Débruitage générique multi-échelle

Marc Lebrun, Jean-Michel Morel

CMLA, Ecole Normale Supérieure de Cachan
61 avenue du Président Wilson, 94235 Cachan cedex, France
marc.lebrun.ik@gmail.com, morel@cmla.ens-cachan.fr

Résumé – Nous proposons une stratégie multi-échelle pour étendre tous les algorithmes de débruitage, qui a *a priori* trois avantages : 1) pour les algorithmes utilisant des patches, elle favorise une meilleure comparaison entre patches, 2) le bruit décroît aux sous-échelles, ce qui donne un avantage compétitif aux algorithmes de l'état-de-l'art sachant traiter les faibles valeurs de bruit, 3) pour les algorithmes basés sur les patches, effectuer un zoom arrière d'une image revient à augmenter la taille des patches et des voisinages, et donc à mieux traiter le bruit basse fréquence, en particulier le bruit coloré. L'avantage visuel est clair bien que le PNSR n'augmente pas.

Abstract – We propose a multi-scale image denoising strategy, applicable to any one scale denoising algorithm. It has *a priori* three advantages: 1) it favors a better patch comparison in patch based algorithm, 2) at lower scales the noise decreases, which is quite a good situation since for low noise state-of-the-art algorithms work very well, 3) for patch-based algorithms, doing a sub-sampling of an image comes down to enlarge the size of patches and neighborhoods and then remove low frequency noise and better handle colored noise. Experimental comparison proves that the multi-scale strategies improve significantly image quality while not yet increasing the PSNR.

1 Introduction

Le débruitage est la première étape de toute chaîne de traitement d'images. Bien que la plupart des algorithmes de débruitage état-de-l'art actuels tels que BM3D (Dabov et al. [1]), NL-means (Buades et al. [2]), K-SVD (Mairal et al. [3], [4]), filtres de Wiener appliqués sur TCD (Yaroslavsky et al. [5], [6]) ou transformées en ondelettes (Donoho et al. [7]) ou même la minimisation de variation totale (Rudin et al. [8]) obtiennent de très bons résultats, il reste néanmoins un problème. En effet, pour de fortes valeurs de bruits (typiquement pour des images codées en 8 bits, $\sigma \geq 40$) beaucoup d'artefacts inhérents à chaque méthode commencent à apparaître. Pour de très fortes valeurs de bruit, des artefacts de bruit basse fréquence sont clairement laissés par les algorithmes, et ne peuvent être supprimés à l'échelle la plus fine. Nous présentons un cadre d'extension multi-échelle valable pour la plupart des algorithmes mono-échelle, et en particulier une extension multi-échelle de l'algorithme NL-Bayes, qui est un bon exemple de l'état de l'art, étant nettement plus performant que BM3D sur les images couleur. En particulier nous indiquerons comment calculer la matrice de covariance du bruit à chaque échelle, ce qui est évidemment requis pour toute extension multi-échelle. La section 2 décrit les opérations de sur- et sous-échantillonnage. La matrice de covariance est calculée dans la section 3. La section 4 compare les résultats mono-échelle et multi-échelle.

2 Sur- et sous-échantillonnage

Les algorithmes de débruitage échouent à enlever le bruit basse fréquence parce que, même quand ils se disent "non-locaux", ils restent encore assez locaux même s'ils travaillent sur un voisinage un peu plus large. Il faut donc une statégie multi-échelle pour traiter toutes les tailles de voisinage.

Soit s l'échelle courante de l'algorithme multi-échelle. On veut obtenir à partir de $\tilde{u}_s$ une image $\tilde{u}_{s+1}$ où l'écart-type du bruit a divisé par deux celui de $\tilde{u}_s$. Il faut pour cela avoir un filtre $f(i,j)$ tel que

$$\sum_{i,j} f(i,j) = 1 \text{ et } \sum_{i,j} f(i,j)^2 = \frac{1}{4}.$$

Par exemple le filtre moyenne locale $\mathbf{B}$ convient, défini par

$$\mathbf{B}(i,j) = \begin{cases} \frac{1}{4} & \text{si } (i,j) \in [(0,0),(0,1),(1,0),(1,1)], \\ 0 & \text{sinon.} \end{cases}$$

Il fait la moyenne de quatre pixels contigus et garde avantageusement un bruit blanc après sous-échantillonnage si le bruit initial était blanc. Cette propriété est commode car la plupart des algorithmes de débruitage sont pensés pour du bruit blanc. On peut obtenir quatre résultats disjoints par sous-échantillonnage de deux qui peuvent donc être disjoints séparément (notre but n'est pas d'accélérer la stratégie multi-échelle, mais de débruiter au mieux). Remarquons que si $\tilde{u}_s$ est bien échantillonnée, $\tilde{u}_s * \mathbf{B}$ le reste et donc l'image différence n'est pas aliasée.

