

# ESTIMATION DE LA LUMINOSITÉ D'UNE ÉTOILE VARIABLE à partir d'une photographie.

Exercices préparatoires : filtrage d'une image et décompte de nombre d'objets

## 1. Le petit robot à lunettes

Pour cette séquence, l'image va être notre sujet principal.

Et l'on va partir d'une histoire...

*Dans un lointain labo préindustriel, on peut rencontrer des objets et personnages curieusement assortis : un informaticien mal peigné, un curieux robot à lunettes, et tout un tas de petites figurines représentant des maisons.*

En fait le robot n'a pas de lunettes mais une petite caméra qui ne prend des images qu'en mode monochrome (en gamme de gris) de qualité médiocre.

Voilà comment travaillent ces "gens" :

- le mal-peigné dispose des figurines sur un fond blanc, puis indique un seuil de perception au robot.
- celui-ci prend une photo de la scène, applique le seuil à l'image puis il décompte le nombre de figurines. Il annonce alors le résultat à son compère humain.

Remarque : ce genre de mécanisme peut être utilisé pour compter tout un tas d'objets vendus selon un nombre précis (boulons, fruits...).

Voir ci-dessous un image telle que celles de prend le robot.

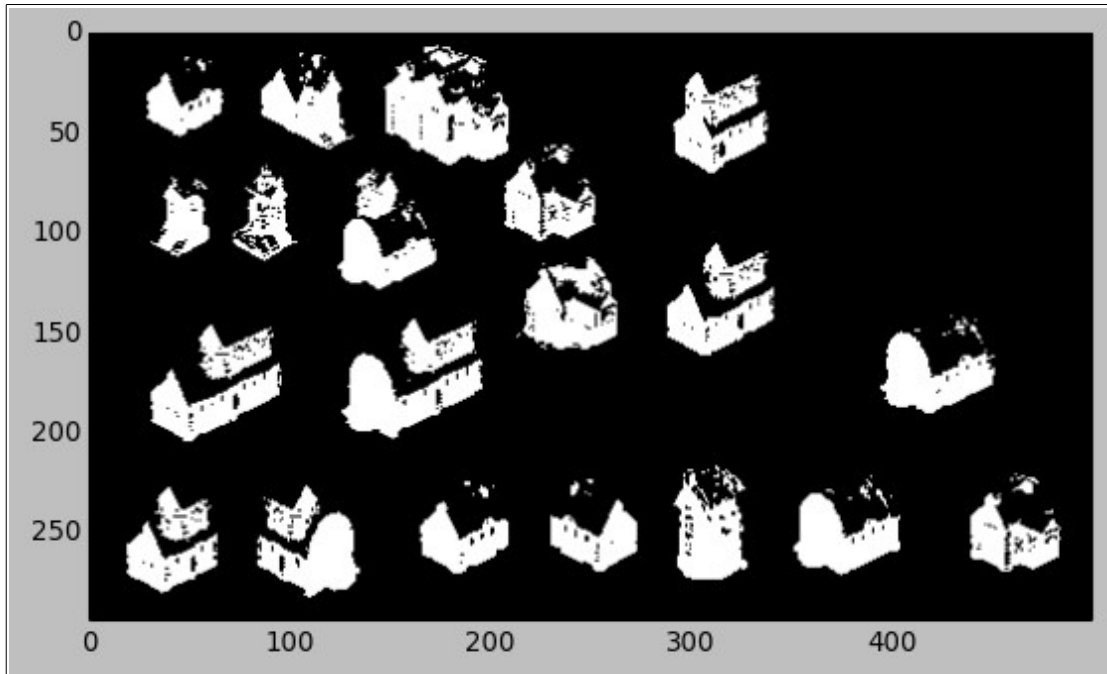


## 2. Visualiser l'importance du seuil

Chacun des pixels peut avoir une valeur allant de 0 à 255.

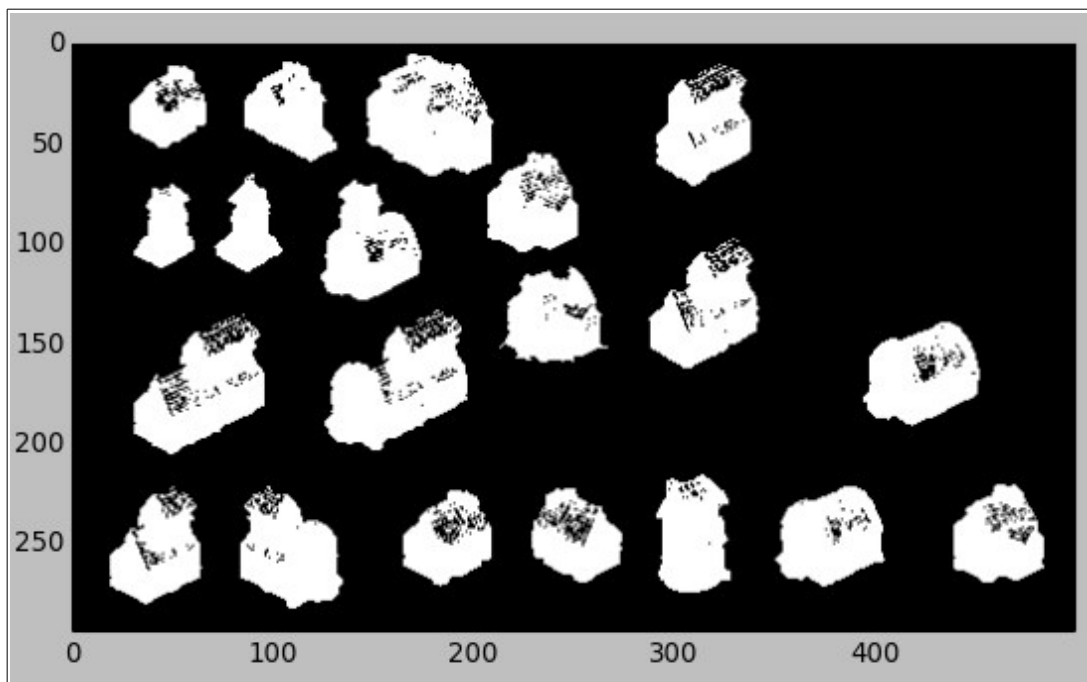
Ici, le seuil est fixé à 16. Tous les pixels qui ont une valeur inférieure à 16 sont mis à 0 (affiché ici par du noir) les autres sont mis à 255 (affiché ici par du blanc).

