
Atelier traitement d'images

- Une image numérique est composée de petits carrés appelés des **pixels**.
- Chaque pixel possède une couleur.
- Cette couleur est noir ou blanc dans le cas d'une image en noir et blanc, gris plus ou moins foncé dans le cas d'une image en niveaux de gris et une vraie couleur dans le cas d'une image en couleurs.
- La couleur d'un pixel est représentée par un nombre (ou par trois nombres dans le cas d'une image en couleurs).
- De plus, la position de chaque pixel dans une image est repérée par un numéro de ligne i et un numéro de colonne j .
- Pour une image de hauteur H et de largeur L , les numéros de lignes vont de 0 à $H - 1$ et les numéros de colonnes vont de 0 à $L - 1$.
- Attention, la ligne numéro 0 est celle d'en haut !
- La colonne numéro 0 est celle de gauche.

Images en noir et blanc

Exercice 1.

1. Ouvrir le fichier `Maison.txt` avec le bloc-note. Voyez-vous la maison ?
2. Ouvrir le logiciel **GIMP**, et ouvrir le fichier `Maison.txt` avec ce logiciel.
3. Quel est le nombre représentant la couleur noire pour un pixel ? Et pour la couleur blanche ?
4. Pouvez-vous ajouter une fenêtre au rez-de-chaussée à droite de la porte ?

- Pour effectuer des traitements plus compliqués sur des images plus grandes, on ne change pas les pixels un par un à la main, mais on utilise des logiciels de traitement d'images.
- Le programme `TraitementNoirEtBlanc.py` est écrit en langage Python. Il peut être ouvert, modifié et exécuté à l'aide du logiciel **Python-idle**.
- Il s'agit d'un programme de traitement d'images très simple.
- À partir d'un fichier image d'entrée, il permet de parcourir chaque ligne de l'image, colonne par colonne, et de modifier chaque pixel pour effectuer le traitement souhaité.
- L'image obtenue est enregistrée dans un fichier de sortie appelé `monimage.txt`, qui peut être lu par un éditeur de texte ou par un visionneur d'images comme **GIMP**.
- Certaines informations sur l'image sont également affichées par le programme.
- Au départ, aucun traitement n'est réalisé, et l'image de sortie est simplement une copie de l'image d'entrée.

Exercice 2.

1. Saurez-vous modifier la partie traitement du programme `TraitementNoirEtBlanc.py` pour que la couleur de chaque pixel de l'image de sortie soit l'inverse de la couleur du pixel correspondant de l'image d'entrée ? Vous pouvez vérifier le résultat en visualisant l'image obtenue à partir du fichier `Maison.txt` comme image d'entrée.
2. Dessiner une inondation qui atteint le quartier de la maison.
3. Dessiner la pluie sur la maison.

4. Ajouter un soleil (carré...) dans la partie en haut à droite de l'image de la maison.
5. *Attention, plus difficile !* Un pixel du bord du dessin est un pixel noir dont au moins un des 8 pixels voisins est blanc. Dans cette question, sur l'image de sortie, on veut afficher les pixels du bord du dessin en noir et tous les autres pixels de l'image en blanc.
6. En changeant le nom du fichier au début du programme, vous pouvez tester le résultat de certains de vos traitements sur l'image du perroquet `Coco400NB.txt`.

Images en niveaux de gris

- Dans une image en niveaux de gris, le gris plus ou moins clair d'un pixel est représentée par un nombre entre 0 et 255.
- Attention, un pixel noir a pour valeur 0 et un pixel blanc a pour valeur 255 !
- Le programme `TraitementGris.py` fonctionne sur le même principe que le programme précédent pour les images en noir et blanc.

Exercice 3.

1. Dessiner l'image `Coco400Gris.txt` en négatif.
2. On veut dessiner le perroquet en noir et blanc. Pour cela, on décide que tous les pixels clairs sur l'image d'entrée deviendront blancs sur l'image de sortie, et que tous les pixels sombres deviendront noirs.
3. On décide maintenant de fixer 4 paliers de couleurs : noir (0), gris foncé (85), gris clair (170) et blanc (255).
4. Les pixels trop foncés deviennent gris foncé (85) et les pixels trop clairs deviennent gris clair (170).
5. Dessiner le perroquet la tête en bas.
6. *Attention, plus difficile !* On se fixe un seuil (par exemple 90), et on décide qu'un pixel du bord du dessin est un pixel dont la couleur diffère d'au moins ce seuil de la couleur d'au moins un de 4 pixels voisins. Dans cette question, sur l'image de sortie, on veut afficher les pixels du bord du dessin en noir et tous les autres pixels de l'image en blanc.

Images en couleurs

- Dans une image en couleurs au format RVB (pour Rouge, Vert, Bleu, en Anglais on dit RGB, je vous laisse deviner pourquoi), la couleur de chaque pixel est codée par trois nombres compris entre 0 et 255.
- Le premier nombre correspond au rouge, le deuxième au vert et le troisième au bleu.
- Un pixel rouge a pour valeur 0 255 255, un pixel vert a pour valeur 255 0 255 et un pixel bleu a pour valeur 255 255 0.
- Un pixel de valeur 10 200 200 sera plutôt rouge, un pixel de valeur 10 200 10 sera plutôt violet, un pixel de valeur 10 10 10 sera gris foncé et un pixel de valeur 200 200 200 sera gris clair.
- Un pixel noir a pour valeur 0 0 0 et un pixel blanc a pour valeur 255 255 255.
- Le programme `TraitementCouleurs.py` fonctionne sur le même principe que les deux précédents.

Exercice 4.

1. Le perroquet de l'image `Coco400Couleurs.txt` est rouge. Que peut-on dire des valeurs des pixels ?
2. Pouvez-vous transformer Coco en perroquet vert ou en perroquet bleu ?