

0.Introduction

Au cours de ce rapport, nous allons réaliser une analyse comparative de trois méthodes de détection de contours : Sobel, Laplacien et Canny. L'objectif de cette étude est d'évaluer la précision de chaque méthode en testant plusieurs images. Pour chaque image, nous comparerons les contours générés par chaque méthode à l'image de contours de référence, tracée manuellement. Nous utiliserons des critères comme le taux de faux positifs, le taux de faux négatifs et la performance pour quantifier et comparer l'efficacité des différentes techniques de détection de contours.

1. Méthodes de Détection de Contours

1.1 Laplacien

La méthode du Laplacien est une technique de détection de contours basée sur le calcul de la dérivée seconde d'une image. Contrairement aux méthodes comme Sobel ou Canny, qui utilisent la dérivée première, le Laplacien permet de détecter les zones de forte variation d'intensité dans toutes les directions.

Le Laplacien est très sensible au bruit, donc si on ne floute pas, il peut détecter des contours non pertinents. Ainsi, avant d'appliquer le Laplacien, l'image est souvent floutée avec un filtre gaussien.

Après le flou, on applique le filtre Laplacien.

Le Laplacien utilise la convolution avec un noyau Laplacien :

Laplacian(blurred, generated, CV_16S, laplacianSize);

- *blurred* → L'image floutée en entrée
- *generated* → L'image de sortie contenant les contours
- *CV_16S* → On utilise un format 16 bits signé pour éviter les débordements
- *laplacianSize* → La taille du noyau (impair)

La convolution consiste à faire glisser un noyau de taille fixe (une matrice) sur l'image, pixel par pixel, et à appliquer une opération arithmétique entre les valeurs des pixels de l'image et les valeurs du noyau.

Les résultats du Laplacien peuvent être négatifs, donc on les ramène entre 0 et 255 pour l'affichage en utilisant la méthode `minMaxLoc` :

minMaxLoc(generated, &minVal, &maxVal);

*generated.convertTo(generated, CV_8U, 255.0 / (maxVal - minVal), -minVal * 255.0 / (maxVal - minVal));*

Avantages :

- Simple et rapide
- Détecte les contours dans toutes les directions

Inconvénients :

- Très sensible au bruit → nécessité du flou avant application
- Produit des bords épais au lieu de contours fins

1.2 Sobel

La méthode Sobel repose sur le calcul du gradient de l'intensité de l'image. Elle permet de mettre en évidence les transitions brusques dans les pixels de l'image, ce qui correspond aux contours des objets présents dans l'image.

Le filtre Sobel calcule les dérivées de l'image dans deux directions : horizontale et verticale. Cela permet de détecter les variations d'intensité lumineuse dans ces directions et de localiser les bords.

Le gradient mesure l'intensité du changement à travers l'image. Dans le cas des bords, les gradients sont élevés.

Le filtre Sobel est appliqué à chaque pixel de l'image pour estimer les dérivées partielles dans les directions x (horizontale) et y (verticale).

Sobel utilise donc deux noyaux pour calculer les dérivées en x et en y :

```
Sobel(blurred, sobelX, CV_16S, 1, 0, ksize);
```

```
Sobel(blurred, sobelY, CV_16S, 0, 1, ksize);
```

Pour chaque pixel de l'image, le noyau est appliqué à la région de l'image autour de ce pixel.

Une fois les gradients en x et y calculés, on combine ces deux informations pour obtenir l'intensité du gradient global à chaque pixel :

```
Mat sobel = abs(sobelX) * 0.5 + abs(sobelY) * 0.5;
```

Les résultats sont enfin normalisés pour mieux visualiser les bords.

Avantages :

- Bonne détection des bords

Inconvénients :

- Sensibilité au bruit
- Réponse moins précise aux bords diagonaux
- Perte de détails fins

1.3 Canny

Canny repose sur la même méthode que Sobel mais contient une dernière étape.

Dans cette dernière, elle applique des seuils avec hystérésis pour finaliser la détection des bords. Deux seuils sont utilisés :

- Un seuil bas : Si un pixel a un gradient supérieur à ce seuil, il est considéré comme faisant partie d'un bord.
- Un seuil haut : Si un pixel a un gradient supérieur à ce seuil, il est considéré comme faisant partie d'un bord.

Canny(blurred, generated, threshold1Canny, threshold2Canny);

Avantages :

- Haute précision dans la détection des bords
- Réduction efficace du bruit
- Contrôle avec les seuils d'hystérésis

Inconvénients :

- Temps de calcul élevé

2. Comparaison des Méthodes

Méthode	Avantages	Inconvénients
Sobel	Simple et rapide, bon pour des bords grossiers	Sensible au bruit, moins précis pour des bords fins
Laplacian	Détecte bien les transitions nettes	Sensible au bruit
Canny	Très précis, gère bien les bords faibles et bruités	Plus lent

3. Comparaison des Contours Détectés

Comparaison des contours détectés avec les contours de référence d'une image

- Contours détectés : nombre de pixels non nuls (pixels contours) dans l'image générée par la méthode de détection (qui correspondent aux contours détectés)

double contours_detectes = countNonZero(generated);

- Contours de référence : nombre de pixels non nuls (pixels contours) dans l'image de référence, qui représente les contours "corrects"

double contours_reference = countNonZero(reference);

