



# L3 I

## Python

### TD n°5 et 6 : traitement d'images (partie n°2)

## Manipulations de fichiers images (partie n°2)

**Objectifs :** Implémenter les transformations du photomaton et du boulanger.

### Exercice N° 1 :

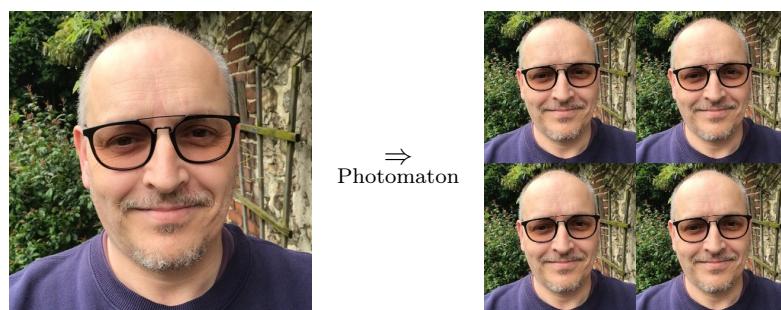
Aller sur le site internet et télécharger, au choix, l'une des images **Joconde**, **Picasso** ou **One** associées à ce TP. C'est sur cette image que vous allez opérer la transformation du photomaton et du boulanger. Dans la bibliothèque **PIL.Image**, on dispose des fonctions

- `(l,h)=image.size` retourne les dimensions de l'image `image` (qui a été chargé au préalable à l'aide de la fonction `open`). `l` correspond à la longueur dans le sens abscisse et `h` correspond à la longueur dans le sens ordonné.
- `image.show()` affiche l'image `image` (qui a été chargé au préalable à l'aide de la fonction `open`).
- `image.getpixel((x,y))` récupère la valeur de la couleur du pixel situé en ligne `y` et colonne `x` de l'image `image` (qui a été chargé au préalable à l'aide de la fonction `open`). À savoir que les indices des lignes et colonnes commencent à 0.
- `image.putpixel((x,y),valeur)` remplace la valeur de la couleur du pixel situé en ligne `y` et colonne `x` de l'image `image` (qui a été chargé au préalable à l'aide de la fonction `open`) par `valeur`.
- `new_image=new('RGB',(l,h))` crée une image nommée `new_image` de taille `(l,h)` toute noire.
- `image.save("Nom.png")` enregistre l'image `image` de Python sur le PC sous le nom de `Nom` au format `png`.

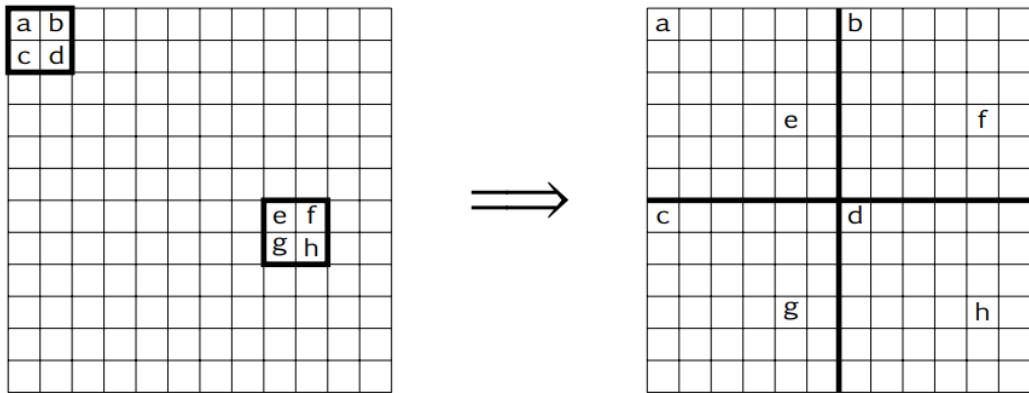
### 1 La transformation du photomaton

Cette transformation « réduit » la taille de l'image de moitié pour obtenir quatre morceaux analogues que l'on place en carré pour obtenir une image de même taille que l'image d'origine.

Voici un exemple du résultat escompté.



Pour que cette transformation soit bijective, on a découpé l'image initiale en paquets carrés de quatre pixels ( $2 \times 2$ ) puis pour chaque paquet carré de quatre pixels, on utilise celui en haut à gauche pour l'image réduite en haut à gauche, celui en haut à droite pour l'image réduite en haut à droite, etc.



On notera que pour que cette transformation soit définie il est nécessaire que les dimensions horizontale et verticale de l'image soient paires

### Exercice N° 2 :



Prendre une feuille et un stylo.

