

BM3D "Image denoising by sparse 3D
transform-domain collaborative filtering"
sur GPU

Stéphane Cuenat

January 25, 2016

Abstract

Dans ce papier nous proposons une implementation open-source de l'algorithme BM3D porté sur GPU. Nous discutons du choix de l'ensemble des paramètres et de l'influence de ceux-ci . Chaque phase est étudié séparément afin d'en déterminer les valeurs optimales en ce qui concerne la qualité et la rapidité de l'algorithme. Nous démontrons que BM3D est parallélisable, donc exécutable dans un monde GPU.

Table des matières

1	Introduction	3
2	Algorithme BM3D	5
2.1	Concepts généraux	5
2.2	Phase 1 - Estimation basic	5
2.2.1	Block matching	5
2.2.2	Filtre "Hardthreshold"	5
2.2.3	Aggregation	5
2.3	Phase 2 - Estimation finale	5
2.3.1	Block matching	5
2.3.2	Wien filter	5
2.3.3	Aggregation	5
3	Réalisation	6
3.1	Block matching	6
3.1.1	Calcul de la distance euclidienne	6
3.1.2	Tri des blocks	6
3.2	"3D transform"	6
3.3	Filtre "Hardthreshold"	6
3.4	Filtre de "Wien"	6
3.5	Aggregation	6
4	Résultats	7
4.1	BM3D vs BM3D-GPU	7
4.2	Influence de la taille de la fenêtre de recherche	7
4.3	Influence de la Kaiser-Window	7
4.4	Influence de la transformée 2D sur les blocks avant "Block-Matching"	7
4.5	Influence du calcul de distance sur la rapidité de l'algorithme	7
4.6	Influence du tri des block sur la rapidité de l'algorithme	7

5 Conclusion	8
5.1 Block matching	8

Chapitre 1

Introduction

Collaborate Filtering est le nom de la méthode de groupage et de filtrage de l'algorithme BM3D. Cette méthode est réalisée en 4 étapes: 1) Trouver un bloc (portion de l'image) similaire à un bloc de référence et ensuite grouper les ensembles pour former un bloc 3D. 2) Appliquer une transformée 3D sur l'ensemble des blocs 3D. 3) Réduire les coefficients du monde spectral. 4) Appliquer la transformée 3D inverse.

En atténuant le bruit, le *Collaborate Filtering* révèle les plus petits détails partagés par les groupes de blocs. Chaque bloc filtré est alors remis à sa position d'origine. Sachant que deux blocs peuvent se chevaucher, nous pouvons obtenir plusieurs estimations pour un pixel donné. C'est pourquoi, nous devons les combiner. Cette étape finale est l'*Aggrégation*.

Le premier filtre collaboratif est considérablement amélioré par un second filtre *Wiener Filtering*. Cette seconde étape mime la première étape avec deux différences. Le *Block-Matching* est appliqué sur l'image filtrée et non sur l'image bruitée. Nous n'appliquons plus un filtre *Hardthreshold* mais un filtre *Wiener Filtering*. L'étape finale d'*Aggrégation* reste inchangée.

L'algorithme BM3D détaillé ici est directement tiré de l'article originale [1]. Plusieurs analyses préalables de l'algorithme BM3D démontrent une diminution de la performance de débrouillage lorsque la déviation standard du bruit dépasse 40. Il a été démontré que pour de grandes valeurs de bruit (standard deviation), les paramètres doivent être revus. Le seuil de filtrage de la première phase doit être augmenté. Il a aussi été démontré que les transformées 2D appliquées peuvent influencer

l'algorithme. Dans [2], ils montrent que la meilleure qualité d'image est atteinte en appliquant une transformée *2D bi-orthogonal spline wavelet*.

Dans ce papier nous nous sommes focalisés sur la qualité, mais plus particulièrement sur l'implémentation sur GPU (parallélisation de l'algorithme BM3D). C'est pourquoi nous nous appliquons des DCT-2D. Pour ce qui est de la transformée 1D selon Z, nous avons la transformée d'Hadamard-Welsh [3]. Nous verrons que cette méthode est efficace sur GPU.

Dans le chapitre 2, nous présentons en détails l'algorithme BM3D. Le chapitre 3 est dédié à la réalisation sur GPU. Au chapitre 4, nous parlerons des résultats obtenus en comparaison aux travaux menés par [4]. Nous présentons nos conclusions et les prochaines étapes au chapitre 5.

Chapitre 2

Algorithme BM3D

2.1 Concepts généraux

2.2 Phase 1 - Estimation basic

2.2.1 Block matching

2.2.2 Filtre "Hardthreshold"

2.2.3 Aggregation

2.3 Phase 2 - Estimation finale

2.3.1 Block matching

2.3.2 Wien filter

2.3.3 Aggregation

Chapitre 3

Réalisation

3.1 Block matching

3.1.1 Calcul de la distance euclidienne

3.1.2 Tri des blocks

3.2 "3D transform"

3.3 Filtre "Hardthreshold"

3.4 Filtre de "Wien"

3.5 Aggregation

Chapitre 4

Résultats

- 4.1 BM3D vs BM3D-GPU**
- 4.2 Influence de la taille de la fenêtre de recherche**
- 4.3 Influence de la Kaiser-Window**
- 4.4 Influence de la transformée 2D sur les blocks avant "Block-Matching"**
- 4.5 Influence du calcul de distance sur la rapidité de l'algorithme**
- 4.6 Influence du tri des block sur la rapidité de l'algorithme**

Chapitre 5

Conclusion

5.1 Block matching

Liste des figures

Bibliographie

[1] test. test, 1992.