A priori Plug-and-Play profond pour la restauration de vidéos

Antoine MONOD[1, 2], Julie DELON[1], Matias TASSANO[2]

[1]Laboratoire MAP5
Université Paris Cité, 45 rue des Saint-Pères, 75006 France

[2]GoPro Technology France SAS
115 rue Rouget de Lisle, 92130 Issy-les-Moulineaux, France

antoine.monod@u-paris.fr, julie.delon@u-paris.fr
tasso.matias@gmail.com

Résumé – Cet article présente une méthode de restauration de vidéos numériques via une approche Plug-and-Play (PnP). La méthode consiste à utiliser un réseau convolutionnel profond de débruitage pour remplacer l'opérateur proximal de l'a priori dans un schéma d'optimisation alterné. Elle permet de réutiliser un réseau uniquement entrainé pour du débruitage pour d'autres tâches de restauration comme l'interpolation ou le déflouage. Nos expériences montrent l'intérêt d'utiliser un réseau spécifiquement conçu pour le débruitage vidéo qui, avec la même formulation PnP, permet d'atteindre de meilleures performances de restauration et une meilleure stabilité temporelle qu'un réseau mono-image aux performances de débruitage similaires.

Abstract – This paper presents a method for restoring digital videos via a Plug-and-Play (PnP) approach. The method consists in using a deep convolutional denoising network in place of the proximal operator of the prior in an alternating optimization scheme. This way, a network trained once for denoising can be repurposed for other restoration tasks such as interpolation or deconvolution. Our experiments show the benefit of using a network specifically designed for video denoising, as it reaches better restoration performance and better temporal stability than a single image denoising network with similar denoising performance using the same PnP formulation.

1 Introduction

La majorité des problèmes inverses rencontrés en imagerie ou vidéo numérique peuvent s'écrire sous la forme

$$y = \mathcal{A}(x) + n \tag{1}$$

où y est l'observation dégradée, x est l'image ou vidéo inconnue à retrouver, $\mathcal{A}$ est un opérateur de dégradation, souvent linéaire, et n est un bruit de loi connue. Ces problèmes inverses sont souvent mal posés ou au moins mal conditionnés. Afin de construire des estimateurs de x fiables et robustes à partir de l'observation y, il est classique de faire appel à un formalisme bayésien, dans lequel on fait l'hypothèse que l'inconnue x suit une loi (dite "a priori") de densité $p(x)$. En la combinant avec $p(y|x)$, la vraisemblance de y sachant x (donnée par le modèle de dégradation (1)), on obtient la densité a posteriori $p(x|y)$, dont on cherche généralement à calculer le maximum a posteriori (MAP) :

$$\hat{x} = \underset{x}{\operatorname{argmax}}\, p(x|y) = \underset{x}{\operatorname{argmax}} \log p(y|x) + \log p(x). \tag{2}$$

Dans le cas où le bruit est Gaussien i.i.d. de variance σ_n^2, le problème précédent se réécrit sous forme variationnelle

$$\hat{x} = \underset{x}{\operatorname{argmin}} \frac{1}{2\sigma_n^2} \|y - \mathcal{A}(x)\|_2^2 + \alpha \mathcal{R}(x), \tag{3}$$

où la log-vraisemblance de l'observation est $-\frac{1}{2\sigma_n^2}\|y-\mathcal{A}(x)\|_2^2$ (aussi appelée attache aux données), et le log-prior sur l'inconnue est le terme $-\alpha\mathcal{R}(x)$.

La restauration bayésienne en imagerie ou vidéo numérique a longtemps reposé sur des a priori explicites (comme la variation totale), exprimant des hypothèses de régularité sur x soit dans l'espace de l'image soit dans des espaces transformés (transformations en ondelettes, espaces de patchs, etc). Lorsque $\mathcal{R}$ est connu et convexe, il existe de nombreux schémas numériques très efficaces pour trouver les solutions de (3) [2].

Depuis quelques années, ces méthodes traditionnelles de restauration se sont vues dépassées numériquement par les réseaux de neurones profonds. Les réseaux dits *end-to-end* (comme [15, 11] pour le débruitage) sont entraînés uniquement à partir de couples (x_i, y_i) vérifiant le modèle de dégradation (1), mais n'utilisent la connaissance de ce modèle qu'à travers ces bases d'exemples. L'entraînement de ces réseaux requiert de larges volumes de données, et des coûts en ressources de calcul très importants. Par ailleurs, un réseau entraîné pour un modèle donné de dégradation doit être réentraîné lorsque la dégradation (ou ses paramètres) change.

