10/07/2023

Filtre de Sobel

Plan de Validation

Atmani Hicham & Kamal Kherchouch
FORMATION SAFRAN CHEZ AJC

Date	Version	Remarque	Auteur
10/07/2023	A0	Création du Document de plan de validation pour le projet d'implémentation du filtre de Sobel	AHI & KKH

Table des matières

1. Plan de validation de la solution proposée

1.1. Présentation du filtre de gauss

Le filtre Gaussien fait partie de la famille des filtres d'atténuations. Ce filtre permet d'uniformiser les images en donnant un côté flouté aux détails présents sur l'image. Ce qui revient à dire en donnant un côté flouté et harmonieux à l'image.

Dans notre cas, notre damier ne sera donc plus net dans les zones de transitions blanc/noir ou noir/blanc mais plutôt flouté dans ces mêmes zones de transitions.

Pour ce faire, le filtre gaussien va (pixel par pixel) réaliser une convolution avec tous les pixels avoisinants le pixel à traiter. Dans notre cas, nous utiliserons donc le cœur de convolution suivant :

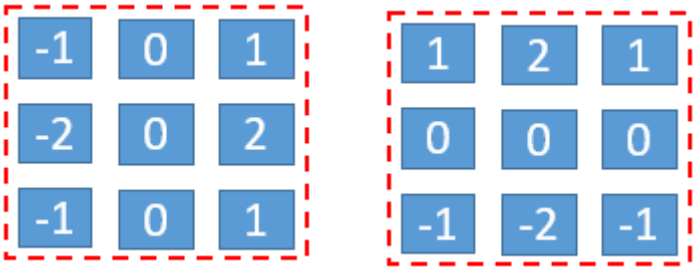
1	2	1	K1	K2	K3
2	4	2	K4	K5	K6
1	2	1	K7	K8	K9

1.1. Présentation du filtre de Sobel

Le filtre de Sobel est un filtre de traitement d'image. Ce filtre utilise deux noyaux de convolution, l'un en horizontale, le second en vertical ce qui permet de faire de la détection de contour. Ces noyaux permettront de calculer la convolution bit par bit de l'image à traiter. Ces convolutions nous permettront de calculer le gradient de l'image qui représentera l'intensité et la direction des contours.

Les deux Noyau de Sobel sont les suivants :

- en Horizontal / en Vertical



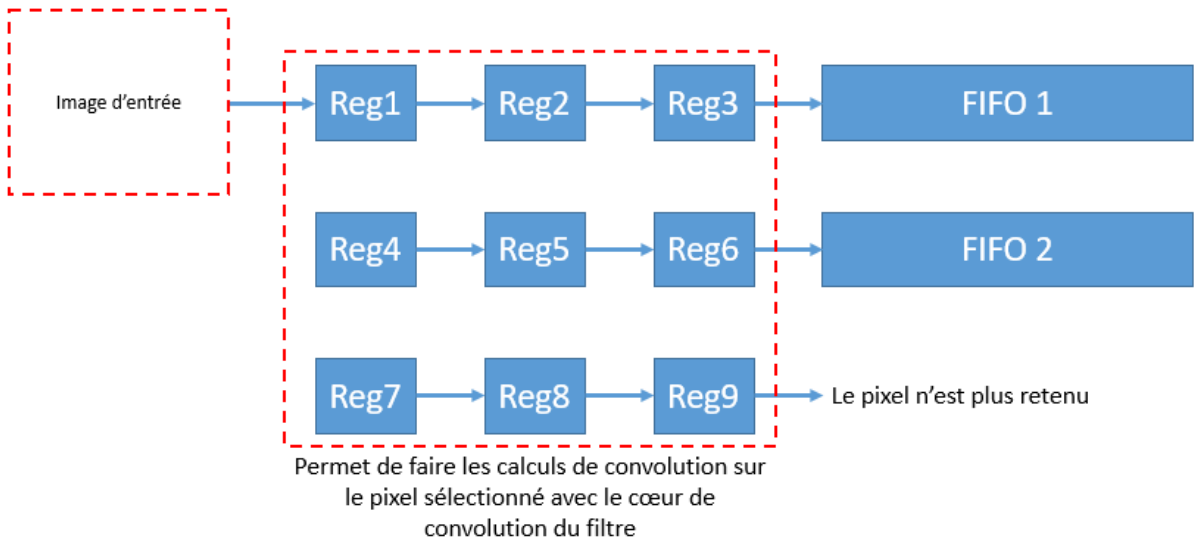
Le résultat final est une image où les pixels représentent la magnitude du gradient, ce qui met en évidence les contours de l'image. Les valeurs plus élevées dans l'image filtrée de Sobel indiquent des changements plus importants d'intensité des pixels et donc des contours plus prononcés.

