

---

## Rapport de Traitement d'Image

Jérémy Jean, Victor Cillerios, Maxime Gey

---

### Abstract :

Les modèles standards convolutifs pour déflouter des images partent du postulat de flou circulaire. Peu réaliste, cette hypothèse présente un intérêt algorithmique en implémentant l'opérateur de convolution par FFT.

L'approche proposée combine un masque avec un modèle de flou circulaire pour conserver l'intérêt en complexité des modèles standards pour un modèle résultant non circulaire. Résolu par une approche lagrangienne et avec un schéma de fractionnement de variable, nous pouvons découpler les deux opérateurs impliqués pour résoudre classiquement le système linéaire. Les sous-opérations de l'ensemble convergent plus rapidement que l'état de l'art en palliant les effets de bords.

### Detailed explanations :

Pour la restauration d'image, l'estimateur est obtenu en minimisant une fonction de coût (formulation analytique) de la forme :

$$\hat{x} = \arg \min_x \Psi(x) = J(x) + \lambda \Phi(Rx)/$$

avec  $J(x)$  le terme de fidélité aux données et  $\Phi(\cdot)$  une fonction de régularisation.

L'approche commune consiste à émettre l'hypothèse d'un flou circulaire. L'expression du terme de fidélité aux données vaut alors :

$$J(x) = \|y - Ax\|_2^2$$

avec la condition que  $A$  soit une matrice circulante c'est-à-dire une matrice de la forme

$$A = \begin{bmatrix} c_0 & c_{n-1} & & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ & c_{n-1} & c_{n-2} & & c_1 & c_0 \end{bmatrix}$$

Cette hypothèse bien que pratique n'est pas réaliste car la plupart des images ne la vérifient pas. De plus, la reconstruction d'images sous ce postulat provoque des discontinuités aux bords. Pour pallier ce problème, nous introduisons un masque qui tronque l'enveloppe circulaire aux bords sous la forme d'une matrice  $T$  telle que :

$$J(x) = \|y - TAx\|_2^2$$

La fonction de coût que nous souhaitons minimiser devient la suivante :

$$\hat{x} = \arg \min_x \left\{ \frac{1}{2} \|y - TAx\|_2^2 + \Phi(Rx) \right\}$$

De nombreux algorithmes comme *Nonlinear Conjugate Gradient (NCG)* ou *(MF)/ISTA* permettent d'optimiser la fonction précédente mais la catégorie d'algorithmes présentant la plus convergence la plus rapide sont les algorithmes de fractionnement de variables (*ADMM*). Le principe est de découpler le terme de régularisation du terme de fidélité aux données en introduisant de nouvelles variables et de nouvelles conditions dans la fonction.

L'approche par *SALSA* considère la nouvelle variable  $u$  telle que  $u = x$  qui transforme le problème sous contraintes :

$$\min_{x,u} \{ \Psi(x, u) = \frac{1}{2} \|y - TA x\|_2^2 + \lambda \Phi(Ru) \} \text{ s.t. } u = x$$

ce qui permet de définir le Lagrangien associé au problème d'optimisation :

$$\mathcal{L}(x, u, \mu, \eta) = \Psi(x, u) + \frac{\mu}{2} \|u - x - \eta\|_2^2$$

où  $\mu > 0$  est défini comme étant un paramètre de pénalisation et  $\eta$  est lié linéairement au multiplicateur de Lagrange pour la contrainte. On aboutit de cette méthode à la définition à chaque itération de  $x$ ,  $u$  et  $\eta$  qui permet de converger vers une solution du problème.

L'approche basée sur la méthode *Split-Bregman* utilise quant à elle la nouvelle variable  $v$  telle que  $v = Rx$  pour découpler les deux termes de la fonction. Le problème d'optimisation sous contraintes devient alors :

$$\min_{x,v} \{ \Psi(x, v) = \frac{1}{2} \|y - TA x\|_2^2 + \lambda \Phi(v) \} \text{ s.t. } v = Rx$$

qui permet d'associer l'expression Lagrangienne :

$$\mathcal{L}(x, v, \mu, \eta) = \Psi(x, v) + \frac{\mu}{2} \|v - Rx - \eta\|_2^2$$

Si cette nouvelle approche semble très proche de *SALSA* elle permet néanmoins de calculer  $v$  à chaque itération avec un seuillage doux en une seule équation :

$$v^{(k+1)} = \text{shrink}\{Rx^{(k)} + \eta^{(k)}, \frac{\lambda}{\mu}\}$$

Comparativement, la mise à jour de  $u$  utilise un algorithme de type Chambolle qui utilise plusieurs sous-opérations.

