

# MSPA GUIDE

[Peter Vogt](#)

European Commission, Joint Research Centre  
I-21027 Ispra (VA), Italy

This document provides information on the usage of *Morphological Spatial Pattern Analysis* ([MSPA](#)); its implementation in the MSPA-standalone version and in the [GuidosToolbox](#) software package, the MSPA input data requirements, the MSPA-Parameter settings, and the resulting MSPA output data.

## 1. MSPA input data

The input data must be a raster (grid) map of the target or study area. Vector (polygon) maps are not supported; please convert them to raster format first. The maximum supported map file size for direct MSPA-processing depends on the operating system:

MS-Windows: ~ 10000×10000 pixels for a square map.

Linux/Mac: depending on the available free RAM, ~25000×25000 for 16GB RAM and 75000×75000 for 128GB RAM. Images larger than the maximum size for MSPA can be processed via a buffered tiling process called *MSPA-tiling*, more details are available in the [Help → GuidosToolbox Manual](#).

The map does not have to be square but it is necessary that both, the number of rows and columns, are at least 10 and the total number of pixels does not exceed the upper limit outlined above.

The input map **must** be of type **BYTE (8-bit)** and **must** contain the two data classes **Foreground** and **Background**. In addition, it **may** contain one data class **Missing**. The MSPA segmentation will be performed for the Foreground data class, and the class Missing will be ignored during this process. The particular definitions of Foreground, Background, and Missing depend on your situation. For example, to classify forest patterns on a forest/non-forest map, the Foreground is forest and the Background is non-forest. To classify wetland patterns on a wetland map, the Foreground is wetland and the Background is non-wetland. In general, Foreground corresponds to the features which should be classified, Background is complementary to Foreground, and in addition there may be data which are neither Foreground nor Background, those belong in the class Missing. Please use your preferred GIS or image processing software to recode the pixels (cells) in your input map as follows:

0 byte = Missing (optional)

1 byte = Background (mandatory)

2 byte = Foreground (mandatory)

Further notes on the input map format:

No color code is needed. Export your (recoded) map as an 8-bit GeoTiff, or 8-bit tiff. Please use either no compression or LZW compression. Please ensure that your exported map is an 8-bit (geo)-tif because some software may not automatically produce 8-bit for maps with cell values of {0,1,2}. It may be possible to process other image format types (png, gif, etc.) but we strongly suggest to use the default image type tif, or when passing geo-information GeoTiff. You will only need the 8-bit tif/GeoTiff file, **not** any additional file created by your software like tfw, .hdr, etc.

## 2. MSPA-Parameters

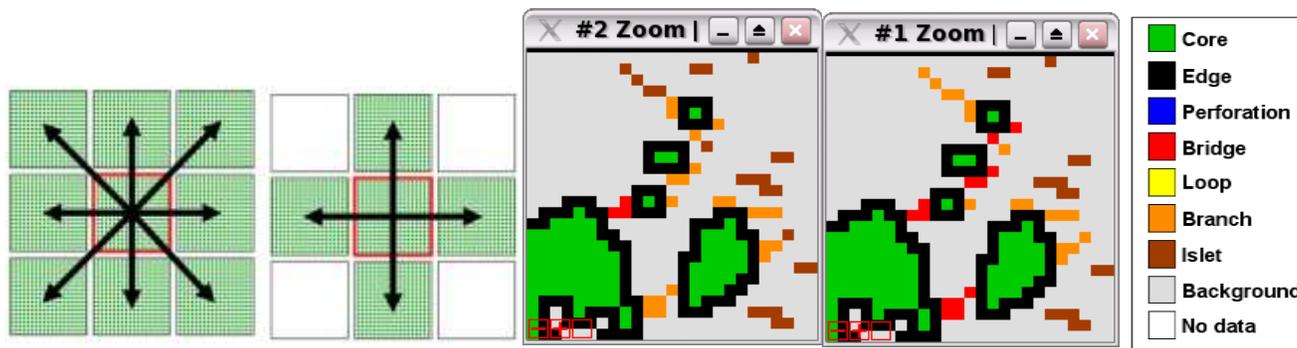
The result of the spatial pattern analysis depends on the settings of the four MSPA-Parameters described below:

### MSPA-Parameter 1: *Foreground Connectivity* (Options: 8, 4)

For a set of 3×3 pixels the center pixel (surrounded in red below) is connected to its adjacent neighboring pixels by having either,

- a) a pixel border and a pixel corner in common (8-connectivity) or,
- b) a common pixel border only (4-connectivity).

