

TI : TP11 : Acquisition d'image couleur et dé-matriçage

binôme : Benjamin Ruytoor et Aurore Allart

date : 5 avril 2013

Introduction

Dans ce TP, nous allons implémenter 2 méthodes permettant de restituer une image couleur à partir d'une image en niveau de gris. Ces méthodes sont : l'interpolation bilinéaire et le dé-matriçage basé sur le gradient.

1. Image CFA

◦ Interprétation d'une image CFA

Le filtre CFA contient 2 fois plus de composante verte que de rouge et bleue. Cela est dû au fait, que la composante verte représente la luminance de l'image : c'est à dire l'intensité lumineuse, et que la sensibilité spectrale de l'œil est à son maximum, autour de la longueur d'onde proche du vert.

Dans l'image CFA donnée, les 3*3 premiers pixels du filtre, c'est à dire du coin en haut de l'image, nous donnent :

B	V	B
V	R	V
B	V	B

B =bleu

V = vert

R = rouge

◦ Simulation d'une image CFA à partir d'une image couleur

Un plugin d'imageJ nous permet de calculer à partir de l'image couleur, l'image CFA suivant la disposition des couleurs des 3 premiers pixels : dans notre exemple G-R-G pour (vert, rouge, vert). Nous obtenons cette image :



Figure 1 : image CFA du phare.

Par la suite, nous allons programmer le dé-matçage, où l'on pourra comparer l'image de référence avec l'image obtenue après dé-matçage.

◦ **Exercice optionnel**

Les modifications apportaient à la méthode cfa, permettront, en fonction des dispositions possibles de CFA, de décaler les indices des pixels des composantes couleurs. Par exemple pour la disposition GBG, on a ce code :

```
if(row_order==0){  
    //indice du premier pixel vert  
    startGX1=0;  
    startGY1=0;  
    //indice du deuxième pixel vert  
    startGX2=1;  
    startGY2=1;  
    //indice du pixel rouge  
    startRX=0;  
    startRY=1;  
    //indice du pixel bleu  
    startBX=1;  
    startBY=0;  
}
```

2. Dé-matçage par interpolation bilinéaire

Le dé-matçage par interpolation bilinéaire permet de calculer les valeurs des composantes manquantes au pixel du milieu du voisinage. Par exemple pour un voisinage 3*3 dans une configuration RGR, si on cherche la valeur de rouge du pixel central, on divise par 2 la somme des composantes rouges de gauche et droite.

Les masques de convolution permettent justement de calculer, à partir d'images contenant une composante chacune (une image pour la composante rouge, une image pour la composante verte et une image pour la composante bleue), les valeurs des composantes manquantes pour chaque pixel (sauf le bord de l'image à cause du manque d'informations).

Le masque HG permet de retrouver la couleur verte de chaque pixel. Il peut avoir 2 cas dans une image, voici les 2 cas :

0	G	0	G	0	G
G	0	G	0	G	0
0	G	0	G	0	G

Le filtre HG est le suivant :

0	0.25	0
0.25	1	0.25
0	0.25	0

On ne considère que le pixel du centre, avec le filtre par convolution, on prend un voisinage 3*3. En vert, le premier cas où il manque la valeur du pixel du milieu, avec le filtre on calcule donc la somme des G autour divisé par 4. En orange, le deuxième cas où la valeur du milieu est présente, dans ce cas le filtre recopie la valeur dans l'image qui reçoit les valeurs. Ce qui revient au calcul de l'interpolation bilinéaire pour la composante verte. On peut donc constater que toutes les pixels, sauf le bord, contiennent leur valeur de couleur verte.

Pour les masques HR et HB, il y a 4 cas possibles. Dans l'explication suivante, nous allons considérer que le masque HB.

0	B	0	B	0	B	0	0	0	0	0	0
0	0	0	0	0	0	B	0	B	0	B	0
0	B	0	B	0	B	0	0	0	0	0	0

Le masque pour la composante bleue est le suivante :

0.25	0.5	0.25
0.5	1	0.5
0.25	0.5	0.25

Dans notre première cas en bleu, on calcule la somme des 2 B ($0.5 + 0.5$) verticaux, ce qui nous donne 1. Dans le 2^e cas en violet, on calcule la somme des 4 B en diagonale. Dans le 3^e cas en rouge, la somme des 2B horizontaux. Et dans le dernier cas, le B du centre. On constate que ce masque revient aussi au calcul de l'interpolation bilinéaire.

On peut donc utiliser ces masques à la place de l'interpolation bilinéaire.

○ Programmation du dé-matriçage

A partir des images phi k, il est possible de récupérer une image couleur. Malheureusement, cette image comporte des défauts (on le voit très bien dans l'image ci-après au niveau des barrières). Ces défauts sont dus aux calculs des composantes couleurs par l'interpolation bilinéaire. Il est nécessaire de faire un post-traitement (correction des couleurs, augmentation de la netteté, réduction du bruit) entre l'image CFA et l'image couleur.



Figure 2 : image couleur obtenue par l'image CFA.

3. Dé-matriage basé sur l'estimation locale d'un gradient

La méthode de l'estimation locale d'un gradient permet de mieux estimer la composante verte des pixels. Cela a pour conséquence de réduire le bruit : par exemple au niveau des bruits de la barrière de l'image de phare.

○ Implémentation de la méthode

Dans un premier temps, on calcule l'image CFA de l'image couleur de départ grâce à la fonction *cfa* écrite précédemment.

A partir de cette image CFA, on calcule la composante verte par l'appel de la méthode *est_G_hamilton*. Dans cette méthode, on parcourt les pixels avec une composante rouge ou bleue déjà existante de l'image CFA. Pour un pixel on calcule le gradient en x et y. Ensuite, en fonction de la valeur de ces gradients, on calcule l'interpolation du niveau de vert.

Ensuite, on convole l'image contenant que les pixels de niveau de bleu (ou rouge) avec le filtre de l'exercice précédent.

Donc on obtient une pile de 3 images de chaque composante couleur : rouge, verte et bleue. Enfin, on utilise la macro d'imageJ *stack To RGB*. On obtient cette image :



Figure 3 : image couleur obtenue par la méthode d'Hamilton.

On peut voir qu'au niveau de la barrière, les pixels jaunes et bleus sont légèrement atténués.

- **Evaluation de la qualité de l'image estimée**

Voici ce que donne la différence entre les 2 images couleurs des 2 méthodes précédentes :



Figure de gauche : les différences entre les 2 images.

Figure de droite : traitement de l'image de gauche par un ajustement de la brillance et du contraste.

Nous avons fait un petit traitement pour visualiser mieux les différences entre les 2 images. C'est juste pour mieux voir.

On constate donc une nette différence au niveau des contours de chaque objet.

Conclusion

La méthode bilinéaire à l'avantage de demander moins de calcul, mais fourni une image avec beaucoup de bruit, quand il y a une succession de contour (voir la barrière).

Le calcul d'Hamilton est plus gourmand en calcul mais fourni des contours plus détaillés et avec moins de bruit.