Une image contenant plein air, ciel, bâtiment, architecture

Description générée automatiquement

COMPRESSION D’IMAGES PAR TRANSFORME DE FOURRIER

A destination de Didier Auroux et Cédric Boulbe, enseignants à Polytech Nice

TALBO Ahmed

KECHID Adel

PARSEKIAN Adrien

**SOMMAIRE**

INTRODUCTION 3

[ORGANISATION 4](#_Toc153830202)

[Planning de la semaine 4](#_Toc153830203)

[Répartition des tâches 5](#_Toc153830204)

[Mise en commun 6](#_Toc153830205)

[COMPREHENSION ET ORGANISATION DE L’ALGORITHME 7](#_Toc153830206)

Théorie du projet………………………………………………………………………………………….7

[Objectif de l’algoritme 8](#_Toc153830207)

[Comment réaliser le bon algorithme 8](#_Toc153830208)

[Optimisation, résultat et interprétation 9](#_Toc153830209)

[CONCLUSION 13](#_Toc153830210)

## 

**INTRODUCTION**

Le projet de MAM 3 en analyse numérique était de compresser et décompresser une image à l’aide de la transformée de Fourier. Nous devions écrire notre algorithme en Python. Ce projet est très intéressant de nos jours. En effet, tous nos appareils électroniques se servent de ce principe de compression et de décompression afin de ne pas occuper trop d’espace de mémoire, d’améliorer la lisibilité des images et d’en faciliter l’interprétation. Cet ensemble d’opérations se nomme le traitement d’image.

Pour mener à bien notre projet, nous allons utiliser la symétrie. En effet, par symétrisation et périodisation d’une image, on peut la rendre infinie dans les deux directions, périodique et paire. Toute fonction périodique peut se décomposer dans une base de fonctions cosinus et sinus, et si elle est paire, seulement avec des fonctions cosinus.

Pour notre projet et afin de simplifier les calculs, il était demandé de travailler sur des blocs de matrices de taille 8x8 provenant de l’image. Pour effectuer notre traitement d’image, nous devions appliquer à chaque bloc de la matrice la Transformée en Cosinus Discrète afin de passer dans le domaine des fréquences.

Ensuite, deux méthodes étaient possibles pour la compression. La première était de diviser chaque bloc par une matrice donnée (Q), afin de garder les modes les plus importants et de couper les hautes fréquences considérées comme du bruit. Les hautes fréquences sont présentes en bas à droite de chaque bloc. Chaque bloc deviendra donc une matrice très creuse. La deuxième méthode était d’utiliser une fréquence de coupure. En effet, au lieu de diviser les blocs de matrices par Q, le but de cette méthode est de tronquer l’information au-delà d’une certaine fréquence. Ainsi, elle mettra à 0 tous les coefficients des matrices de fréquences si i+j ≥ f\_coupure, avec i comme indice de ligne et j comme indice de colonne. Ce qui nous donnera la matrice creuse.

Pour effectuer la décompression, il suffisait de suivre le chemin inverse de la compression. En effet, on multiplie chaque élément de bloc de matrices de fréquences par chaque élément de Q. Ensuite, on applique le changement de base pour sortir du domaine des fréquences. On regroupe ensuite les blocs afin de former la grande matrice et afficher l'image originale qui sera légèrement modifiée en raison de l'erreur que nous expliquerons dans la suite du rapport.

**ORGANISATION**

## **Planning de la semaine**

Lundi :

* Réunion sur le projet ;
* Appropriation du sujet ;
* Répartition des tâches ;
* Compréhension du mécanisme de compression/décompression ;
* Initialisation du code ;

Mardi :

* Approche mathématique pour obtenir les matrices P ;
* Fonctions compression du code avec quantification ;
* Rédaction du rapport ;

Mercredi :

* Fonction décompression ;
* Réalisation de la présentation ;

Jeudi :

* Unification du code ;
* Test du code ;
* Relecture du code et du rapport ;
* Entrainement pour la soutenance ;

## **Répartition des tâches**

Étant habitués à travailler ensemble et connaissant les qualités de chacun, la répartition des tâches s’est faite naturellement :

TALBO Ahmed :

* Initialisation du code
* Partie mathématique pour la transformée discrète

PARSEKIAN Adrien :

* Fonction compression

KECHID Adel :

* Fonction décompression
* Fonction recomposition

La rédaction et la préparation de la soutenance ont été effectuées à parts égales.

## **Mise en commun**

Après avoir réalisé chaque fonction, nous avons mis en commun notre code en utilisant l'interface Spider. Nous avons dû renommer les variables afin que notre code compile. Après plusieurs tests, notre code fonctionnait.

Ensuite, après avoir unifié le code, nous avons partagé les complications rencontrées par chacun et les solutions trouvées pour les résoudre.

**Problèmes** **rencontrés** : Nos problèmes rencontrés ont été la compréhension du changement de base, la compréhension de la compression d'une image, et notre principale difficulté a été la partie du code qui permet de recomposer la grande matrice à partir des blocs de matrices pour effectuer la décompression.

Afin de pallier ces difficultés, nous avons consulté des forums pour y trouver des solutions, et nous nous sommes également rendus à la bibliothèque afin de trouver des documentations qui pourraient nous être utiles.