Les méthodes dites *Plug-and-Play* (PnP) tentent de combiner les avantages des deux mondes précédents. Elles associent une vraisemblance définie explicitement selon le modèle direct (1), et un a priori $\mathcal{R}$ défini implicitement par un algorithme de débruitage. Cette combinaison est faite algorithmiquement, typiquement au sein d'un schéma d'optimisation alterné où l'opérateur proximal de $\mathcal{R}$ (ou parfois son gradient) est rem-

placé par le débruiteur. De multiples formulations PnP d'algorithmes d'optimisation ont été proposées, comme ADMM [12], SGD [4], PDHG [6] ou encore HQS [14].

Les méthodes PnP les plus récentes utilisent généralement comme débruiteurs des réseaux de neurones profonds [6, 10, 4, 14]. Alors que la taille des réseaux convolutionnels et les moyens matériels utilisés pour entraîner ceux-ci tendent à continuer d'augmenter, ces méthodes permettent de réutiliser un réseau de débruitage entrainé une seule fois, pour différentes tâches de restauration sans avoir à le réentrainer. Elles permettent également de répondre à des contraintes de mémoire, par exemple dans un système embarqué (comme une caméra d'action ou un téléphone mobile) où les poids d'un seul réseau peuvent être stockés pour plusieurs cas d'utilisation.

A notre connaissance, cet article est le premier décrivant l'utilisation de réseaux de neurones au sein d'une approche PnP pour restaurer une vidéo numérique à partir d'une observation elle aussi sous forme de vidéo. L'article [13] utilise FastDVDnet [11] comme débruiteur vidéo dans un schéma PnP, mais pour résoudre un problème bien spécifique de Snapshot Compressive Imaging, où l'observation est une seule image. [3] utilise l'algorithme PnP-ADMM pour de la super-résolution vidéo, mais fait appel à un débruiteur d'image par patchs.

2 Plug-and-Play vidéo

PnP-ADMM. Commençons par rappeler le principe de l'algorithme *Alternating Directions Method of Multipliers* [1] (ou ADMM). Supposons que l'on souhaite minimiser (3). On commence par définir le Lagrangien augmenté

$$L_\epsilon(x,z,v) = \frac{1}{\alpha}\underbrace{\frac{1}{2\sigma_n^2}\|y-\mathcal{A}(x)\|_2^2}_{F(x,y)} + \mathcal{R}(z) + \frac{1}{2\epsilon}\|x-z\|_2^2 + v^T(x-z) \tag{4}$$

que l'on souhaite minimiser en (x,z) et maximiser en v. Lorsque $\epsilon \to 0$, les solutions (x,z,v) vérifient $x-z \to 0$, et nous donnent donc des solutions de (3). La formulation précédente s'optimise de manière alternée par un schéma du type (en posant $u = \epsilon v$)

$$\begin{aligned} x_{k+1} &\leftarrow \underset{x}{\operatorname{argmin}} L_\epsilon(x, z_k, u_k/\epsilon) = \operatorname{prox}_{\frac{\epsilon}{\alpha}F(.,y)}(z_k - u_k) \\ z_{k+1} &\leftarrow \underset{z}{\operatorname{argmin}} L_\epsilon(x_{k+1}, z, u_k/\epsilon) = \operatorname{prox}_{\epsilon\mathcal{R}}(x_{k+1} + u_k) \\ u_{k+1} &\leftarrow u_k + x_{k+1} - z_{k+1}. \end{aligned} \tag{5}$$

Supposons qu'on sache construire un débruiteur $\mathcal{D}_\epsilon$ qui s'écrit comme l'estimateur MAP pour un problème de débruitage (pour un bruit Gaussien i.i.d de variance ϵ) pour le log-prior $\mathcal{R}$. Par définition du MAP, on a exactement $\mathcal{D}_\epsilon = \operatorname{prox}_{\epsilon\mathcal{R}}$, on peut donc "plugger" directement ce débruiteur dans le schéma d'optimisation précédent. En pratique, ces schémas PnP sont utilisés avec succès même avec des débruiteurs ne vérifiant pas cette propriété, et l'étude de leur convergence est un champ de recherche très actif [4].