La véritable contribution du papier est dans la définition d'une nouvelle approche :

*ADMM-P2*. Dans cet algorithme, on définit deux variables auxiliaires  $u_0$  et  $u_1$  telles que  $u_0 = Ax$  et  $u_1 = Rx$  pour séparer le terme de régularisation, ce qui définit un nouveau problème d'optimisation sous contraintes :

$$\min_{x,u_0,u_1} \{ \Psi(u_0, u_1) = \frac{1}{2} \|y - Tu_0\|_2^2 + \lambda \Phi(u_1) \} \text{ s.t. } u_0 = Ax, u_1 = Rx$$

L'expression du Lagrangien associé est donc :

$$\mathcal{L}(x, u, \mu, \nu, \eta) = \Psi(u_0, u_1) + \frac{\mu}{2} \|u_0 - Ax - \eta_0\|_2^2 + \frac{\mu\nu}{2} \|u_1 - Rx - \eta_1\|_2^2$$

Sous cette définition du problème, on peut montrer que la minimisation des coefficients à chaque étape est donné par les expressions suivantes :

$$u_0^{(k+1)} = \arg \min_{u_0} \{ \frac{1}{2} \|y - Tu_0\|_2^2 + \frac{\mu}{2} \|u_0 - Ax^{(k)} - \eta_0^{(k)}\|_2^2 \}$$

$$u_1^{(k+1)} = \arg \min_{u_1} \{ \lambda \Phi(u_1) + \frac{\mu\nu}{2} \|u_1 - Rx^{(k)} - \eta_1^{(k)}\|_2^2 \}$$

$$x^{(k+1)} = \arg \min_x \{ \frac{\mu}{2} \|u_0^{(k+1)} - Ax - \eta_0^{(k)}\|_2^2 + \frac{\mu\nu}{2} \|u_1^{(k+1)} - Rx - \eta_1^{(k)}\|_2^2 \}$$

$$\eta_0^{(k+1)} = \eta_0^{(k)} - u_0^{(k+1)} + Ax^{(k+1)}$$

$$\eta_1^{(k+1)} = \eta_1^{(k)} - u_1^{(k+1)} + Rx^{(k+1)}$$

Pour  $u_0$ , la fonction de coût est quadratique donc l'optimisation à l'itération  $k + 1$  peut s'écrire sous la forme :

$$u_0^{(k+1)} = (T'T + \mu I_N)^{-1} [T'y + \mu(Ax^{(k)} + \eta_0^{(k)})]$$

Similairement, la fonction de coût relative à l'optimisation de  $x$  est quadratique et l'expression de  $x$  au temps  $k + 1$  peut donc s'écrire :

$$x^{(k+1)} = H_\nu^{-1} [A'(u_0^{(k+1)} - \eta_0^{(k)}) + \nu R'(u_1^{(k+1)} - \eta_1^{(k)})]$$

où  $H_\nu = A'A + \nu R'R$  et l'inversion peut-être réalisée par des FFT.

On peut également montrer que l'optimisation relative à  $u_1$  peut être effectuée avec un

seuillage doux de seuil  $\frac{\lambda}{\mu}$  :

$$u_1^{(k+1)} = shrink\{Rx^{(k)} + \eta^{(k)}, \frac{\lambda}{\mu}\}$$

En théorie, la convergence est assurée quelque soit le choix des paramètres faits pour l'algorithme *ADDM-P2* mais on constate expérimentalement que l'on souhaite avoir :

$$v_{min} = \arg \min_{\nu} \{\kappa(A'A + \nu R'R)\}$$

$$\mu\nu = 2^8 \frac{\lambda\nu_{min}}{x_{max}}$$

où  $x_{max}$  est le maximum d'intensité dans l'image floue et  $\kappa(\cdot)$  représente un numéro d'état.

