**Traitement Image**

*Compte Rendu TP2*

PATRY Alan, ED-DAHMANY Ayoub

**Exercice 1 : modèle de débruitage par variation totale**

Ce premier exercice consiste à implémenter une autre méthode de débruitage par variation totale, elle est paramétrable par un lambda.

Cette méthode diffère essentiellement de Perona-Malik par la matrice diagonale utilisée dans l’approximation du Laplacien. Ses coefficients étant Une image contenant Police, symbole, ligne, blanc

Description générée automatiquement contre Une image contenant Police, ligne, blanc, diagramme

Description générée automatiquement pour Perona-Malik.

**Une image contenant texte, capture d’écran

Description générée automatiquement**

Figure 1 : Débruitage avec lambda = 0.125

Avec un lambda de 0.125 on élimine bien le bruit mais on perd également beaucoup d’informations sur les hautes-fréquences, surtout de basse intensité (le building à droite). En diminuant la valeur de lambda on devrait pourvoir trouver un compromis.

Une image contenant texte, capture d’écran, trépied, noir et blanc

Description générée automatiquement

Figure 2 : Débruitage avec lambda = 0.04

Avec un lambda de 0.04 on lisse suffisamment le bruit dans les basses fréquences et on maintien de manière de très correct les contours même de faible intensité (le building à droite).

Une image contenant texte, croquis, noir et blanc, noir

Description générée automatiquement

Figure 3 : Comparatif méthode Perona-Malik

En comparant les deux méthodes on s’aperçoit d’abord que les contours sont plus nets avec Perona-Malik. Quelques bruits résiduels sont toujours présents et le lissage abime significativement les textures. De plus Perona-Malik est considérablement plus long à s’exécuter (15-40s contre moins de 2s pour la variation totale).

Une image contenant texte, affiche, trépied

Description générée automatiquement

Figure 4 : Différence des images originale et non bruitée

On remarque que la perte d’informations sur les contours est significative.

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

Figure 5 : Cas couleur (RGB)

Dans le cas d’une image couleur rgb les contours sont encore mieux conservés, ce qui apparait logique puisque nous disposons de 2 canaux supplémentaires d’information permettant d’identifier les contours.

**Exercice 2 : modèle d'inpainting par variation totale**

Il est cette fois-ci question d’inpainting, le but de cet exercice est de supprimer un certain domaine « parasite » de l’image tout en restaurant les informations perdues par l’ajout initial de ce domaine.

Pour cela il est nécessaire de construire une matrice diagonale dont les coefficients seront la représentation du domaine à effacer. Cette matrice sera utilisée à la place de la matrice identité de l’exercice 1 et permettra ainsi d’appliquer le traitement uniquement sur la zone du domaine.

L’identification de ce domaine se fait dans un premier temps à l’aide d’une image binaire de segmentation donnée.

Une image contenant plante, fleur, orchidée, texte

Description générée automatiquement

Figure 6 : Restauration avec image binaire

On voit rapidement que le traitement est très efficace, il restaure avec une grande qualité les informations perdues de l’image, seules certaines coupures de contours ne sont pas restaurées et l’on peut apercevoir les vestiges du domaine en zoomant.

Une image contenant texte, capture d’écran, Magenta, Caractère coloré

Description générée automatiquement

Figure 7 : Démonstration des défauts

Il pourrait toutefois être imaginé une solution se basant sur l’analyse morphologique de l’image pour tenter de reconstruire les contours coupés comme dans le cas présent.

Il est également possible de segmenter le domaine nous-mêmes grâce à ses propriétés colorimétriques, ici le domaine est d’un jaune prononcé, on peut donc tenter de le segmenter en testant la valeur de chacun des canaux comme suit :



On test ici si les composantes R et G ont une valeur supérieure à 0.8 sur 1 et si le B est inférieur à 0.15 sur 1. Il est également à précisé qu’il aurait été possible d’utiliser un outil color picker pour déterminer la valeur exacte en rgb du jaune, mais nous laissons volontairement ici un spectre de couleur plus large pour observer les effets qu’un domaine non-uniforme pourrait avoir.

Une image contenant texte, amphibien, capture d’écran, grenouille

Description générée automatiquement

Figure 8 : Restauration par segmentation sur texte "facile"

Ici la restauration est plus efficace que sur la fleur précédente, en effet les zones à restaurer sont moins vastes et plus d’information est éparpillé au sein du domaine. La conséquence est qu’il n’est pas possible de voir de résidus du domaine comme sur la fleur, même en zoomant.

Une image contenant texte, capture d’écran

Description générée automatiquement

Figure 9 : Agrandissement

On peut toutefois se demander que donnerait ce processus sur un texte qui se fond plus efficacement dans l’image.