Cas de la vidéo. Dans cet article, on propose d'utiliser le schéma précédent directement sur toute la vidéo x. Cela permet de tenir compte de cas où l'opérateur $\mathcal{A}$ ne s'écrit pas de manière séparable sur toutes les images (cas d'un flou temporel par exemple), mais aussi d'utiliser des débruiteurs spécifiques à la vidéo (comme le récent [11]).

Remarquons que si l'opérateur $\mathcal{A}$ est appliqué de manière séparable sur chaque image de x et le débruiteur utilisé est un débruiteur mono-image, un schéma itératif ADMM sur l'ensemble de la vidéo et une succession de schéma itératifs ADMM sur chaque image de la vidéo deviennent équivalents. Ce n'est plus le cas lorsque $\mathcal{A}$ est appliqué sur la vidéo de manière non-séparable.

3 Expériences

3.1 Débruiteurs utilisés

Nous nous intéressons ici à DRUNet [14] et FastDVDnet [11], deux réseaux de l'état de l'art en débruitage Gaussien. DRUNet est un réseau mono-image conçu spécifiquement pour être intégré au sein d'une approche PnP pour la restauration d'image. FastDVDnet est un réseau conçu pour le débruitage vidéo : il utilise l'information supplémentaire des images voisines pour fournir une estimation débruitée de meilleure qualité et plus stable temporellement, sans recalage explicite des images. Les deux réseaux font appel à des autoencodeurs de type U-Net [9] : FastDVDnet combine deux petits blocs U-Net avec connexions résiduelles et batch-norm dans une architecture en cascade ; DRUNet comporte un seul bloc plus profond et avec davantage d'étapes de sous-échantillonnage, et remplace les couches de convolution standard par des ResBlocks [5]. Nous récupérons les réseaux tels qu'ils ont été fournis par les auteurs des publications originales, sans les réentrainer.

Avant d'étudier la performance de ces réseaux en restauration vidéo PnP, il est intéressant d'évaluer leur performance en débruitage. Il semble en effet raisonnable de penser que la performance de débruitage du réseau a un impact sur la performance maximale atteignable en restauration PnP [14]. DRUNet débruite toutes les images du film séparément. Pour produire une version débruitée de l'image à l'instant t, FastDVDnet utilise les images aux instants t_{-2}, t_{-1}, t, t_{+1} et t_{+2}. Pour garder un film de taille identique en sortie, la vidéo bruitée en entrée est augmentée de 4 frames par miroir :

$$y = \{I_0, \dots, I_{n-1}\} \to y' = \{I_2, I_1, I_0, \dots, I_{n-1}, I_{n-2}, I_{n-3}\}.$$

FastDVDnet est entraîné avec des niveaux de bruit $\sigma \in [5, 55]/255$, et DRUNet avec $\sigma \in [0, 50]/255$.

Nous évaluons la performance en débruitage vidéo des deux réseaux sur l'ensemble de test du dataset DAVIS-2017 [8], dans sa version 480p. Ce dataset comporte 30 séquences vidéo de longueur variable. Ni DRUNet ni FastDVDnet n'ont vu ces séquences en entraînement. Les résultats sont consultables dans le tableau 1. Pour un temps d'exécution et un nombre de pa-

TABLE 1 – Débruitage : PSNR/SSIM sur DAVIS-2017-test-480p [8] (8 CPU AMD 7F52 / 1 NVIDIA Tesla T4 / 16GB RAM)

Réseau	$\sigma = 10/255$	$\sigma = 25/255$	$\sigma = 50/255$	$\sigma = 100/255$	# params	tps. exec. (s/image)
bruité	28.13/0.634	20.17/0.314	14.15/0.146	08.13/0.053		
DRUNet	38.90/0.967	34.40/0.921	31.27/0.861	**28.32/0.781**	32.6410M	0.48
FastDVDnet	**39.20/0.969**	**35.05/0.931**	**31.97/0.878**	26.84/0.655	2.4791M	0.23

ramètres moindres, FastDVDnet est légèrement plus performant. DRUNet est toutefois le réseau dont la performance se dégrade le moins à $\sigma = 100/255$, niveau de bruit non vu à l'entraînement pour les deux réseaux. DRUNet est un réseau dont les couches de convolution sont sans biais, et qui n'a pas de batch-norm ; ces résultats sont cohérents avec ceux de [7], où il a été montré que des réseaux sans biais généralisent mieux à des niveaux de bruits en dehors de l'intervalle d'entrainement.

