

Implémentation d'article de recherche
Removing Camera Shake via Weighted Fourier
Burst Accumulation

Sofiane Horache, Roman Fenioux

April 2017

Introduction

Ce projet est une implémentation de l'article Removing Camera Shake via Weighted Fourier Burst Accumulation proposé par Mauricio Delbracio et Guillermo Sapiro.

Le problème adressé ici est le flou de bougé, présent notamment dans les photographies amateur prises à la main dans des environnements mal éclairés. Pour avoir une photographie suffisamment exposée il est souvent nécessaire d'augmenter le temps d'exposition lorsqu'on ne peut ou ne veut pas agir sur l'ouverture ou la sensibilité de l'appareil. Mais si la position ou l'orientation de l'appareil change durant l'acquisition, on voit apparaître du flou dans la direction du mouvement.

Approche de l'article

Le mouvement de l'appareil durant la prise de vue est modélisé par une convolution de l'image par un noyau de flou. La direction du mouvement est aléatoire d'une image à l'autre et on veut donc récupérer l'information que contient chaque image pour obtenir une image nette. L'approche utilisée ici consiste à prendre une rafale d'images au lieu d'une seule, puis à les combiner pour éliminer ce flou. Cette technique permet même de débruiter l'image.

Pour une séquence de M images on a donc :

$$v_i = u_i * k_i + n_i$$

Pour ce faire l'article propose de calculer une moyenne pondérée de leurs transformées de Fourier pour éviter d'avoir à faire une estimation explicite du noyau de flou.

$$u_p(x) = \mathcal{F}^{-1} \left(\sum_{i=1}^M w_i(\zeta) \cdot \hat{v}_i(\zeta) \right) (x)$$
$$w_i(\zeta) = \frac{|\hat{v}_i(\zeta)|^p}{\sum_{j=1}^M |\hat{v}_j(\zeta)|^p}$$

Le paramètre p est un entier qui contrôle la manière dont sont agrégés les résultats. Si $p = 0$ la somme se ramène à une moyenne des transformées de Fourier des différentes images. Lorsque $p \rightarrow \infty$ cela revient à prendre pour chaque fréquence la maximale qu'elle atteint dans une des images.

Notre implémentation

Nous avons travaillé en C++ avec la librairie OpenCV.

Nous avons d'abord implémenté la méthode de l'article sur des images superposées bien recalées, puis nous avons implémenté l'algorithme ORSA pour le recalage des images. Nous n'avons pas implémenté la détection des homographies dégénérées.

Rappeler formules ORSA

Nos résultats

Images déjà recalées

Nous avons d'abord testé la méthode sur des images déjà recalées que nous avons créées nous même en convoluant une image avec différents noyaux de flous correspondant pour simuler le mouvement de l'appareil. Dans la figure 1 (page 2) on peut comparer le résultat obtenu à l'image originale utilisée et à la meilleure des images floues.

