

Séquence 2 :

PIXEL ART ET PHOTOGRAPHIE NUMÉRIQUE

1. Réalisation d'un « Pixel Art »

Définition du Pixel Art (art du pixel) : désigne une composition numérique qui utilise une résolution d'écran basse et un nombre de couleurs limité.

→ Diaporama diapositives 1 à 9

Activité d'introduction : activité de Rémy Pages (académie de Montpellier)

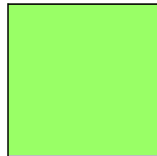
Le « Pixel Art » obtenu est à coller et conserver pour la suite.

Projeter aux élèves les diapositives 1 à 9 du diaporama où l'activité est présentée.

2. Composantes Rouge, Verte, Bleue d'un pixel

Activité :

1) Sur l'application RVB (fichier apk joint, ou application équivalente sur le Playstore) ajuster la luminosité des couleurs *rouge*, *verte* et *bleue* afin d'obtenir un mélange qui s'approche au mieux de la teinte du « pixel » vert utilisé dans l'activité précédente.



Compléter ci-dessous la luminosité obtenue pour chacune des couleurs :

Rouge	←
Vert	←
Bleu	←

Correction avec Paint ou projection de la diapositive 5

→ Diaporama diapositive 5

2) Calculer la luminosité moyenne des valeurs obtenues précédemment.

Moyenne ←

3) Sur l'application RVB régler ainsi la luminosité pour chacune des couleurs :

Rouge	←	Moyenne (calculée à la question 2))
Vert	←	Moyenne
Bleu	←	Moyenne

Qu'obtient-on ?

4) a) Sur l'application RVB ajuster la luminosité des couleurs *rouge*, *verte* et *bleue* afin d'obtenir du blanc et relever alors les luminosités obtenues.

b) Relever les luminosités qui permettent d'obtenir du noir.

5) Compléter tous les ... du tableau-ci dessous, en utilisant les résultats de 1), 3) et 4) :

Pixel	Rouge	Vert	Bleu	Valeurs pour Gris
Rouge	255	51	51	...
Vert
Bleu	102	102	255	...
Cyan	102	255	255	...
Magenta	255	51	255	...
Jaune	255	255	153	...
Noir
Blanc

6) Transformer votre image « Pixel Art » en niveau de gris.

7) Remplacer chaque pixel gris de votre « Pixel Art » par son négatif (négatif = $255 - \text{gris}$)

8) En justifiant votre démarche, transformer votre image « Pixel Art » en noir et blanc.

9) Cette dernière image doit être transmise à un autre groupe sous forme codée :
une seule ligne est utilisable, les deux premières valeurs sont libres, les autres valeurs ne doivent contenir que des 0 ou des 1.

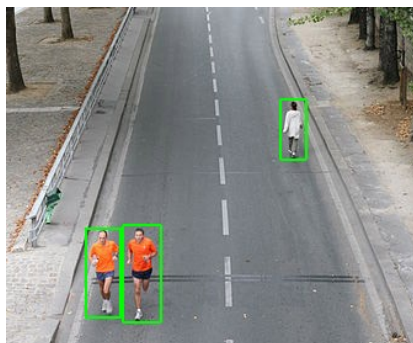
Par exemple : ?, ?, 1,0,1,0,1,0,0,1,1,1,0, ...etc

3. Traitement d'image : détection de contour

→ Diaporama diapositives 10 à 15

Activité 1 :

La détection de contours est utilisée pour détecter des objets ou des personnes sur des images numériques. Les applications sont nombreuses : voitures autonomes, vidéosurveillance...



Dans la question 1), vous allez comprendre comment fonctionne un filtre horizontal.
On appliquera ensuite dans la question 2) ce filtre horizontal afin de détecter les contours d'une image.

1) Dans cet exemple chaque pixel peut prendre comme valeur 0 (noir) ou 1 (blanc).

PIXEL 1	PIXEL 2	PIXEL 3	PIXEL 4	PIXEL 5	PIXEL 6
0	1	0	1	1	0

Pour calculer l'image du Pixel 4 on effectue le calcul suivant :

$$-1 \times (\text{Pixel 3}) + 0 \times (\text{Pixel 4}) + 1 \times (\text{Pixel 5}).$$

On obtient 1.