The two sets of images below show 8- and 4-connectivity and its impact on the MSPA segmentation results.

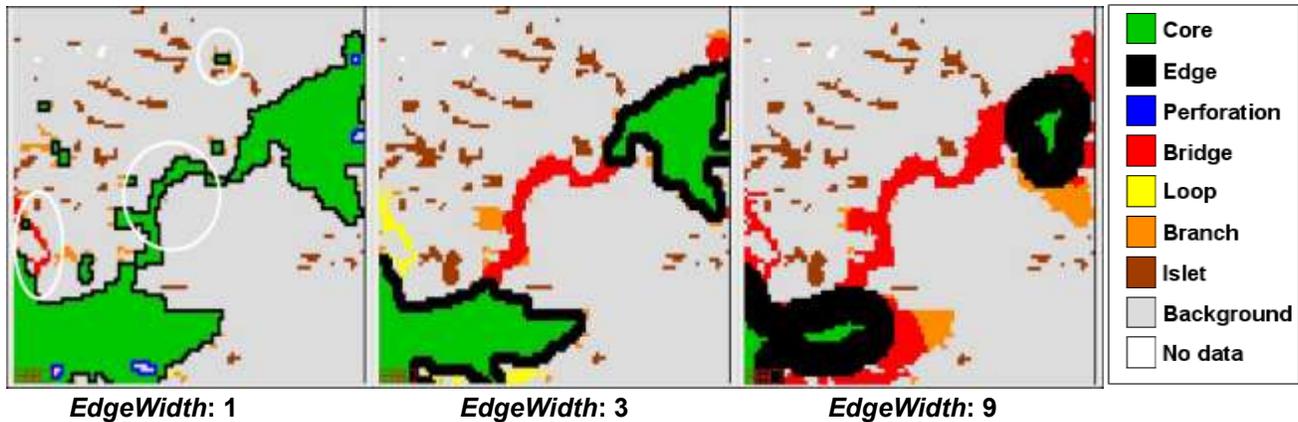


**8-connectivity ↔ 4-connectivity      MSPA segmentation: 8100 ↔ 4100**  
**Parameter 1: *Foreground Connectivity* (<*Foreground Connectivity*>, 1, 0, 0).**

***The default value for MSPA-Parameter 1, Foreground Connectivity, is 8.***

## **MSPA-Parameter 2: *EdgeWidth* (Options: 1, 2, 3, 4, ...)**

The ***EdgeWidth*** parameter defines the width or thickness of the non-core classes in pixels. The actual distance corresponds to the number of edge pixels multiplied by the pixel resolution of the data.



**Parameter 2: *EdgeWidth* (8, <EdgeWidth>, 0, 0).**

The ***EdgeWidth*** distance used in MSPA corresponds to the radius of a Euclidean disk. Imagine to insert a pin in the center point of this disk and drive the pin along the perimeter line of a Foreground object. Those Foreground pixels which are within the radius of this disk are boundary pixels (edge + perforation) and those with a distance from the pin larger than the radius of the disk are core pixels.

For example, if the spatial resolution of your input data is 25m and you wish to get a 100m wide boundary in MSPA (4 pixels thick), then  $EdgeWidth = 4$ . Within GuidosToolbox you can use a drop-down menu to select a predefined or insert a custom value for the MSPA-Parameter 2 ***EdgeWidth***.

