Package 'OTBsegm'

May 6, 2025

Type Package

Title Apply Unsupervised Segmentation Algorithms from 'OTB'

Version 0.1.0

Description
Apply unsupervised segmentation algorithms included in 'Orfeo ToolBox' software (<https://www.orfeo-toolbox.org/>), such as mean shift or watershed segmentation.

Encoding UTF-8

Imports cli, terra, link2GI

RoxygenNote 7.3.2

License MIT + file LICENSE

URL https://cidree.github.io/OTBsegm/

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Adrián Cidre González [aut, cre]

Maintainer Adrián Cidre González <adrian.cidre@gmail.com>

Repository CRAN

Date/Publication 2025-05-06 09:00:05 UTC

Contents

segm_lsms *Large-scale segmentation using Mean-Shift*

Description

Applies the Mean-Shift segmentation algorithm to an image file or a SpatRaster. Suitable for large images

Usage

```
segm_lsms(
  image,
  otb,
  spatialr = 5L,
  ranger = 15,
  minsize = 100L,
  tilesize = 500L,
  mode = "vector",
  mask = NULL,
  ram = 256L
)
```

Arguments

image	path to raster, or SpatRaster
otb	output of link2GI::linkOTB()
spatialr	integer. Spatial radius of the neighborhood
ranger	range radius defining the radius (expressed in radiometry unit) in the multispectral space
minsize	integer. Minimum size of a region (in pixel unit) in segmentation. Smaller clusters will be merged to the neighboring cluster with the closest radiometry. If set to 0 no pruning is done
tilesize	integer. Size of the tiles during the tile-wise processing
mode	processing mode, either 'vector' or 'raster'. See details
mask	an optional raster used for masking the segmentation. Only pixels whose mask is strictly positive will be segmented
ram	integer. Available memory for processing (in MB)

Details

Mean-Shift is a region-based segmentation algorithm that groups pixels with similar characteristics. It's a non-parametric clustering technique that groups pixels based on spatial proximity and feature similarity (color, intensity). This method is particularly effective for preserving edges and defailt while simplifying textures in high-resolution images. Steps:

1. Initialization: Each pixel is treated as a point in a multi-dimensional space (combining spatial and color features).
2. Mean Shift Iterations: For each pixel, a search window moves toward the region with the highest data density (local maxima) by calculating the mean of neighboring pixels within the window.
3. Convergence: The process repeats until the movement of the window becomes negligible, indicating convergence.
4. Label Assignment: Pixels that converge to the same mode (local maxima) are grouped into the same region.

The most important parameters are:

- spatialr: defines the size of the neighborhood
- ranger: determines similarity in the feature space
- maxiter: limits the number of iterations for convergence
- thresh: defines the convergence criterion based on pixel movement

The processing mode 'vector' will output a vector file, and process the input image piecewise. This allows performing segmentation of very large images. IN contrast, 'raster' mode will output a labeled raster, and it cannot handle large data. If mode is 'raster', all the 'vector_*' arguments are ignored.

Value

sf or SpatRaster

Examples

```
## Not run:
## load packages
library(link2GI)
library(OTBsegm)
library(terra)

## load sample image
image_sr <- rast(system.file("raster/pnoa.tiff", package = "OTBsegm"))

## connect to OTB (change to your directory)
otblink <- link2GI::linkOTB(searchLocation = "C:/OTB/")

## apply segmentation
results_ms_sf <- segm_lsms(
    image = image_sr,
    otb   = otblink,
    spatialr = 5,
    ranger   = 25,
    minsize  = 10
)

plotRGB(image_sr)
```

```
plot(st_geometry(results_ms_sf), add = TRUE)

## End(Not run)
```

---

segm_meanshift *Mean-Shift Segmentation*

Description

Applies the mean-shift segmentation algorithm to an image file or a SpatRaster

Usage

```
segm_meanshift(
  image,
  otb,
  spatialr = 5L,
  ranger = 15,
  thresh = 0.1,
  maxiter = 100L,
  minsize = 100L,
  mode = "vector",
  vector_neighbor = FALSE,
  vector_stitch = TRUE,
  vector_minsize = 1L,
  vector_simplify = 0.1,
  vector_tilesize = 1024L,
  mask = NULL
)
```

