Package 'fasteraster'

March 20, 2017

Type Package

Title Raster Image Processing and Vector Recognition

Version 1.1.1

Date 2017-03-19

Author Andy Bosyi

Maintainer Andy Bosyi <andy@bosyi.com>

Description If there is a need to recognise edges on a raster image or a bitmap or any kind of a matrix, one can find packages
that does only 90 degrees vectorization. Typically the nature of artefact images is linear and can be vectorized in much more
efficient way than draw a series of 90 degrees lines. The fasteraster package does recognition of lines using only one pass.
It also allows to calculate mass and the mass centers for the recognized zones or polygons.

License GPL (>= 2)

URL http://bosyi.com/craft/

Imports Rcpp (>= 0.12.5)

LinkingTo Rcpp

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-03-19 23:35:39 UTC

R topics documented:

`fasteraster` *Raster Image Processing and Vector Recognition*

Description

If there is a need to recognise edges on a raster image or a bitmap or any kind of a matrix, one can find packages that does only 90 degrees vectorization. Typically the nature of artefact images is linear and can be vectorized in much more efficient way than draw a series of 90 degrees lines. The fasteraster package does recognition of lines using only one pass. It also allows to calculate weight and the mass centers for the recognized zones or polygons.

Details

Use raster2vector to vectorize input matrix. Use rasterZoneAnalyzer to break the input matrix by zones and get their weight and mass center.

Author(s)

Andy Bosyi <andy@bosyi.com>.

Maintainer: Andy Bosyi <andy@bosyi.com>

See Also

http://bosyi.com/craft/

Examples

```
## Not run:
library(fasteraster);
library(datasets);

polygons <- raster2vector(volcano, 120, 200, 20, 1);
image(volcano, col = rev(grey.colors(100)), useRaster = TRUE)
plot(0, type = "l", xlim = c(0, nrow(volcano)), ylim = c(0, ncol(volcano)))
a <- lapply(polygons, function(x) lines(rbind(x, x[1,])))

zones <- rasterZoneAnalyzer(volcano, 120, 200, 20);
a <- text(zones[ , 3], zones[ , 4], labels = zones[ , 2]);

## End(Not run)
```

raster2vector *Converts raster image matrix into list of polygons.*

Description

Takes a raster matrix and vectorize it by polygons. Typically the nature of artefact images is linear and can be vectorized in much more efficient way than draw a series of 90 degrees lines.

Usage

```
raster2vector(raster, from = 0, to = 1, step = 0.1, precision = 1L,
  exclave = TRUE)
```

Arguments

raster	input matrix of numeric values.
from	lower (greater than) margin for recognizable polygon values.
to	upper (less than or equal) margin.
step	values gradient.
precision	linearization precision in matrix cells.
exclave	polygons may have other inside.

Value

list of integer matrixes that represent polygon coordinates in two columns.

Examples

```
library(datasets)
raster2vector(volcano, 100, 200, 60, 1, FALSE)
raster2vector(volcano, 120, 200, 20)
```

rasterZoneAnalyzer *Recognizes zones in the raster matrix and return a list of their analysis.*

Description

Takes a raster matrix and recognize zones exactly as the raster2vector function does. However does not return polygon vectors, but weight and center of mass for each polygone (or so called zone).

Usage

```
rasterZoneAnalyzer(raster, from = 0, to = 1, step = 0.1)
```

Arguments

raster	input matrix of numeric values.
from	lower (greater than) margin for recognizable zone values.
to	upper (less than or equal) margin.
step	values gradient.

Value

matrix of zones by four values: fill (ceiling values), weight (in cells), x and y of the mass center.

Examples

```
library(datasets)
rasterZoneAnalyzer(volcano, 120, 200, 20)
```

Index