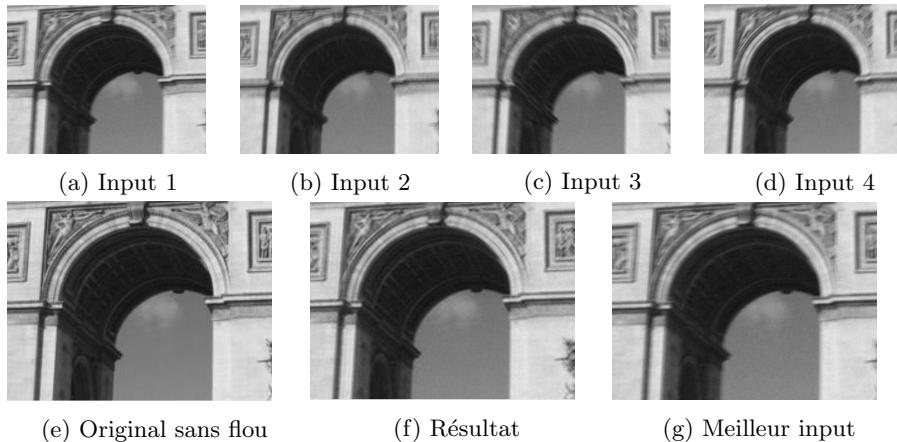


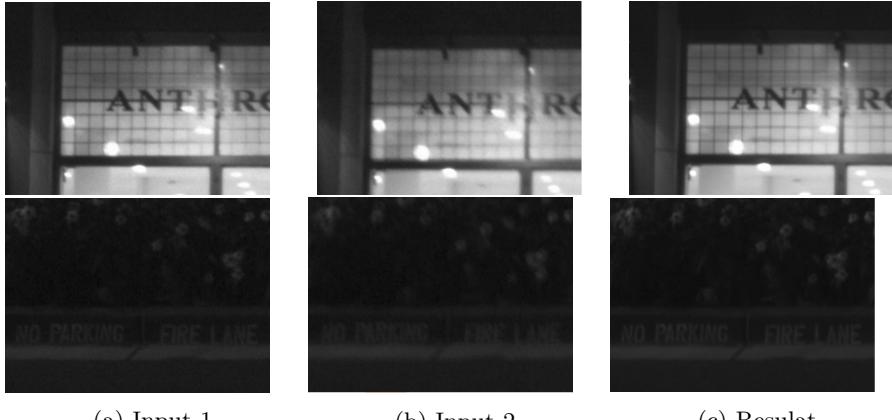
Figure 1: Résultat de l'algorithme pour 4 images recalées

Nous avons ajouté du bruit gaussien uniforme sur les images pour vérifier la robustesse de la méthode et comme annoncé par l'article cela ne dégrade pas le résultat. On observe même comme attendu un effet de débruitage plus ou moins efficace selon la quantité d'images utilisées.

Nous avons également appliqué l'algorithme aux images anthropologie utilisées par l'article (voir figure 2 page 3)

Influence des paramètres

On observe que le paramètre p a une grande influence sur la qualité du résultat (voir figure 3 page 3). Le choix de p correspond à un compromis entre le biais et la variance du résultat. Lorsque p est proche de 1, on obtient une image peu bruitée mais avec un biais fort : la valeur de fréquence choisies tient plus compte des différentes images. Au contraire avec une grande valeur de p le biais est plus



(a) Input 1

(b) Input 2

(c) Resultat

Figure 2: Résultat de l'algorithme sur les images anthropologies

faible mais la variance augmente, puisqu'on sélectionne la valeur maximale de la fréquence parmi les différentes images.

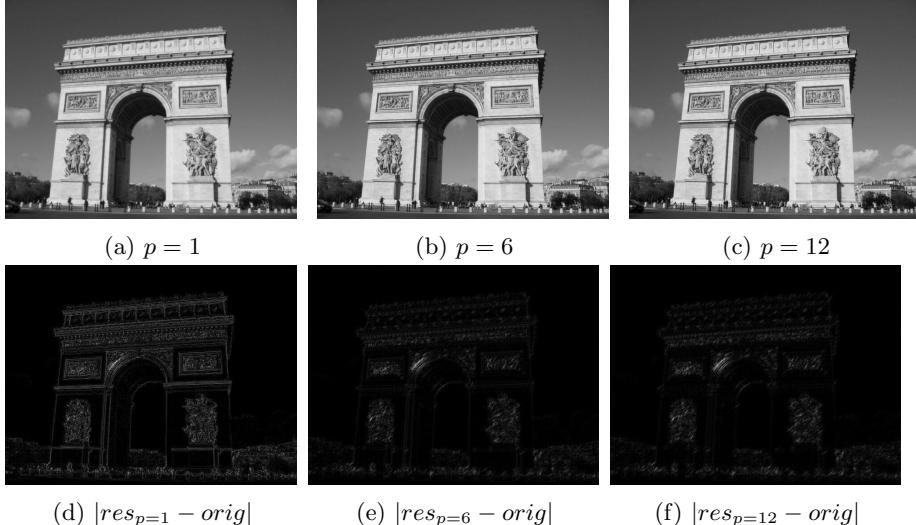


Figure 3: Influence du paramètre p

Images non recalées

Mettre une image d'orsa avec inliers

Dans la phase de recalage, le nombre d'itération utilisé joue un rôle très important. Comme on le voit sur la figure 4 (page 4), en doublant le nombre d'itérations on obtient un recalage plus fin qui permet un meilleur résultat. En effet la moindre erreur de recalage (quelques pixels) se traduit par des artefacts dus à des effets de déphasage (Figure 4c page 4).

