

Projet: Détection de points d'intérêts dans une image à l'aide du filtre de Sobel

Binôme : Jean Baptiste NARI et Thi Huong DANG

L'objectif de ce projet est de créer un détecteur de points d'intérêts d'une image. Pour répondre à l'objectif du projet, nous avons choisi de travailler avec un filtre de Sobel vertical et horizontal qui permet de détecter le contour d'une image. En post-traitement, en réalisant un seuillage, nous obtenons les points d'intérêts de notre image.

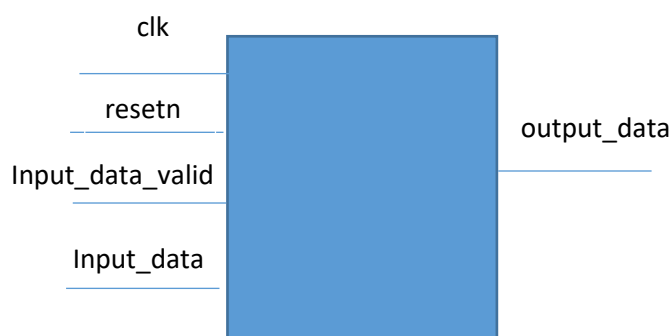
Dans ce projet il n'est pas demandé d'implémenter le programme sur la carte CoraZ7 mais de se limiter à des résultats de simulation.

I. Plan de validation

Durant ce projet nous avons suivi un plan de validation.

Analyse :

- 1) Vérifier le fichier testbench de lecture et écriture d'image.



Les signaux input_data et output_data sont identiques.

Image entrée

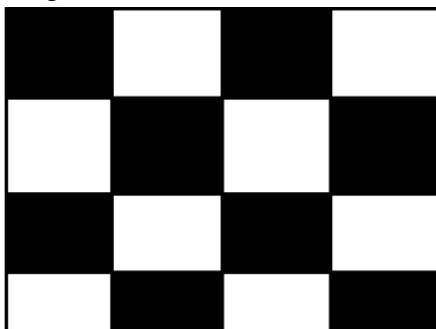
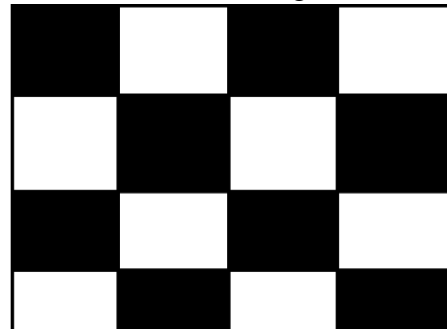
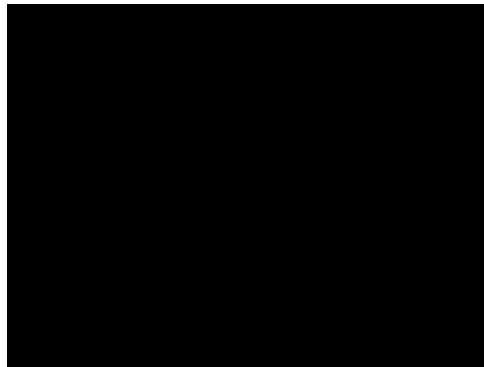


Image sortie



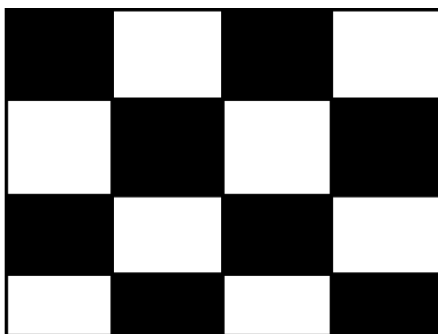
La différence entre l'image sortante et l'image entrante :



2) Test du filtre unitaire

Avant d'appliquer le filtre de Sobel à notre image, nous utilisons le filtre unitaire pour tester notre programme. L'image de sortie doit être identique à l'image d'entrée.