Arguments

image	path or SpatRaster
otb	output of link2GI::linkOTB()
spatialr	integer. Spatial radius of the neighborhood
ranger	range radius defining the radius (expressed in radiometry unit) in the multispectral space
thresh	algorithm iterative scheme will stop if mean-shift vector is below this threshold or if iteration number reached maximum number of iterations
maxiter	integer. Algorithm iterative scheme will stop if convergence hasn't been reached after the maximum number of iterations
minsize	integer. Minimum size of a region (in pixel unit) in segmentation. Smaller clusters will be merged to the neighboring cluster with the closest radiometry. If set to 0 no pruning is done
mode	processing mode, either 'vector' or 'raster'. See details

`vector_neighbor`
: logical. If FALSE (the default) a 4-neighborhood connectivity is activated. If TRUE, a 8-neighborhood connectivity is used

`vector_stitch`
: logical. If TRUE (the default), scans polygons on each side of tiles and stitch polygons which connect by more than one pixel

`vector_minsize`
: integer. Objects whose size in pixels is below the minimum object size will be ignored during vectorization

`vector_simplify`
: simplify polygons according to a given tolerance (in pixel). This option allows reducing the size of the output file or database.

`vector_tilesize`
: integer. User defined tiles size for tile-based segmentation. Optimal tile size is selected according to available RAM if NULL

`mask`
: an optional raster used for masking the segmentation. Only pixels whose mask is strictly positive will be segmented

Details

Mean-Shift is a region-based segmentation algorithm that groups pixels with similar characteristics. It's a non-parametric clustering technique that groups pixels based on spatial proximity and feature similarity (color, intensity). This method is particularly effective for preserving edges and defailt while simplifying textures in high-resolution images. Steps:

1. Initialization: Each pixel is treated as a point in a multi-dimensional space (combining spatial and color features).
2. Mean Shift Iterations: For each pixel, a search window moves toward the region with the highest data density (local maxima) by calculating the mean of neighboring pixels within the window.
3. Convergence: The process repeats until the movement of the window becomes negligible, indicating convergence.
4. Label Assignment: Pixels that converge to the same mode (local maxima) are grouped into the same region.

The most important parameters are:

- spatialr: defines the size of the neighborhood
- ranger: determines similarity in the feature space
- maxiter: limits the number of iterations for convergence
- thresh: defines the convergence criterion based on pixel movement

The processing mode 'vector' will output a vector file, and process the input image piecewise. This allows performing segmentation of very large images. IN contrast, 'raster' mode will output a labeled raster, and it cannot handle large data. If mode is 'raster', all the 'vector_*' arguments are ignored.

Value

`sf` or `SpatRaster`

Examples

```
## Not run:
## load packages
library(link2GI)
library(OTBsegm)
library(terra)

## load sample image
image_sr <- rast(system.file("raster/pnoa.tiff", package = "OTBsegm"))

## connect to OTB (change to your directory)
otblink <- link2GI::linkOTB(searchLocation = "C:/OTB/")

## apply segmentation
results_ms_sf <- segm_meanshift(
    image    = image_sr,
    otb      = otblink,
    spatialr = 5,
    ranger   = 25,
    maxiter  = 10,
    minsize  = 10
)

## End(Not run)
```

---

segm_mprofiles *Morphological profiles segmentation*

Description

Applies the morphological profiles segmentation algorithm to an image file or a SpatRaster

Usage

```
segm_mprofiles(
  image,
  otb,
  size = 5L,
  start = 1L,
  step = 1L,
  sigma = 1,
  mode = "vector",
  vector_neighbor = FALSE,
  vector_stitch = TRUE,
  vector_minsize = 1L,
  vector_simplify = 0.1,
  vector_tilesize = 1024L,
  mask = NULL
)
```

Arguments

image	path or SpatRaster
otb	output of link2GI::linkOTB()
size	integer. Size of the profiles
start	integer. Initial radius of the structuring element in pixels
step	integer. Radius step in pixels along the profile
sigma	profiles values under the threshold will be ignored
mode	processing mode, either 'vector' or 'raster'. See details
vector_neighbor	logical. If FALSE (the default) a 4-neighborhood connectivity is activated. If TRUE, a 8-neighborhood connectivity is used
vector_stitch	logical. If TRUE (the default), scans polygons on each side of tiles and stitch polygons which connect by more than one pixel
vector_minsize	integer. Objects whose size in pixels is below the minimum object size will be ignored during vectorization
vector_simplify	simplify polygons according to a given tolerance (in pixel). This option allows reducing the size of the output file or database.
vector_tilesize	integer. User defined tiles size for tile-based segmentation. Optimal tile size is selected according to available RAM if NULL
mask	an optional raster used for masking the segmentation. Only pixels whose mask is strictly positive will be segmented

