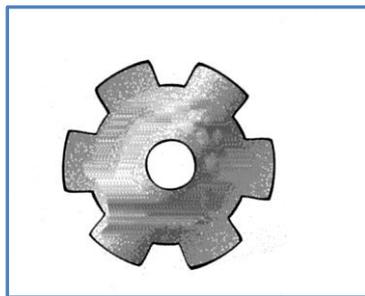


CONTROLE D'ALGORITHMIQUE

1H30— TOUS DOCUMENTS AUTORISES A L'EXCEPTION DES SUPPORTS NUMERIQUES

Une chaîne de production d'engrenages est équipée d'un système de contrôle qualité par vision artificielle. La première vérification sur laquelle ce sujet est basé, consiste à vérifier que la surface de l'engrenage correspond à ce qui est attendu.

Le dispositif d'acquisition d'image retourne une image de taille variable et dont chaque pixel correspond à un niveau de gris codé entre 1 et 255 (plus le chiffre est élevé, plus le pixel est proche du blanc). Ci-dessous une image type :



Pour commencer le travail vous disposez de 2 fonctions et d'un début de script :

- **camera** : fonction qui ne prend pas d'arguments d'entrée et qui retourne l'image sous la forme d'un tableau à 2 dimensions.
- **affiche** : fonction qui prend en argument d'entrée l'image et qui l'affiche. Cette fonction ne retourne rien.
- Le script **Script_Base** appelle ces deux fonctions. Vous aurez à le compléter.

PARTIE N°1 : BINARISATION

Ecrivez une fonction **binaire** qui réalise la binarisation de l'image par rapport à un seuil **S**. La binarisation de l'image consiste à construire une image de même dimension mais qui ne contienne que 2 valeurs :

- 1 si le pixel est inférieur à la valeur du seuil **S**
- 255 si le pixel est supérieur ou égal à la valeur du seuil **S**

L'appel à la fonction dans le script **Script_Base** doit correspondre à la syntaxe suivante :

```
PhotoBW = binaire(Photo, Seuil)
```

Complétez ensuite le script pour demander à l'utilisateur le niveau de seuil à appliquer. L'appel à la fonction **binaire** est ensuite faite et on affichera (avec la fonction **affiche**) la photo ainsi binarisée. On pensera à n'accepter lors de la saisie que des valeurs de seuil correspondantes à des valeurs possibles.

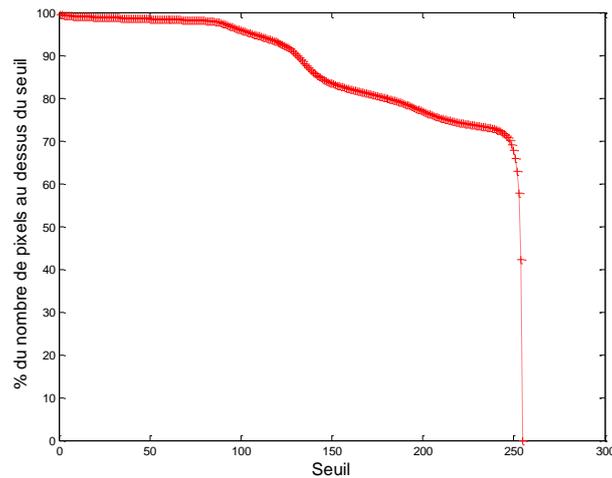
Cette opération (saisie, binarisation, affichage) sera répétée jusqu'à ce que l'utilisateur saisisse pour la valeur de seuil -1 auquel cas c'est la valeur de seuil précédemment saisie qui sera utilisée et on affichera l'image binarisée avec ce seuil.

PARTIE N°2 : DENOMBREMENT

Ecrivez une fonction (`denombre`) qui prenne comme argument d'entrée l'image précédemment binarisé et qui calcule le pourcentage de pixel qui sont à 1. La fonction retournera ce pourcentage.

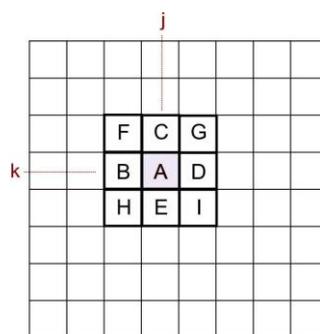
Complétez le script pour fabriquer un tableau (que l'on nommera `Ventile`) de 255 valeurs. La $N^{\text{ième}}$ case tableau contiendra le pourcentage de pixels à 1 quand le seuil de niveau N est appliqué sur l'image d'origine par la fonction `binnaire`.

Tracez la courbe correspondant au tableau ainsi calculé. On doit donc obtenir une courbe qui ressemble à :



PARTIE N°3 : FILTRAGE

Comme il est compliqué de déterminer la valeur de seuil idéal à cause du « bruit » dans l'image, il est décidé d'appliquer un filtrage sur celle-ci. Ce filtrage consiste à créer une image de même dimension que l'image d'origine et dont chaque pixel de coordonnées (j, k) prendra pour valeur $V(j, k) = (4*A + 2*(B+C+D+E) + 1*(F+G+H+I)) / 16$. Les cases A, \dots, I correspondent au quadrillage suivant sur l'image d'origine :



On appliquera ce filtre à l'ensemble de l'image d'origine, à l'exception des pixels du bord qui conserveront leur valeur initiale.

Ecrivez la fonction `Filtre` qui réalise cette opération. Proposez un passage d'arguments (entrée et sortie) qui vous paraît cohérent.

Complétez le script pour que, comme dans la partie 2, il trace le pourcentage de pixel à 1 en fonction du seuil sur l'image filtré.