

Université de Montpellier

Année 2022-2023

Faculté des Sciences

30 Place E.Bataillon, 34095 Montpellier

Rapport de Travaux Pratiques

TP n°4

Détection des contours d'une image avec utilisation du gradient (1er ordre)

par

Romain GALLERNE

Lien Git du TP :

<https://github.com/RomainGallerne/DonneesMultimedia/tree/main/TP4>

Encadrant de TP : M Marc HARTLEY

Responsable du module : M William PUECH

Table des matières

1	Introduction	2
2	Création de la carte de gradient d'une image	3
2.1	Images d'origine	3
2.2	Image de la norme de gradient.	4
2.3	Profil des deux images	5
3	Extraction des maximums locaux par seuillage	6
4	Seuillage par hystérésis des maximums locaux	8
5	Prétraitement par filtrage	10
5.1	Filtre Moyenneur	10
5.2	Filtre Gaussien	11
5.3	Comparaison	11
6	Conclusion	12

Introduction

Pour ce quatrième TP, nous allons travailler sur la détection de contours d'une image en utilisant des gradient. Pour cela il nous faudra tout d'abord pouvoir calculer les valeurs de gradients dans l'image. À l'aide de ces gradients, nous pourrons mettre en place un nouveau type de seuillage afin de détecter au mieux les contours de l'image.

Création de la carte de gradient d'une image

Dans cette première partie, nous nous intéressons à la norme du gradient des pixels de l'image. Pour cela, nous créons une fonction `norme_gradient.cpp` qui va, à partir d'une image donnée, retourner une image de même dimension ayant pour valeur du pixel (i,j) la norme du gradient du pixel (i,j) .

2.1 Images d'origine

Voici l'image de base qui sera utilisé durant l'ensemble de ce TP :