Il convient de noter que le filtre de Sobel est sensible au bruit. Par conséquent, il est souvent utilisé en conjonction avec d'autres techniques de traitement d'image, telles que la suppression du bruit ou la binarisation, pour améliorer les résultats de détection des contours.

Pour faire simple, l'opérateur calcule le gradient de l'intensité de chaque pixel. Ceci indique la direction de la plus forte variation du clair au sombre, ainsi que le taux de changement dans cette direction. On connaît alors les points de changement soudain de luminosité, correspondant probablement à des bords, ainsi que l'orientation de ces bords.

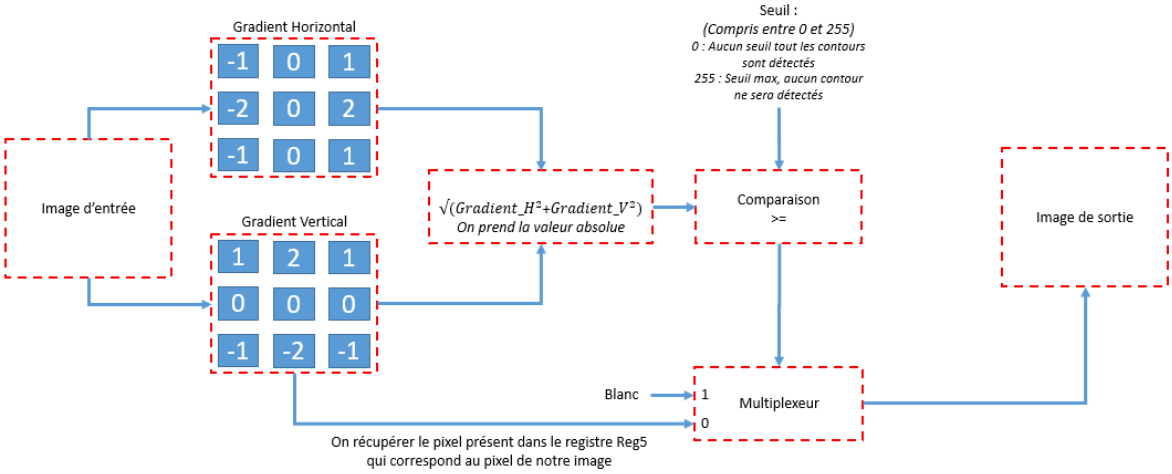
1.2. Stockage des pixels dans les registres

Pour pouvoir réaliser les calculs de convolution (peu importe le filtre que nous utiliserons) , il nous faut stocker les valeurs des pixels d'entrés (Image d'entrée). L'enregistrement des pixels se fait ligne par ligne avec des passages par des registres afin de pouvoir réaliser le calcul de convolution du filtre. Pour ce faire, nous utiliserons deux Fifos ainsi que 9 registres comme le montre le schéma suivant :



1.3. Réalisation de la solution proposée

L'architecture de la solution finale sera la suivante :



La description du fonctionnement de notre solution est la suivante :

On vient récupérer les pixels de notre image d'entrée en réalisant un sliding window à l'aide du stockage des pixels dans nos registres et nos FIFOs.
Une fois les pixels récupérés, on réalise le calcul de convolution en horizontal et en vertical.
Ces résultats passent ensuite par la détermination de la valeur absolue du gradient du pixel, à l'aide de la formule suivante :

$$\sqrt{(Gradient\_H^2+Gradient\_V^2)}$$

Pour finir, le résultat du gradient sera ensuite comparé à un seuil à l'aide d'un multiplexeur. Si la comparaison est supérieure à ce dernier on détecte le contour en fixant la couleur blanche, sinon on garde la valeur d'entrée de notre pixel.

