Package: nmhsa (via r-universe)

August 25, 2024

Version 0.0.0.9000

Title Nested multiresolution hierarchical simulated annealing

Description Porous media reconstruction using an advanced simulated annealing approach.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.2

Config/reticulate list(packages = list(list(package = ``scipy"), list(package = ``numpy")))

Imports reticulate, rui (>= 0.0.0.9000)

SystemRequirements Python (>= 3.0.0)

Depends R (>= 2.10)

Remotes rogiersbart/rui

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://rogiersbart.github.io/nmhsa

BugReports https://github.com/rogiersbart/nmhsa/issues

Repository https://rogiersbart.r-universe.dev

RemoteUrl https://github.com/rogiersbart/nmhsa

RemoteRef HEAD

RemoteSha 09980d8976303987d00ec89f6911054e9b66734d

Contents

`cement` *A 2D cement paste image*

Description

A 2D segmented cement paste image, as an integer matrix, which carries the `nmhsa_array` S3 class for more friendly printing and plotting.

Usage

```
cement
```

Format

An integer matrix of 128 rows and 128 columns, with the following labels for the different segmented phases:

1. Pores
2. Portlandite
3. Clinker
4. CSH

Source

https://doi.org/10.1103/PhysRevE.100.053316

hsa *Hierarchical simulated annealing*

Description

This is the simplest algorithm available at the moment, which is capable of multiphase reconstruction using the hierarchical approach (phases are added one by one).

Usage

```
hsa(
  ti,
  dimensions = NULL,
  start_after = NULL,
  stop_after = NULL,
  stop_rejected = NULL,
  stop_at = 1e-06,
  order = NULL,
  distance = NULL,
  cool = 0.9
)
```

Arguments

`ti` The original training image (2D array).

`dimensions` Dimensions for the reconstruction (vector of length 2 or 3). Defaults to the dimensions of `ti`.

`start_after` Number of iterations for determining the initial simulated annealing temperature. Defaults to 0.1% of the total amount of pixels in the reconstruction.

`stop_after` Maximum number of iterations. Defaults to the amount of pixels in the reconstruction. Can be a vector for different values per phase.

`stop_rejected` Maximum number of consecutively rejected iterations. Defaults to 1% of the total amount of pixels in the reconstruction.

`stop_at` Target precision. Defaults to `1e-6`.

`order` Integer vector with the phase order for hierarchical simulation. Defaults to the least to the most occurring phase.

`distance` Distance, in pixels, up to which to investigate the structural descriptors. Defaults to 1/4th of the minimum of `dim(ti)` and `dimensions`.

`cool` Simulated annealing cooling factor. Must be lower than one. Defaults to `0.9`.

Value

A reconstructed 2D or 3D array.

References

https://doi.org/10.1103/PhysRevE.100.053316

See Also

mhsa(), nmhsa()

---

mhsa *Multiresolution hierarchical simulated annealing*

Description

This algorithm extends hsa() by introducing different grid levels, or resolutions, which should result in improved and more efficient reconstruction of larger particles.

Usage

```
mhsa(
  ti,
  dimensions = NULL,
  start_after = NULL,
  stop_after = NULL,
  stop_rejected = NULL,
  stop_at = 1e-06,
  order = NULL,
  distance = NULL,
  cool = 0.9,
  levels = 2
)
```

Arguments

ti	The original training image (2D array).
dimensions	Dimensions for the reconstruction (vector of length 2 or 3). Defaults to the dimensions of ti.
start_after	Number of iterations for determining the initial simulated annealing temperature. Defaults to 0.1% of the total amount of pixels in the reconstruction.
stop_after	Maximum number of iterations. Defaults to the amount of pixels in the reconstruction. Can be a vector for different values per phase.
stop_rejected	Maximum number of consecutively rejected iterations. Defaults to 1% of the total amount of pixels in the reconstruction.
stop_at	Target precision. Defaults to 1e-6.
order	Integer vector with the phase order for hierarchical simulation. Defaults to the least to the most occurring phase.

distance — Distance, in pixels, up to which to investigate the structural descriptors. Defaults to 1/4th of the minimum of `dim(ti)` and `dimensions`.

cool — Simulated annealing cooling factor. Must be lower than one. Defaults to `0.9`.

levels — Amount of grid levels (or resolutions) to use for the reconstruction. Defaults to `2`. Can be a vector to use different levels for different phases.

Value

A reconstructed 2D or 3D array.