- Contours corrects : nombre de pixels contours correctement détectés dans l'image calculée selon l'image de référence (c'est-à-dire les contours qui sont présents dans les deux images)

double contours_corrects = countNonZero(generated & reference);

- Faux positifs : pixels qui ont été détectés comme des contours par la méthode, mais qui ne sont pas présents dans l'image de référence

double faux_positifs = contours_detectes - contours_corrects;

- Faux négatifs : pixels qui devraient être des contours (selon l'image de référence), mais qui n'ont pas été détectés

double faux_negatifs = contours_reference - contours_corrects;

Calcul des performances :

- Performance : mesure la proportion de contours correctement détectés par rapport au nombre total de pixels. Plus ce pourcentage est élevé, meilleure est la détection des contours.

*double performance = (contours_corrects / (contours_corrects + faux_positifs + faux_negatifs)) * 100;*

- Taux de faux positifs

*double taux_faux_positifs = (faux_positifs / (contours_corrects + faux_positifs + faux_negatifs)) * 100;*

- Taux de faux négatifs

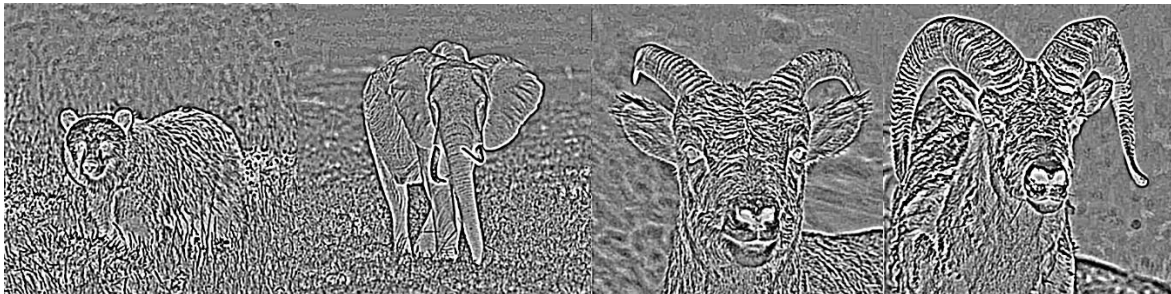
*double taux_faux_negatifs = (faux_negatifs / (contours_corrects + faux_positifs + faux_negatifs)) * 100;*

4. Résultats & Commentaires

Canny



Laplacien



Sobel



Les résultats ci-dessous sont observable dans le fichier texte se trouvant dans le dossier résultat y figurant aussi les images des contours

Image	Méthode	Contours détectés	Contours référence	Contours corrects	Faux positifs	Faux négatifs	Performance (%)	Taux faux positifs (%)	Taux faux négatifs (%)
elephant_3	Canny	3681	260324	2997	684	257327	1.14824	0.262061	98.5897
goat_5	Canny	11047	260117	10494	553	249623	4.02578	0.212146	95.7621
bear_5	Canny	4156	260610	3797	359	256813	1.45496	0.137564	98.4075
goat_8	Canny	9165	259216	8295	870	250921	3.18933	0.334505	96.4762
elephant_3	Laplacian	248303	260324	246544	1759	13780	94.071	0.671161	5.25788
goat_5	Laplacian	243456	260117	241103	2353	19014	91.8593	0.896483	7.24426

bear_5	Laplacian	235560	260610	234186	1374	26424	89.3894	0.52446	10.0861
goat_8	Laplacian	230649	259216	227748	2901	31468	86.8879	1.10676	12.0053
elephant_3	Sobel	258098	260324	255314	2784	5010	97.0377	1.05812	1.90416
goat_5	Sobel	259912	260117	256849	3063	3268	97.5944	1.16384	1.24174
bear_5	Sobel	261243	260610	259152	2091	1458	98.649	0.795962	0.555004
goat_8	Sobel	258692	259216	254312	4380	4904	96.4779	1.66163	1.86042

Méthode Canny

Bonne précision sur des bords nets, mais très faible performance (1,15% à 4,03%) à cause d'un taux élevé de faux négatifs (jusqu'à 98,59%). Canny n'a donc pas bien performé.

Méthode Laplacian

Très bonne détection des contours avec des performances atteignant jusqu'à 94,07%. Faible taux de faux négatifs (5,26% à 12%). Laplacian a montré une très bonne capacité à détecter les contours, avec une forte détection mais parfois légèrement susceptible aux faux positifs.

Méthode Sobel

Meilleure performance (97,04% à 98,65%), faible taux de faux négatifs (0,56% à 1,90%). Sobel a bien capturé les contours tout en maintenant des valeurs de faux positifs corrects, certains faux positifs (jusqu'à 1,66%). Sobel a offert les meilleurs résultats dans ces tests, avec une excellente détection des contours, peu de faux négatifs et une bonne précision.

5. Conclusion

La méthode **Sobel** est la plus fiable et performante parmi les trois, elle possède une détection précise des contours avec un faible taux de faux négatifs. La méthode **Laplacian** a bien fonctionné aussi, mais était plus sensible aux faux positifs. Enfin, la méthode **Canny** a montré des résultats moins satisfaisants à cause d'un taux élevé de faux négatifs et d'une faible performance.