La valeur du seuil sera à déterminer lors de la validation du projet. Le seuil est compris entre 0 et 255. Avec 0 étant aucun seuil, on détecte tous les contours et 255 étant le seuil max, on ne détecte aucun contour.

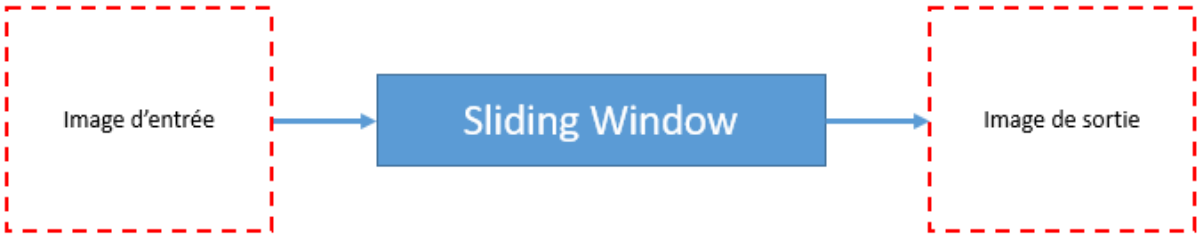
2. Plan de validation de la solution finale

2.1. Validation du passe plat

L'objectif de cette étape est de valider le passe plat. C'est-à-dire valider que l'image de sortie reste conforme à l'image d'entrée.

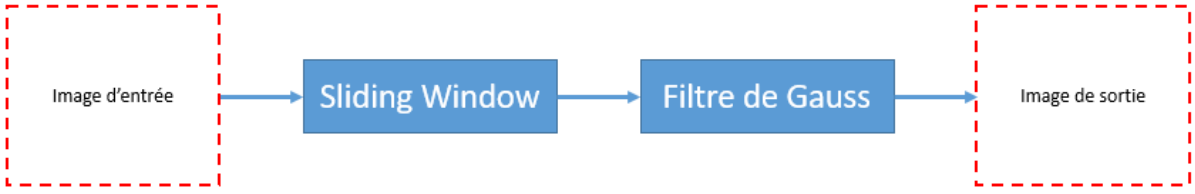
2.2. Validation du sliding window

L'objectif est d'intégrer à notre solution le principe de sliding window. C'est-à-dire le stockage des pixels de notre image sont stockés dans les registres et les fifos.
Cette partie va nous permettre de valider l'adaptation de la sortie d'image. C'est-à-dire que la sortie ne subit pas la latence des Fifos



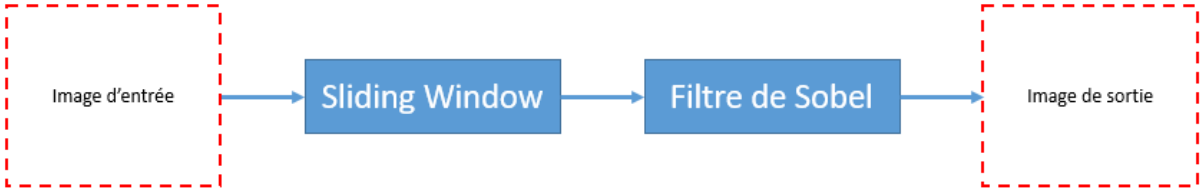
2.3. Validation du filtre de Gauss

Cette étape intermédiaire nous permet d'intégrer le filtre de Gauss réalisé lors du projet précédent et donc de valider l'utilisation de ce dernier (avec les produits de convolutions).



2.4. Validation du filtre de Sobel

Cette étape intermédiaire nous permet d'intégrer le filtre de Sobel à réaliser.



La validation du filtre de Sobel se fera en plusieurs étapes :

- Validation de la convolution en H
- Validation de la convolution en V
- Validation du calcul de gradient
- Validation de la mise en place du seuil avec la détermination de ce dernier.