POLITECNICO
MILANO 1863

Reference based defect detection using foveated and classical NL-means

Professor: Boracchi Giacomo

Student: Bertarini Andrea

Introduction

- Two images:
  - Ref : image without defect
  - Src : Image with a possible defect, it is the image the algorithm has to check
- Reconstruction of pixels in the src as a weighted average of pixel in the ref exploiting two patch similarity measure
  - Windowed-distance from NL-means
  - Foveated-distance from foveation and foveated-NLM
- Many fields of application
  - Define error rate in a product
  - Helps in parameters tuning
  - Discriminate good/bad products
- Our solution is build upon the method proposed by M. Zontak for anomaly detection in silicon wafer
  - In practice it may works in multiple cases
  - It does not need image registration

NL-means

Compute the value of a pixel as a weighted sum of other pixels

$$v(s) = \frac{1}{\sum \boxed{w(s,s')}} * \sum_{s' \in \boxed{Ns}} w(s,s') * v(s')$$

W(s,s') represent the weight betwen pixel s ans s', the more they are distant the more the weight is low

Search neighborhood, limit the used pixel to a small area due to computational cost, without loosing in performance

NL-means

This represent the windowed quadratic distance

$$w(s,s') = e^{\frac{\boxed{||\boxed{k}*P_s - k*\boxed{P_{s'}}||_2^2}}{\boxed{h}^2}}$$

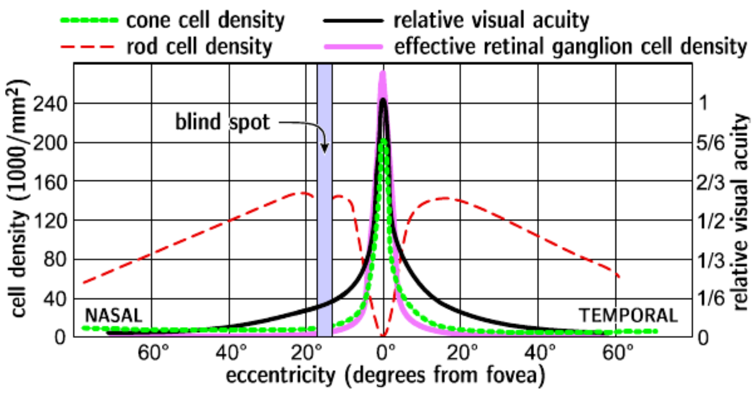
K is a kernel, it is used to weight pixels in the patch, the idea is to have a kernel that weight more the pixels in the center of the patch. A good choice is a gaussian kernel.

The patch is a square around of the pixel, the dimension depends on the application and may change the performance of the algorithm.

h is a filtering parameter controlling weight decay

Foveation

- Based on a peculiarity of Human Visual System (HVS)
- Maximal acuity in the middle of the retina called fovea or fixation point
- Acuity decrease toward the perifery, implies an increase of blur
- Improve the reconstruction quality for high level of noise



Graph representing the density of cells in the retina

Foveated image of Lena at two different fixation point

Foveated Patch Distance

- Replace windowed-distance with foveated-distance

Foveated patch of pixel x

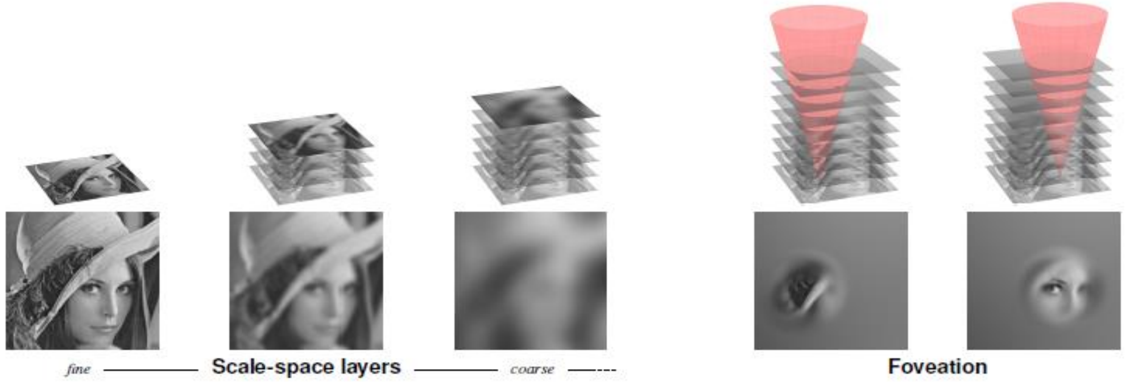
F is the foveation operator, it receive as input an image v and a fixation point point x and outputs a foveated patch