Le sur-échantillonnage Son but est de reconstruire une image à partir des quatre images sous-échantillonnées comme indiqué ci-dessus. Les centres des pixels des images $\tilde{u}_1$, $\tilde{u}_2$, $\tilde{u}_3$ et $\tilde{u}_4$ (montrés en rouge, violet, vert et bleu dans la figure 1) sont situés à l'intérieur des pixels de $\tilde{u}$ (dont les centres sont en noir). Donc leurs coordonnées sont décalées de $\pm\frac{1}{2}$.

La reconstruction des pixels de $\tilde{u}$ (voir par exemple le pixel marqué en jaune) se fait tout simplement comme une moyenne des quatre voisins appartenant à chacune des sous-images.

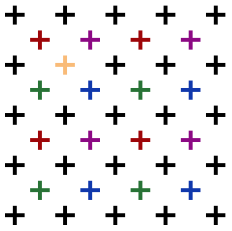

FIGURE 1 – Les centres des pixels des images $\tilde{u}_1$, $\tilde{u}_2$, $\tilde{u}_3$ et $\tilde{u}_4$ (en rouge, violet, vert et bleu) sont situés à l'intérieur des pixels de $\tilde{u}$ (dont les centres sont en noir). La reconstruction des pixels de $\tilde{u}$ (voir par exemple le pixel marqué en jaune) se fait en moyennant les quatre voisins colorés.

3 Calculer la matrice de covariance du bruit

3.1 Revue rapide de NL-Bayes

Le calcul de la matrice de covariance est nécessaire pour NL-Bayes, comme pour la plupart des algorithmes, car elle caractérise un bruit gaussien, pas nécessairement blanc.

NL-Bayes s'effectue en deux étapes : d'abord une estimation de base, ensuite une estimation finale de type Wiener basée sur l'estimation de base. Les deux étapes sont données par les formules

$$P^{\mathbf{base}} = \overline{\tilde{P}} + [\mathbf{C}_{\tilde{P}} - \mathbf{C}_n]\,\mathbf{C}_{\tilde{P}}^{-1}(\tilde{P} - \overline{\tilde{P}})$$

pour la première étape, où $\mathbf{C}_n$ est la matrice de covariance du bruit, $\mathbf{C}_{\tilde{P}}$ celle des patches semblables au patch de référence $\tilde{P}$, $P^{\mathbf{base}}$ est l'estimation de base de P, et

$$P^{\mathbf{final}} = \overline{\tilde{P}}^{\mathbf{base}} + \mathbf{C}_{\tilde{P}}^{\mathbf{base}}\left[\mathbf{C}_{\tilde{P}}^{\mathbf{base}} + \mathbf{C}_n\right]^{-1}(\tilde{P} - \overline{\tilde{P}}^{\mathbf{base}})$$

pour la seconde étape, où $P^{\mathbf{final}}$ est la restauration finale de $\tilde{P}$. Ces formules utilisent la matrice de covariance du bruit. Si le bruit initial est blanc gaussien, cette matrice est $\sigma^2\mathbf{I}$. Cela restera vrai pour l'échelle la plus grossière, mais aux échelles plus fines, il faut prendre en compte que les fréquences les plus basses ont été éliminées par le procédé multi-échelle. Donc le bruit n'est plus blanc.