2.2 Image de la norme de gradient.

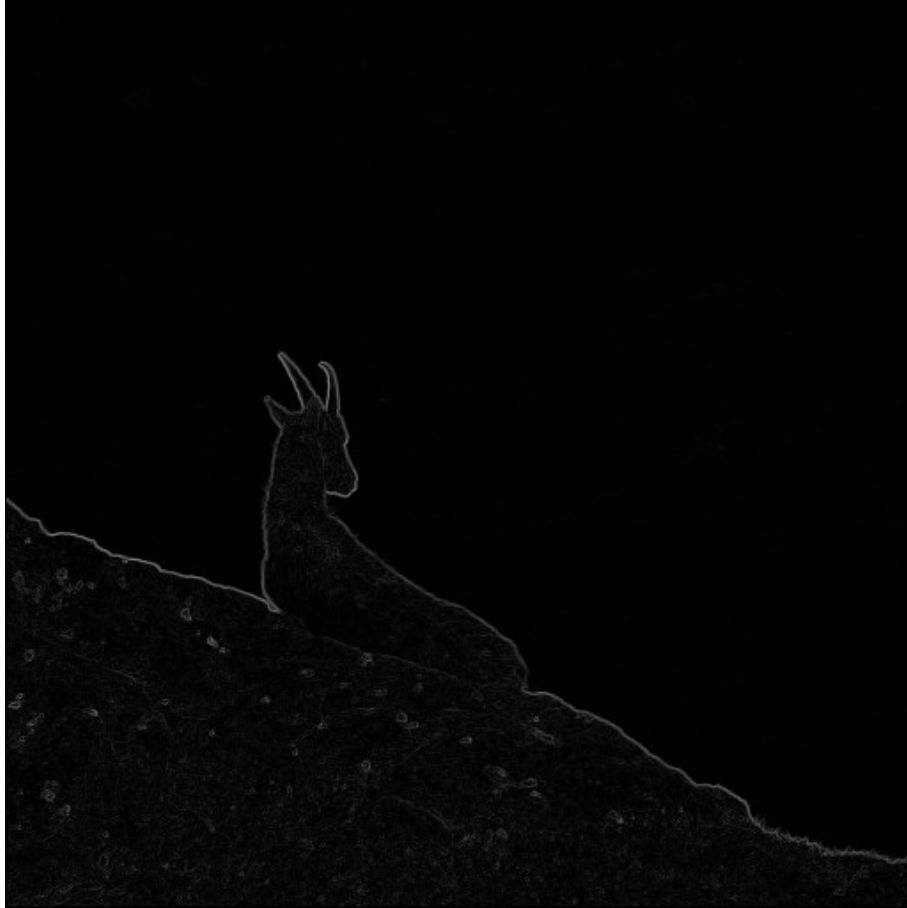
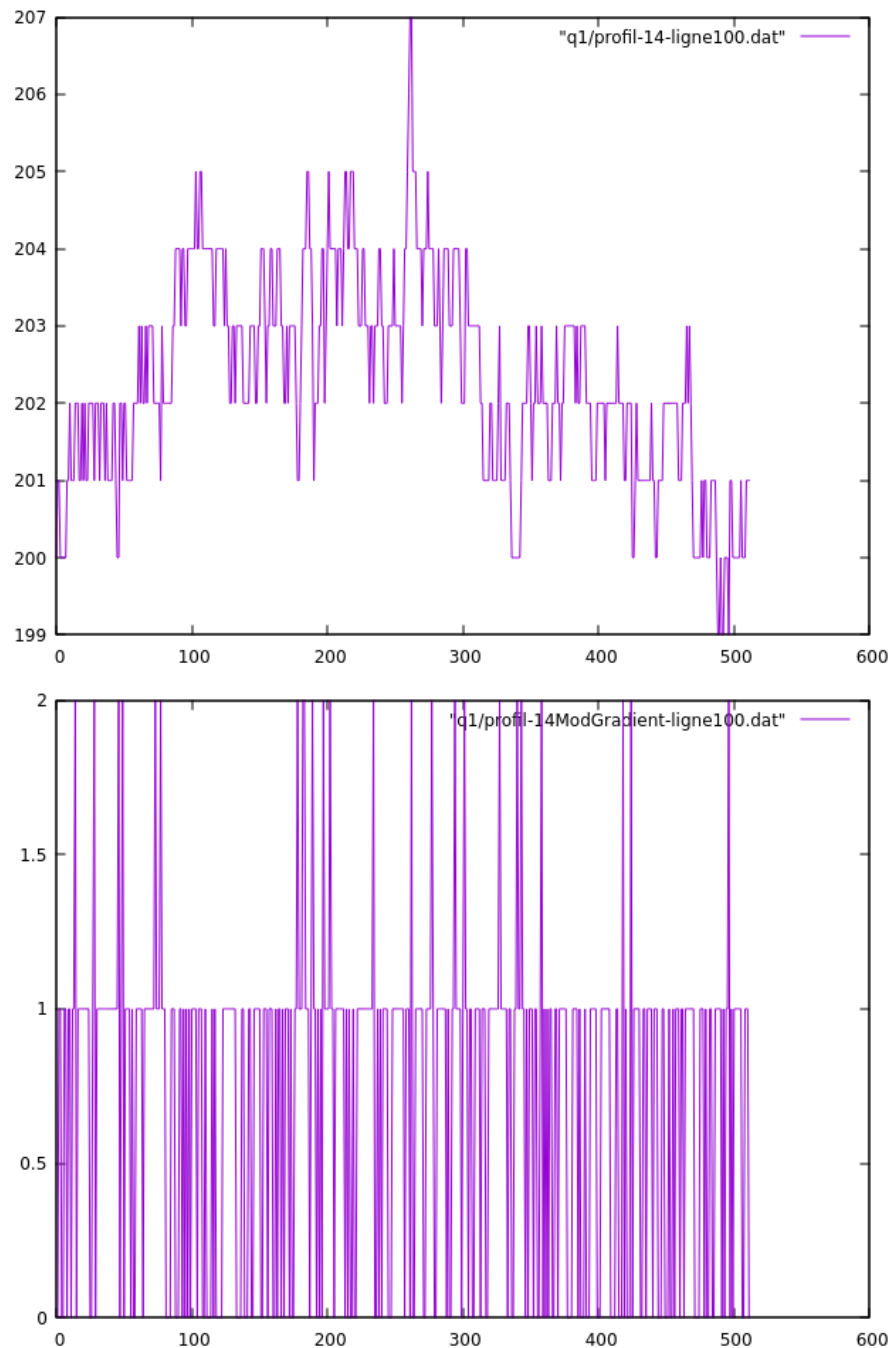


Image obtenu après application de la fonction calculant la norme de gradient.

