

IFT 6150 - Projet

Filtre guidé (Guided Image Filtering)

Bowen Peng

15 Décembre 2020

0 Introduction

Le filtre guidé, comme le filtre bilatéral, est un filtre de lissage qui préserve les contours. Ce filtre est dérivé d'un modèle linéaire local qui prend en entrée deux images, une première image qui est filtrée et une deuxième qui sert de guide. Tout comme le filtre bilatéral, ce filtre est très versatile et peut servir pour du débruitage, colorisation, raffinement des bordures (eg. pour la segmentation), super-résolution jointe, etc. De plus, contrairement au filtre bilatéral, le filtre guidé a une implémentation exacte qui roule en temps linéaire $O(N)$ et ne souffre pas du problème d'inversion de gradient aux bords des objets.

1 Filtres de lissage

Un fameux exemple de filtre de lissage qui préserve les contours est le filtre bilatéral. Le filtre est une moyenne pondérée des pixels en prenant compte de la distance spatiale euclidienne et une autre distance en intensité. La valeur de l'intensité peut provenir de l'image filtrée ou d'une autre image qui sert de guide. En utilisant cette moyenne pondérée qui prend en compte l'intensité, le filtre bilatéral ignore et ne moyenne pas les intensités trop différentes, et préserve mieux les contours comparé à un simple filtre gaussien.

Le filtre guidé est aussi un filtre de lissage qui préserve les contours, mais la définition de ce filtre n'est pas explicite comme le filtre bilatéral, il est plutôt défini comme un problème de minimisation. Les auteurs définissent le filtre comme ceci:

$$q_i = a_k I_i + b_k, \forall i \in \omega_k$$

où q est l'image de sortie, I l'image guide et ω_k une fenêtre centré au pixel k .

Pour trouver les paramètres a_k et b_k , on résoud un problème d'optimisation linéaire en minimisant la fonction de coût suivante:

$$E(a_k, b_k) = \sum_{i \in \omega_k} ((a_k I_i + b_k - p_i)^2 + (\epsilon a_k)^2)$$

Ceci va trouver les meilleurs paramètres (a, b) qui minimize la distance L2 entre l'image guide I et l'image à filtrer p , plus un facteur de régularisation ϵ . Ce facteur de régularisation réduit les valeurs grandes dans a , et fait en sorte que le filtre ignore les contours, et va lisser davantage l'image.

L'équation en haut est un modèle de "linear ridge regression" couramment utilisé en statistique et en apprentissage automatique. La solution de ce problème est donné par:

$$a_k = \frac{\frac{1}{|\omega|}(\sum_{i \in \omega_k} I_i p_i) - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon^2} = \frac{\text{Cov}(I_{\omega_k}, p_{\omega_k})}{\sigma_k^2 + \epsilon^2}$$

$$b_k = \bar{p}_k - a_k \mu_k$$

où μ_k et σ_k^2 est la moyenne et la variance de I dans la fenêtre ω_k , et \bar{p}_k est la moyenne de p dans ω_k .

Finalement, pour trouver l'image q:

$$q_i = \bar{a}_i I_i + \bar{b}_i$$

où \bar{a}_i et \bar{b}_i est la moyenne de a et b dans la fenêtre ω_k .

2 Implémentation

Pour l'exécution rapide du filtre guidé, nous avons besoin d'une méthode rapide pour calculer la moyenne, la variance et la covariance. Heureusement, les trois peuvent être calculé rapidement avec un filtre moyen rapide. Nous utilisons un filtre moyen à temps linéaire $O(N)$.

$$\begin{aligned} \text{Avg}(X) &= E[X] \\ \text{Var}(X) &= E[X^2] - E[X]^2 \\ \text{Cov}(X, Y) &= E[XY] - E[X]E[Y] \end{aligned}$$

Donc l'algorithme implémenté est:

$$\begin{aligned} \text{Var}(I) &= \text{Avg}(I^2) - \text{Avg}(I)^2 \\ \text{Cov}(I, p) &= \text{Avg}(Ip) - \text{Avg}(I) * \text{Avg}(p) \\ a &= \text{Cov}(I, p) / (\text{Var}(I) + \epsilon^2) \\ b &= \text{Avg}(p) - a * \text{Avg}(I) \\ q &= \text{Avg}(a) * I + \text{Avg}(b) \end{aligned}$$

où $\text{Avg}()$ est un filtre moyen rapide.