References

https://doi.org/10.1103/PhysRevE.100.053316

See Also

`hsa()`, `nmhsa()`

nmhsa *Nested multiresolution hierarchical simulated annealing*

Description

This algorithm extends `mhsa()` by introducing the nested approach, meaning it automatically handles identified subphases (which are split by the user, and merged again by the algorithm), and allows handling things like inclusions by a splitting step at the end of the simulation (phases merged, or rather rearranged, by the user, and split again by the algorithm).

Usage

```
nmhsa(
  ti,
  dimensions = NULL,
  start_after = NULL,
  stop_after = NULL,
  stop_rejected = NULL,
  stop_at = 1e-06,
  order = NULL,
  distance = NULL,
  cool = 0.9,
  levels = 2,
  ti_merged = NULL,
  ti_splitted = NULL,
  merge_pairs = NULL,
  merge_distance = NULL,
  merge_stop_after = NULL,
  split_distance = NULL,
  split_stop_after = NULL
)
```

Arguments

`ti` The original training image (2D array).

`dimensions` Dimensions for the reconstruction (vector of length 2 or 3). Defaults to the dimensions of `ti`.

`start_after` Number of iterations for determining the initial simulated annealing temperature. Defaults to 0.1% of the total amount of pixels in the reconstruction.

`stop_after` Maximum number of iterations. Defaults to the amount of pixels in the reconstruction. Can be a vector for different values per phase.

`stop_rejected` Maximum number of consecutively rejected iterations. Defaults to 1% of the total amount of pixels in the reconstruction.

`stop_at` Target precision. Defaults to `1e-6`.

`order` Integer vector with the phase order for hierarchical simulation. Defaults to the least to the most occurring phase.

`distance` Distance, in pixels, up to which to investigate the structural descriptors. Defaults to 1/4th of the minimum of `dim(ti)` and `dimensions`.

`cool` Simulated annealing cooling factor. Must be lower than one. Defaults to `0.9`.

`levels` Amount of grid levels (or resolutions) to use for the reconstruction. Defaults to `2`. Can be a vector to use different levels for different phases.

`ti_merged` Merged version of `ti`. Typically obtained through `phase_merge()`.

`ti_splitted` Splitted version of `ti_merged`. Typically obtained through `phase_split()`.

`merge_pairs` List of length 2 vectors, indicating for which pairs of phases optimisation should again be performed after merging all subphases again to their original phase (transition from the state of `ti_splitted` to the state of `ti_merged`).

`merge_distance` Distance, in pixels, up to which to investigate the structural descriptors for the merging step. Defaults to 1/4th of the minimum of `dim(ti)` and `dimensions`. Should be a vector of the same length as `merge_pairs`.

`merge_stop_after` Maximum number of iterations for the merging step. Defaults to the amount of pixels in the reconstruction. Can be a vector for different values per phase.

`split_distance` Distance, in pixels, up to which to investigate the structural descriptors for the splitting step. Defaults to 1/4th of the minimum of `dim(ti)` and `dimensions`.

`split_stop_after` Maximum number of iterations for the splitting step. Defaults to the amount of pixels in the reconstruction.

Details

A more general approach to the phase merging and splitting would be possible, where the user has control of every step in the simulation, but this is currently not implemented.

Value

A reconstructed 2D or 3D array.

References

https://doi.org/10.1103/PhysRevE.100.053316

See Also

`hsa()`, `mhsa()`, `phase_merge()`, `phase_split()`

`phase_merge` *Merge neighbouring pixels*

Description

This functionality was originally introduced to remove inclusions of certain phases within another phase. We are however simply looking for neighbouring pixels here. As this is still effective with small inclusions, for larger ones, repeated calls of this function may be required. Note however that this also affects pixels of the phase to modify that are not inclusions but just neighbouring pixels of the target phase.

Usage

```
phase_merge(img, phase, into)
```

Arguments

`img`	A 2D array.
`phase`	The phase to modify.
`into`	The phase to change pixels to.

Value

The modified 2D array.

See Also

`phase_split()`, `nmhsa()`

`phase_split` *Split into subphases*

Description

This function splits a certain phase into two subphases based on a size threshold, which makes sense in the reconstruction framework if the two subphases behave different structurally.

Usage

```
phase_split(img, phase, larger_than, into = max(c(img)) + 1)
```

Arguments

`img`	A 2D array.
`phase`	The phase to modify.
`larger_than`	The threshold for splitting (in pixels).
`into`	The phase to change pixels to.

Value

The modified 2D array.

See Also

`phase_merge()`, `nmhsa()`

`wrapper` *Access to the Python wrapper examples*

Description

Access to the Python wrapper examples

Usage

```
wrapper(algorithm)
```

Arguments

`algorithm`	One of "hsa", "mhsa", "nmhsa" or "3d".

Value

A 2D or 3D array

Index