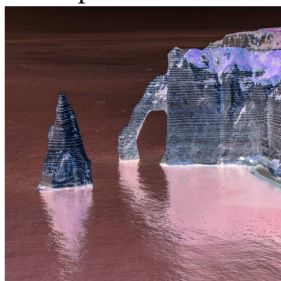
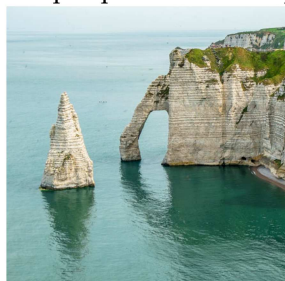


## TP 6 - Traitement d'image matricielle (bitmap)

**Exercice 1 Négatif d'une image** Définir une fonction `negatif(f_in, f_out)` qui retourne le négatif de l'image : chaque quantité est remplacée par son complément à 1.



### ROTATION D'UNE IMAGE

**Exercice 2 Rotation** Créer une fonction `rotation90(f_in, f_out, a)` qui retourne une nouvelle image ayant subi une rotation d'angle  $a$  degrés compris entre 0 et 90.

**Attention !** Il faudra créer un nouveau *cadre* et utiliser le paramètre de transparence pour afficher uniquement l'image.

**Attention !** Le pixel (0,0) est en haut à gauche du cadre.

**Rappel :** Considérant une rotation d'angle  $\theta$  de centre  $\Omega = (x_0, y_0)$ , l'image du point  $M(x, y)$  est le point  $M'(x', y')$  défini par :

$$\overrightarrow{\Omega M}, \overrightarrow{\Omega M'} = \theta$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$



**Exercice 3** Proposer une fonction `rotation(f_in, f_out, a)` qui retourne une nouvelle image ayant subi une rotation d'angle  $a$  degrés (sans contrainte sur  $a$ ).

### FILTRE PAR CONVOLUTION

On modifie un pixel en considérant une moyenne pondérée des pixels qui l'entourent (le travail est à faire sur chaque composante). Pour cela, on introduit une matrice contenant les coefficients de pondération.

Par exemple pour une matrice (3,3) :  $M = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & k \end{pmatrix}$

Posons  $s = a + b + c + d + e + f + g + h + k$  ; et si  $s = 0$  alors on pose  $s = 1$ .

Le pixel en position  $[i, j]$  est modifié par :

$$p_{i,j} = \frac{1}{s} (ap_{i-1,j-1} + bp_{i-1,j} + cp_{i-1,j+1} + dp_{i,j-1} + ep_{i,j} + fp_{i,j+1} + gp_{i+1,j-1} + hp_{i+1,j} + kp_{i+1,j+1})$$

**Exercice 4** Écrire une fonction `ajust(f)` qui considère un tableau de type image mais dont les coefficients peuvent déborder du domaine  $[0,1]$  et qui normalise les coefficients par la règle suivante :

- si la quantité est inférieure à 0, elle est ramenée à 0 ;
- si la quantité est supérieure à 1, elle est ramenée à 1.

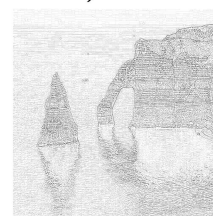
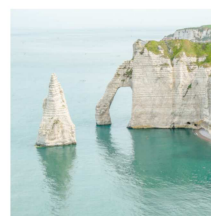
**Exercice 5** Écrire une fonction `convolution(M, f_in, f_out)` qui applique le filtre de convolution de matrice  $M$  de taille  $(2r + 1, 2r + 1)$  en rognant les  $r$  premières et dernières lignes et colonnes de `f_in`.

Tester différents filtres et succession de filtres :

- floutage de l'image :  $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

- filtre passe haut (accentue les détails) :  $\begin{pmatrix} 0 & -4 & 0 \\ -4 & 18 & -4 \\ 0 & -4 & 0 \end{pmatrix}$

- filtre passe bas (adoucit les détails) :  $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 6 & 1 \\ 1 & 1 & 1 \end{pmatrix}$



**Exercice 6 Détection des contours** Le laplacien ou le gradient permettent de mesurer les variations de teintes et donc de détecter les contours des objets de l'image.

Donner la matrice de convolution associée au laplacien afin de repérer les variations de teinte. Appliquer ensuite le passage au négatif pour obtenir une esquisse associée à l'image.