$$\boxed{z_x^{FOV}} = \boxed{F[v, x]}$$

$$w(s, s') = e^{\frac{||z_s^{Fov} - z_{s'}^{Fov}||_2^2}{h^2}}$$

Foveated Patch Distance - How F works

- Scale Space Layer : convolve the image with a set of kernels, to get a stack of progressively blurred images
- Imagine F as cone, make its vertex overlapping with the fixation point on the less blurred image in the stack, the patch is composed by pixels intersecting the cone



Reference Based Defect Detection

- Proposed by M. Zontak
- Reconstruct pixel in the src using the ref, then detect anomalous region
- The algorithm is composed by two main procedure:
  - Reconstruction
  - Defect Detection

Reference Based Defect Detection - Reconstruction

- Based on NL-means
- Compute weight using windowed-distance

The weights are computed using the
Windowed-distance

The search neighborhood
Is moved to the ref image

$$\hat{v}_{src}(s) = \frac{1}{w(s,s')} * \sum_{\boxed{s' \in Ns}} \boxed{w(s,s')} * v_{ref}(s')$$

Reference Based Defect Detection - Detection

- Based on the weights
- If the sum of the weight in the search window of a pixel is 0 then the pixel cannot be reconstructed

$$S_w\,(s) = \sum_{s\prime \in Ns} w(s, s')$$

- S_w will never reach 0
- A small value is used as a threshold

$$S_w\,(s) \le \epsilon$$

Our implementation

- Based on the method propose by M. Zontak
- Two implementation for the reconstruction phase
- Added a post-processing phase

Our Implmentation - Reconstruction

- Two implementation for the reconstruction phase:
  - NL-impl: use the windowed-distance
  - Foveation-impl: foveation and foveated-distance may help improving the performance

$$\hat{v}_{src}(s) = \frac{1}{\boxed{w(s,s')}} * \sum_{s' \in Ns} \boxed{w(s,s')} * v_{ref}(s')$$

The weights can be computed either with foveated-distance or with windowed-distance.

Our Implmentation - Detection

- Detection is based on

$$Anomaly(s) = ||z_s - \hat{z}_s||_2^2$$

- Patch-wise difference between the original and the reconstructed src
- Anomaly is just an intermediate result, which is used in the post-processing phase

Our Implementation - Post processing

- Threshold anomaly

$$temp(s) = \ Anomaly(s) \geq \zeta$$

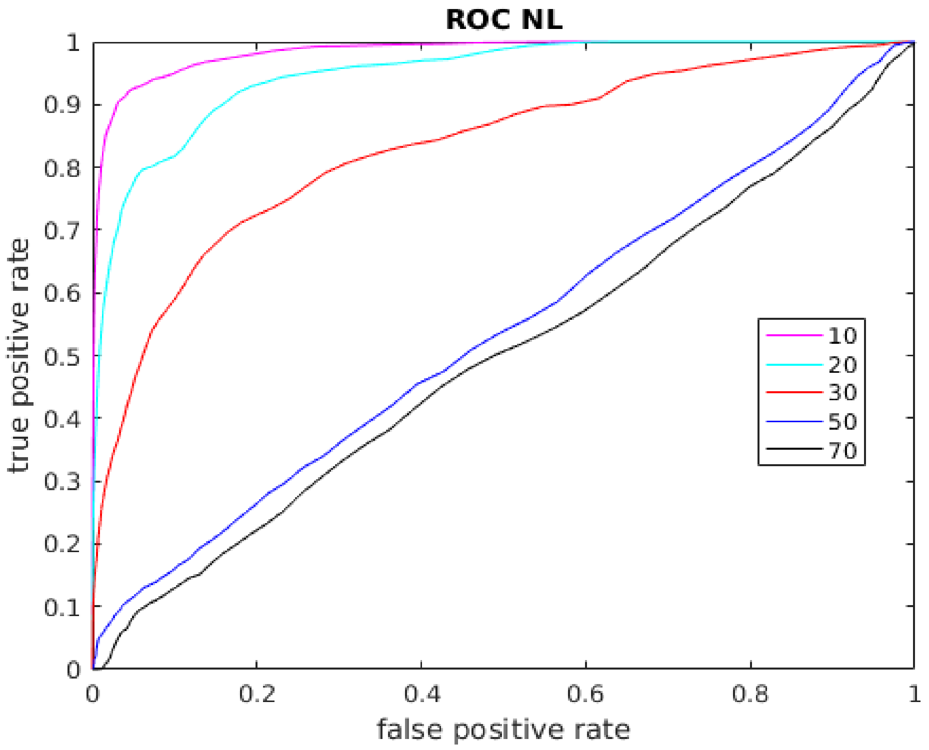
- Majority Voting
- Considering an area as big as the patch around pixel s
- A(s) = # anomalous pixels in the area
- N(s) = # not anomalous pixels in the area