Le robot a trouvé 198 images.



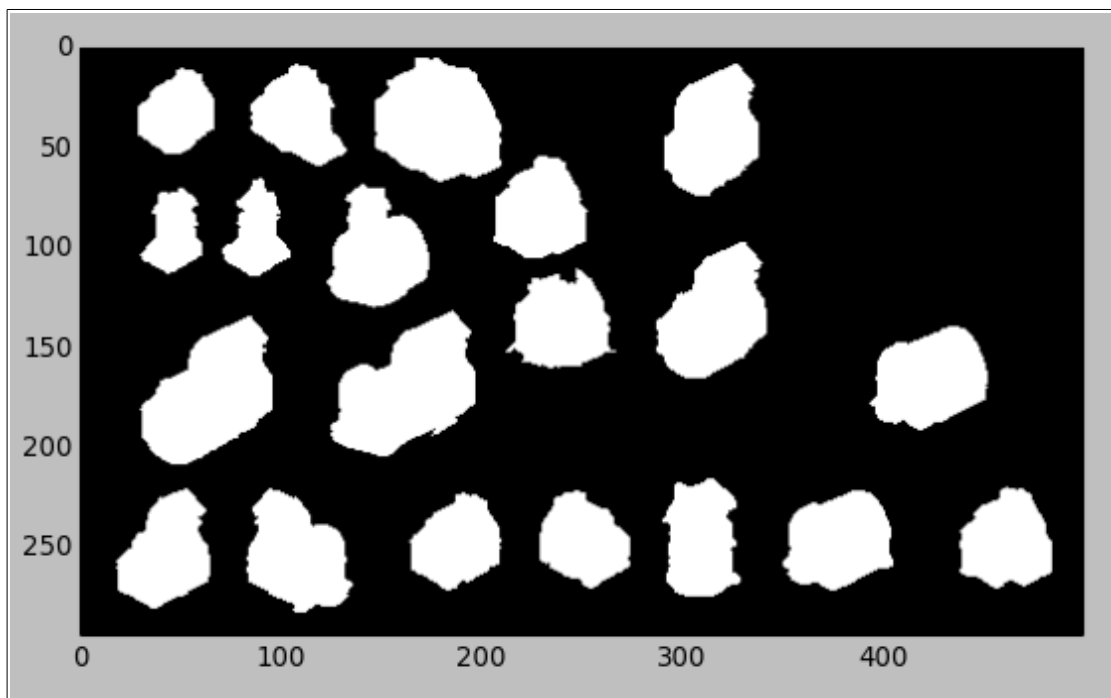
C'est normal : certains petits morceaux blancs sont individualisés et décomptés.

Ici, le seuil est fixé à 64. Tous les pixels qui ont une valeur inférieure à 64 sont mis à 0 (affiché ici par du noir) les autres sont mis à 255 (affiché ici par du blanc).



Le robot a décompté 254 objets. C'est pire.

Ici, le seuil est fixé à 128. Tous les pixels qui ont une valeur inférieure à 128 sont mis à 0 (affiché ici par du noir) les autres sont mis à 255 (affiché ici par du blanc).



Le robot a compté 21 objets, ce qui est la bonne réponse.

### 3. Conclusion

Si l'on veut faire identifier des objets individuellement, et c'est le cas pour des étoiles sur un cliché, le filtrage des données par application d'un seuil est utile, voire nécessaire.

Cette démarche peut être combinée avec la localisation du centre de chaque objet identifié ainsi qu'avec le calcul de la somme des valeurs des pixels (voir note précédente).

### 4. En annexe : le code du programme ayant réalisé les images.

Les remarques et commentaires sont en rouge.

```
# -*- coding:Utf-8 -*-
"""
    Charger une image en niveaux de gris
    la transformer en niveaux binaires par application d'un seuil
    compter le nombre d'objets ainsi isolés
    afficher les images qui résultent du traitement
"""
from PIL import Image
from pylab import *
from scipy.ndimage import measurements,morphology,find_objects
im = array(Image.open("houses.png").convert("L"))

def applique_seuil_et_cree(im,s):
    """
```

applique un seuil aux données de l'image  
crée une image à partir des données ainsi modifiées  
imprime le nombre d'objets comptés

\*\*\*\*\*

```
image = 1*(im<s)
labels,nbr_obj = measurements.label(image)
print "nombre d'objets", nbr_obj
```

\*\*\*\*\*

Crée une image à partir de données modifiées par  $image = 1*(im < s)$

\*\*\*\*\*

```
figure()    # définit une fenêtre pour afficher l'image
gray()      # sera affichée en gamme de gris
imshow(image)
```

\*\*\*\*\*

Crée une image à partir des "labels" qui résultent de  
labels,nbr\_obj = measurements.label(image)

\*\*\*\*\*

```
seuil = 16
applique_seuil_et_cree(im,seuil)
seuil = 64
applique_seuil_et_cree(im,seuil)
seuil = 128
applique_seuil_et_cree(im,seuil)
```

```
show()    # affiche une fenêtre
```