Une image contenant texte, capture d’écran, amphibien, grenouille

Description générée automatiquement

Figure 10 : Restauration par segmentation sur texte moins "facile"

Ici le jaune du texte précédent a été remplacé par un rouge assez proche de celui présente sur le dos de la grenouille, segmenté comme suit : 

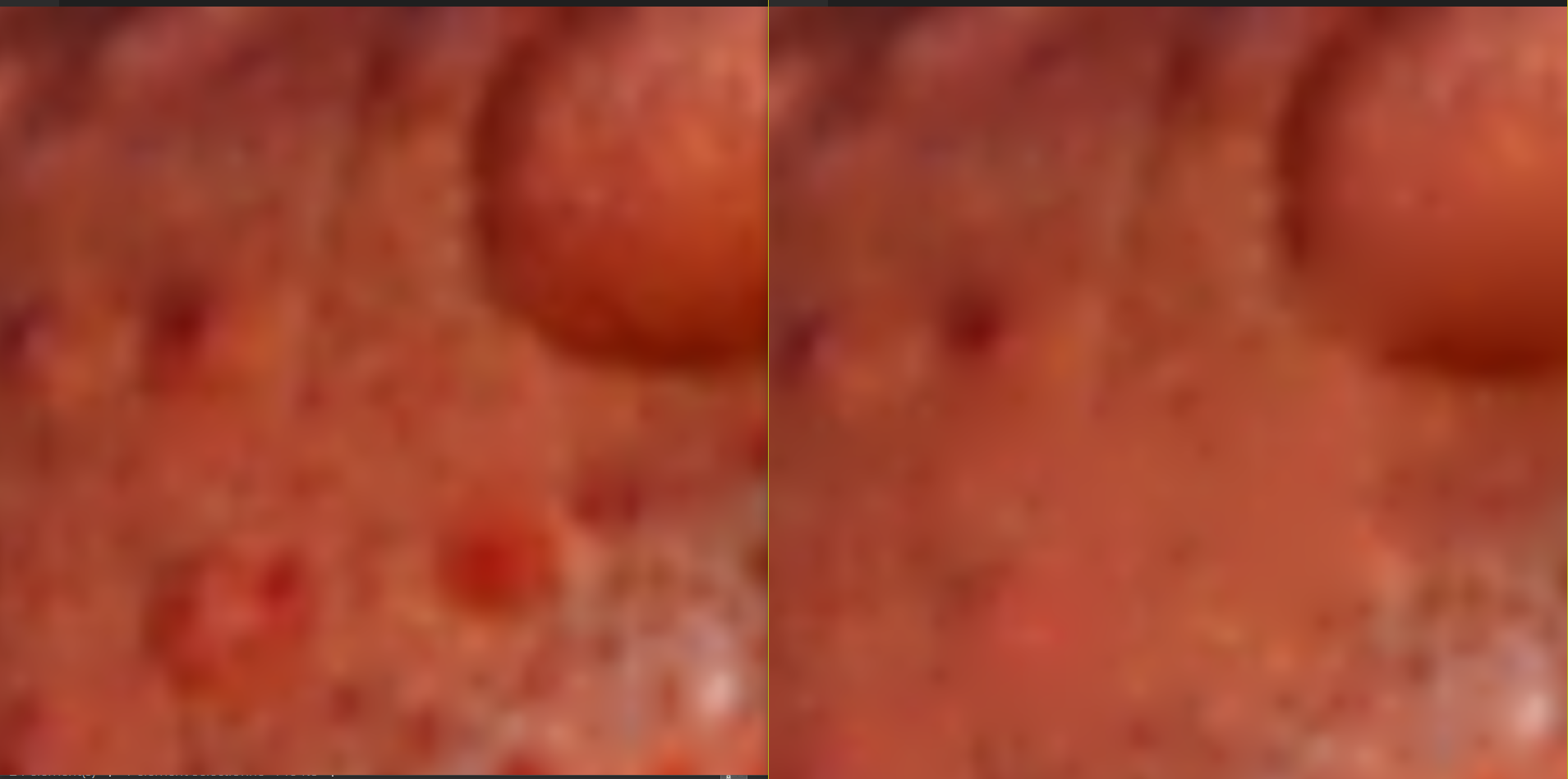


Figure 11 : Agrandissement zone restaurée avec texte jaune (à gauche) et rouge (à droite)

On voit que sur l’image de droite on perd plus de texture de de contraste sur celle de gauche, ce qui s’explique par le fait que des pixels n’appartenant pas au domaine soient traité comme si c’était le cas dû à l’erreur de segmentation. Nous rappelons que cette approximation lors de la segmentation est utilisée pour identifier les conséquences qu’aurait un domaine non-uniforme sur l’image après restauration.