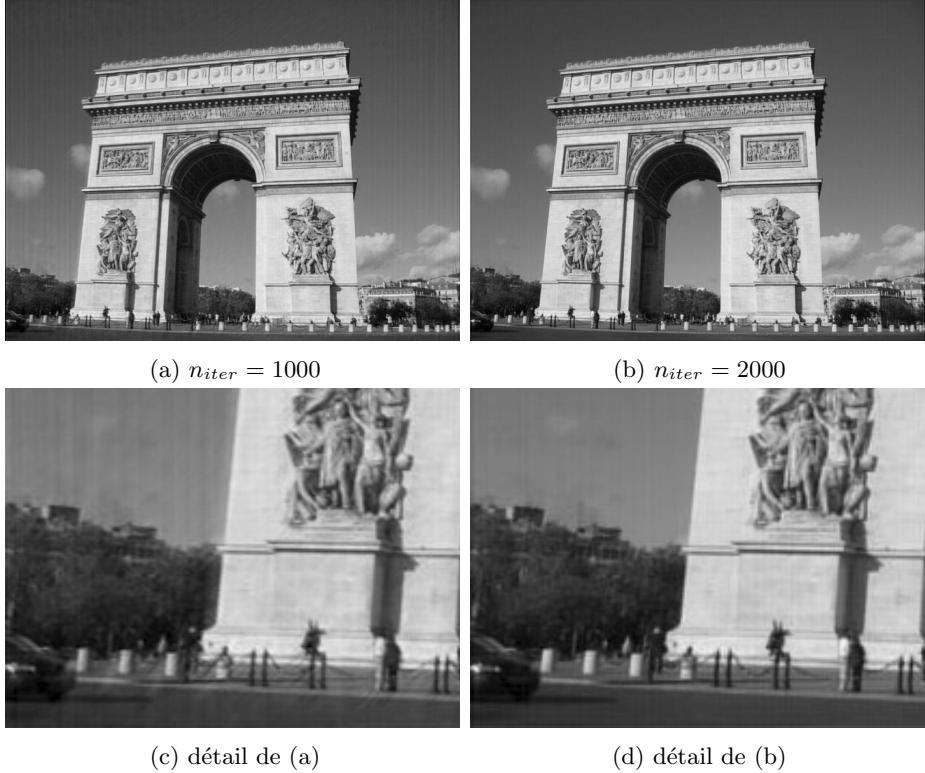


Figure 4: Influence du nombre d’itérations

Analyse et Commentaires

La méthode proposée par l’article donne de bons résultats dans le cas d’images bien recalées. Elle est très robuste au bruit et à des noyaux de flous relativement grands (jusqu’à 5 pixels), pas nécessairement en mouvement rectiligne. Cependant il y a quelques points qui peuvent poser problème.

Recalage des images

Le recalage s’avère être le point critique de notre implémentation. Une erreur lors du recalage d’une seule des images engendre de très gros défauts dans le résultat comme on le voit sur la figure 5 (page 5). Ce problème nous a conduit à rejeter les images dont l’estimation d’homographie s’appuie sur un nombre de points trop faibles. Nous utilisons donc ici le critère : image rejetée si $n_{inliers} < 100$. Nous sommes bien sûr conscient qu’un meilleur critère devrait être utilisé.

La phase de recalage peut prendre beaucoup de temps, mais est extrêmement importante pour avoir de bons résultats. Sur nos exemples il est nécessaire d’avoir au moins 2000 iterations dans l’algorithme ORSA pour obtenir un recalage satisfaisant et dans certains cas le recalage échoue lorsque les images sont trop floues. L’article évite cette question en supposant que le recalage



Figure 5: Résultat suite à un mauvais recalage

est fait immédiatement grâce aux gyroscopes des smartphones, mais cela peut constituer une limite à l'application de la méthode en temps réel dans d'autres appareils. Et effectivement si on se restreint à ce cas d'application on a une méthode rapide et efficace.

Flou non homogène : mouvement rotatif

Une limite intrinsèque à cette méthode est l'hypothèse que le flou peut être modélisé comme une convolution par un noyau. C'est le cas pour les mouvements de translation et de certaines rotations mais ce n'est plus possible lorsque l'appareil subit une rotation autour de son axe optique. La PSF dépend alors de la position du pixel dans l'image, ce qui ne peut pas être modélisé par une convolution avec un noyau.

Ce cas de figure peut pourtant se produire en situation réelle comme nous l'avons constaté (voir Figure 6, page 6) Le résultat est très bon au centre de l'image mais se dégrade vers les bords. Une solution envisageable pourrait être d'appliquer l'algorithme localement sur des portions de l'image suffisamment petites pour que l'hypothèse de convolution par un noyau de flou soit valable.

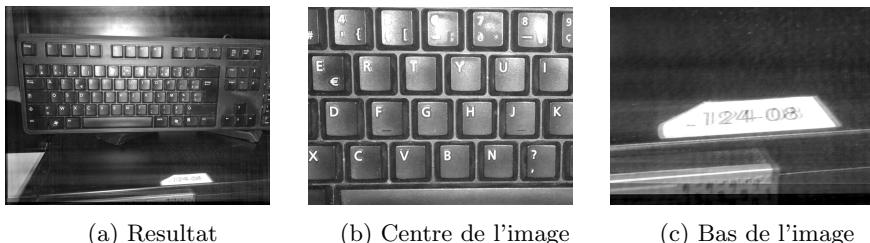


Figure 6: Résultat suite à un flou de rotation