
Co-conception Optimisation d'un flutter shutter

PEÑARANDA Yohan¹
GOSTIAUX Gabriel¹

We certify that this work is original, that we quote in reference all the sources used and that it does not contain plagiarism.

January, 2025

¹University of Paris-Saclay, Institut d'Optique Graduate School

Contents

1	Introduction	2
2	Simulation d'un flutter shutter	3
3	Optimisation	4
4	Conclusion	7
A	Codes	9

1 Introduction

Un flutter shutter est une technique reposant sur l'ouverture et la fermeture successive du diaphragme durant l'acquisition afin d'améliorer la restauration de l'image par déconvolution. L'objectif de ce TP est d'étudier les paramètres influençant la qualité de l'image restauré et d'optimiser le flutter shutter. On dispose pour cela de deux PSF, une simulant une acquisition classique, l'autre celle par un flutter shutter. Dans les deux cas, le flou de bougé provient uniquement d'un mouvement rectiligne uniforme dans la direction horizontale. L'objet se déplace avec une vitesse de $1 \text{ pixel} \cdot \Delta t^{-1}$, avec Δt l'écart temporel entre deux valeur du code. Le code en question correspond au signal d'ouverture et fermeture du diaphragme, en binaire, avec 1 et 0 respectivement. De cette manière, la PSF correspond directement au code temporel, à une normalisation près.[1]

L'étude débute par la simulation du flutter shutter, par la visualisation d'une image net convoluée aux PSF. Un premier critère d'optimisation sera dégagé et optimisé. Ensuite, un second critère sera étudié.

2 Simulation d'un flutter shutter

Densité spectrale de puissance Les deux PSF se différencient d'abord par leur densité spectrale de puissance. Pour les visualiser, on trace le module de la transformée de fourier des PSF en échelle semi-logarithmique, Fig.1. Dans la suite sera simulé la réponse d'une image parfaite avec le système d'imagerie, ce qui viendra à ajouté un bruit blanc gaussien d'écart-type $\sigma = 2$ à la convolution de l'image de référence avec une PSF.

En rouge la PSF avec le shutter, en vert l'acquisition continue. L'acquisition continue est reconnaissable par la forme de sa TF en sinus cardinal, due à sa forme de porte dans le domaine temporelle. La PSF se définit comme un étalement de la taille de la tache image d'un point source. Une densité spectrale de puissance plus étroite signifie moins de perte de détails dans les hautes fréquences, car cela résulte en une PSF plus localisée contenant plus de hautes fréquences spatiales, facilitant la récupération des détails lors de la déconvolution. La PSF la plus adaptée à la déconvolution est celle avec le shutter. Cet effet est visible grâce au tracé bleu de la DSP du bruit blanc. La DSP de la PSF avec le shutter est majoritairement au dessus du niveau de bruit, ce qui facilitera la déconvolution.

