

Synthèse de *Rain Streak Removal Using Layer Priors*, Yu Li, Robby T. Tan, Xiaojie Guo, JiangBo Lu et Michael S. Brown

La suppression de la pluie qui dégrade les images avec des flous, des reflets et des effets de diffusion, permet d'améliorer les résultats des algorithmes de vision par ordinateur en fournissant en entrée des images avec une meilleure visibilité.

L'article *Rain Streak Removal Using Layer Priors* de Yu Li, Robby T. Tan, Xiaojie Guo, JiangBo Lu et Michael S. Brown, publié en 2016, propose une nouvelle méthode pour retirer les traînées de pluie d'une image en modélisant la pluie et le fond par deux modèles de mélange gaussien.

Nous nous proposons dans ce qui suit de placer l'article dans le contexte de la littérature lors de sa publication, de détailler l'approche d'optimisation mise en œuvre et de résumer les points forts de l'article et ses quelques points faibles qui nous sont apparus. Enfin, nous présentons en quelques lignes notre code et ses résultats.

I) Contexte, état de l'art et contributions de *Rain Streak Removal Using Layer Priors*

Dans la littérature, deux grandes classes de méthodes existent pour supprimer les traînées de pluie : celles qui prennent en entrée une vidéo et celles qui prennent en entrée une seule image. Le problème de la suppression de pluie pour une image à une dimension (luminescence ou niveau de gris) est formulé comme un problème de superposition : étant donné une image $O \in \mathbb{R}^{M \times N}$, trouver $B \in \mathbb{R}^{M \times N}$ et $R \in \mathbb{R}^{M \times N}$ modélisant respectivement l'image sans la pluie et l'image de la pluie, telles que $O = B + R$, $B > 0$ et $R > 0$. La dimension de la sortie est *a priori* deux fois plus grande que celle de l'entrée. Une solution pour dépasser ce problème est de prendre plusieurs images en entrée avec un fond identique mais une pluie différente. Il s'agit des méthodes à vidéo. Néanmoins, celles-ci sont limitées par les fonds qui ne sont jamais tout à fait identiques d'une image à une autre. L'autre classe de méthode se contente d'une seule image et ajoute des hypothèses sur la pluie pour parvenir à trouver B et R , bien que, combinées, celles-ci aient une dimension plus grande que O .

Plusieurs hypothèses ont été essayées sur la pluie dans la littérature. [1] propose de modéliser la pluie comme les hautes fréquences de l'image et le fond comme les basses fréquences. Le défaut de cette méthode est que le fond obtenu est flou. [2] a proposé une méthode de minimisation de vraisemblance mais avec des résultats limités. [3] a modélisé la forme elliptique et verticale de la pluie. Néanmoins, il ne parvient pas à retenir la diversité des formes des traînées de pluie.

Rain Streak Removal Using Layer Priors se place dans le cadre des méthodes à une seule image. Il retient l'approche d'optimisation et de modélisation de la pluie de [2] et [3]. Par ailleurs, ils reprennent l'approche de [4] qui modélise les patches d'images avec un modèle de mélange gaussien. *Rain Streak Removal Using Layer Priors* modélise le fond et la pluie par deux modèles de mélange gaussien, celui de la pluie étant réappris pour chaque image. Ainsi, la méthode se généralise davantage que [2].

Finalement, les résultats de *Rain Streak Removal Using Layer Priors* se plaçaient à la tête de l'état de l'art au moment de la publication, sans pour autant avoir complètement résolu le problème car des restes de pluie subsistent dans les images de sortie, en particulier si la pluie était initialement dense en entrée.

II) Optimisation dans *Rain Streak Removal Using Layer Priors* pour retirer les traînées de pluie

Rain Streak Removal Using Layer Priors optimise deux modèles pour supprimer la pluie d'une image. Le premier est le mélange gaussien modélisant les traînées de pluie d'une image. Le deuxième est la minimisation de l'erreur de reconstruction $\|O - B - R\|_F^2$ soumise à des priors $\phi(B)$ pour le fond et $\varphi(R)$ pour la pluie, sous-tendus notamment par des mélanges gaussiens.

Pour le fond, $\phi(B) = -\gamma \sum_{B's \text{ patches } p} \log G_B(p) + \alpha \|\nabla B\|_1$ où $G_B(p)$ est la vraisemblance du patch p sous le mélange gaussien G_B . Ce modèle gaussien est repris de [29]. $\phi(B)$ s'interprète comme le fait que chaque patch de l'image doit être proche des patches de fond modélisés par G_B et que l'image sans pluie doit présenter peu de brusques fluctuations d'intensité.

