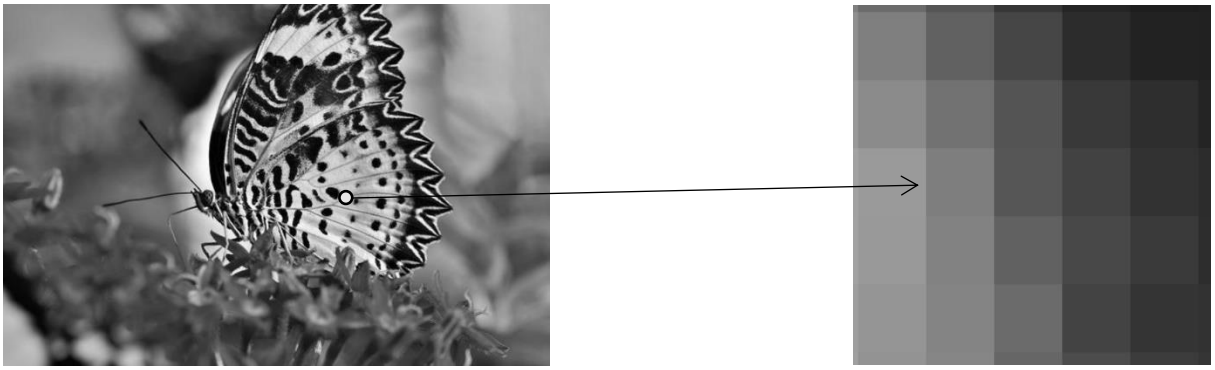
Traitement des images : contour d'une image

Contenus	Capacités attendues
Traitement d'image	Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels.

Activité n°1 : Situation : qu'est-ce qu'un contour ?

On a vu qu'il était possible de traiter une image numérique. Pour cela il suffit de lire les différents pixels, puis de leur appliquer un traitement.



Valeurs relevées avec Gimp
De chaque pixel zoomé

53	50	40	37	30
82	57	39	38	34
107	85	54	37	46
154	97	70	44	50
153	130	97	72	58

- ***Un contour sera en Noir et Blanc : quelles sont les deux valeurs de pixel que vous allez utiliser pour tracer un contour ?***
- ***Comment pourriez-vous tracer le contour dans ce carré de valeurs en niveau de gris ?***
- ***Tracez ce contour.***

Un contour définit la limite d'un objet dans une image. Cette limite est caractérisée par un changement dans l'image : un changement de couleur ou de contraste. Ce changement se traduit dans la valeur des pixels qui sont localisés de part et d'autre de la limite. Nous sommes donc à la recherche d'un moyen de détecter et de localiser un changement :

Les mathématiques nous donnent ce moyen sous la forme de la différentiation.

Considérons un pixel p(x,yj) dans une image couleur :

- Ce pixel est-il semblable, de même couleur, que ses voisins ?
- Si non, quelle est la différence de couleur entre lui et ses voisins *?*
- Est-elle grande, ce qui signifierait qu'il est situé à la limite d'un objet ?
- Que signifie une "grande" ou une "petite" différence ?
- Comment la mesurer pratiquement ?

C'est ce que nous allons essayer de traduire en algorithme. Cependant afin de simplifier le problème nous travaillerons sur une image en niveau de gris.

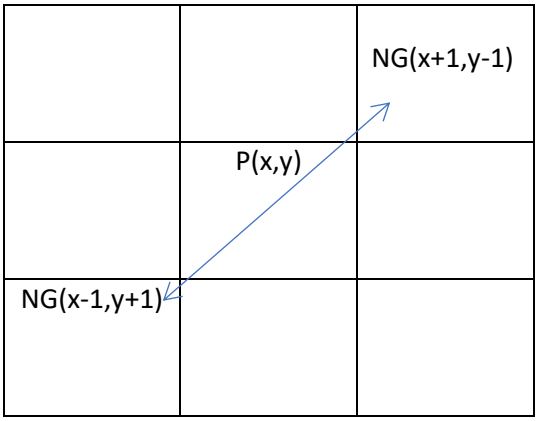
Activité n°2 : Détection de contours d'une image

La discontinuité :

Si le niveau de gris des pixels voisins est différent du pixel P(x,y), alors on considérera que ce pixel P(x,y) fait partie d'un contour.

Pour cela il faut donc calculer une différence qui soit la plus significative.

Il faudra alors prendre les 2 pixels qui sont diamétralement opposés au pixel central P(x,y). On va donc faire une soustraction entre les deux niveaux de gris des pixels voisins diamétralement opposés.