En particulier, les auteurs se sont rendus compte que l'on pouvait fixer la valeur de

$$\mu = \mu_0 = 2^{-4} \text{ puis choisir } \nu = 2^8 \frac{\lambda\nu_{min}}{\mu_0 x_{max}}$$

### Algorithm description :

L'ensemble du code réalisé pour ce projet se trouve sur GitHub à l'adresse :

[https://github.com/VictorCilleros/Image\\_Restoration\\_SDI/](https://github.com/VictorCilleros/Image_Restoration_SDI/)

L'idée de l'algorithme est le suivant :

1. Sélectionner  $x^{(0)}$ ,  $\nu > 0$ , et  $\mu > 0$
2. On calcule le padding de l'image réelle :  $T'y$
3. On fixe les paramètres  $\eta_0^{(0)} = 0$ ,  $\eta_1^{(0)} = 0$  et  $k = 0$
4. Tant que le critère d'arrêt n'est pas rempli :
  - a. On calcule  $u_0^{(k+1)}$  avec l'équation vue plus haute
  - b. On calcule  $u_1^{(k+1)}$  avec l'équation vue plus haute
  - c. On calcule  $x^{(k+1)}$  avec l'équation vue plus haute
  - d. On calcule  $\eta_0^{(k+1)}$  avec l'équation vue plus haute

e. On calcule  $\eta_1^{(k+1)}$  avec l'équation vue plus haute

Le flou choisi est un flou gaussien qui nous permet de constituer notre image bruitée. Si la structure de l'algorithme était déjà donnée dans le papier de recherche, nous avons choisi les paramètres suivants :

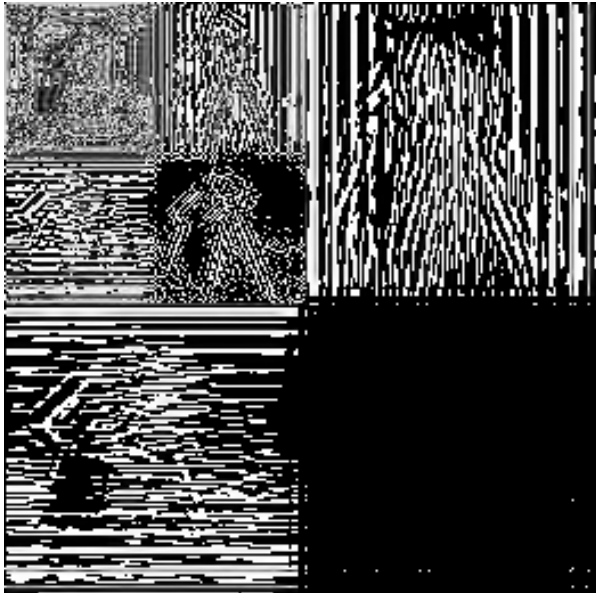
- $\mu = 2^{-4}$
- $\nu_{min} = 4.3$
- $\lambda = 2^{-17}$

