



Alves Vincent  
Jarret Guillaume  
Analyse Numérique - Groupe B10

## **Reconstruction d'Images par FFT et SVD**

8/06/14

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation mathématique du problème</b>	<b>3</b>
2.1	Méthode FFT . . . . .	3
2.2	Méthode SVD . . . . .	3
<b>3</b>	<b>Réalisation informatique</b>	<b>4</b>
3.1	Fonction FFT . . . . .	4
3.2	Fonction SVD . . . . .	6
<b>4</b>	<b>Résultats obtenus</b>	<b>7</b>
<b>5</b>	<b>Conclusion générale</b>	<b>9</b>
<b>6</b>	<b>Références</b>	<b>9</b>
<b>7</b>	<b>Annexe - Guide d'utilisation du programme</b>	<b>9</b>

## 1 Introduction

L'objectif de ce TP d'Analyse Numérique est d'utiliser deux méthodes pour la reconstruction approchée des images. La transformée rapide de Fourier (FFT) et la décomposition en valeurs singulières (SVD).

Nous avons implémenté ces deux méthodes sous Scilab et les avons testées sur l'image fournie *im34.txt*. Nous allons d'abord vous présenter ces méthodes, puis notre implémentation et les résultats que nous en avons tiré.



*Image originale.*

## 2 Présentation mathématique du problème

### 2.1 Méthode FFT

Cette méthode consiste à compresser en enlevant certaines valeurs de la transformée de Fourier de l'image.

Soit notre image transformée en une matrice bidimensionnelle  $M$ . Pour appliquer FFT, les étapes à suivre sont les suivantes.

- Convertir  $M$  en une matrice monodimensionnelle  $T$ .
- Appliquer la transformée de Fourier à ce tableau.
- Retirer la première valeur, usuellement trop élevée, et la seconde moitié du tableau obtenu, car elle peut être réobtenue par symétrie.
- Appliquer notre compression en retirant un pourcentage des valeurs restantes, par exemple 70% : On retirera les valeurs après 3/10 de la longueur de la matrice.
- Appliquer une symétrie à la matrice restante pour obtenir une matrice monodimensionnelle de taille égale à  $T$ .
- Appliquer la transformée de Fourier inverse.
- Reconvertir la matrice en une matrice bidimensionnelle.

### 2.2 Méthode SVD

Cette méthode consiste à décomposer notre image en trois matrices  $U, S, V$  de mêmes dimensions, avec les valeurs singulières de  $M$  sur sa diagonale, classées par ordre décroissant.  $U$  et  $V$  sont deux matrices représentant les vecteurs singuliers de  $M$ .

- Récupérer la diagonale de  $S$  dans un tableau monodimensionnel.
- La taille de ce tableau représente le nombre de valeurs singulières pour l'image étudiée
- Conserver un pourcentage de valeurs singulières
- Calculer  $U * S * V'$  pour obtenir l'image reconstruite,  $V'$  étant les valeurs singulières préservées.

### 3 Réalisation informatique

Nous avons utilisé le programme Scilab pour réaliser les codes suivants.

#### 3.1 Fonction FFT

```
//Passage en Matrice 1D. -1 = length(original)
go = matrix(img,1,-1);

//Application de Fourier
A=fft(go);

figure(4, "Figure_name", "FFT : Spectre Original");
clf(4);
plot(abs(A));

//On prend de 2 à 1/2 de la matrice(160000)
for (i=length(A)/2:length(A))
    A(i)=0;
end
A(1)=0;

figure(2, "Figure_name", "FFT : Spectre Tronqué");
clf(2);
plot(abs(A));

//-----Reconstitution de l'image via FFT inverse-----

//On prend un pourcentage de fréquences (ici, 30%) pour la compression.
for (i=length(A)*(30/100):length(A))
    A(i)=0;
end

//A=A(2:length(A)*(30/100));

//Reconstruire un spectre complet par symétrie:
A_recons = zeros(1, 500*640); //Longeur complete
A_recons(1:length(A)) = A; //On remplit le début avec ce qui reste de A

A_recons((500*640-length(A)+2):(500*640)) = conj(A($:-1:2));
A_recons(1:length(A)/2) = A(1:length(A)/2);
//symétrie.
```

```
figure(5, "Figure_name", "FFT : Spectre complet reconstruit");
clf(5);
plot(abs(A_recons));

B=ifft(A); //On applique la transformée inverse

B=abs(B);

temp = matrix(B,500,640); //500x640 Taille image originale
Result = uint8(255*temp);
figure(3, "Figure_name", "Image compressée");
clf(3);
ShowImage(Result,'Image compressée');
```