53	50	40	37	30
82	57	39	38	34
107	85	54	37	46
154	97	70	44	50
153	130	97	72	58

- ***Quelle est cette différence pour le pixel P(x,y) encadré en rouge. Ecrire l'opération effectuée ?***
- ***Même question pour le pixel encadré en jaune.***
- ***Parmi ces deux pixels, lequel ou lesquels semble(nt) appartenir à un éventuel contour et pourquoi ?***

Pour éviter d'avoir un signe négatif dans la soustraction, il suffira d'élever au carré le résultat qui donnera toujours un résultat positif.

- ***Refaire les calculs précédents avec le carré pour les deux pixels. Comment alors pourrait 'on décider qu'un pixel appartient au pas à un contour ?***

Nous n'avons pris en compte que 2 pixels voisins. Cela privilégie qu'une seule direction. Pour éviter cela on va donc prendre 4 pixels.

53	50	40	37	30
82	57	39	38	34
107	85	54	37	46
154	97	70	44	50
153	130	97	72	58

- ***Refaire tous tes calculs de la même façon que précédemment pour les 2 pixels rouge et jaune, mais en prenant compte maintenant 4 pixels. On obtiendra 2 résultats (soustraction élevée au carré) que l'on additionnera. Enfin on prendra la racine carrée du résultat final obtenu. On appelle cela une norme.***

$$norme = \sqrt{(soustraction1)^2 + (soustraction2)^2}$$

Soustraction1= différence de niveau de gris entre 2 pixels voisins opposés.

Le seuil.

- ***A partir que quel niveau de la norme (différence entre 4 pixels voisins) pourrait-on décider qu'un pixel fait partie du contour ?***

Activité n°3 : Algorithme de détection et programmation en python

Algorithme

Pour cet algorithme, il faudra :

- Ouvrir un fichier en niveaux de gris
- Créer un fichier vide qui affichera le contour. On rappelle qu'un pixel sera noir si le pixel de l'image de départ fait parti du contour et blanc dans le cas contraire.
- Lire un niveau de gris d'un pixel de coordonnée(x,y).
- Puis calculer la norme(x,y) sur 4 pixels voisin
- Si la norme(x,y) autour d'un pixel donné P(x,y) > seuil alors on mets un pixel noir dans le fichier image contour, sinon on met un pixel blanc.

Remettre à leur place les lignes de l'algorithme suivant et le compléter afin qu'il réponde à nos attentes.

```
DEBUT
        POUR ligne ALLANT de 1 à nb_lignes-1
        Image_niveaux_gris_depart ← ouvrir fichier de départ en niveaux de gris
        nb_colonnes←nombre de colonnes de l'image de départ

                POUR colonne ALLANT de 1 à nb_colonnes-1
                        Niveau_gris← couleur pixel image départ de coordonnées (colonne,ligne)
                        norme←calculer la norme autour du pixel (colonne,ligne) de départ

                        SINON
                                pixel(colonne,ligne) du contour←(blanc ou noir ??)
                        FIN SI

        FIN POUR
        Afficher image_contour
        Fermer image_contour
FIN
```

- ***Lignes qui se sont perdues, à remettre dans l'algorithme précédent :***

- SI norme > seuil ALORS

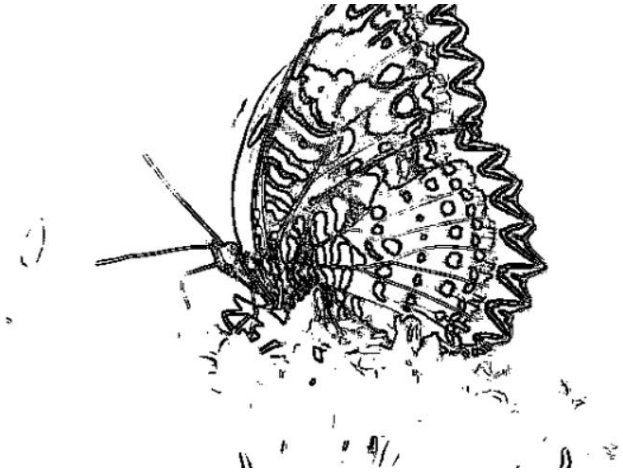
pixel(colonne,ligne) du contour←(blanc ou noir ??)

- FIN POUR
- Image_contour ← créer un fichier image vierge qui mémorisera le contour.
- nb_lignes←nombre de lignes de l'image de départ

Programmation de l'algorithme en Python et conclusion.

- ***A l'aide des programmes réalisés dans les séances précédentes (plus particulièrement butterfly_NB.py) programmez cet algorithme en python (on se servira indifféremment d'édupython ou de spyder)***

Vous devriez obtenir cela avec un seuil de 30 :



- ***Faire varier le seuil et repérer quelle est, pour vous, la meilleure valeur.***
- ***Si on a le temps on pourra comparer notre programme à la détection de seuil d'un logiciel comme Gimp.***