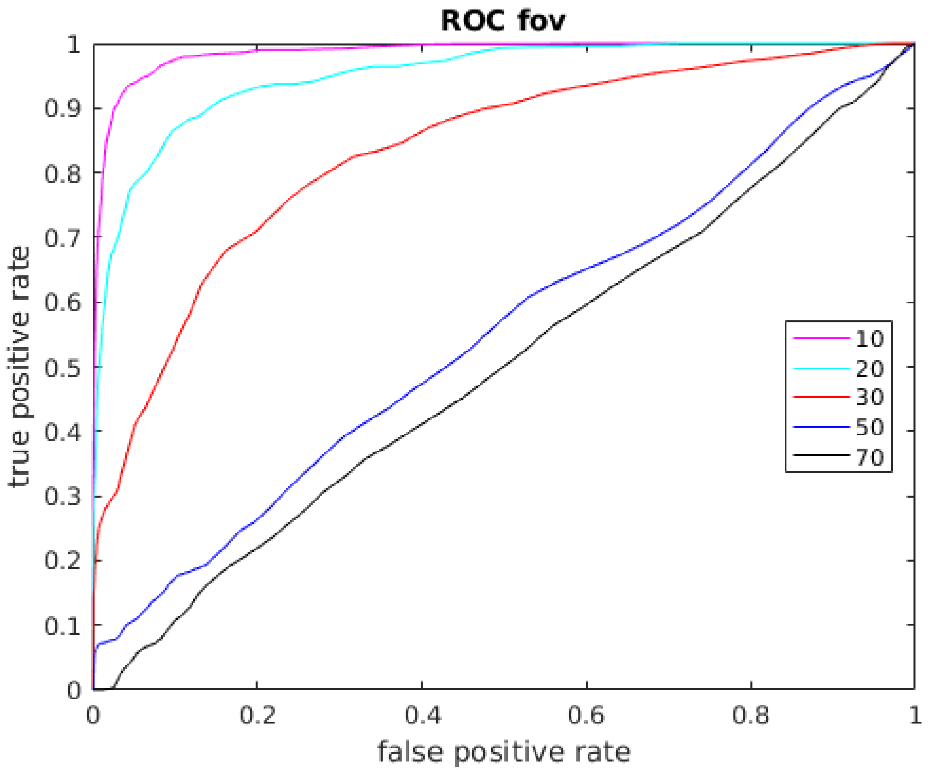
$$final(s) = \ 0 \ \leftrightarrow \ A(s) < N(s)$$
$$final(s) = \ 1 \ \leftrightarrow A(s)N(s)$$

Experiments

- Train test composed by 26 images, with manually inserted anomaly
- Study performance
- Find threshold
- Relation AUC-PSNR
- Apply our two proposed method
- Repeate the process for different noise with different sigma :
  - 10 , 20 , 30 , 50 , 70 : are the values of sigma used