### Numerical illustrations :

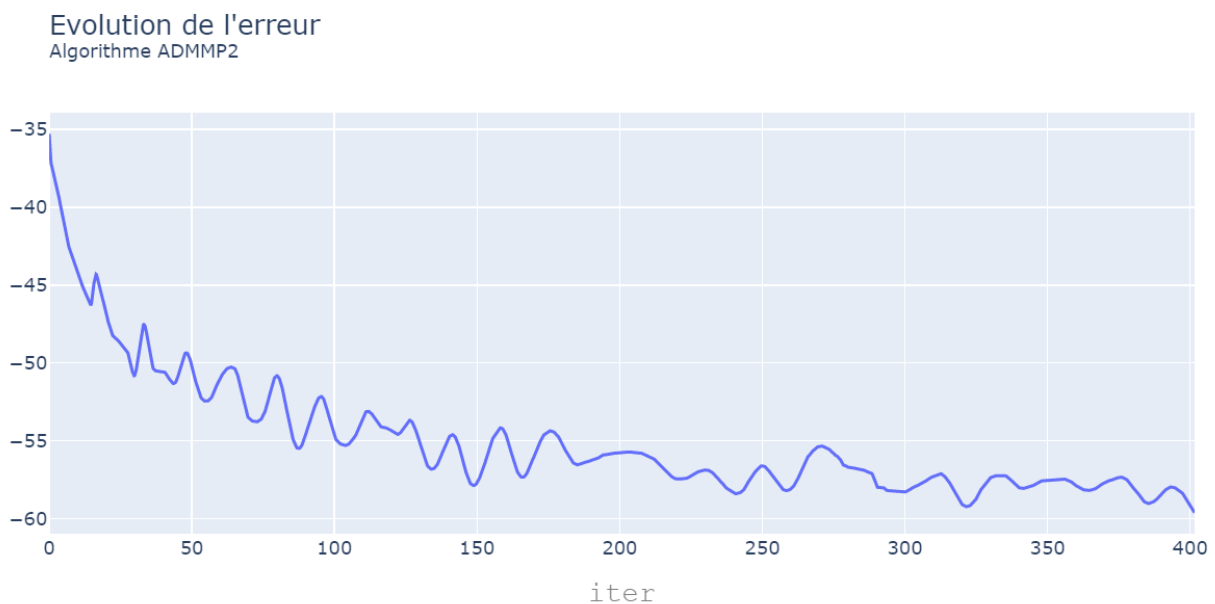
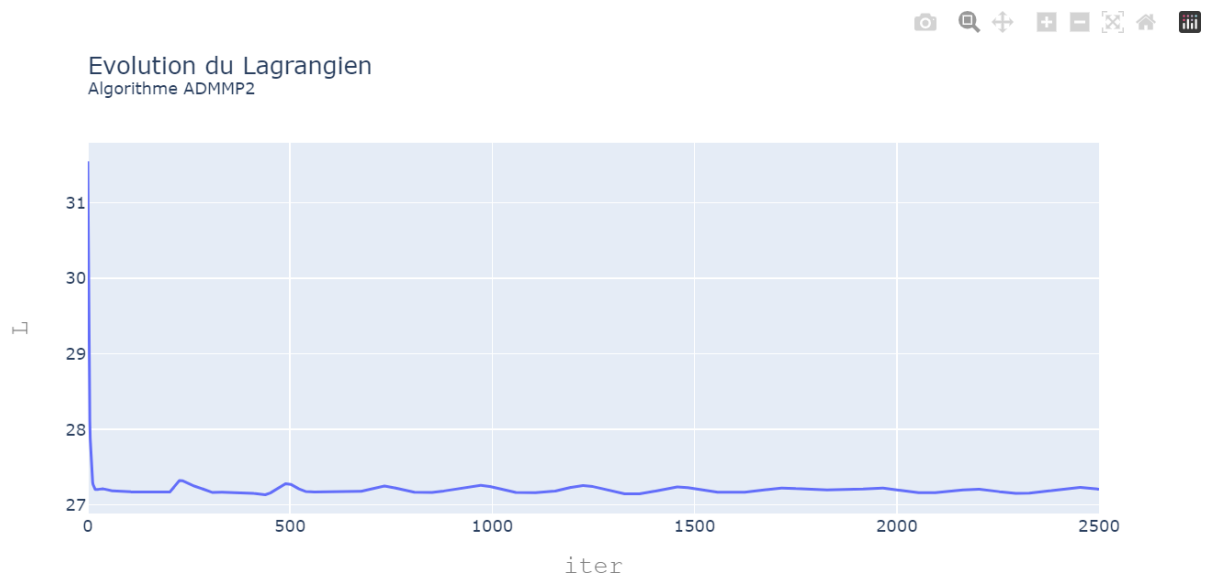
On montre ici l'image choisie pour les expérimentations ainsi que son équivalent floutée par un filtre gaussien et l'image défloutée :



Nous avons également représenté l'opérateur  $R$ , opérateur de *Sparsifying Transform* :



Enfin, nous avons obtenu des résultats exprimant l'évolution du Lagrangien en fonction du nombre d'itérations ainsi que l'évolution de l'erreur en fonction du nombre d'itérations :



La stratégie de l'algorithme *ADMM-P2* de définir deux variables auxiliaires pour séparer le terme de régularisation permet de calculer beaucoup plus rapidement l'inversion matricielle de  $H_{\nu}$  par des transformées de Fourier de façon directe et non itérative. Si le nombre d'itérations avec les algorithmes de l'état de l'art sont similaires, son temps de calcul à chaque itération est bien diminué.

### Conclusion :

L'approche proposée par *ADMM-P2* de découpler le terme de régularisation par deux variables auxiliaires permet d'utiliser les transformées de Fourier pour calculer les inversions matricielles nécessaires. Cela apporte une convergence en un temps beaucoup plus intéressant que les algorithmes de l'état de l'art même si le nombre d'itérations ne change pas. Cela permet également une très bonne reconstruction d'image sans hypothèse circulante initiale

De plus, nous pourrions utiliser la TV comme méthode de régularisation qui semblerait plus adaptée à notre image.

**Références :**

- Antonios Matakos, Student Member, IEEE, Sathish Ramani, Member, IEEE, and Jeffrey A. Fessler, Fellow, IEEE, Accelerated Edge-Preserving Image Restoration Without Boundary Artifacts
- B. Hunt, A matrix theory proof of the discrete convolution theorem
- Fessler, The Discrete Fourier Transform