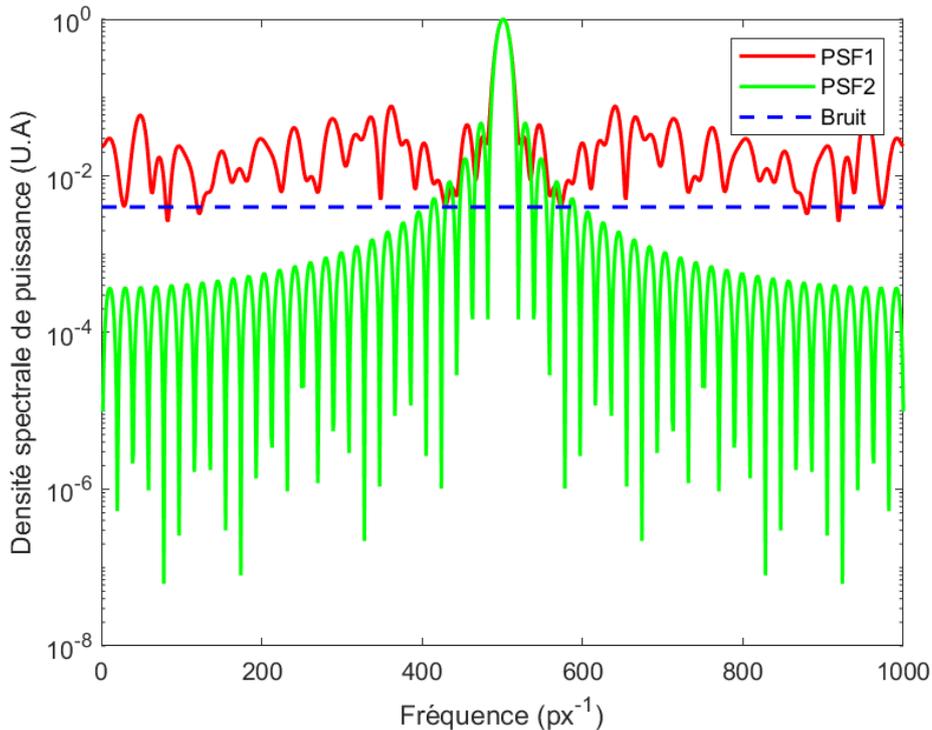


Figure 1: Densité spectrale de puissance, en unité arbitraire, des deux PSF étudiées. En rouge le flutter shutter, en vert l'acquisition continue. En bleu, la DSP d'un bruit blanc gaussien d'écart-type $\sigma = 2$.

Images de la scène après acquisition La simulation de l'image de la scène obtenue par un flou de bougé est calculé par convolution de l'image de référence avec un PSF et l'ajout d'un bruit blanc gaussien ($\sigma = 2$). L'image de référence est considérée comme image parfaite de la scène. Le bruit ajouté correspond au bruit du système de détection, à savoir l'ajout quadratique des bruits de photons, d'obscurité, de Johnson, et toute autres sources de bruit excepté le flou de bougé dont l'information est contenue dans la PSF. Pour la convolution, la

fonction `imfilter` est utilisé avec comme paramètre de convolution “circular” afin de limiter les effets de bords [2].

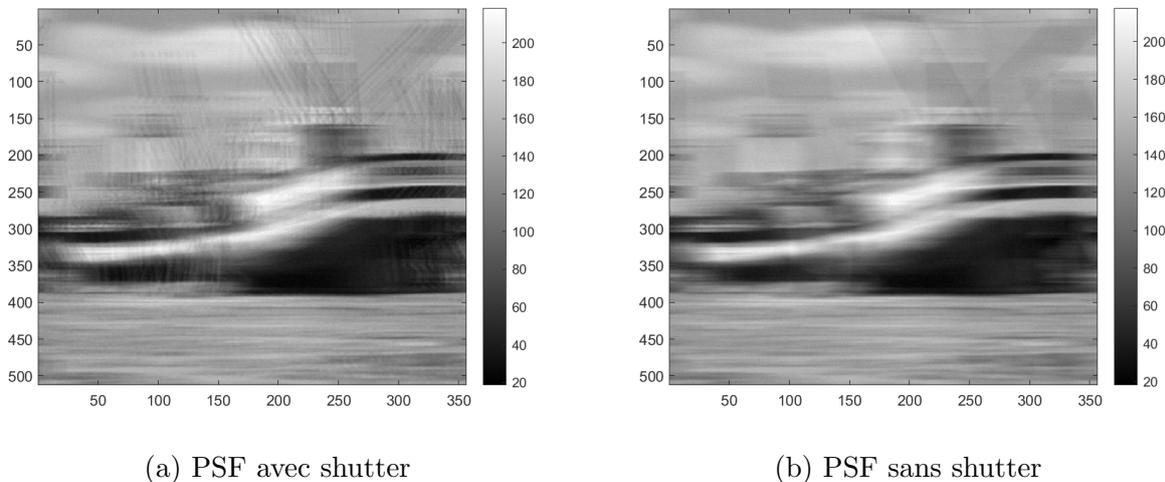


Figure 2: Image floue et bruité, résultat de la convolution de l’image de référence avec une PSF et de l’ajout d’un bruit blanc gaussien d’écart-type $\sigma = 2$.

Images de la scène après déconvolution La déconvolution de l’image acquise (flouté et bruité) se fait avec un filtre de Wiener. Conformément à Fig.3, l’image par la PSF avec shutter se déconvolue et l’image de référence est reconnaissable. En revanche, l’image formée sans shutter dégradent les hautes fréquences spatiales, qui sont noyées dans le bruit et filtrées. Pour quantifier la qualité des images et permettre une comparaison qualitative dans l’objectif d’une optimisation, un critère de qualité de reconstruction RQ est utilisé selon la formule suivante :

$$RQ = 10 \log \left(\frac{Var(x)}{Var(x - u)} \right)$$

RQ est défini comme le rapport des variances de l’image reconstruite et de la variance des écarts entre les images, avec x l’image reconstruite et u l’image de référence. Plus RQ est élevé, plus l’image reconstruite est proche de l’image de référence. Dans cette étude, les valeurs de RQ calculés sont données Tab.1. Conformément à la description précédente, l’image acquise avec le shutter permet une meilleure reconstruction de l’image d’après ce critère.