Experiments: Performance

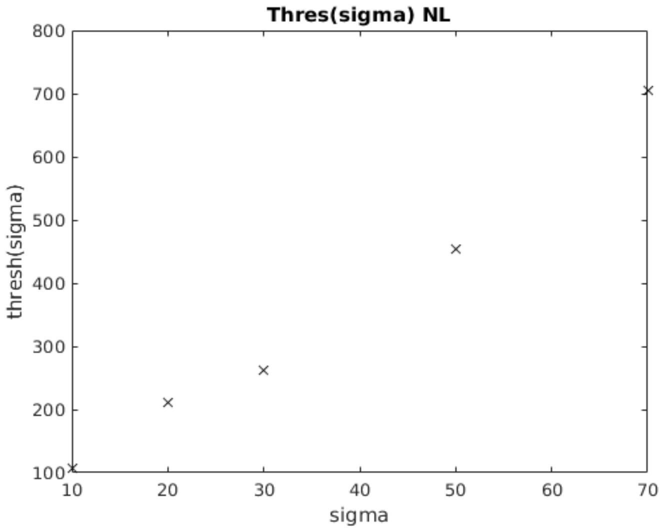


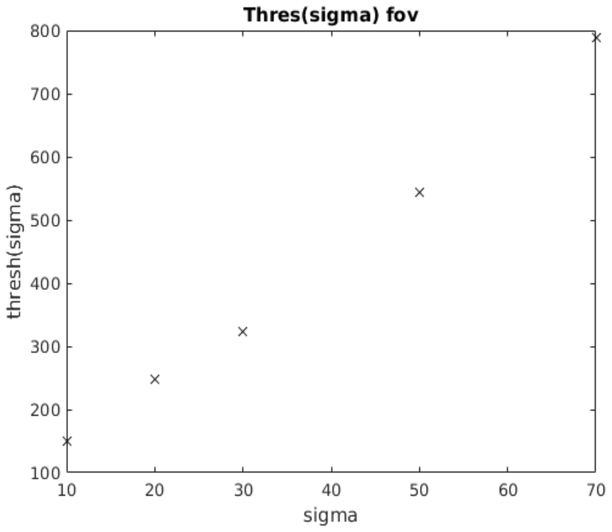


Sigma Value	10	20	30	50	70	Average
AUC - NL	0.9871	0.9453	0.8210	0.5359	0.4894	0,75574
AUC - fov	0.9907	0.9494	0.8163	0.5559	0.4978	0,76264

Experiments: Find threshold

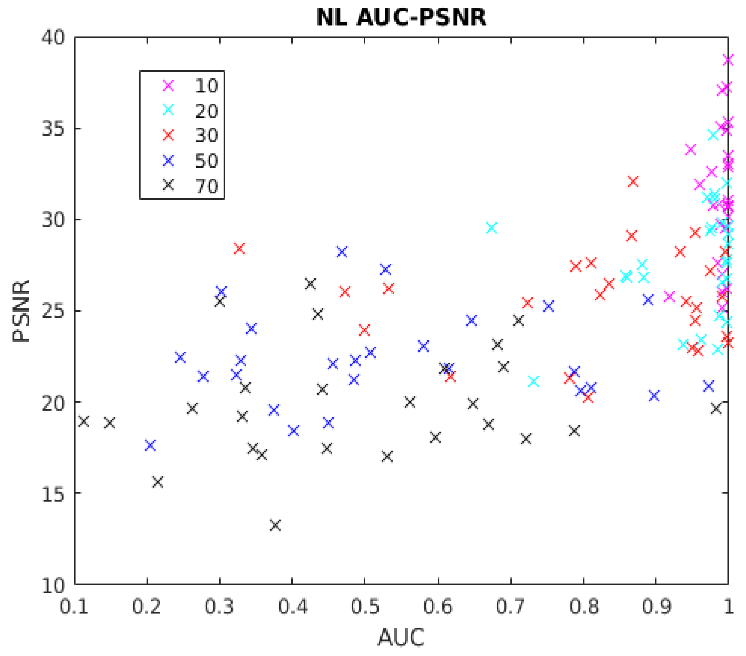
- From the train set take the intermediate results Anomaly(s)
- From Anomaly(s) remove value coresponding to anomalous pixels
- The threshoold is found as the 99 percentile of the remaining value
- The threshold is sigma dependent