2.3 Profil des deux images

On cherche maintenant à comparer le profil de l'image d'origine et le profil de la carte des gradient obtenu ci-avant.



Profil de la ligne 100 de l'image d'origine puis de la ligne 100 de la carte des normes de gradient.

Extraction des maximums locaux par seuillage

On souhaite à présent réaliser une extraction des maximums locaux par seuillage. Pour cela on applique l'algorithme suivant :

Si (norme du gradient $<$ seuil) alors 0, sinon 255

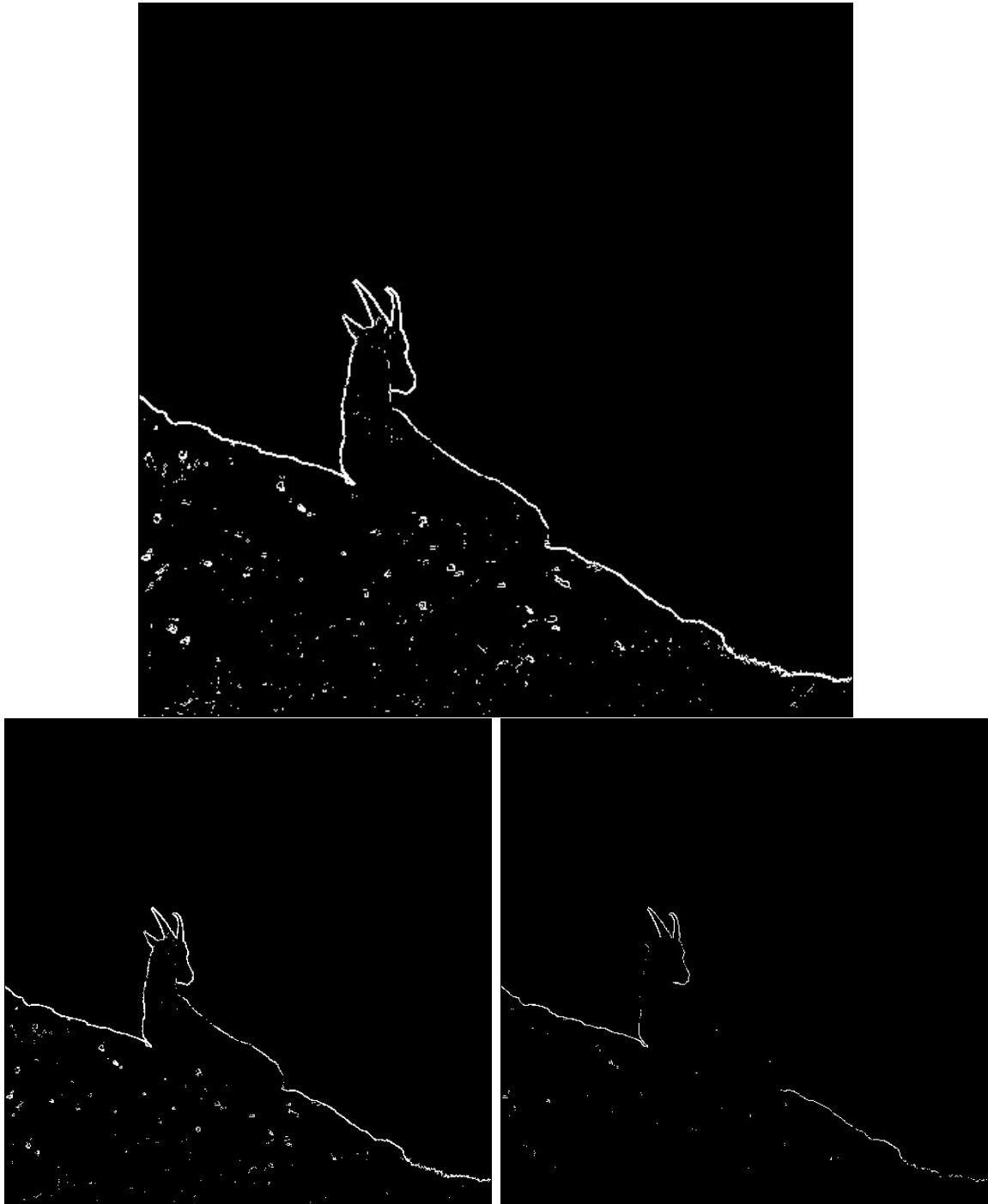


Image de la carte des gradient seuillée par 32 puis 40 et enfin 64.