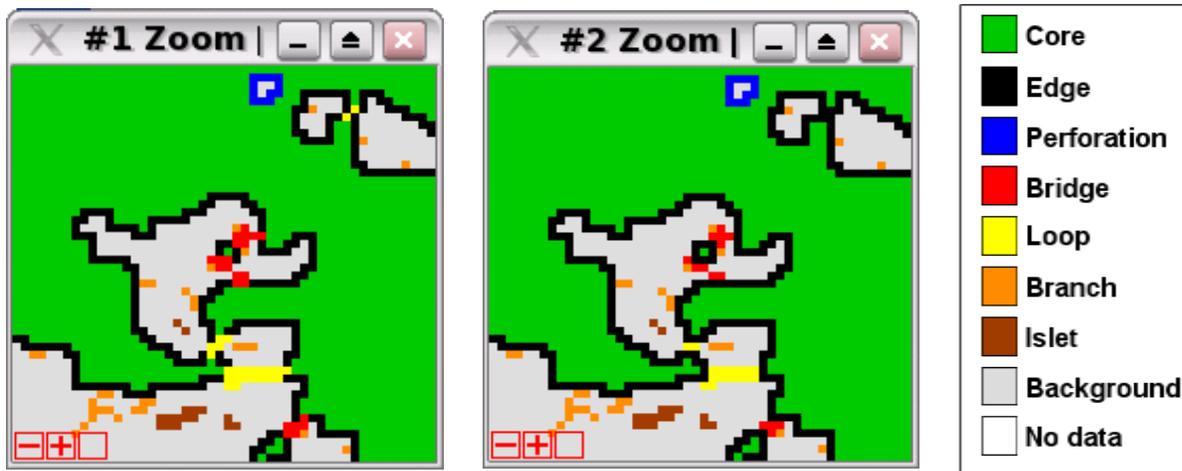
An increase of the width of the non-core classes has two effects:

- a) the remaining core area will decrease and may even disappear,
- b) relatively small image components may change their spatial pattern class.

***The default value for MSPA-Parameter 2, EdgeWidth, is 1.***

### MSPA-Parameter 3: *Transition* (Options: 0, 1)

Transition pixels are those pixels of an edge or a perforation where the core area intersects with a loop or a bridge. If ***Transition*** is set to 0 ( $\leftrightarrow$  do *not* show transition pixels) then the perforation and the edges will show closed core boundaries. Note that a loop or a bridge of length 2 will not be visible for this setting since it will be hidden under the edge/perforation.



Transition: On

Transition: Off

Parameter 3: *Transition* (8, 1, <*Transition*>, 0).

Left: *Transition* on (8110). Right: *Transition* off (8100).

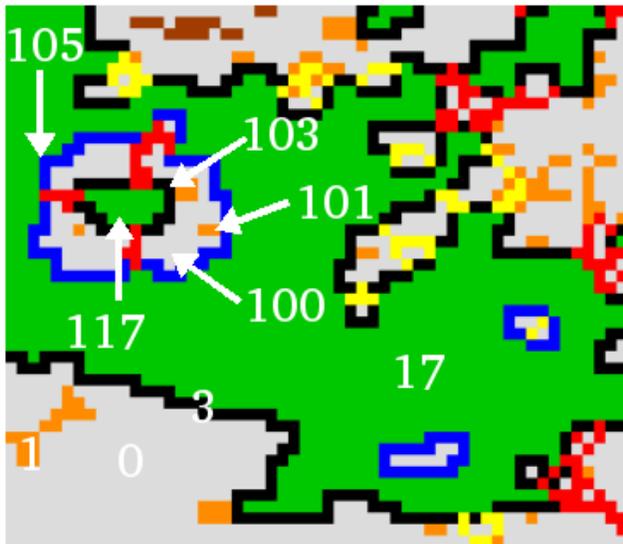
The default is to show (***Transition*** = 1) transition pixels in order to illustrate all detected connections. For larger ***EdgeWidth*** parameters, the connectors as well as the area covered by transition pixels strongly increases. While the results are mathematically correct, they may look confusing for the unexperienced user. The figure above shows the impact of ***Transition*** on/off for the settings of ***Foreground Connectivity***, ***EdgeWidth*** and ***Intext***: 8110/8100. Note how for ***Transition***=0 the loop (yellow pixels) in the upper right part of the image is hidden under the edge pixels (black).

Changing the value for ***Transition*** will use a different color table only. The actual pixel values of the MSPA segmentation are independent of the selected setting for ***Transition*** (see also the table in section 3 below).

