scipy.ndimage.filters.minimum_filter

scipy.ndimage.filters.**minimum_filter(**_input, size=None, footprint=None, output=None, mode='reflect', cval=0.0, origin=0_**)**   **[source]** (http://github.com/scipy/scipy/blob/v0.14.0/scipy/ndimage/filters.py#L913)

Calculates a multi-dimensional minimum filter.

Parameters: **input** : *array_like*

> Input array to filter.

size : *scalar or tuple, optional*

> See footprint, below

footprint : *array, optional*

> Either *size* or *footprint* must be defined. *size* gives the shape that
> is taken from the input array, at every element position, to define
> the input to the filter function. *footprint* is a boolean array that
> specifies (implicitly) a shape, but also which of the elements within
> this shape will get passed to the filter function. Thus `size=(n, m)` is
> equivalent to `footprint=np.ones((n, m))`. We adjust *size* to the
> number of dimensions of the input array, so that, if the input array
> is shape (10,10,10), and *size* is 2, then the actual size used is (2,2,2).

output : *array, optional*

> The *output* parameter passes an array in which to store the filter
> output.

mode : *{ 'reflect' , 'constant' , 'nearest' , 'mirror' , 'wrap' }, optional*

> The *mode* parameter determines how the array borders are
> handled, where *cval* is the value when mode is equal to
> 'constant' . Default is 'reflect'

cval : *scalar, optional*

> Value to fill past edges of input if *mode* is 'constant' . Default is
> 0.0

origin : *scalar, optional*

> The *origin* parameter controls the placement of the filter. Default
> 0.0.

Previous topic

Next topic