Pour la pluie, $\varphi(R) = \beta \|R\|_F^2 - \gamma \sum_{R's \text{ patches } p} \log G_R(p)$ permet de s'assurer que la pluie est proche de la forme modélisée par G_R . G_R est appris sur les patches d'une sous-fenêtre de l'image. Cette sous-fenêtre de taille arbitraire est choisie pour présenter un fond uniforme avec des traînées de pluie. Cela se fait en choisissant celle de plus faible variance. Le terme $\beta \|R\|_F^2$ représente le fait que la pluie n'est qu'une faible partie de l'image.

Le problème est finalement

$$\min_{B>0, R>0} \|O - B - R\|_F^2 + \beta \|R\|_F^2 + \alpha \|\nabla B\|_1 - \gamma \sum_{B's \text{ patches } p} \log G_B(p) - \gamma \sum_{R's \text{ patches } p} \log G_R(p).$$

Celui-ci est résolu avec des méthodes classiques de la littérature. Il est approximé par la méthode *half quadratic splitting* : on minimise

$$\begin{aligned}
& \|O - B - R\|_F^2 + \beta \|R\|_F^2 + \alpha \|H\|_1 - \gamma \sum_{g_B's \text{ patches } p} \log G_B(p) \\
& - \gamma \sum_{g_R's \text{ patches } p} \log G_R(p) + \omega \|H - \nabla B\|_F^2 + \omega \sum \|patch_{B_i} - g_{B_i}\|_F^2 \\
& + \omega \sum \|patch_{B_i} - g_{B_i}\|_F^2
\end{aligned}$$

successivement pour H , B et R , g_B et g_R en augmentant ω au fur et à mesure pour que H et ∇B coïncident ainsi que g_B avec les patches de B et g_R avec les patches de R . L'optimisation pour H est un problème LASSO avec une solution analytique calculable rapidement et précisément. L'optimisation pour B et R se fait avec L-BFGS et celle de g_B et g_R en suivant la méthode de [29] qui approxime le problème en ne conservant que la composante la plus probable du mélange gaussien.

III) Apports et faiblesses de *Rain Streak Removal Using Layer Priors*

Finalement, Yu Li, Robby T. Tan, Xiaojie Guo, JiangBo Lu et Michael S. Brown ont combinés plusieurs idées développées dans la littérature, notamment la modélisation des patches d'une image par un mélange gaussien, pour obtenir des résultats meilleurs que l'état de l'art.

La qualité de leur méthode réside dans la qualité de leurs résultats où les traînées de pluie sont mieux retirées que dans les autres méthodes de la littérature et dans la généralisation potentielle de leur méthode qui apprend la forme de la pluie de chaque image.

Néanmoins, en toute modestie, **nous relevons quelques faiblesses qui ne sont pas mentionnées dans l'article. La généralisation n'est pas si bonne qu'avancée car la taille de la sous-fenêtre choisie pour entraîner le mélange gaussien de la pluie est cruciale, varie d'une image à une autre et doit être choisie à la main.** Par ailleurs, malgré la parallélisation possible de certaines étapes de l'optimisation, **le traitement d'une image est lent et prend plusieurs heures sur notre ordinateur.**

Enfin, **nous regrettons le manque de transparence scientifique de l'article : les auteurs ne donnent pas la valeur des hyperparamètres de l'optimisation et ne mettent pas à disposition les images sur lesquelles ils ont travaillées.**

IV) Implémentation de la méthode de *Rain Streak Removal Using Layer Priors*

Nous avons reproduit la méthode de suppression des traînées de pluie et obtenu des résultats similaires que les auteurs, que ce soit sur une image utilisée par les auteurs ou sur notre propre image. Les résultats sont présentés dans la figure 1 en annexe. Pour notre image, il reste des traces de pluie. Cela correspond tout à fait aux résultats de l'article. Un exemple de performances similaires issu de l'article est donné figure 2. La méthode proposée dans l'article n'est pas parfaite.

Nous présentons succinctement dans ces dernières lignes la structure de notre code. Les résultats peuvent être reproduits en lançant le notebook main.ipynb. Cela prend six et sept heures par image. main.ipynb fait appel aux fichiers main ainsi qu'aux fichiers tools_preprocess_image.py, tools_gmm.py et tools_opti.py qui contiennent les fonctions relatives au prétraitement de l'image, à l'optimisation des modèles gaussiens et à l'optimisation de l'erreur de reconstruction soumises à des priors. Dans le dossier data se trouve les images sur lesquelles nous avons travaillées ainsi que les poids du modèle gaussien pour le fond.