Dans le cas où la taille de l'image guide est différente de l'image à filtrer, nous rééchantillonnons l'image guide pour qu'elle corresponde à la taille de l'image à filtrer, puis ensuite nous calculons les statistiques et rééchantillonnons les moyennes de a et b pour qu'ils correspondent à la taille de l'image guide. Ceci nous permet de faire du super-résolution et débruitage jointe.

3 Images couleur

Pour une image p couleur, il suffit d'appliquer le filtre sur chaque couche RGB individuellement. Mais pour utiliser le filtre guidé avec plusieurs guides I (c'est le cas lorsque I est une image couleur RGB), la solution du problème devient un peu plus complexe:

$$\begin{aligned} a_k &= (\Sigma_k + 3\epsilon^2 U)^{-1} \left(\frac{1}{|\omega|} \sum_{i \in \omega_k} (I_i p_i) - \mu_k \bar{p}_k \right) \\ b_k &= \bar{p}_k - a_k^T \mu_k \end{aligned}$$

où a_k et I_i sont des vecteurs 3x1, Σ_k est une matrice 3x3 de covariance de I et U une matrice 3x3 d'identité.

Finalement, pour trouver l'image q:

$$q_i = \bar{a}_i^T I_i + \bar{b}_i$$

4 Résultats

4.1 Effet du filtre guidé sur des images p sans guide I .



Figure 1: Effet du filtre guidé avec des différents paramètres sur une image p où $I = p$.