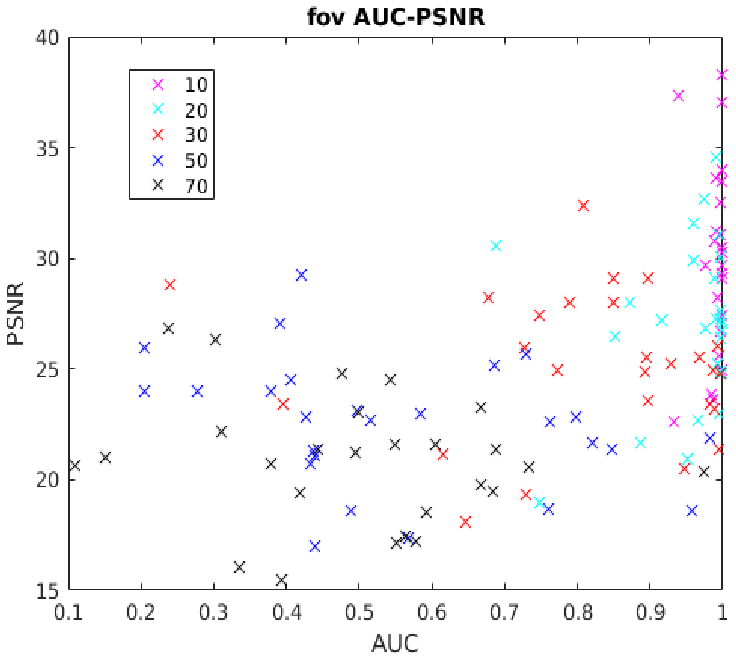




Experiments: AUC-PSNR

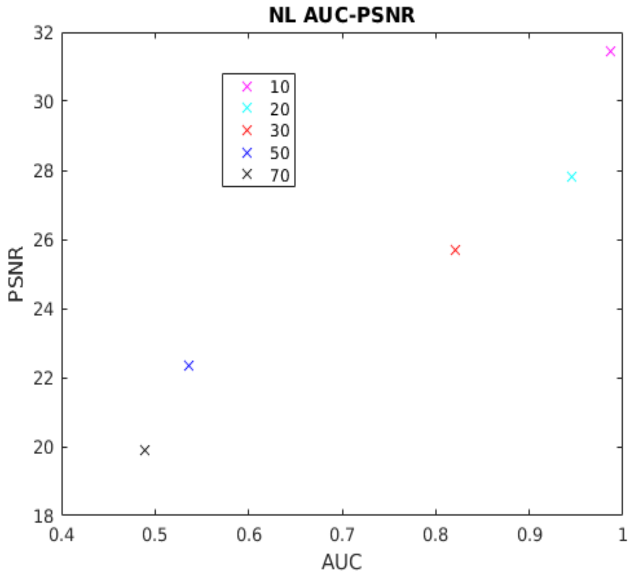
- The idea is: High PSNR means high AUC, the better an image will be reconstructed, the better will be the detection