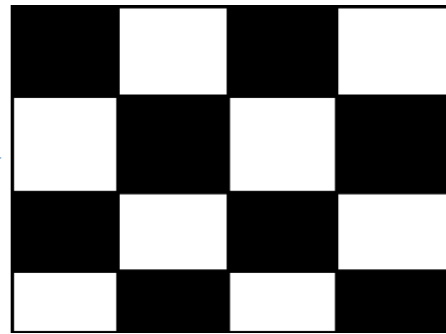
Image avant filtrage



Filtre unitaire



Image après filtrage

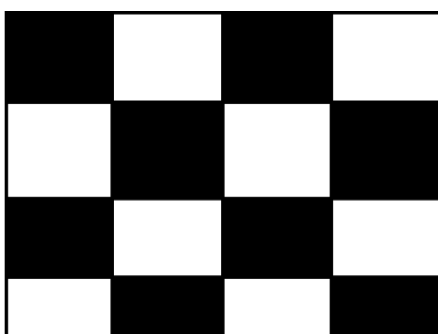


3) Test du filtre de Sobel en direction verticale

La matrice de filtre de Sobel G_x est :

-1	0	1
-2	0	2
-1	0	1

Image avant filtrage

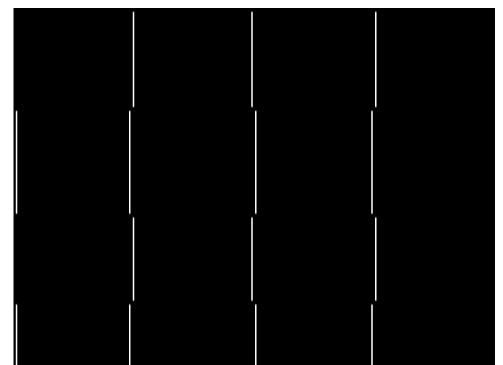


Filtre G_x

-1	0	1
-2	0	2
-1	0	1



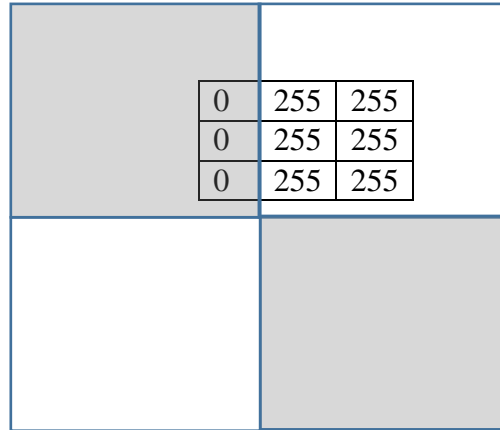
Image après filtrage



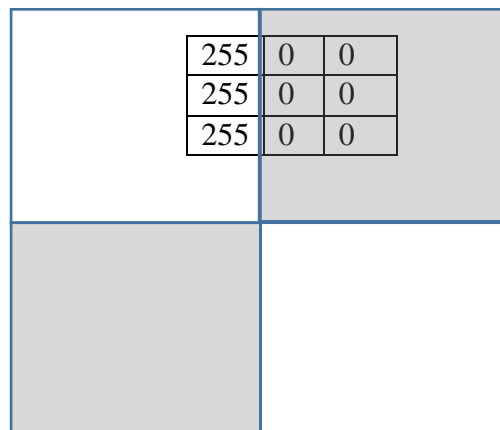
Lorsque la fenêtre reste entièrement dans la zone blanche ou noire, la valeur du pixel après filtrage =0 (output_data=0).

Au bord des carreaux de 2 couleurs différentes :

-) Les pixels aux bords des carrés noirs et blancs prennent 1020 après filtrage



-) Les pixels au bord des carrés blancs et noirs prennent valeurs négatives après filtrage -1020. Par définition, les pixels prendront les valeurs absolues de ces valeurs négatives.

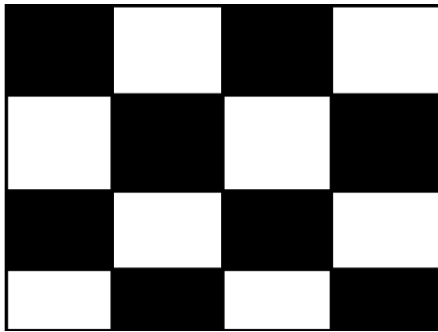


4) Test du filtre de Sobel en direction horizontale

La matrice de filtre de Sobel G_y est :