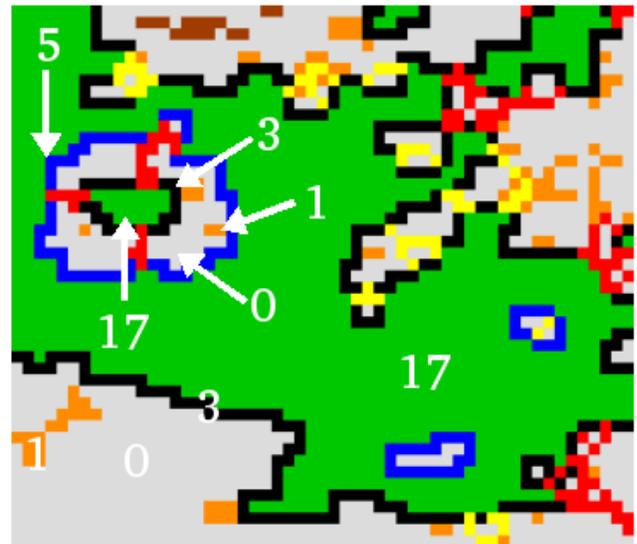
***The default value for MSPA-Parameter 3, Transition, is 1.***

### MSPA-Parameter 4: *Intext* (Options: 0, 1)

The parameter *Intext* allows separating internal from external features, where internal features are defined as being enclosed by a Perforation. The default is to enable this distinction, which will add a second layer of classes to the seven basic classes. All classes, with the exception of Perforation, which by default is always internal (105b), can then appear as internal or external.



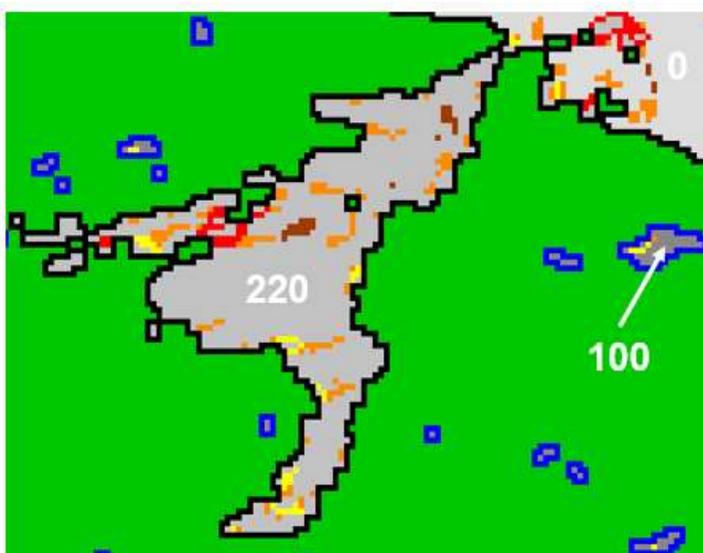
Intext: 1 (On)



Intext: 0 (Off)

Parameter 4: *Intext* (8, 1, 1, <*Intext*>). Left: *Intext* on (8111). Right: *Intext* off (8110).

The image above shows the difference in foreground pixel values only. In addition, and when using the default setting *Intext=1* the internal Background is segmented into the two categories **Core-Opening** and **Border-Opening** (see below).



- Background: 0 byte outside Foreground
- Border-Opening: 220 byte surrounded by FG-pixels (mainly Edge)
- Core-Opening: 100 byte surrounded by blue Perforation pixels (inside Core)

*The default value for MSPA-Parameter 4, Intext, is 1.*

### **3. MSPA output data**

MSPA will conduct a segmentation of the input image Foreground objects. It will assign each and every Foreground pixel into one of the mutually exclusive geometric feature classes. This principle implies that first, the spatial coverage of the initial Foreground and the resulting MSPA classes is identical, and second, MSPA is a true mathematical segmentation due to the assignment in mutually exclusive classes comprising the Foreground area. Please note that MSPA will only conduct a **segmentation** of the input data, it does not do magic and interpret the pattern, nor can it find user-desired features which are not available in the input data. It is up to the user to provide appropriate input data containing all features of interest, as well as doing the final interpretation of the pattern classes provided by MSPA.

The MSPA result is visualized using the following colors:

- 7 colors: corresponding to the pattern classes of Foreground,
- 3 gray tones: corresponding to external Background, Border-Opening, Core-Opening,
- 1 color white: corresponding to Missing, if this class was in the input map.