Seuillage par hystérésis des maximums locaux

On remplace à présent le seuillage simpliste de la question ci-dessus par une seuillage par hystérésis. On appliquera donc désormais un nouvel algorithme de seuil :

- **Première lecture de l'image de la norme des gradients :**

si norme du gradient $\leq SB$ alors 0, si norme du gradient $\geq SH$ alors 255

- **Deuxième lecture de l'image de la norme de gradients pré-seuillée avec SB et SH :**

si $SB < \text{norme du gradient} < SH$ et qu'au moins 1 de ses voisins = 255 alors 255 sinon 0.





Images obtenus avec 3 valeurs de seuils (S_b , S_h) différents : (20,80), (30,60) et (40,50).

Prétraitement par filtrage

On répète les étapes précédentes en appliquant tout d'abord sur l'image un filtre moyennneur puis un filtre Gaussien.

5.1 Filtre Moyennneur



Image après application du filtre moyennneur puis du gradient.



Image moyenné gradué après application de trois seuils : 20, 25 et 32.

5.2 Filtre Gaussien



Image après application du filtre Gaussien puis du gradient.

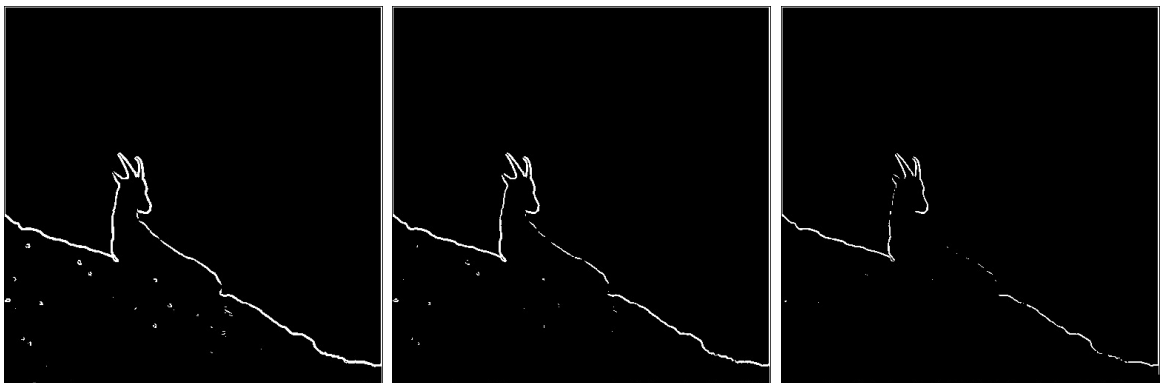


Image filtré par un filtre Gaussien gradué après application de trois seuils : 20, 25 et 32.

5.3 Comparaison

On remarque que l'application du filtre moyenneur permet de diminuer fortement le nombre de parasites après application du seuil. Ainsi les contours de l'images sont clairs et avec bien moins de parasites. Les bords de l'image sont ainsi bien plus nettes et faciles à observer puisque on retire la majorité des tâches qui venaient perturber l'image.

Le filtre Gaussien permet quand à lui d'obtenir des contours bien plus épais et nettes au prix d'un peu plus de parasites que le filtre moyenneur mais toujours moins que sans filtrage. On obtient ainsi des rebords particulièrement lisibles.

Conclusion

Dans ce TP nous avons donc utilisé différentes méthodes afin de détecter les contours d'une image. Pour cela il a été possible de modifier le type de seuillage, la valeur du seuil ainsi que d'appliquer un prétraitement sur l'image avant le seuillage (filtre moyenneur ou Gaussien).