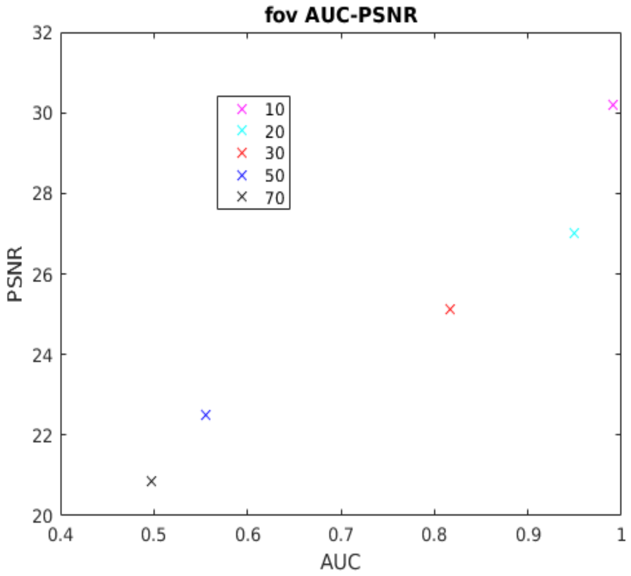




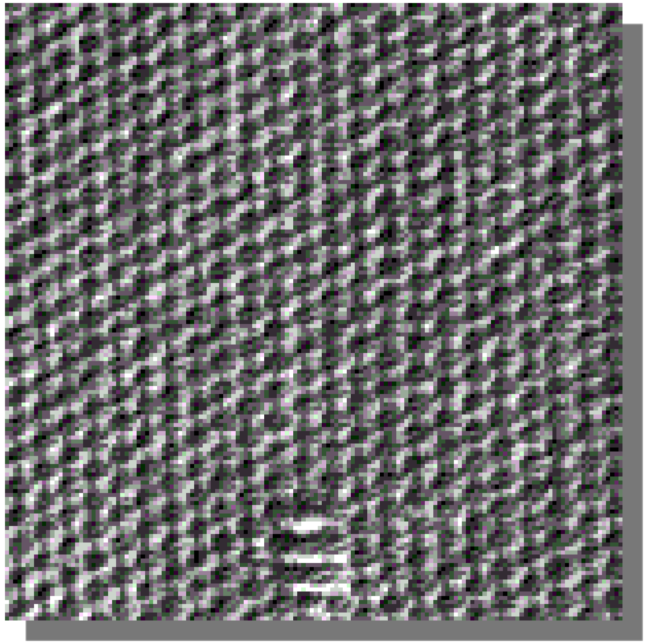
Experiments: AUC-PSNR

- For both method an high value of PSNR correspond to a low value of sigma

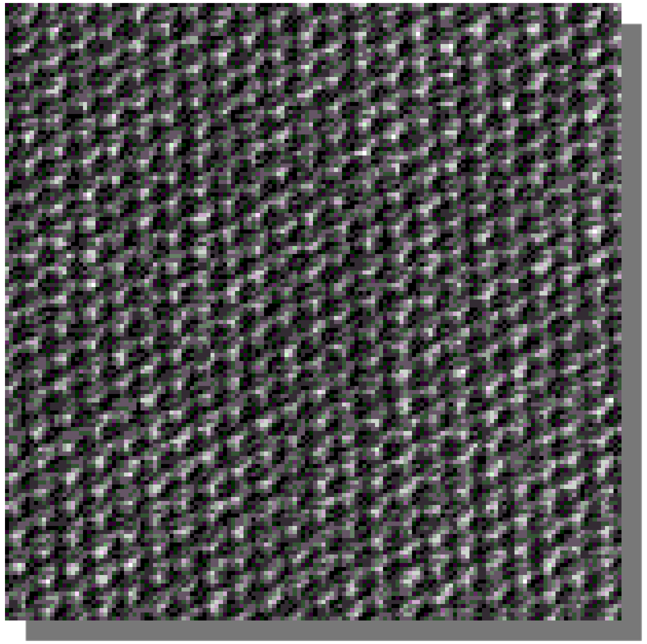




Experiments: Test1



Src image



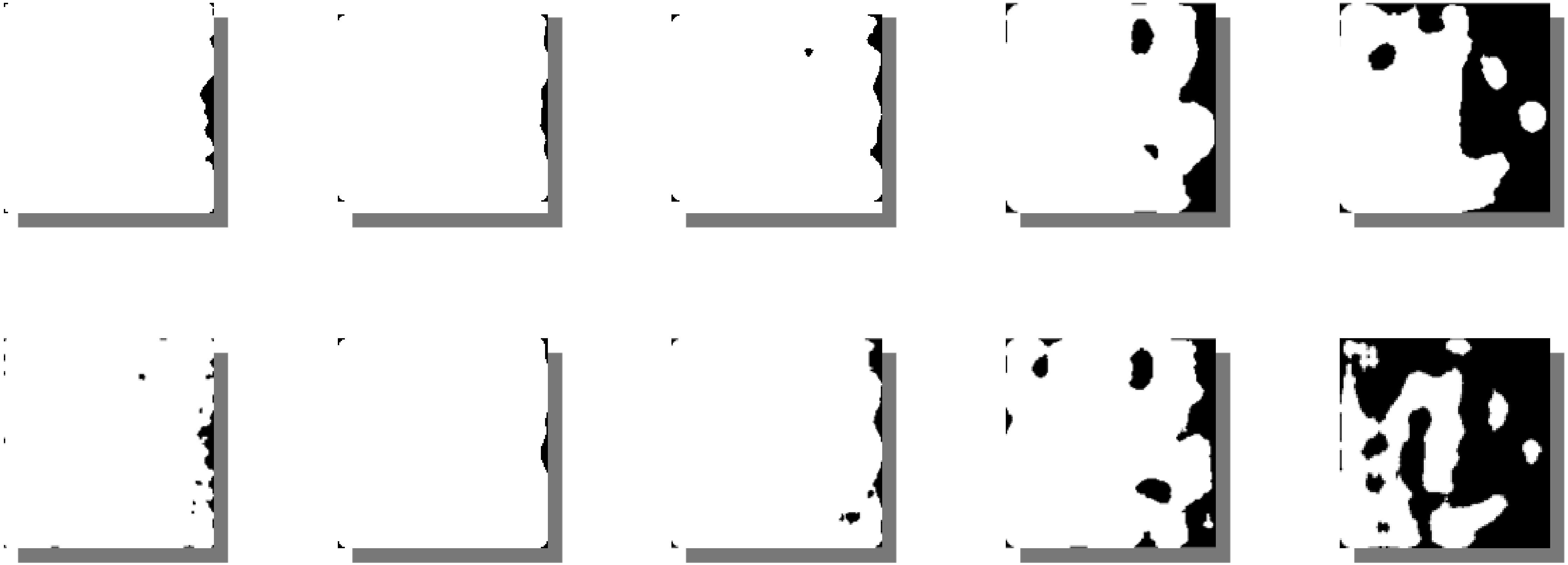
Ref Image

Experiments: Test1

Result of Test1

Fov

Nl



σ=10 σ=20 σ=30 σ=50 σ=70

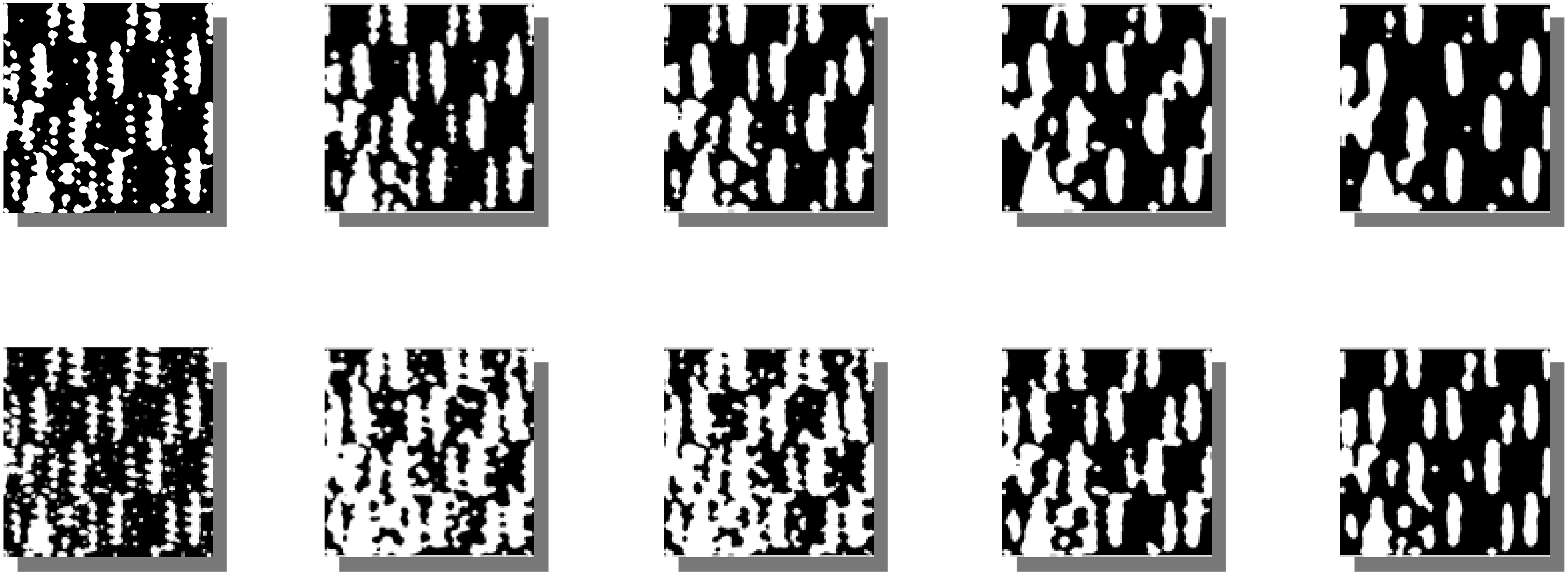
Experiments: Test2

Src image

Ref Image

Experiments: Test2

Result of Test2



Fov

Nl

