**Traitement Image**

*Compte Rendu TP1*

PATRY Alan, ED-DAHMANY Ayoub

**Exercice 1 : matrices de différences finies**

Cet exercice implémente une fonction d’approximation des opérateurs de dérivation en x et y et du Laplacien d’une matrice de taille quelconque.

Le scaling de cette méthode est assez mauvais puisque pour une image de N pixels les opérateurs matriciels sont de taille N\*N, il est donc nécessaire de réduire au maximum le temps de calcul de ces opérations, sans quoi le traitement d’image de grande taille sera trop long.

L’implémentation des matrices est donc faite à l’aide de matrices creuses, diminuant ainsi l’espace mémoire utilisé et les performances de calcul.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Figure 1 : Benchmark de l'algo

**Exercice 2 : gradient et laplacien d’une image**

Cet exercice implémente une fonction de calcul du gradient et du Laplacien d’une image à l’aide des opérateurs construis ci-dessus. Les résultats sont ici comparés à ceux fournis par les fonctions matlab remplissant ce rôle. Nos résultats leurs sont très similaires (à l’exception du Laplacien les images sont 100% identiques)

Une image contenant texte, carte, capture d’écran, diagramme

Description générée automatiquement

Figure 2 : Similarité des opérateurs

**Exercice 3 : diffusion linéaire**

Il est maintenant temps d’implémenter deux méthodes de flous Gaussien à l’aide du Laplacien calculé précédemment. Ces méthodes sont dites « implicite » et « explicite ».

Les exemples ci-dessous se font avec 100 itérations de l’algorithme.

Ces deux fonctions sont paramétrables par un delta\_t qui permet de régler l’intensité de la diffusion et donc de converger plus vite vers une image unie.

La méthode explicite ne laisse cependant pas un degré de liberté trop grand puisqu’avec des valeurs trop importantes de delta\_t le système diverge.

Une image contenant texte, capture d’écran, film radiographique

Description générée automatiquement

Figure 3 : Méthode explicite avec delta\_t = 0.1

Une image contenant texte, croquis, diagramme

Description générée automatiquement

Figure 4 : Divergence de la méthode explicite avec delta\_t = 1

La méthode implicite, quant à elle, ne se limite pas à des valeurs faibles de delta\_t, qui peut ici augmenter autant que souhaité sans risque de divergence.

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

Figure 5 : Pas de divergence de la méthode implicite avec delta\_t = 1

L’inconvénient majeur de ces deux méthodes est que la diffusion s’effectue de manière uniforme, elles ont donc tendance à générer une perte d’informations significative au niveau des contours de l’image. On peut facilement observer cette perte grâce à la différence des images originale et bruitée, dans laquelle on voit le bruit éliminé mais également les contours de l’image qu’on peut y distinguer.

**Exercice 4 : diffusion non linéaire**

Pour remédier à cela il est possible d’utiliser une diffusion non linéaire (Perona-Malik), proportionnelle au gradient local de l’image. Ainsi dans les directions de fort gradient la diffusion sera très faible. Cette méthode permet la conservation d’un maximum de la structure de l’image grâce au maintien des contours.

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

Description générée automatiquement

Figure 6 : Comparaison d'une zone réduite avec lamda = 30, delta\_t = 0.05, 100 itérations

On observe bien sur la carte de diffusivité que la diffusion s’effectue majoritairement dans les zones de faible gradient en évitant de détériorer les contours. L’inconvénient majeur étant que le bruit ajouté à l’image créer parfois de fortes différences entre deux pixels adjacents, l’algorithme n’est pas donc en mesure de déterminer s’il s’agit du bruit ou d’un contour. Cela donne sur la 3ème image une texture lisse mais parsemée de points non lissés.

Il est également possible de régler ce souci en jouant avec les paramètres lambda et delta\_t et le nombre d’itération, la difficulté réside dans l’équilibre entre ces paramètres.

Une image contenant texte, film radiographique

Description générée automatiquement

Figure 7 :: Comparaison avec lamda = 50, delta\_t = 0.07, 50 itérations

Une image contenant texte, instrument de mesure rigide, capture d’écran, noir et blanc

Description générée automatiquement

Figure 8 : Comparaison d'une zone réduite avec lamda = 50, delta\_t = 0.07, 50 itérations

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Figure 9 : Différence entre les images originelle et bruitée

On peut aisément voir que contrairement à la méthode linéaire, la perte d’information dans les contours est très limitée.

En augmentant le lambda on parvient désormais à éliminer la grande majorité du bruit présent dans les zones à basses fréquences tout en conservant l’essentiel des contours. En diminuant le nombre d’itérations on évite également la perte de trop d’informations dans ces mêmes zones, on conserve donc une partie de la texture.

Une amélioration potentielle du système serait de faire un edge sharpening des contours les plus marqués avant le débruitage pour les conserver d’autant plus. Néanmoins ce processus doit être précisément paramétré pour ne pas renforcer le bruit dans l’image.