

Reconstruction d'images par FFT et SVD

LESIEUR Christophe; MONGODIN Maxime

9 juin 2014

Table des matières

Introduction	2
1 Présentation des méthodes utilisées	3
1.1 Transformée rapide de Fourier	3
1.2 Décomposition en valeurs singulières	4
2 Analyse des résultats	6
2.1 Par transformée rapide de Fourier	6
2.2 Par décomposition en valeurs singulières	7
Conclusion	10

Introduction

Dans le cadre de ce TP noté d'Analyse numérique, ils nous a été proposé de procéder à la reconstruction d'images de deux méthodes différentes : la transformée rapide de Fourier (FFT) et la décomposition en valeurs singulières (SVD). Ces deux méthodes seront implémentées sous Scilab et seront testées sur l'image `im34.txt`.

Nous allons tout d'abord vous présenter ces deux méthodes, leur fonctionnement et leurs particularités, puis nous vous proposerons notre code, ainsi que les résultats qu'il produit, et enfin nous analyserons ces résultats en termes de qualité visuelle, d'erreur numérique et enfin d'économie de stockage.

Chapitre 1

Présentation des méthodes utilisées

1.1 Transformée rapide de Fourier

Cette méthode consiste à travailler sur la transformée rapide de Fourier des valeurs de la matrice dont est composée l'image. Il nous faut tout d'abord lire l'image, ici stockée dans un fichier texte .txt simple. Une fois la matrice M lue, il est nécessaire de la mettre en forme : en effet, M est bidimensionnelle, de taille $m \times n$ (où m et n sont les dimensions de l'image) et, pour pouvoir lui appliquer une transformée rapide de Fourier, nous avons besoin d'un tableau monodimensionnel T . Nous transformons donc notre matrice en tableau T de taille $N = m \times n$, auquel nous appliquons une transformée rapide de Fourier.

De ce point, nous pouvons obtenir le spectre de notre image en calculant les modules des valeurs contenues dans T , et l'afficher à l'écran. Grâce au spectre, nous pouvons décider du pourcentage de fréquences à garder, et donc du pourcentage de fréquences à supprimer. A noter que le tableau des modules étant symétrique, seulement sa première moitié sera affichée à l'écran pour plus de lisibilité. Une fois ce pourcentage décidé, notre seuil est calculé, et les valeurs correspondant à ces fréquences sont supprimées de T . Enfin, pour relire notre image, une FFT inverse est appliquée à T , puis ses valeurs sont mises au modules pour s'assurer qu'elles soient réelles. T est ensuite retransformé en matrice $m \times n$ et est enfin affiché sous forme d'image.

La transcription de cet algorithme sous Scilab donne le code suivant :

FIGURE 1.1 – Code Scilab de la reconstruction d’images par FFT

```
//Lecture et affichage de l'image source
N = 500*640;
a = read("im34.txt",-1,640);
ap = uint8(255*a);
figure();
ShowImage(ap,'Image originale');

//Traitement de l'image

a=matrix(a,1,N);
A = fft(a,-1);
Amod = abs(A);

figure();
plot2d([2:N/2],Amod(1,2*N/2))

seuil = 0.3.*max(Amod)

for (i=seuil:(N-seuil))
    A(i)=0;
end

//FFT inverse et affichage de l'image reconstruite

A1 = fft(A,1);
Amod2 = abs(A1);
a2 = matrix(Amod2,500,640);
ap1 = uint8(255*a2);
figure();
ShowImage(ap1,'Image-Reconstruite');
```

1.2 Décomposition en valeurs singulières

Après avoir lu la matrice M correspondant à notre image, il est possible de la décomposer en trois matrices U , S et V . S est une matrice de même dimension que M , avec les valeurs singulières de M sur sa diagonale, classées par ordre décroissant. U et V sont deux matrices carrées unitaires représentant les vecteurs singuliers de M .

On peut alors récupérer la diagonale de S dans un tableau monodimensionnel, puis récupérer la taille de ce tableau pour connaître le nombre de valeurs singulières pour l’image étudiée. Ainsi, on peut alors choisir le pourcentage de valeurs singulières à garder. Une fois les valeurs singulières supprimées dans la matrice S , il suffit de calculer le produit matriciel $U \times S \times {}^tV$ pour récupérer la matrice M' correspondant à l’image reconstruite. On peut alors l’afficher.

La transcription de cet algorithme sous Scilab donne le code suivant :

FIGURE 1.2 – Code Scilab de la reconstruction d’images par SVD

```
N=500*640;
a= read('im34.txt',-1,640);
[U,S,V]=svd(a);
ap= uint8(255*a);
figure();
ShowImage(ap,'Image-originale');

vals=diag(S);
seuil=length(vals)*0.3;

for(i=seuil:length(vals))
    S(i,i)=0;
end

a= U*S*V';
ap=uint8(255*a);

figure();
ShowImage(ap,'Image-Reconstruite');
```

Chapitre 2

Analyse des résultats

2.1 Par transformée rapide de Fourier

La transformée rapide de Fourier a pour avantage de nous donner accès au spectre de l'image, outil très important, notamment dans le traitement d'images (d'un point de vue graphique donc). Les résultats donnés sont plutôt bons, sur bien des aspects : pour obtenir notre image optimale, nous avons choisi de garder seulement 30% du spectre, ce qui signifie qu'après reconstruction, 70% des valeurs ont été supprimées, soit une économie de stockage équivalente.