1	2	1
0	0	0
-1	-2	-1

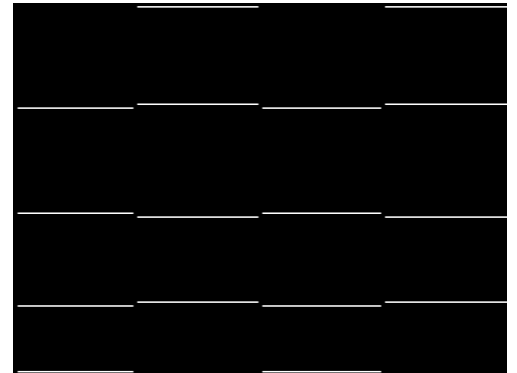
Image avant filtrage



Filtre G_y



Image après filtrage



Très similaire au filtre G_x ,

Lorsque la fenêtre reste entièrement dans la zone blanche ou noire, la valeur du pixel après filtrage = 0 (output_data=0).

Au bord des carreaux de 2 couleurs différentes :

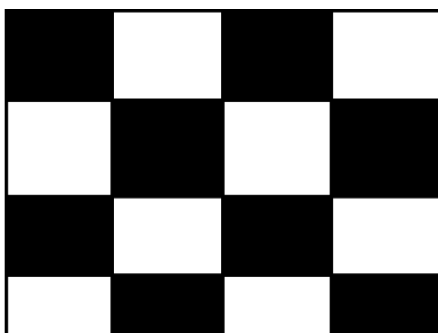
-) Les pixels au bord des carrés noirs et blancs prennent 1020 après filtrage.

-) Les pixels au bord des carrés blancs et noirs prennent valeurs négatives -1020 après filtrage. Par définition, les pixels prendront les valeurs absolues de ces valeurs négatives.

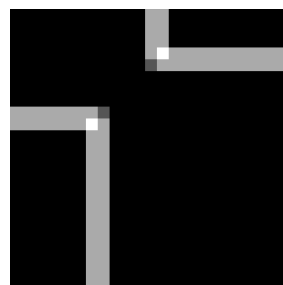
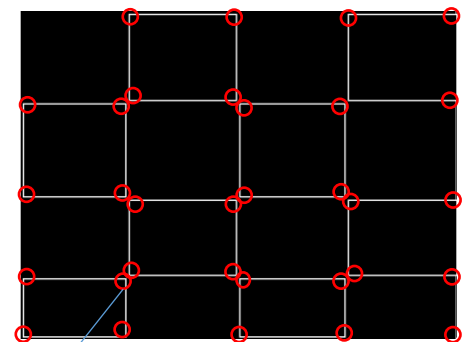
5) Teste le filtre de Sobel en 2 directions

Le filtre de Sobel en 2 directions est créé par addition des filtres verticaux et horizontaux, $G = |G_x| + |G_y|$.

Nous nous attendons à détecter les coins représentés par les cercles rouges.



Filtre G_y et G_x



6) Utilisation du filtre de Sobel sur une image naturelle.

Nous allons choisir une image et lui appliquer le filtre Sobel, par exemple l'image ci-dessous.



Filtre de Sobel



Démonstration :

Affichage des images avant et après l'application du filtre de Sobel à l'aide du logiciel Fiji

II. Résultat

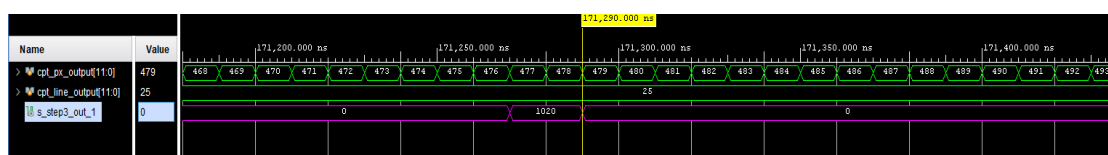
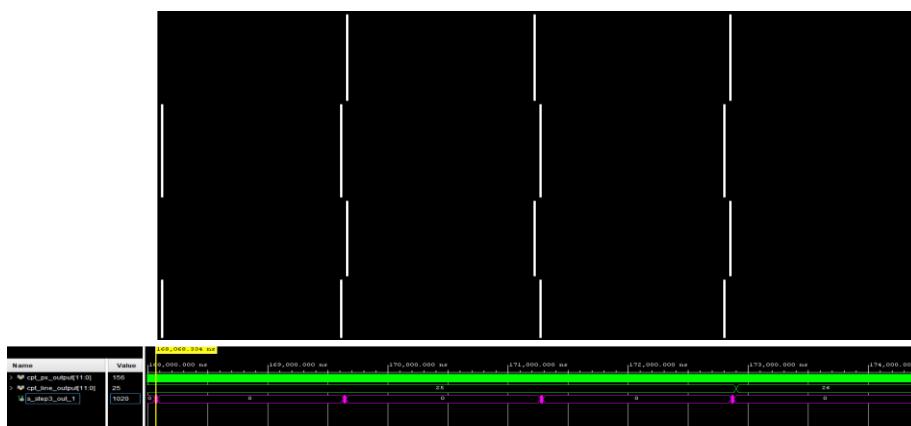
1) Test du fichier testbench de lecture et écriture d'image