Pour travailler en groupe, nous nous retrouvions à la bibliothèque afin d'écrire sur un tableau les idées de chacun, apportant ainsi de la clarté aux concepts de chaque membre.

# **COMPREHENSION ET REALISATION DE L’ALGORITHME**

## **Théorie du projet**

Avant de commencer à rédiger l’algorithme, il fallait comprendre la théorie derrière le projet. La théorie du projet est basée sur la transformée de Fourier.

En effet, la transformée de Fourier est utilisée pour passer du domaine temporel ou spatial au domaine des fréquences. Dans le projet, nous allons l’utiliser pour lire les fréquences de l’image afin d’appliquer la compression.

Cependant, des pertes seront présentes et nous obtiendrons une erreur que nous allons expliquer ensuite.

Pour appliquer la transformée de Fourier, le sujet nous indique de passer par un changement de base afin de simplifier les calculs. En effet, en passant dans une base où la matrice de passage est orthogonale, les calculs informatiques seront plus simples.

Calcul de la matrice de passage : On sait que : D = PMP’ avec P’ qui est la transposée de P

Une image contenant texte, Police, ligne, écriture manuscrite

Description générée automatiquement

De plus, on sait que :

Avec C0 = 1/sqrt(2) et Ci = 1 pour tout i = ! 0

On a donc D = PC  C = MP’

Une image contenant Police, texte, blanc, conception

Description générée automatiquement

On a donc

Une image contenant texte, Police, reçu, blanc

Description générée automatiquement

D’où :

## 

En faisant une identification, on a :

## **Objectif de l’algorithme**

Le but de notre algorithme est de compresser et décompresser une image afin de libérer de l’espace mémoire. Nous devions suivre l’algorithme donné dans le sujet. Ainsi, nous devions initialiser le code et créer les fonctions de compression et de décompression.

## **Comment réaliser le bon algorithme**

Pour mener à bien notre algorithme, nous avons raisonné comme avec le logiciel Matlab. En effet, le plus gros problème rencontré était dans la fonction de décompression pour effectuer la recomposition de la matrice. Nous avons choisi d'utiliser la fonction ravel et reshape afin de recomposer la matrice à l’aide d’un vecteur. De plus, durant tout le déroulement, nous avons décidé d'appliquer nos fonctions à des matrices en 3D au lieu de les diviser pour former des matrices 2D, dans le but d'optimiser le code

## **Optimisation, résultat et interprétation**

OPTIMISATION :

On s'est penché sur l'optimisation de la complexité temporelle et spatiale de l'algorithme. En effet, un algorithme doit pouvoir résoudre un problème, mais il doit le faire en le moins de temps possible pour des raisons écologiques et expérimentales. Cela a passé par plusieurs étapes, dont les suivantes :

Complexité temporelle :

* Élimination des redondances : Suppression des calculs redondants en stockant les résultats intermédiaires lorsque cela était possible.
* Suppression de boucles inutiles : Éviter autant que possible les boucles inutiles. Utilisez des opérations vectorielles ou matricielles là où c'est possible pour éviter d'itérer manuellement sur chaque élément

Complexité spatiale :

* Nettoyage des objets inutiles : Libération de la mémoire utilisée par des objets temporaires ou des variables non nécessaires, en particulier dans des boucles importantes.

RESULTAT :

On a également changé de méthode, on traitait d'abord chacune des couleurs séparément, puis on a décidé de les traiter toutes ensemble. La raison principale est qu'on gagnait en complexité dans certaines de nos fonctions non-optimales.

Concrètement, voici une comparaison vidéo :

Remarque : Le logiciel d'enregistrement consomme un peu de temps d'exécution.

Version initiale : environ 5 secondes (hors vidéo : 4.5 secondes)

# Une image contenant texte, capture d’écran, logiciel, ordinateur Description générée automatiquement

Version finale : environ 1 seconde (hors vidéo : 0.7 secondes)

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

On constate qu'on obtient un gain de plus de 4 secondes, ce qui n'est pas du tout négligeable.

Nous obtenons aussi un taux de compression de 97 %, ce qui est plus qu'acceptable.

Pour conclure, durant cette étape, nous avons réussi à alléger au maximum notre code pour associer une bonne exécution du programme à une optimisation efficace.

INTERPRETATION :

Image originale :

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

Description générée automatiquement

Image compressée :

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

Description générée automatiquement

On peut bien voir ci-dessus que notre image est compressée, notre algorithme répond donc bien au problème donné en un temps très faible.

# **CONCLUSION**

Pour conclure, ce projet nous a permis de nous familiariser avec Python et d'appliquer la théorie à un projet concret. En effet, pendant ce projet, nous avons remarqué que la transformée de Fourier est un outil très utilisé en traitement du signal. Cette étape est essentielle pour stocker les images dans tous les appareils électroniques.

Ce projet nous a aussi permis d’aborder toutes les qualités qu’un ingénieur doit avoir. En effet, un projet de groupe permet de se familiariser avec le travail en entreprise, notamment avec le compte rendu oral et écrit du projet.

Si l'on pouvait développer ce projet, on pourrait également envisager d'ajouter le traitement du son qui peut être décomposé en une fréquence fondamentale et plusieurs harmoniques.

Nous avons apprécié ce projet car nous avons développé la capacité de travailler en groupe tout en acquérant de nouvelles compétences dans l'utilisation de Python et en appliquant nos connaissances mathématiques à un projet concret.