PSF	RQ
avec shutter	10,07
sans shutter	7,13

Table 1: RQ des images en fonction de la PSF.

3 Optimisation

1^{er} critère Le premier critère d’optimisation est proposé par [4]. Le flutter shutter est optimisé en maximisant la valeur minimale de la densité spectrale de puissance de la PSF et en minimisant son écart-type. Soit n correspondant au nombre de 1, ils imposent deux contraintes sur les codes potentiels :

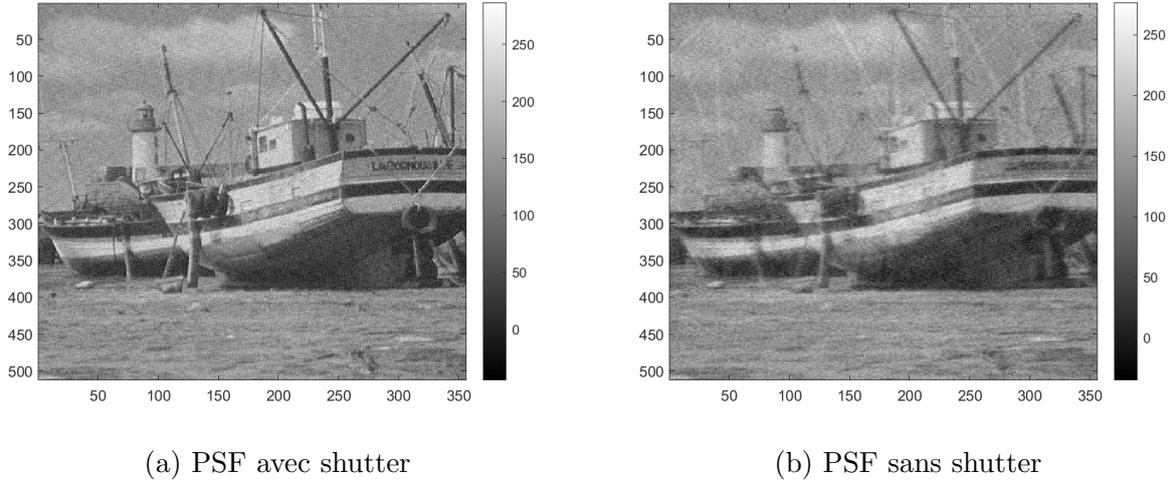


Figure 3: Déconvolution par un filtre Wiener de chaque image acquise.

- Pour perdre le moins possible de flux lumineux, ils considèrent que le diaphragme est ouvert la moitié du temps, autrement dit soit n la longueur du code, il existe $s = \frac{n}{2}$ valeurs égales à 1 dans le code du flutter shutter.
- Pour chaque code, la première et la dernière valeur est égale à 1.

Compte tenu de ces contraintes, le nombre de code possible en fonction de n suit une combinaison de $\frac{n}{2}$ parmi $n - 2$, soit :

$$N = \binom{n-2}{n/2} = \frac{(n-2)!}{(n/2)!(n-2-n/2)!}$$

Pour $n = 14$, il existe 792 codes possibles. Pour $n = 20$, il existe 43 758 codes possibles. Cela montre une contrainte sur l'optimisation, pour une optimisation fine, le nombre n doit être suffisamment grand, ce qui implique un temps de calcul plus important. Dans la suite, la valeur retenue sera $n = 12$, donnant 210 possibilités de codes.

L'optimisation consiste à sélectionner le code de PSF minimisant l'écart-type de sa DSP correspondante, afin d'affiner la PSF, tout en maximisant la valeur minimal de cette DSP. Le code optimal est celui en haut à gauche de la Fig.4. Ce point correspond en réalité à deux codes symétriques :

- Code 1 : 110110000101.
- Code 2 : 101000011011.