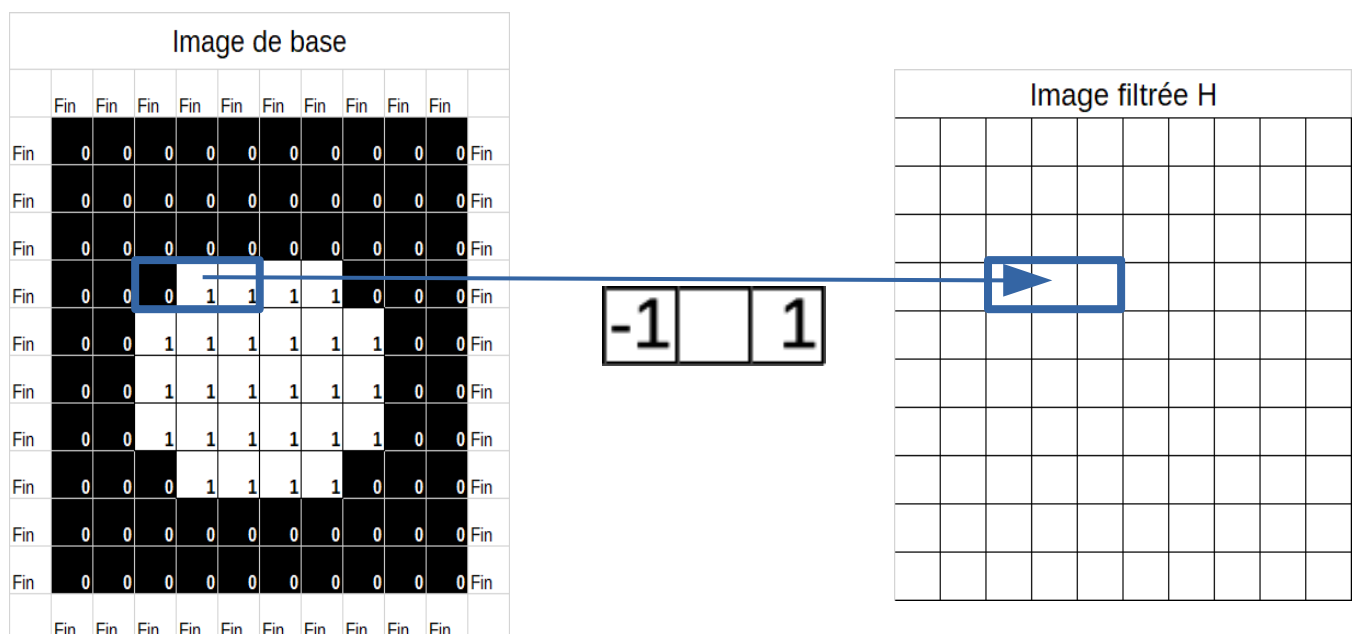
-1		1
----	--	---

			1		
PIXEL 1	PIXEL 2	PIXEL 3	PIXEL 4	PIXEL 5	PIXEL 6

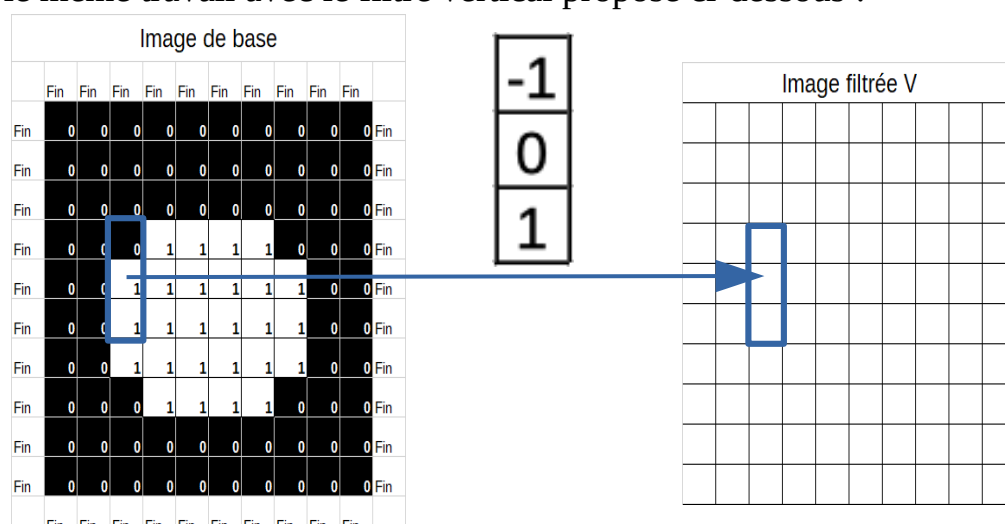
Compléter le tableau pour les pixels 2,3,4 et 5. Attention si le résultat du calcul donne -1, on remplace par 1.