### 3.2 Fonction SVD

```
//Obtention de l'image originale
img = read('im34.txt',-1,640);
ap = uint8(255*img);
figure(1, "Figure_name", "Image originale");
clf(1);
ShowImage(ap,'Image originale');

//-----Application de la SVD-----

[U, S, V] = svd(img);
S = diag(S);
// On récupère les min(m, n) valeurs singulières (extraction valeurs diagonales de S)

    // Début : partie graphique
    figure(2, "Figure_name", "Valeurs Singulières");
    clf(2);
    x = 1:length(S);
    plot(x,S);

// Réduction à 30\% des valeurs
nb_val = length(S) * (5/100);
U = U(:, 1:nb_val);
S = S(1:nb_val);
V = V(:, 1:nb_val);

svdmat = [S,U',V'];

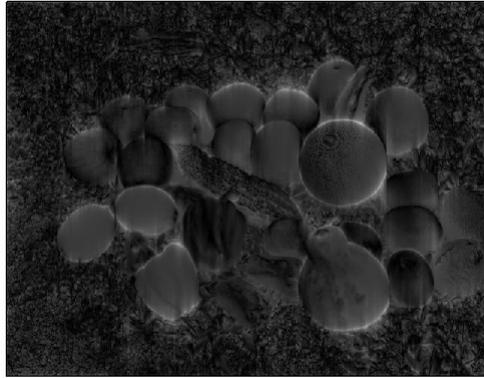
//-----SVD Inverse-----

    // Extraction des matrices U, S, V
    S = diag(svdmat(:, 1));
    U = svdmat(:, 2:501)'; //Commencer à 2 <=> Ajouter 1 à la largeur
    V = svdmat(:, 502:size(svdmat,2));
    tab = U * S * V;

    Resultat = uint8(255*tab);
    figure(3, "Figure_name", "Image reconstruite à partir de SVD");
    clf(3);
    ShowImage(Resultat,'Image reconstruite à partir de SVD');
```

## 4 Résultats obtenus

Avec un taux de compression de 70%(30% des valeurs conservées) :

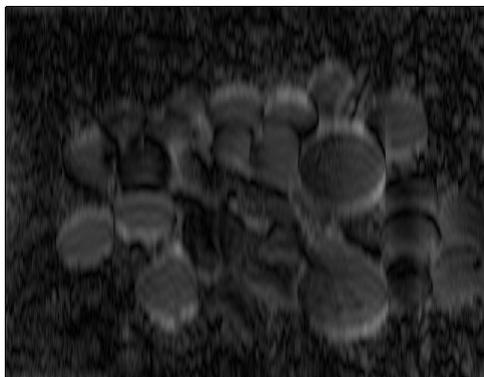


*Transformation FFT appliquée.*

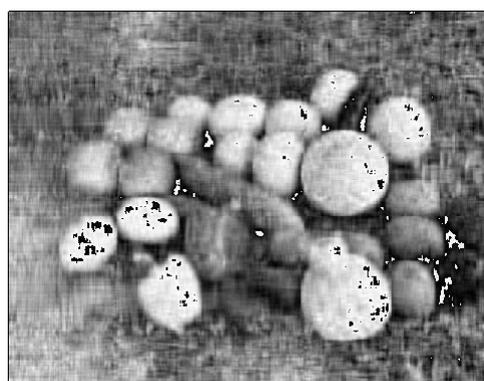


*Transformation SVD appliquée.*

Avec un taux de compression de 95%(5% des valeurs conservées) :



*Transformation FFT appliquée.*



*Transformation SVD appliquée.*

Notre implémentation de FFT est fautive, au vu de l'image obtenue. Cependant, SVD semble bien fonctionner.

## 5 Conclusion générale

Malgré les erreurs de notre implémentation, nous pouvons voir que SVD est une méthode de compression correcte jusqu'à un certain pourcentage. Cependant, les valeurs singulières supprimées apportent une erreur numérique visible par l'apparition de points blancs ou noirs sur toutes les valeurs saturées de l'images (blancs/noirs quasi-parfaits).

Passé un certain pourcentage (20% de valeurs gardées), on observe un phénomène de pixélisation de l'image. On ne peut donc pas pousser l'algorithme trop loin si l'on désire conserver la qualité générale de l'image.

## 6 Références

Support de cours Scilab, Cycle ingénieur, EISTI Première année.

## 7 Annexe - Guide d'utilisation du programme

- Dans Scilab, sélectionner **Fichier** → **Changer le répertoire courant**
- Sélectionner le répertoire dans lequel ont été extraits les fichiers `Fourier.sce` et `ValSingu.sce`, ainsi que le fichier `im34.txt` (que nous ne pouvons fournir à cause des restrictions de Sifoci...)
- Entrer dans Scilab `exec('Fourier.sce')`; ou `exec('ValSingu.sce')`; , selon la méthode désirée.
- Scilab récupèrera l'image, exécutera l'algorithme et affichera le résultat.