3.2 Déconvolution et interpolation

Nous nous intéressons désormais à la performance de notre méthode PnP-ADMM en utilisant les réseaux de la section 3.1, pour deux problèmes de restauration de vidéo : le déflouage par un noyau fixe, et l'interpolation. Ces résultats ont pour but de présenter la viabilité de notre méthode, et d'évaluer l'impact du débruiteur choisi sur le résultat final ; des cas d'utilisation plus réalistes, accompagnés de comparaisons à des méthodes spécifiquement dédiées à ces problèmes, seront l'objet d'une future publication.

Dans les expériences qui suivent, nous fixons le niveau du débruiteur à $\epsilon = (50/255)^2$ (en fournissant à DRUNet et FastDVDnet une carte de bruit constante de valeur ϵ lors du débruitage), et on choisit $\alpha = 1$. Ces valeurs, au même titre que le nombre d'itérations maximales du schéma ADMM, ont été trouvées empiriquement ; à terme, il serait intéressant de développer des stratégies pour en estimer des versions optimales selon le problème et le débruiteur utilisé. Nous limitons également chaque séquence à ses 30 premières images pour harmoniser les temps de calcul et la complexité par vidéo. Le code source de ces expériences est disponible sur GitHub [1].

3.2.1 Déflouage spatial

Notre premier cas d'utilisation est le suivant : chaque image de la vidéo est convoluée par un noyau fixe (uniforme ou Gaussien) et un bruit d'écart type σ_n est ajouté au résultat. Ce type de problème inverse peut être résolu image par image, mais est tout de même rencontré en vidéo, par exemple pour augmenter la netteté d'une vidéo à partir de la fonction d'étalement du point du système optique utilisé. Les résultats quantitatifs de notre méthode PnP-ADMM pour des noyaux uniforme et Gaussien sont présentés dans le tableau 2. Nous avons constaté un PSNR de FastDVDnet très légèrement supérieur à DRUNet dans la quasi totalité des vidéo observées, et cette différence est quasi-constante image par image au sein d'une vidéo.

1. https://github.com/amonod/grestsi_pnp_video

TABLE 2 – Déflouage : PSNR/SSIM après 20 itérations de PnP-ADMM sur DAVIS-2017-test-480p [8] (chaque vidéo limitée aux 30 premières frames)

Noyau / Réseau	Uniforme 9×9, $\sigma_n = 5/255$	Gaussien ($\sigma = 5$) 11×11, $\sigma_n = 1/255$
flou	24.41/0.554	12.66/0.533
DRUNet	30.89/0.838	31.42/0.846
FastDVDnet	**31.09/0.846**	**31.79/0.865**

3.2.2 Interpolation de pixels manquants

Nous étudions un second cas d'utilisation de notre méthode de restauration vidéo Plug-and-Play : l'interpolation, qui consiste à estimer les valeurs de pixels masqués ou manquants. Dans notre cas, nous masquons une proportion ρ des pixels de la vidéo selon un motif spatio-temporel aléatoire. Ce problème peut être apparenté à un cas particulier de *compressed sensing*. Les résultats de notre méthode pour $\rho = 0.9$ sont présentés dans le tableau 3. Un exemple visuel est proposé en Figure 1. Dans cette expérience, étant donné que la quantité de pixels manquants et leur position changent d'une frame à l'autre, le schéma basé sur DRUNet (équivalent ici à traîter chaque image séparément) est clairement en retrait. Le schéma basé sur FastDVDnet tire partie de son étape de débruitage utilisant les frames voisines pour fournir une meilleure interpolation.

