- gBuffer(spgeom=spoint, width=0.25)
spdf <- SpatialPolygonsDataFrame(Sr=spols, data=data.frame(value=1), match.ID=FALSE)
## Rasterize, set background to -1
vx$rasterize(spdf=spdf, field="value", background=-1)
```

---

**VeloxRaster_sumFocal**

**Sum focal**

Description

Applies a focal sum with weights matrix weights to a VeloxRaster.

Arguments

weights A numeric matrix of weights. Both dimensions must be uneven.
bands Numeric vector indicating bands where filter is applied.

Details

Padding is currently not implemented.
VeloxRaster_write

Value
Void.

Examples
### Make VeloxRaster with two bands
mat1 <- matrix(1:100, 10, 10)
mat2 <- matrix(100:1, 10, 10)
vx <- velox(list(mat1, mat2), extent=c(0,1,0.1), res=c(0.1,0.1),
    crs="+proj=longlat +datum=WGS84 +no_defs")
### Sum focal
weights <- matrix(1, 5, 5)
vx$sumFocal(weights=weights, bands=c(1:2))

---

**VeloxRaster_write**

Write a VeloxRaster to disk as a GeoTiff file

Description
Write a VeloxRaster to disk as a GeoTiff file

Arguments
- **path**: Output filename as character string.
- **overwrite**: Boolean indicating whether target file should be overwritten.

Value
Void.
Index

velox, 2
VeloxRaster (VeloxRaster-class), 3
VeloxRaster-class, 3
VeloxRaster_aggregate, 3, 4
VeloxRaster_as.matrix, 3, 4
VeloxRaster_as.RasterLayer, 3, 5
VeloxRaster_as.RasterStack, 3, 5
VeloxRaster_col2im, 3, 6
VeloxRaster_crop, 3, 7
VeloxRaster_drop, 3, 7
VeloxRaster_extract, 3, 8
VeloxRaster_getCoordinates, 3, 9
VeloxRaster_im2col, 3, 9
VeloxRaster_meanFocal, 3, 10
VeloxRaster_medianFocal, 3, 11
VeloxRaster_rasterize, 3, 11
VeloxRaster_sumFocal, 3, 12
VeloxRaster_write, 3, 13