σ=10 σ=20 σ=30 σ=50 σ=70

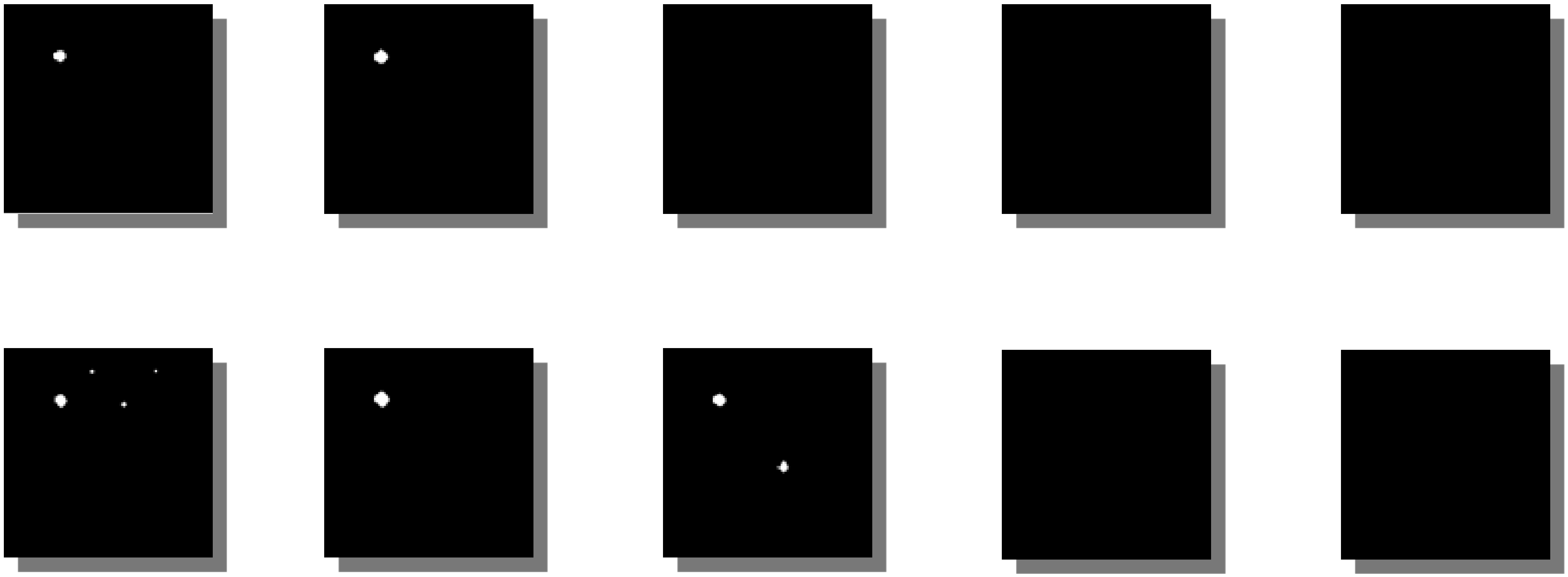
Experiments: Test3

Experiments: Test3

Result of Test3

Fov

Nl



σ=10 σ=20 σ=30 σ=50 σ=70

Conclusion

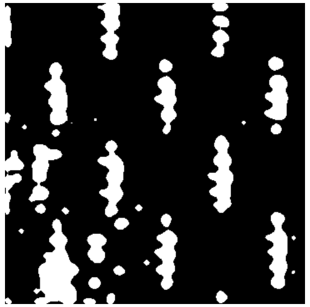
- Performance:
  - Good for both method for the lower value of sigma
  - Higher value (50 and 70) of sigma have an $AUC \leq 0.5$
- AUC-PSNR relation is not so clear, however high AUC implies high PSNR
- There are some problem on the threshold (look at test2)
- Increasing the search window may increase the performance, but also increase the computational time