3.2 La matrice de covariance du bruit

L'image bruitée $\tilde{u}$ se décompose en $\tilde{u} = u + n$ où u est l'image idéale et $n \sim \mathcal{N}(0, \sigma^2)$ un bruit gaussien. L'image bruitée est décomposée en deux parties : la haute fréquence et la basse fréquence, avec deux filtres complémentaires $\mathbf{H}$ et $\mathbf{L}$

$$\tilde{u} = \mathbf{H}(\tilde{u}) + \mathbf{L}(\tilde{u}) = \mathbf{H}(u) + \mathbf{H}(n) + \mathbf{L}(u) + \mathbf{L}(n).$$

Sous l'hypothèse que l'algorithme multi-échelle a déjà éliminé les basses fréquences du bruit, on a

$$\tilde{u} = \mathbf{H}(u) + \mathbf{H}(n) + \mathbf{L}(u) = u + \mathbf{H}(n).$$

Donc, pour débruiter $\tilde{u}$ par NL-Bayes, on doit calculer la matrice de covarince $\mathbf{C}$ des hautes fréquences du bruit. Pour être simple, faisons le calcul en dimension 1. On note $*$ le produit de convolution discret. On a

$$\mathbf{C}\,(\mathbf{H} * n)\,(x,y) = \mathbb{E}\left[(\mathbf{H} * n)\,(x)\,(\mathbf{H} * n)\,(y)\right]$$

$$= \mathbb{E}\left[\left(\sum_{i=-L}^{L} \mathbf{H}(i)n(x-i)\right)\left(\sum_{j=-L}^{L} \mathbf{H}(j)n(y-j)\right)\right]$$

où (x,y) sont des pixels distincts de l'image 1D, $2L+1$ est la longueur du support du filtre passe-haut $\mathbf{H}$. Comme $\mathbf{C}(n) = \sigma^2\mathbf{I}$, on obtient

$$\begin{aligned}
\mathbf{C}\,(\mathbf{H} * n)\,(x,y) &= \sigma^2 \sum_{x-i=y-j} \mathbf{H}(i)\mathbf{H}(j) \\
&= \sigma^2 \sum_{j=-L}^{L} \mathbf{H}\,(x-y+j)\,\mathbf{H}\,(j) \\
&= \sigma^2 \mathbf{H} * \mathbf{H}(y-x),
\end{aligned}$$

car $\mathbf{H}(-x) = \mathbf{H}(x)$. Le sous-échantillonnage a utilisé le filtre en boîte *centré* (i.e. $\mathbf{H} = \mathbf{B}$) de coefficients $\frac{1}{4}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ sur les $(-\frac{1}{2},-\frac{1}{2})$, $(\frac{1}{2},\frac{1}{2})$, $(\frac{1}{2},-\frac{1}{2})$, $(-\frac{1}{2},\frac{1}{2})$. Donc nous avons

$$\begin{aligned}
(n * \mathbf{B})(x,y) &:= \tfrac{1}{4}(n(x+\tfrac{1}{2},y+\tfrac{1}{2}) + n(x-\tfrac{1}{2},y+\tfrac{1}{2}) \\
&+ n(x+\tfrac{1}{2},y-\tfrac{1}{2}) + n(x-\tfrac{1}{2},y-\tfrac{1}{2})).
\end{aligned}$$

Nous calculons la matrice de covariance du bruit filtré par le filtre en boîte :

$$\mathbf{B} := \frac{1}{4}\left[\delta_{-\frac{1}{2},-\frac{1}{2}} + \delta_{-\frac{1}{2},\frac{1}{2}} + \delta_{\frac{1}{2},-\frac{1}{2}} + \delta_{-\frac{1}{2},-\frac{1}{2}}\right].$$

Donc

$$\begin{aligned}
&\mathbf{C}_s((n(x,y)n(x+1,y)) = \tfrac{1}{16}\mathbb{E}(n(x-\tfrac{1}{2},y-\tfrac{1}{2}) \\
&+n(x-\tfrac{1}{2},y+\tfrac{1}{2}) + n(x+\tfrac{1}{2},y+\tfrac{1}{2}) + n(x+\tfrac{1}{2},y+\tfrac{1}{2})) \\
&(n(x+\tfrac{1}{2},y-\tfrac{1}{2}) + n(x+\tfrac{1}{2},y+\tfrac{1}{2}) + n(x+\tfrac{3}{2},y-\tfrac{1}{2}) \\
&+n(x+\tfrac{3}{2},y-\tfrac{1}{2})) = \frac{1}{8}\sigma^2.
\end{aligned}$$

TABLE 1 – Comparaison entre NL-Bayes à une échelle et à quatre échelles. Les PSNR et RMSE sont le résultat moyen sur huit images.