Bibliographie

- [1] L.-W. Kang, C.-W. Lin, and Y.-H. Fu. Automatic single-image-based rain streaks removal via image decomposition. IEEE Trans. Image Processing, 21(4):1742–1755, 2012.
- [2] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In IEEE Int'l Conf. Computer Vision, 2013.
- [3] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim. Single image deraining using an adaptive nonlocal means filter. In IEEE Int'l Conf. Image Processing, 2013.
- [4] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In IEEE Int'l Conf. Computer Vision, 2011.

Annexe

Figure 1 : Résultats obtenus avec notre code pour une image issue de l'article (celle du dessus) et pour une image choisie par nos soins (image du dessous).

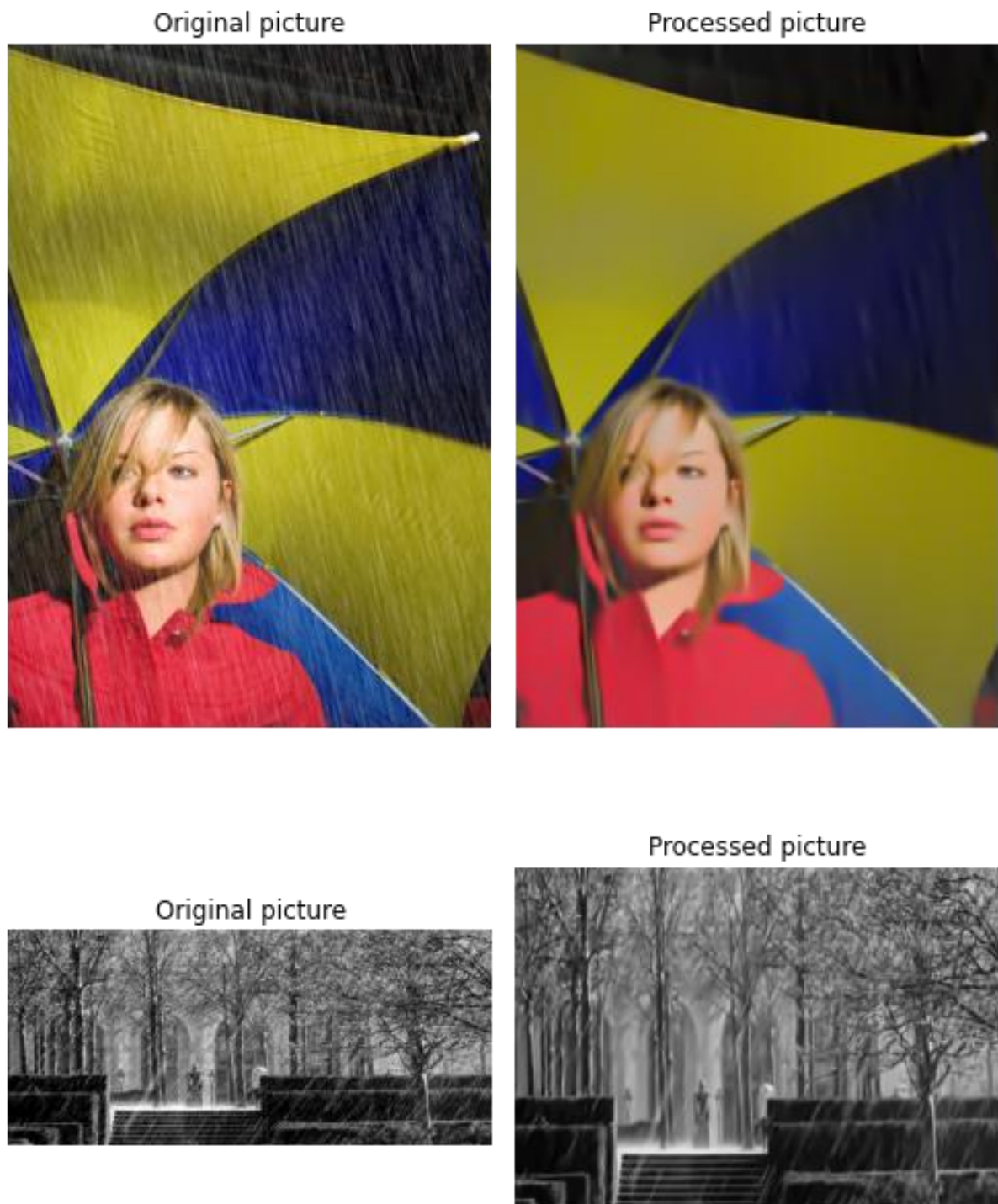


Figure 2 : Résultats obtenus par les auteurs avec des traces de pluie toujours visibles après traitement. L'image originale est celle de gauche et celle traitée est à droite.