Deux codes symétriques donnent la même densité spectrale de puissance. Ceci est dû à la propriété des parité de la TF et l'uniformité du flou. Un mouvement accéléré ne permettrait pas une tel propriété.

Ce critère permet d'optimiser la qualité de reconstruction de l'image en réduisant la largeur de la PSF. Cependant, il fait l'impasse sur une notion abordée précédemment. L'une des hypothèses de travail formulait reposait sur l'ouverture total correspondante à la moitié du temps d'intégration total afin de permettre un signal suffisant. Il apparaît alors une seconde piste d'optimisation, le rapport signal sur bruit (RSB). C'est ce critère qu'explore [3].

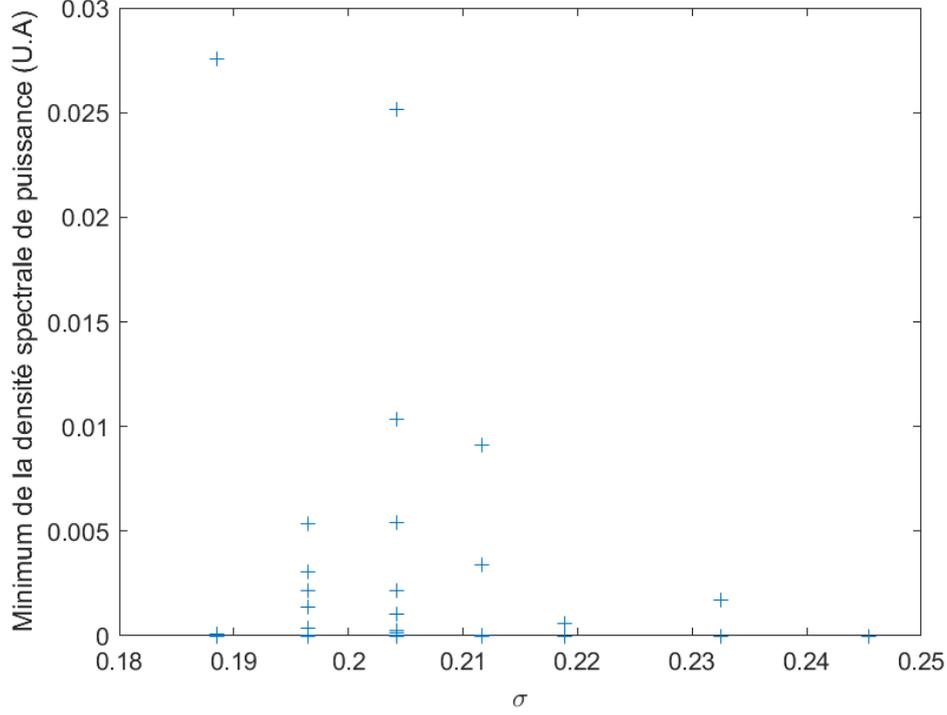


Figure 4: Densité spectrale de puissance de chaque code trié selon leur largeur σ et leur minimum respectif.

2nd critère Le second critère d'optimisation proposé par [3] est basé sur le RSB après déconvolution. Il s'écrit :

$$RSB \approx \frac{st/n}{F_D \sqrt{\sigma^2 + \sigma_{ph}^2}}$$

Avec :

- t : la durée totale de l'acquisition.
- $f_D = \sqrt{\text{Tr}\{(A^t A)^{-1}\}/m}$, où A est la matrice de convolution de taille $(m+n-1)m$ associée à la PSF et m la longueur du vecteur relatif à la scène statique.
- σ^2 : la variance du bruit électronique.
- $\sigma_{ph}^2 = \frac{Cts}{n}$: la variance du bruit de photons, où C est une constante.

Dans la suite, les valeurs considérées seront $\Delta t = 1$ ms, $\Delta t = 1$ ms.

Cette formule (codée en annexe A) permet de calculer les RSB en dB des deux PSF utilisées pour les images figures 2. Les RSB sont : $RSB1 = -35,8705$ dB et $RSB2 = -44,3266$ dB. Pour tenter de trouver le s qui permet d'optimiser le RSB, il est possible de tracer les variations du RMS maximal en fonction du nombre de 1 dans le code. Ces variations sont présentées figure 5. Après essais pour différentes valeurs de n , une règle empirique peut-être dégagée : le rapport signal à bruit est maximiser pour les flutter-shutter ouvert pendant un intervalle de temps correspondant à la moitié du temps maximal possible.