	Mono-échelle		Multi-échelle	
σ	PSNR	RMSE	PSNR	RMSE
2	**46.13**	**1.29**	45.92	1.32
5	**40.88**	**2.43**	40.58	2.52
10	**36.56**	**4.04**	36.40	4.15
20	**32.63**	**6.39**	32.12	6.78
30	**30.11**	**8.46**	29.82	8.76
40	**28.32**	**10.30**	28.08	10.61
60	**27.35**	**11.81**	26.79	12.47
80	**26.33**	**13.40**	25.90	13.96
100	**25.11**	**15.18**	24.89	15.63

Par des calculs semblables on a

$$\mathbf{C}_s((n(x,y)n(x,y\pm 1)) = \frac{1}{8}\sigma^2 = \mathbf{C}_s((n(x,y)n(x\pm 1,y)))$$

$$\mathbf{C}_s((n(x,y)n(x,y)\pm(1,1))) = \frac{1}{16}\sigma^2$$

$$\mathbf{C}_s((n(x,y)n(x,y)\pm(1,-1))) = \frac{1}{16}\sigma^2$$

et

$$\mathbf{C}_s(n(x,y)n(x,y)) = \frac{1}{4}\sigma^2,$$

toutes les autres covariances sont nulles. Comme nous voulons la matrice de covariance des hautes fréquences du bruit après avoir enlevé ses basses fréquences aux échelles plus basses, on a $\mathbf{C}_n = \sigma^2\mathbf{I} - \mathbf{C}_s$.

L'algorithme multi-échelle est synthétisé dans le pseudo-code Algorithme 1.

Algorithm 1 Algorithme de débruitage multi-échelle

Input : image bruitée $\tilde{u}_0$
Input : nombre d'échelles N_s
Output : image débruitée $\hat{u}_0$

Partie 1 : calcul de la pyramide d'échelle, gardant les détails pour chaque échelle
for Chaque échelle $s = 1$ à N_s **do**
  Soit $(\tilde{u}_{s-1,k})_{k=1,\dots,4^{s-1}}$ l'ensemble des sous-images bruitées venant de l'échelle précédente :
  for $k = 1$ à 4^{s-1} **do**
    Sous-échantillonner $\tilde{u}_{s-1,k}$ en quatre sous-images : $\left(\tilde{u}_{s,4(k-1)+i}\right)_{i=1,\dots,4}$ (voir la section 2) ;
    Enregistrer les détails pour cette échelle :

$$\tilde{u}^{diff}_{s-1,k} = \tilde{u}_{s-1,k} - \mathbf{B} * \tilde{u}_{s-1,k}.$$

  end for
  if $s = N_s$ **then**
    Enregistrer l'ensemble des sous-images bruitées : $(v_{N_s,k})_{k=1,\dots,4^{N_s}} = (\tilde{u}_{N_s,k})_{k=1,\dots,4^{N_s}}$.
  end if
end for

Partie 2 : Débruitage *de bas-en-haut*
for $s = N_s$ à 0 **do**
  Obtenir l'écart-type du bruit à cette échelle :

$$\sigma_s = \begin{cases} \dfrac{\sigma}{2^s} & \text{si } s = N_s, \\ \sigma\sqrt{\left(\dfrac{1}{2^s}\right)^2 - \left(\dfrac{1}{2^{s+1}}\right)^2} = \dfrac{\sigma}{2^s}\sqrt{\dfrac{3}{4}} & \text{sinon.} \end{cases}$$

  for $k = 1$ à 4^s **do**
    Débruiter $v_{s,k}$ par l'algorithme de débruitage, en utilisant la matrice de covariance du bruit calculée comme indiqué dans la section 3 et obtenir $\hat{u}_{s,k}$;
  end for
  if $s > 0$ **then**
    for $k = 1$ à 4^{s-1} **do**
      Sur-échantillonner $(\hat{u}_{s,4(k-1)+i})_{i=1,\dots,4}$ comme indiqué dans la section 2 et ajouter les détails sauvegardés $\tilde{u}^{diff}_{s-1,k}$ pour obtenir $v_{s-1,k}$.
    end for
  else
    $\hat{u}_0 = \hat{u}_{0,1}$
  end if
end for

4 Expériences

4.1 Comparaison de l'algorithme multi-échelle avec le mono-échelle

Nous avons comparé l'algorithme mono-échelle et l'algorithme multi-échelle (quatre échelles) sur un ensemble d'images couleur sans bruit ($\sigma_{reel} << 1$), auquel un bruit gaussien blanc connu était ajouté. Les résultats pour un écart-type compris entre 2 et 100 sont dans la table 1.

