

## Tutorial: GANIP

### Note

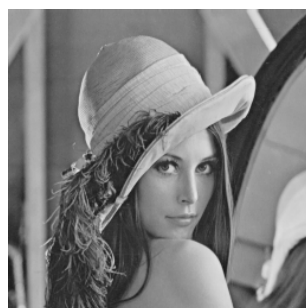
This tutorial aims to test the elementary operators of the GANIP framework.

The General Adaptive Neighborhood Image Processing (GANIP) is a mathematical framework for adaptive processing and analysis of gray-tone and color images. An intensity image is represented with a set of local neighborhoods defined for each pixel of the image to be studied. Those so-called General Adaptive Neighborhoods (GANs) are simultaneously adaptive with the spatial structures, the analyzing scales and the physical settings of the image to be addressed and/or the human visual system. The following figure illustrates the GANs of two points on an image of retinal vessels.

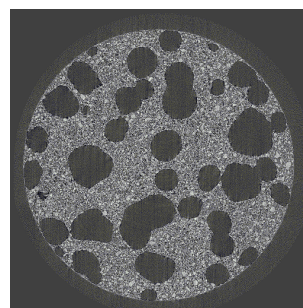


The GANs are then used as adaptive operational windows for local image transformations (morphological filters, rank/order filters...) and for local image analysis (local descriptors, distance maps...).

The different processes will be applied on the following gray-tone images:



(a) lena



(b) cement paste (2-D section / X-ray tomography)

# 1 GAN

Let  $f$  be a gray-tone image. The GAN of a point  $x$  using the luminance criterion and the homogeneity tolerance  $m$  within the CLIP framework is defined as:

$$V_m^f(x) = C_{\{y; |f(y) - f(x)| \leq m\}}(x) \quad (1)$$

where  $C_X(x)$  denotes the connected component of  $X$  holding  $x$ .



1. Load the image 'lena' and compute the GAN of a selected point in the image.
2. Look at the influence of the homogeneity tolerance.
3. Compute the GAN of different points and comment.

# 2 GAN Choquet filtering

The Choquet filters generalize the rank-order filters.

In this exercise, we are going to implement some GAN Choquet filters such as the GAN mean operator. For each point  $x$  of the image, the mean value of all the intensities of the points inside the GAN of  $x$  is computed. So, a first algorithm consists in making a loop on the image points for computing the different GAN and the mean intensity values, but it is time consuming. Nevertheless, by using some properties of the GAN (particularly the one giving that iso-valued points can have exactly the same GAN), it is possible to create a second algorithm by making a loop on the gray-tone range:

**Data:** original (8-bit) image  $f$ , homogeneity tolerance  $m$

**Result:** GAN mean filtered image  $g$

**for**  $s = 0$  **to** 255 **do**

$seed$  = points  $x$  with intensity  $f(x) = s$ ;

$thresh$  = points  $y$  with intensity satisfying  $s - m \leq f(y) \leq s + m$ ;

$threshGAN$  = connected components of  $thresh$  holding  $seed$ ;

**foreach**  $label$  of  $threshGAN$  **do**

$currentLabel$  = current connected component of  $threshGAN$ ;

$meanValue$  = intensity mean value of the image points inside  $currentLabel$ ;

**set**  $g(x) = meanValue$  to the points  $x$  of  $seed$  belonging to  $currentLabel$ ;

**end**

**end**



1. Implement the proposed algorithm.

2. Test this operator on the 'lena' image with different homogeneity tolerances.
3. Compare the result with a classical mean filtering.



See `imfilter` .



See `scipy.ndimage.filters` .

### 3 GAN morphological filtering

As previously explained, it is also possible to compute the GAN dilation and GAN erosion by making a loop on the gray-tone range. The algorithm for the GAN dilation is:

**Data:** original (8-bit) image  $f$ , homogeneity tolerance  $m$

**Result:** GAN dilated image  $g$

**set**  $g(x) = 0$  for all points  $x$ ;

**for**  $s = 0$  **to** 255 **do**

$seed$  = points  $x$  with intensity  $f(x) = s$ ;

$thresh$  = points  $y$  with intensity satisfying  $s - m \leq f(y) \leq s + m$ ;

$threshGAN$  = connected components of  $thresh$  holding  $seed$ ;

**foreach** label of  $threshGAN$  **do**

$currentLabel$  = current connected component of  $threshGAN$ ;

$maxValue$  = intensity maximum value of the image points inside

$currentLabel$ ;

**set**  $g(x) = \max(g(x), maxValue)$  to the points  $x$  belonging to  $currentLabel$ ;

**end**

**end**

The algorithm for the GAN erosion is similar.



1. Implement the GAN dilation and GAN erosion.
2. Test this operator on the 'lena' image with different homogeneity tolerances.
3. Compare the results with the classical morphological dilation and erosion using the Matlab functions `imdilate` and `imerode`.
4. Compute, test, compare and check the properties (idempotence, extensivity/anti-extensivity) of the GAN opening and GAN closing.