D'autre part, l'erreur numérique à ce pourcentage est très faible : peu de tâches apparaissent sur l'image, seulement un léger flou. Une analyse numérique supplémentaire serait donc envisageable, voire même certains traitements d'images. En effet, certains algorithmes de détection de bords ou de points d'intérêts ne nécessitent pas d'image nette. Au contraire, il est par exemple recommandé d'effectuer une opération de lissage de l'image (c'est-à-dire un floutage de l'image) avant d'appliquer le détecteur de coins de Harris. En effet, une telle opération de lissage a pour effet de supprimer tous les points d'intérêts mineurs qui pourraient obscurcir les résultats du détecteur, pour ne laisser que les véritables points d'intérêt de l'image. Une image préalablement reconstruite par transformée rapide de Fourier pourrait alors être nettement avantageuse et faire gagner énormément en temps ou en ressources.

En revanche, descendre sous les 30% du spectre dégrade énormément la qualité de l'image et risque de nuire à tout algorithme appliqué ensuite, comme le montre la figure 2.2 obtenue après n'avoir gardé que 10% du spectre.

FIGURE 2.1 – Image reconstruite avec 30% du spectre



FIGURE 2.2 – Image reconstruite avec 10% du spectre



2.2 Par décomposition en valeurs singulières

La décomposition en valeurs singulières est un traitement purement numérique et mathématique, qui se base sur l'importance des valeurs singulières dans l'image. Chacune d'entre elle est porteuse d'information, et plus celle-ci est grande, plus l'information portée par la valeur est importante.

Ainsi, supprimer des valeurs singulières, en commençant par les plus faibles, permet de retirer une quantité croissante mais mesurée d'information. La matrice S nous offrant déjà les valeurs singulières classées dans l'ordre décroissant, nous gagnons déjà en ressources et en temps. D'autre part, la quantité d'information retirée avec les valeurs singulières les plus faibles est minime, si bien que l'œil humain ne la perçoit qu'à partir d'un certain point.

Cependant, l'erreur numérique est indéniable. En effet, elle est apparente sur toutes la valeurs saturées de l'image (blancs ou noir parfaits ou quasi-parfaits),

faisant apparaître des points blancs et noirs plus ou moins nombreux selon le pourcentage de valeurs singulières gardées. De plus, si le nombre de valeurs singulières supprimées est trop important, on observe alors un phénomène de pixélisation de l'image, comme on peut le voir sur la figure 2.3. Il est alors peu envisageable d'effectuer un traitement d'image quelconque après une telle reconstruction, car l'erreur numérique engendrée fausserait vraisemblablement les résultats.

D'autre part, calculer l'économie de stockage effectuée peut se révéler compliqué, car l'image reconstruite est issue du produit matriciel $M' = U \times S \times {}^tV$. Certes, des valeurs singulières ont été supprimées de la matrice S , mais cela ne suffit pas à faire de M' une matrice creuse. Au contraire, cela ne fait apparaître que très peu de zéros, sauf dans le cas où le nombre de valeurs singulières supprimées est grand (moins de 20%), auquel cas l'image est sensiblement altérée. Cette méthode est donc peu avantageuse au niveau de l'économie de stockage réalisée.

FIGURE 2.3 – Image reconstruite avec 10% des valeurs singulières

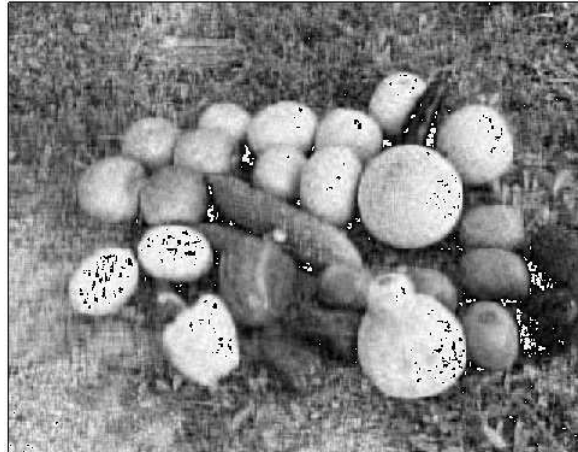


FIGURE 2.4 – Image reconstruite avec 30% des valeurs singulières



FIGURE 2.5 – Image reconstruite avec 50% des valeurs singulières



Conclusion

Après étude des résultats, il semble clair que la reconstruction d'image par transformée rapide de Fourier semble plus avantageuse que celle par décomposition en valeurs singulières, et ce sur tous les points : une erreur numérique limitée, des prédispositions pour de futurs traitements d'image et enfin une économie de stockage non négligeable. Les résultats obtenus par décomposition en valeurs singulières étant, eux, moins probants, bien que tout à fait satisfaisants d'un point de vue strictement numérique.

Cela explique en partie pourquoi une variante de la méthode de transformée rapide de Fourier est utilisée, en parallèle avec d'autres algorithmes, dans le format d'image JPEG (joint Photographic Experts Group), qui est l'un des formats d'images les plus utilisés aujourd'hui, car il est très avantageux au niveau compression des données, aspect primordial de la transmission d'information à l'heure d'Internet.