4.2 Comparaison visuelle

Le PNSR et la RMSE ne reflètent pas nécessairement la gêne visuelle. Aussi nous présentons une comparaison visuelle plus probante dans la Figure 2.

4.3 Évolution de l'erreur en fonction du nombre d'échelles

Avec plus d'échelles on débruite mieux les zones plates, mais on perd quelques détails. Il faut donc trouver un compromis qui peut dépendre de l'image. Nous avons mené

FIGURE 2 – De haut en bas et de gauche à droite : image bruitée ($\sigma = 60$), résultat à une échelle, résultat à quatre échelles sur deux images.

des tests sur huit images (toujours sans bruit, avec bruit ajouté artificiellement). Pour une meilleure illustration de l'évolution du PSNR en fonction du nombre d'échelles N_s et de l'écart-type σ du bruit, la figure 3 donne $f(\text{PSNR}(s))$ en fonction de l'échelle courante s, où f est définie par

$$f(x_s) = \frac{x_s - x_m}{x_M}$$

où $x_m = \min(x_s)$ et $x_M = \max(x_s - x_m)$.

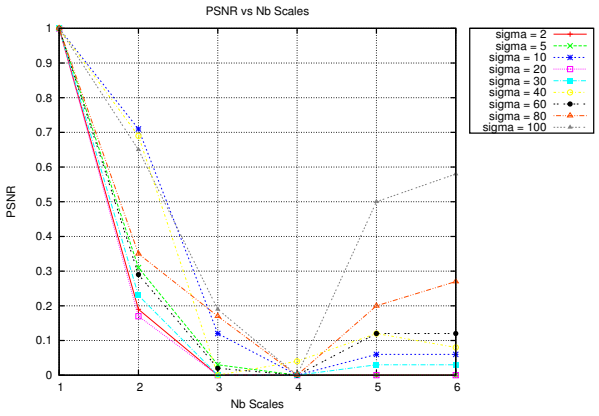

FIGURE 3 – PSNR *vs* nombre d'échelles.

5 Conclusion

L'algorithme multi-échelle présenté est générique, seule l'estimation de la matrice de covariance étant spécifique de NL-Bayes. Les résultats multi-échelles sont légérement inférieurs en PSNR qu'avec une seule échelle, mais c'est le prix à payer pour une indéniable amélioration de la qualité visuelle, particulièrement dans les zones homogènes de l'image. Ceci est particulièrement sensible dans les forts bruits.

Références

[1] K. Dabov, A. Foi, V. Katkovnik et K. Egiazarian. *Image denoising by sparse 3D transform-domain collaborative filtering*. IEEE Transactions On Image Processing, 2007.

[2] A. Buades, B. Coll et J.M. Morel. *A non local algorithm for image denoising*. IEEE Computer Vision and Pattern Recognition, 2005.

[3] J. Mairal, M. Elad et G. Sapiro. *Sparse representation for color image restoration*. IEEE Transactions On Image Processing, 2008.

[4] J. Mairal, G. Sapiro, et M. Elad. *Learning multiscale sparse representations for image and video restoration*. SIAM Multiscale Modeling and Simulation, 2008.

[5] L.P. Yaroslavsky, K.O. Egiazarian et J.T. Astola. *Transform domain image restoration methods : review, comparison, and interpretation*. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 2001.

[6] L.P. Yaroslavsky. *Local adaptive image restoration and enhancement with the use of DFT and DCT in a running window*. Proceedings of SPIE, 1996.

[7] J.L. Starck, E.J. Candès et D.L. Donoho. *The Curvelet Transform for Image Denoising*. IEEE Transactions On Image Processing, 2002.

[8] L.I. Rudin, S. Osher et E. Fatemi. *Nonlinear total variation based noise removal algorithms*. Physica D vol. 60 pages 259-268, 1992.

[9] M. Lebrun, A. Buades et J.M. Morel. *Implementation of the "non-local Bayes" image denoising algorithm*. IPOL, Image Processing On Line (http ://www.ipol.im), 2013.