2) Détection de contour d'une image (filtre horizontal) :

Appliquer le filtre horizontal sur l'image ci-dessous (image de base) afin d'en obtenir les contours (image filtrée H).

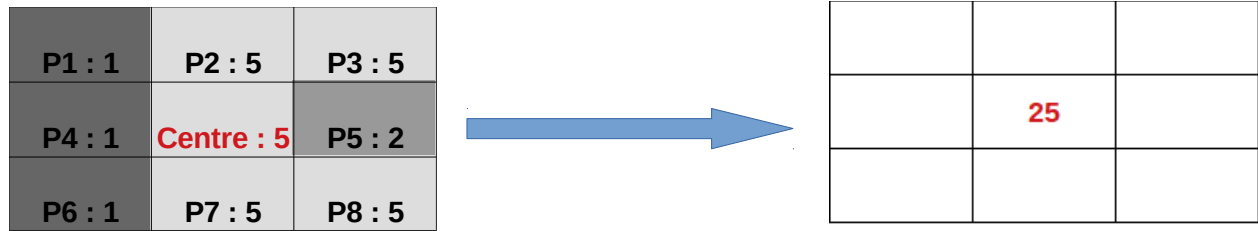


3) Effectuer le même travail avec le filtre vertical proposé ci-dessous :



Activité 2 : Un algorithme plus fin

On va calculer une « distance » par rapport à chacun des 4 pixels adjacents aux côtés d'un pixel donné, en procédant ainsi :



Une grille de 3×3 pixels

Pour calculer la « distance » par rapport au centre, on effectue le calcul suivant :

$$D = (\text{Centre} - P2)^2 + (\text{Centre} - P4)^2 + (\text{Centre} - P5)^2 + (\text{Centre} - P7)^2$$

Ici, on obtient :

$$D = (5 - 5)^2 + (5 - 1)^2 + (5 - 2)^2 + (5 - 5)^2$$

$$D = 25$$

Il peut aussi être judicieux d'utiliser un programme en langage Python comme ci-dessous :

```
def distance(c,p2,p4,p5,p7):
    d = (c-p2)**2+(c-p4)**2+(c-p5)**2+(c-p7)**2
    return d
```

```
>>> distance(5,5,1,2,5)
25
```

1) Compléter la figure de droite en effectuant le calcul des distances pour le pixel manquant.

Image de base

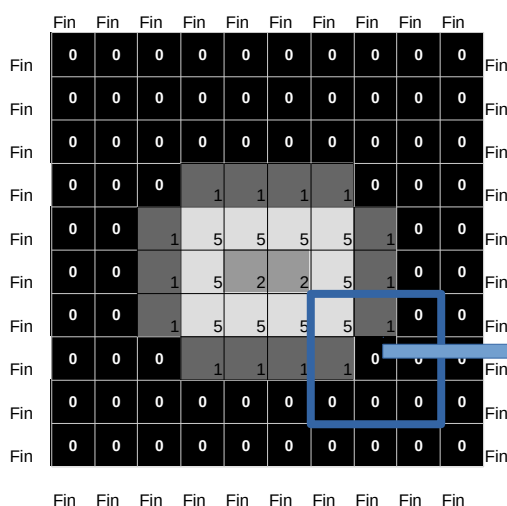
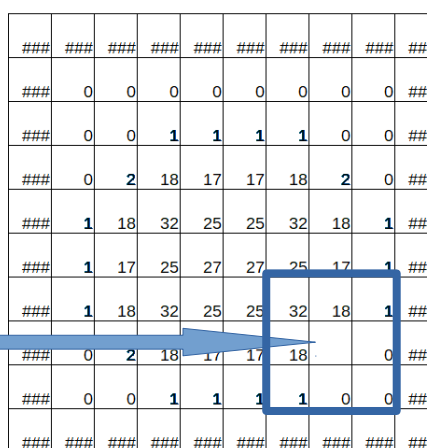


Image filtrée avec calcul de distance



2) Compléter la phrase : « Pour obtenir le contour de la figure, il faut colorer en bleu tous les pixels dont la valeur est comprise entre et ».