En utilisant le logiciel Fiji, nous faisons la soustraction de deux images et le résultat est une image entièrement noire. L'image de sortie est donc identique à l'image d'entrée.

2) Test du filtre unitaire

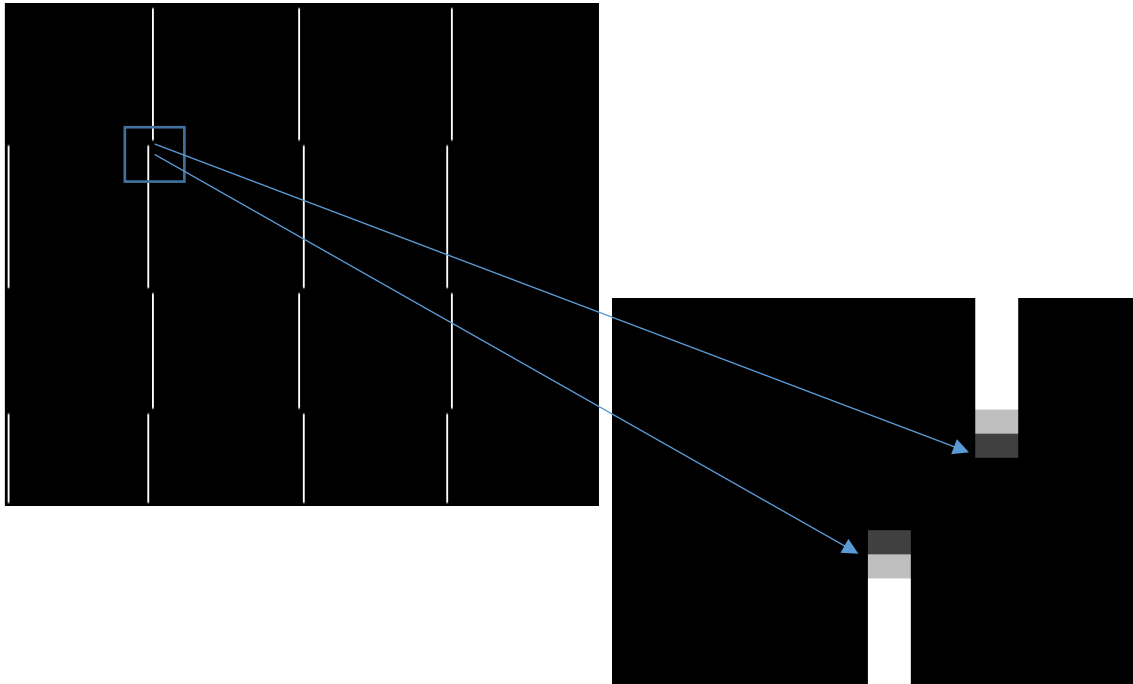
Cette étape est validée dans le projet VGA.

3) Test du filtre de Sobel en direction verticale



L'image filtrée par le filtre de Sobel vertical est exactement comme attendue.