Details

The morphological profiles segmentation algorithm is a region-based image segmentation technique that applies a series of morphological operations using structuring elements of increasing size to capture spatial patterns and textures within the image. Steps:

1. Morphological Filtering: The algorithm applies a sequence of openings (removing small bright structures) and closings (removing small dark structures) to the input image using structuring elements (e.g., disks, rectangles).
2. Profile Generation: It generates a profile for each pixel by recording the response of the morphological operations at different scales.
3. Feature Extraction: These profiles help capture both fine and coarse structures within the image, creating a set of features that can be used for classification or segmentation.
4. Segmentation (Optional): The extracted profiles can be input into a classifier or segmentation algorithm to differentiate between regions with distinct spatial characteristics.

The processing mode 'vector' will output a vector file, and process the input image piecewise. This allows performing segmentation of very large images. IN contrast, 'raster' mode will output a labeled raster, and it cannot handle large data. If mode is 'raster', all the 'vector_*' arguments are ignored.

Value

sf or SpatRaster

Examples

```
## Not run:
## load packages
library(link2GI)
library(OTBsegm)
library(terra)

## load sample image
image_sr <- rast(system.file("raster/pnoa.tiff", package = "OTBsegm"))

## connect to OTB (change to your directory)
otblink <- link2GI::linkOTB(searchLocation = "C:/OTB/")

## apply segmentation
results_ms_sf <- segm_mprofiles(
    image = image_sr,
    otb   = otblink,
    size  = 5,
    start = 3,
    step  = 20,
    sigma = 1
)

## End(Not run)
```

---

segm_watershed *Watershed segmentation*

Description

Applies the watershed segmentation algorithm to an image file or a SpatRaster

Usage

```
segm_watershed(
  image,
  otb,
  thresh = 0.01,
  level = 0.1,
  mode = "vector",
  vector_neighbor = FALSE,
  vector_stitch = TRUE,
  vector_minsize = 1L,
  vector_simplify = 0.1,
  vector_tilesize = 1024L,
```

```
  mask = NULL
)
```

Arguments

image	path or SpatRaster
otb	output of link2GI::linkOTB()
thresh	depth threshold units in percentage of the maximum depth in the image
level	flood level for generating the merge tree from the initial segmentation (from 0 to 1)
mode	processing mode, either 'vector' or 'raster'. See details
vector_neighbor	logical. If FALSE (the default) a 4-neighborhood connectivity is activated. If TRUE, a 8-neighborhood connectivity is used
vector_stitch	logical. If TRUE (the default), scans polygons on each side of tiles and stitch polygons which connect by more than one pixel
vector_minsize	integer. Objects whose size in pixels is below the minimum object size will be ignored during vectorization
vector_simplify	simplify polygons according to a given tolerance (in pixel). This option allows reducing the size of the output file or database.
vector_tilesize	integer. User defined tiles size for tile-based segmentation. Optimal tile size is selected according to available RAM if NULL
mask	an optional raster used for masking the segmentation. Only pixels whose mask is strictly positive will be segmented

Details

The watershed segmentation algorithm is a region-based image segmentation technique inspired by topography. It treats the grayscale intensity of an image as a topographic surface, where brighter pixels represent peaks and darker pixels represent valleys. The algorithm simulates flooding of this surface to separate distinct regions. Steps:

1. Topographic Interpretation: The input image is treated as a 3D landscape, where pixel intensity corresponds to elevation.
2. Flooding Process: Starting from local minima, the algorithm simulates water flooding the surface. As the water rises, distinct regions (basins) are formed.
3. Watershed Lines: When two basins meet, a boundary (watershed line) is formed to prevent merging.
4. Region Labeling: Each basin is assigned a unique label, producing a segmented image where boundaries are clearly defined.

The processing mode 'vector' will output a vector file, and process the input image piecewise. This allows performing segmentation of very large images. IN contrast, 'raster' mode will output a labeled raster, and it cannot handle large data. If mode is 'raster', all the 'vector_*' arguments are ignored.

Value

sf or SpatRaster

Examples

```
## Not run:
## load packages
library(link2GI)
library(OTBsegm)
library(terra)

## load sample image
image_sr <- rast(system.file("raster/pnoa.tiff", package = "OTBsegm"))

## connect to OTB (change to your directory)
otblink <- link2GI::linkOTB(searchLocation = "C:/OTB/")

## apply segmentation
results_ms_sf <- segm_watershed(
    image  = image_sr,
    otb    = otblink,
    thresh = .1,
    level  = .2
)

## End(Not run)
```

Index