- Image registration is not needed, unless the images are rotated

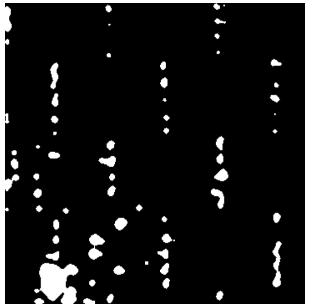
Conclusion - improvement

- Reduce time complexity or GPU implementation
- Resort to a rotational invariant form
- Modify threshold selection or post-processing

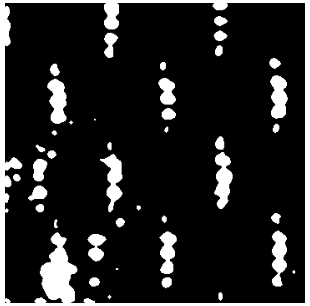
Conclusion - plus

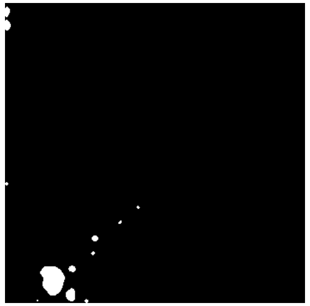
fov





NL





Search window 51x51 101x101

Changing the search window dimension may help in the detection.