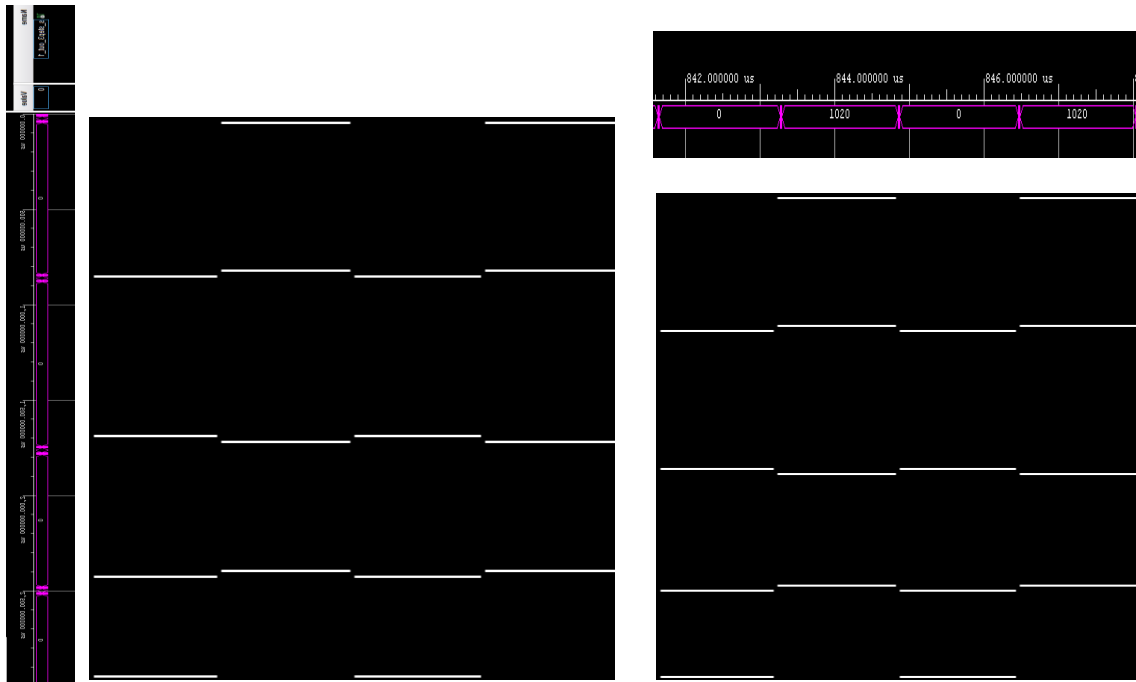
Le signal magenta de chronogramme présente la valeur du pixel après filtrage. La valeur 0 correspond à la partie noire sur l'image. Nous obtenons quatre parties noires de l'image sur le chronogramme. Entre 2 parties noires, nous avons 2 pixels qui prennent la valeur 1020. Ils correspondent à des lignes blanches sur l'image.



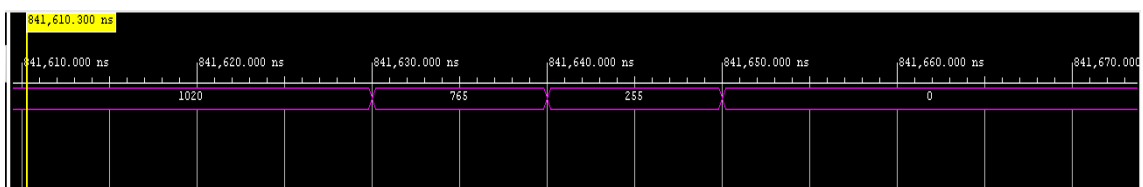
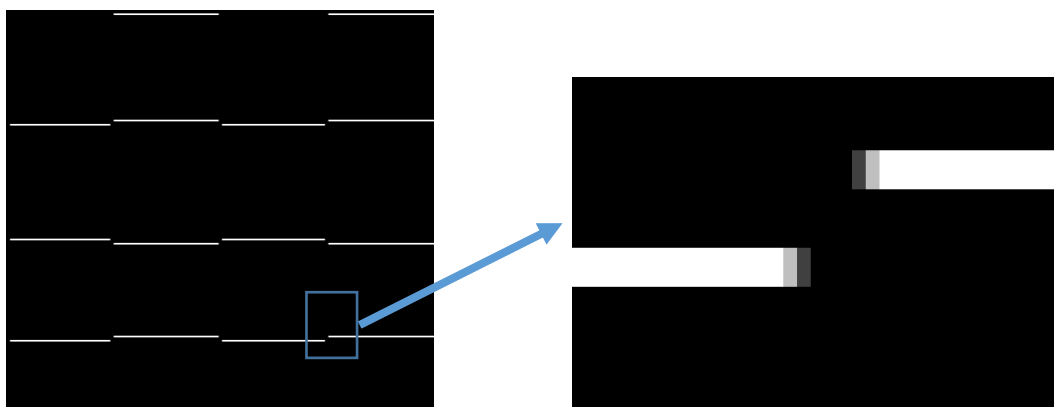
Sur les lignes verticales, nous constatons un changement de couleur au bord de chaque ligne blanche. On le retrouve sur le chronogramme en regardant les mêmes pixels 156 et 157 de 3 lignes consécutives 129, 130 et 131. La valeur après filtrage du pixel diminue de 1020 à 765 et puis à 255.