The legend for these 11 colors is listed in the table below. Please note that:

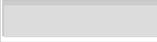
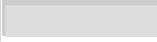
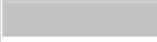
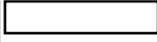
- data area = Foreground area (class 1 → 7) + Background area (gray)
- image area = data area + Missing (white)

#### **Naming scheme of the resulting MSPA classes:**

The resulting spatial pattern classes have a generic naming scheme. The meaning of these classes depends on the nature of the input data and the class names should be amended by the user. For example, the class Perforation denotes Foreground pixels which form the outer perimeter of a 'hole' in a Foreground area: For a forest mask, the area enclosed by a perforation could be a 'clearing' while for a water mask such an area would be an 'island'.

#### **Details on the resulting image:**

The default colors and legend do not fully describe the actual byte values in the output image, because some of the byte values have been combined into one RGB/pattern class for visual simplicity. You can read out the actual pixel values either in GuidosToolbox via the mouse pointer or after importing the output image into another image processing software. The table below lists the spatial pattern classes, their colors, RGB-, and image byte values. Note that the boundary intersecting pixels of Loop and Bridge classes are each represented by three different byte values in order to distinguish between the Loops and Bridges and their intersections with Edge and Perforation:

Class	Color	RGB	Value [byte] internal/external
1) Core		000/200/000	117 / 17
2) Islet		160/060/000	109 / 9
3) Perforation		000/000/255	105 / 5
4) Edge		000/000/000	103 / 3
5a) Loop		255/255/000	165 / 65
5b) Loop in Edge		255/255/000	167 / 67
5c) Loop in Perforation		255/255/000	169 / 69
6a) Bridge		255/000/000	133 / 33
6b) Bridge in Edge		255/000/000	135 / 35
6c) Bridge in Perforation		255/000/000	137 / 37
7) Branch		255/140/000	101 / 1
Background		220/220/220	0
Opening		220/220/220	0 (if Intext=0)
Border-Opening		194/194/194	220 (if Intext=1)
Core-Opening		136/136/136	100 (if Intext=1)
Missing		255/255/255	129 / 129

**Class names, color codes, and byte values for MSPA and other per-pixel categories.**

The table above shows the color code for **Transition**=1. For **Transition**=0, the intersecting pixels of class 5b and 6b are color coded as Edge (black) and the pixels of class 5c and 6c are color coded as Perforation (blue). The statistics in GuidosToolbox are calculated for closed boundaries (**Transition**=0) regardless of the actual **Transition** setting.

**Intext** = 0: the resulting MSPA classes can only have the byte values of the right column in the table above. Background and Openings are not distinguished.

**Intext** =1 (default): all classes can appear as internal or external. This naturally excludes the pixels of the class Perforation, which by definition are always internal. The background is segmented into the three classes Background, Core-Opening and Border-Opening.

In summary, the visual result can have a total of up to 7 unique Foreground colors plus 3 gray tones for the Background area and the color white for Missing or no data areas.

The actual unique image pixel values for **Intext**=0 can be up to 13 (11 for Foreground + 1 for Background + 1 for Missing) and for **Intext**=1 up to 25 (21 for Foreground + 3 for Background + 1 for Missing).

## 4. MSPA-standalone version

The command-line version of MSPA can be used as a standalone application. It is intended for advanced users only, having a very clear understanding of the MSPA parameters and required MSPA-compliant input data. **Beginners should not use the standalone version but the one in GuidosToolbox instead.**

GuidosToolbox acts as a wrapper around the standalone version and handling the **4 limitations of the standalone version**:

1. No support for [tiled tiff images](#) and writing of geoheader (GeoTiff) information,
2. No verification if input data is MSPA-compliant,
3. No summary statistic,
4. No MSPA-Tiling.

This implies that certain tiff file-types, or input data that is not MSPA-compliant, will simply crash the standalone version.

Within GuidosToolbox, the directory *MSPAstandalone* contains operating system specific mspa-executables for 32/64bit MS-Windows, 64bit MacOS, 32/64bit Linux.

The directory *MSPAstandalone* can be found in [GuidosToolbox/idl/save\\_add/](#) (Linux), [GuidosToolbox/GuidosToolbox.app/Contents/Resources/idl/save\\_add/](#) (MacOS), and in [GuidosToolbox](#) (MS-Windows).