À partir des coordonnées  $(x, y)$  d'un pixel de l'image initiale, exprimer ses coordonnées  $(x', y')$  dans l'image résultat. Il sera nécessaire de distinguer quatre cas, suivant la parité de  $x$  et de  $y$ .

### Exercice N° 3 :



Écrire une fonction intitulée `def photomaton(image)` : prenant en entrée une image (qui a été chargé au préalable à l'aide de la fonction `open`) et qui retourne la transformation photomaton de `image`.

Tester cette fonction avec le fichier `image choisi` en début de TP.

## 2 La transformation du boulanger

Cette transformation s'appelle ainsi car elle s'apparente au travail du boulanger qui réalise une pâte feuillettée.

L'image est tout d'abord étirée :



coupée en deux dans le sens vertical pour placer la partie de droite sous la partie gauche en la faisant tourner de  $180^\circ$  :

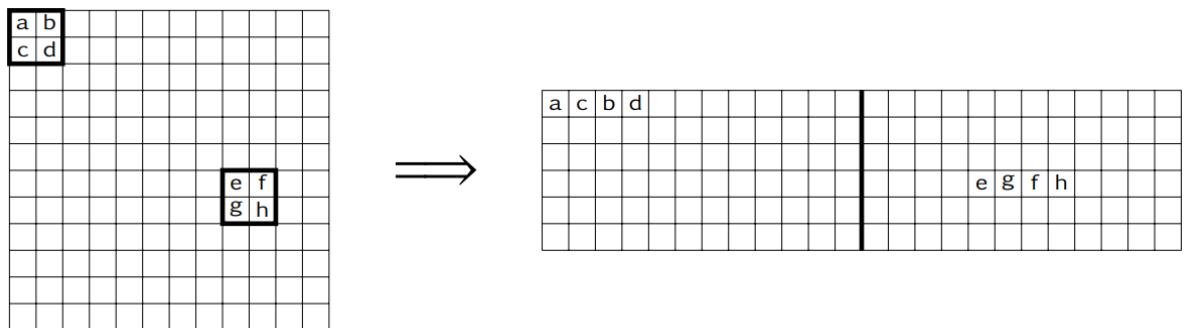


Pour que cette transformation soit bien définie, il est nécessaire là encore que les dimensions de l'image soient paires.

## 2.1 L'étirement

D'une image de taille  $(l, h)$ , on obtient une image de taille  $(2l, h/2)$ .

L'image initiale est découpée en paquets carrés de quatre pixels puis ceux-ci sont entrelacés pour obtenir l'image intermédiaire aplatie.



### Exercice N° 4 :



Prendre une feuille et un stylo.

À partir des coordonnées  $(x, y)$  d'un pixel de l'image initiale, exprimer ses coordonnées  $(x', y')$  dans l'image résultat. Il sera nécessaire de distinguer deux cas suivant la parité de  $y$ .

### Exercice N° 5 :



Écrire une fonction intitulée `def etirement(image)` : prenant en entrée une image (qui a été chargé au préalable à l'aide de la fonction `open`) et qui retourne la transformation `etirement` de `image`. Tester cette fonction avec le fichier `image choisi` en début de TP.

## 2.2 Le repliement

D'une image de taille  $(l, h)$ , on obtient une image de taille  $(l/2, 2h)$ .

L'image initiale est découpée en deux selon la verticale, la partie de droite est placé sous la partie gauche en la faisant tourner de  $180^\circ$ .

### Exercice N° 6 :



Prendre une feuille et un stylo.

À partir des coordonnées  $(x, y)$  d'un pixel de l'image initiale, exprimer ses coordonnées  $(x', y')$  dans l'image résultat. Il sera nécessaire de distinguer deux cas suivant la valeur de  $x$ .

### Exercice N° 7 :



Écrire une fonction intitulée `def repliement(image)` : prenant en entrée une image (qui a été chargé au préalable à l'aide de la fonction `open`) et qui retourne la transformation `repliement` de `image`.

Tester cette fonction avec le fichier `image choisi` en début de TP.



## 2.3 La transformation du boulanger

### Exercice No 8 :

Écrire une fonction intitulée `def boulanger(image)` : prenant en entrée une image (qui a été chargé au préalable à l'aide de la fonction `open`) et qui retourne la transformation du boulanger de `image`. Tester cette fonction avec le fichier `image` choisi en début de TP.

### Exercice N° 9 :



Opérer dix neuf fois la transformation du boulanger sur le le fichier `image` choisi en début de TP.  
Afficher successivement les vingt images obtenues.

Admirez avec étonnement le résultat.

Vérifier que ce résultat n'est pas dû à l'image choisie en début de TP en testant avec n'importe quelle image de taille  $256 \times 256$  trouvée sur le net.