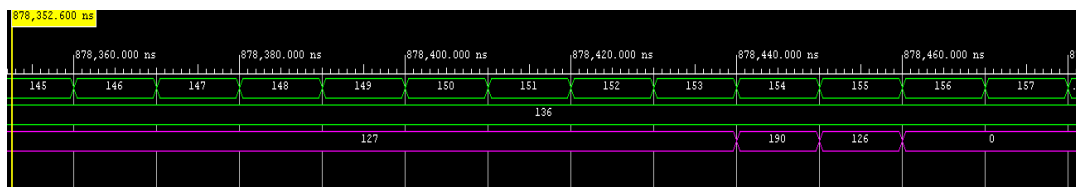
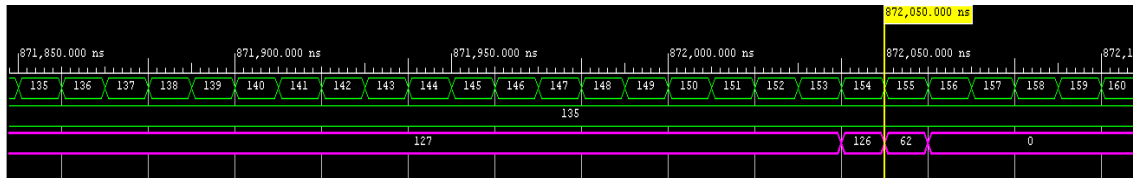
4) Test du filtre de Sobel en direction horizontale



L'image filtrée par le filtre de Sobel dans la direction horizontale est exactement comme attendue. Le chronogramme présente 4 parties noires et des lignes horizontales blanches. À la fin de chaque ligne, on observe un changement de couleur expliqué par le changement de la valeur du pixel après filtrage de 1020 à 765 puis à 255.



5) Test du filtre de Sobel en 2 directions $G=|G_x| + |G_y|$

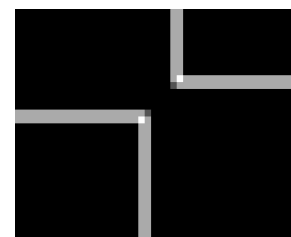
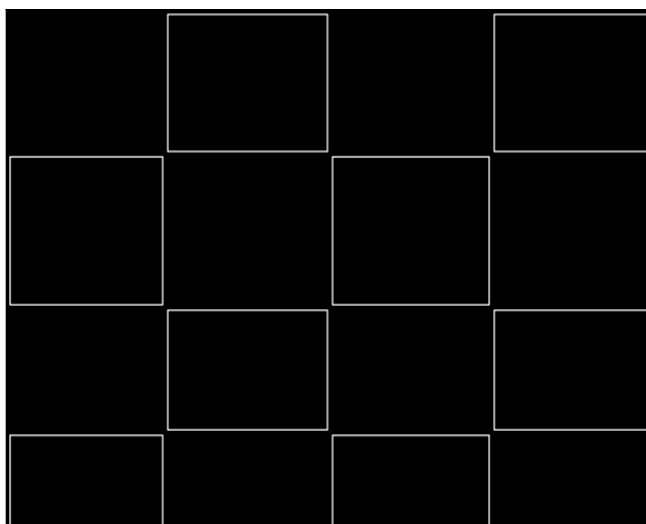


Les coins des carrés se trouvent dans la zone des lignes verticales. En observant deux lignes consécutives 135 et 136, le pixel le plus sombre est représenté par la valeur la plus faible (62) et le pixel le plus clair par la valeur la plus élevée (190) sur le chronogramme.

6) Résultat sur l'image

Dans cette section, nous présentons les résultats du filtrage d'images obtenus avec nos codes vhd.

a) Application du filtre de Sobel en deux directions sur le damier.



- b) Application du filtre de Sobel en deux directions sur une image naturelle.



Ce projet nous a permis de mettre applications nos connaissances et de travailler en binôme sur des problématiques de traitement d'image.