To use the standalone version copy the operating system specific mspa-executable into the directory where you have your data and run it to get an overview of the processing options, i.e.:

```
peter$./mspa_mac
```

```
-----  
Morphological Spatial Pattern Analysis (MSPA)  
by {Pierre.Soille,Peter.Vogt}@ec.europa.eu  
Version 2.2, December 2017  
(c) European Commission, Joint Research Centre
```

### References:

P. Soille & P. Vogt, 2009: Morphological Segmentation of Binary Patterns. Pattern Recognition Letters,  
DOI: <http://dx.doi.org/10.1016/j.patrec.2008.10.015>

P. Vogt & K. Riitters, 2017: GuidosToolbox: universal digital image object analysis. European Journal of Remote Sensing (TEJR),  
DOI: <http://dx.doi.org/10.1080/22797254.2017.1330650>

### Disclaimer:

```
=====
```

1. The use of the software of this interface is made available for non-commercial purposes. For commercial purposes please contact: [Peter.Vogt@ec.europa.eu](mailto:Peter.Vogt@ec.europa.eu)
2. The JRC, on behalf of the Commission does not accept any liability whatsoever for any error, or for any loss or damage arising from the use of the software of this interface.
3. If the results of this interface are passed on to third parties it has to be assured that the conditions of the disclaimer are agreed and fulfilled.

```
-----  
Usage : ./mspa_mac
```

[-v] [-usage] [-help] [-i <input>] [-o <output>] [-eew <psize>] [-graphfg <pgraphfg>]  
[-internal <internal>] [-odir <output\_dir>] [-transition <transition>]  
[-disk]

Parameters:

-----  
-v : Flag for verbose mode.  
-usage : Flag for displaying program usage.  
-help : Flag for help.  
-i : Name of the input image.  
-o : Name of the output image; default: "out.tif".  
-eew : Effective Edge Width; default: 1.000000.  
-graphfg : Foreground connectivity; default: 8.  
-internal : Internal flag; default: 1.  
-odir : Name of the output directory; default: "./".  
-transition : Transition flag; default: 1.  
-disk : Flag for minimum RAM usage

For example, to process your image with the default MSPA-parameters the command would be: `./mspa_mac -i input.tif -o output.tif -odir ./`

or to process your image with 3-pixel wide edges and not showing transition pixels (to show closed boundaries): `./mspa_mac -i input.tif -eew 3 -transition 0 -o output.tif -odir ./`

Note: You can safely ignore the warning message in the terminal output.

The maximum image size that can be processed without having to tile:

- **MS-Windows:** 10000×10000 pixels (use the option "-disk"). Any Windows-MSPA version is significantly slower compared to its Linux/Mac equivalent.
- **Linux 32bit:** 12000×12000 pixels (use the option "-disk")
- **Linux 64bit/Mac OSX:** depends on the amount of free RAM on your PC:
  - ~ 19 x uncompressed image size (with option "-disk")
  - ~ 22 x uncompressed image size (without option "-disk")

Example: MSPA-compliant input image dimension is 75000×75000 pixels → 5.24 GB uncompressed image size → requiring 100 GB free RAM. If you have a PC with 128 GB RAM and Linux (64bit) or MacOS then you could process an image of that size. In general, the maximum size of a square image of dimension Z (ZxZ pixels) that can be processed is:  $Z = \sqrt{1024^3 / 20 * RAM}$ . For example, on a laptop with 16 GB RAM and 64bit Linux or MacOS: 2 GB for the operating system, 14 GB free RAM; then  $Z = \sqrt{1024^3 / 20 * 14} = 27400$  pixels, meaning you can process an image of 27400×27400 pixels. Indicative processing times from such a laptop having these specs and for processing an image of 10000×10000 (0:52 mins), 20000×20000 (3:50 mins). On Linux/Mac OS the maximum image size for MSPA processing is calculated on the fly and displayed in the *Help* → *About GuidosToolbox* window.

More details on the concepts of MSPA are provided in:

Soille, P. and Vogt, P. (2009): [Morphological segmentation of binary patterns](#). Pattern Recognition Letters. DOI: 10.1016/j.patrec.2008.10.015

P. Vogt & K. Riitters, 2017: [GuidosToolbox: universal digital image object analysis](#). European Journal of Remote Sensing (TEJR), DOI: <http://dx.doi.org/10.1080/22797254.2017.1330650>