TABLE 3 – Interpolation : PSNR/SSIM après 500 itérations de PnP-ADMM sur DAVIS-2017-test-480p [8] (crops 256×256, chaque vidéo limitée aux 30 premières frames)

Réseau	$\rho = 0.9$ $\sigma_n = 1/255$
masqué	7.14/0.040
DRUNet	27.54/0.818
FastDVDnet	**31.25/0.885**

Perspectives

D'après nos expériences, plus le problème étudié est difficile, plus le nombre d'itérations nécessaires est grand. La vitesse de convergence de la méthode et la performance finale dépendent aussi du contenu de l'inconnue et du choix de ϵ et α, et cette dépendance varie selon le débruiteur utilisé. Il est nécessaire d'étudier davantage ces phénomènes, et de définir des pratiques permettant d'assurer la stabilité des résultats ; des

FIGURE 1 – Interpolation ($\rho = 0.9$) après 500 itérations de PnP-ADMM. 1er rang : inconnue. 2e rang : observation (PSNR : 7.51 dB). 3e rang : résultat de DRUNet (25.93 dB). 4e rang : résultat de FastDVDnet (27.30 dB). Bien que l'écart en dB sur l'ensemble de la vidéo ne soit pas extrèmement élevé, FastDVDnet produit de meilleurs résultats, temporellement plus stables.

stratégies d'évolution de ϵ et α au cours des itérations sont par exemple envisageables. Une fois ces pratiques en place, cette méthode peut être appliquée à des problèmes plus complexes ou plus réalistes de restauration de vidéo, comme le dématriçage, la déconvolution temporelle ou encore la super-résolution spatiale et/ou temporelle.

Références

[1] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, AND J. ECKSTEIN, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, Foundations and Trends® in Machine Learning, 3 (2011).

[2] A. CHAMBOLLE AND T. POCK, *An introduction to continuous optimization for imaging*, Acta Numerica, 25 (2016).

[3] V. KHORASANI GHASSAB AND N. BOUGUILA, *Plug-and-Play video reconstruction using sparse 3D transform-domain block matching*, Machine Vision and Applications, 32 (2021).

[4] R. LAUMONT, V. DE BORTOLI, A. ALMANSA, J. DELON, A. DURMUS, AND M. PEREYRA, *On Maximum-a-Posteriori estimation with Plug & Play priors and stochastic gradient descent*. 2021.

[5] B. LIM, S. SON, H. KIM, S. NAH, AND K. M. LEE, *Enhanced Deep Residual Networks for Single Image Super-Resolution*, in CVPRW, 2017.

[6] T. MEINHARDT, M. MOELLER, C. HAZIRBAS, AND D. CREMERS, *Learning Proximal Operators : Using Denoising Networks for Regularizing Inverse Imaging Problems*, in ICCV, 2017.

[7] S. MOHAN, Z. KADKHODAIE, E. P. SIMONCELLI, AND C. FERNANDEZ-GRANDA, *Robust and interpretable blind image denoising via bias-free convolutional neural networks*, in ICLR, 2020.

[8] J. PONT-TUSET, F. PERAZZI, S. CAELLES, P. ARBELÁEZ, A. SORKINE-HORNUNG, AND L. VAN GOOL, *The 2017 davis challenge on video object segmentation*, arXiv :1704.00675, (2017).

[9] O. RONNEBERGER, P. FISCHER, AND T. BROX, *U-net : Convolutional networks for biomedical image segmentation*, in MICCAI, 2015.

[10] E. RYU, J. LIU, S. WANG, X. CHEN, Z. WANG, AND W. YIN, *Plug-and-Play Methods Provably Converge with Properly Trained Denoisers*, in PMLR, 2019.

[11] M. TASSANO, J. DELON, AND T. VEIT, *Fastdvdnet : Towards real-time deep video denoising without flow estimation*, in CVPR, 2020.

[12] S. V. VENKATAKRISHNAN, C. A. BOUMAN, AND B. WOHLBERG, *Plug-and-Play priors for model based reconstruction*, in GlobalSIP, 2013.

[13] X. YUAN, Y. LIU, J. SUO, F. DURAND, AND Q. DAI, *Plug-and-play algorithms for video snapshot compressive imaging*, TPAMI, (2021).

[14] K. ZHANG, Y. LI, W. ZUO, L. ZHANG, L. VAN GOOL, AND R. TIMOFTE, *Plug-and-Play Image Restoration with Deep Denoiser Prior*, TPAMI, (2021).

[15] K. ZHANG, W. ZUO, Y. CHEN, D. MENG, AND L. ZHANG, *Beyond a Gaussian Denoiser : Residual Learning of Deep CNN for Image Denoising*, TIP, 26 (2017).