Cette règle empirique est en accord avec le critère de la première partie. Les codes correspondant à la maximisation du RSB sont les codes symétriques :

- Code 1 : 110110000101.
- Code 2 : 101000011011.

Ce sont les mêmes codes que ceux optimisant le premier critère.

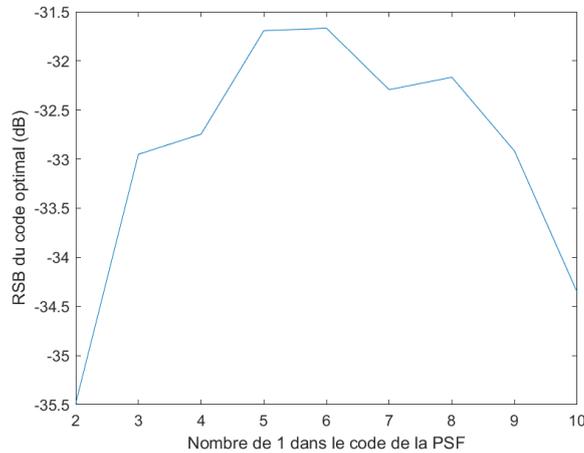


Figure 5: Variation du RSB maximum en décibel en fonction du nombre de 1 dans le code de la PSF.

4 Conclusion

Dans ce travail pratique, l'objectif principal était d'optimiser le shutter en maximisant la densité spectrale de puissance minimale de la PSF, tout en minimisant son écart-type. Ces critères permettent d'optimiser la distribution de la lumière tout en réduisant l'impact du bruit. Le rapport signal à bruit a ensuite été utilisée comme critère de performance pour vérifier les codes optimisés.

En particulier, les résultats montrent que le maximum du RSB est atteint pour certains codes spécifiques. Ces codes maximisent la densité spectrale de puissance tout en maintenant un faible écart-type, ce qui conduit à une meilleure qualité de l'image en termes de signal par rapport au bruit.

En utilisant les contraintes de symétrie (première et dernière valeurs égales à 1) et en maximisant la valeur minimale de la DSP, les codes de shutter qui maximisent le RSB ont été identifiés. Ces codes sont ceux qui non seulement respectent la contrainte sur la DSP, mais aussi ceux qui garantissent un RSB maximal.

References

- [1] François Goudail, Marie-Anne Burcklen, and Pauline Trouvé. *Co-conception : TP 2 Optimisation d'un flutter shutter*. 2024. URL: https://ecampus.paris-saclay.fr/pluginfile.php/2693106/mod_folder/content/0/TP2.pdf?forcedownload=1.
- [2] MATLAB. *24.2.0.2773142 (R2024b) Update 2*. Natick, Massachusetts, 2024.
- [3] Amit Agrawal and Ramesh Raskar. “Optimal single image capture for motion deblurring”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 2560–2567. DOI: [10.1109/CVPR.2009.5206546](https://doi.org/10.1109/CVPR.2009.5206546).
- [4] Ramesh Raskar, Amit Agrawal, and Jack Tumblin. “Coded exposure photography: motion deblurring using fluttered shutter”. In: *ACM Trans. Graph.* 25.3 (July 2006). ISSN: 0730-0301. DOI: [10.1145/1141911.1141957](https://doi.org/10.1145/1141911.1141957). URL: <https://doi.org/10.1145/1141911.1141957>.

A Codes

```
%%% Calcul le RSB en dB
%%%
%%% usage : RSB_dB=Calcul_RSB(code, n, s, m, C, t, sigma)
%%%
%%% Entree:
%%% code : PSF considéré
%%% n : taille du code (PSF)
%%% s :
%%% m : longueur de la scene statique
%%% C : constante arbitraire
%%% t : intervalle de temps
%%% sigma : ecart-type
%%% Sortie:
%%% RSB_dB : rapport signal à bruit en dB
%%%

function RSB_dB = Calcul_RSB(PSF, n, s, m, C, t, sigma)
    A = convmtx(PSF, m);
    AtA_inv = inv(A'*A); % Inverse de A'*A
    f_D = sqrt(trace(AtA_inv)/m);
    sigma_ph = sqrt((C*t*s)/n);
    RSB = (s*t/n)/(f_D*sqrt(sigma^2 + sigma_ph^2));
    RSB_dB = 10*log10(RSB);
end
```

Calcul du RSB