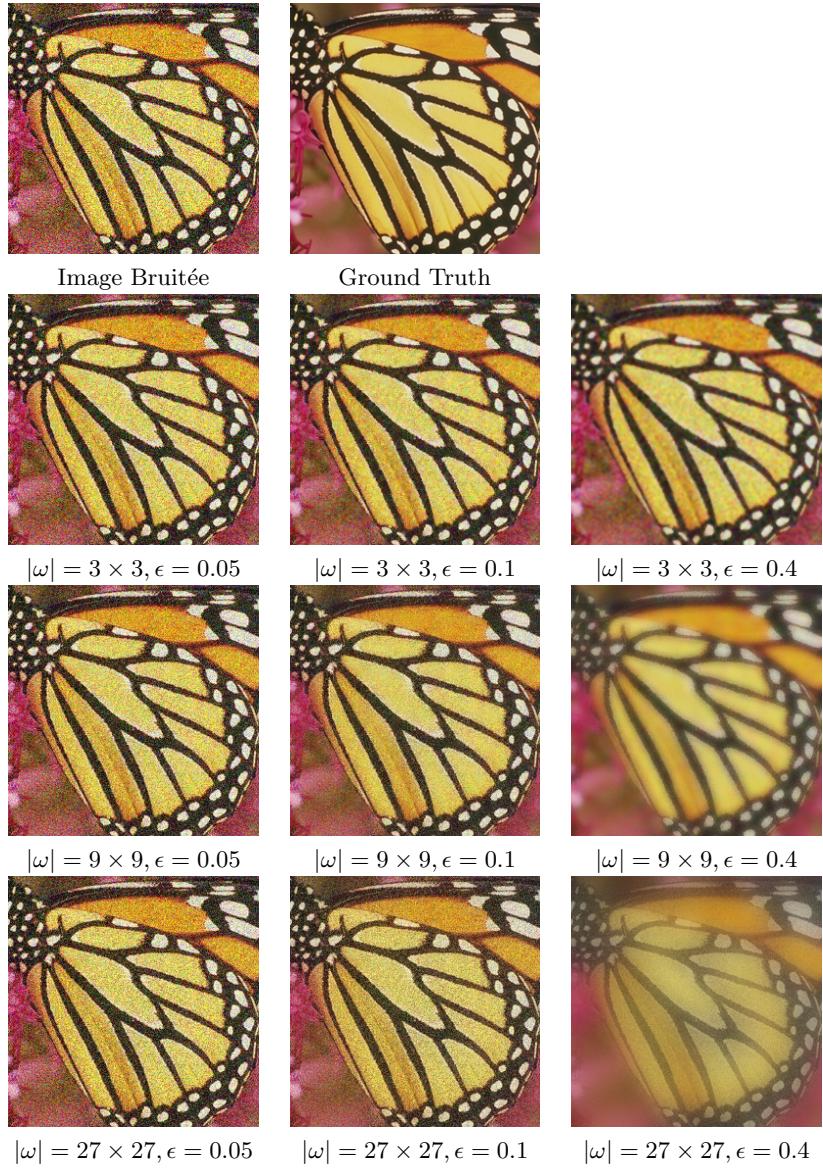


Figure 2: Débruitage utilisant le filtre guidé avec des différents paramètres sur une image p où $I = p$.

4.2 Filtre guidé sur des images p avec guide I .

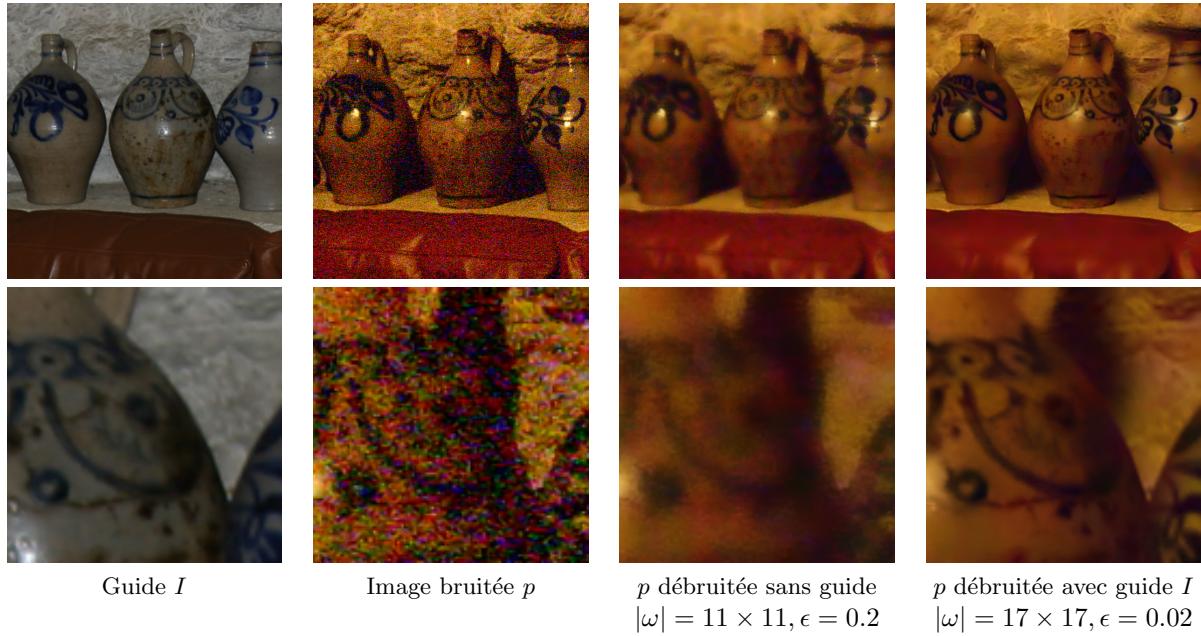


Figure 3: Débruitage d'une photo bruitée prise sans flash avec une photo prise avec flash comme guide.

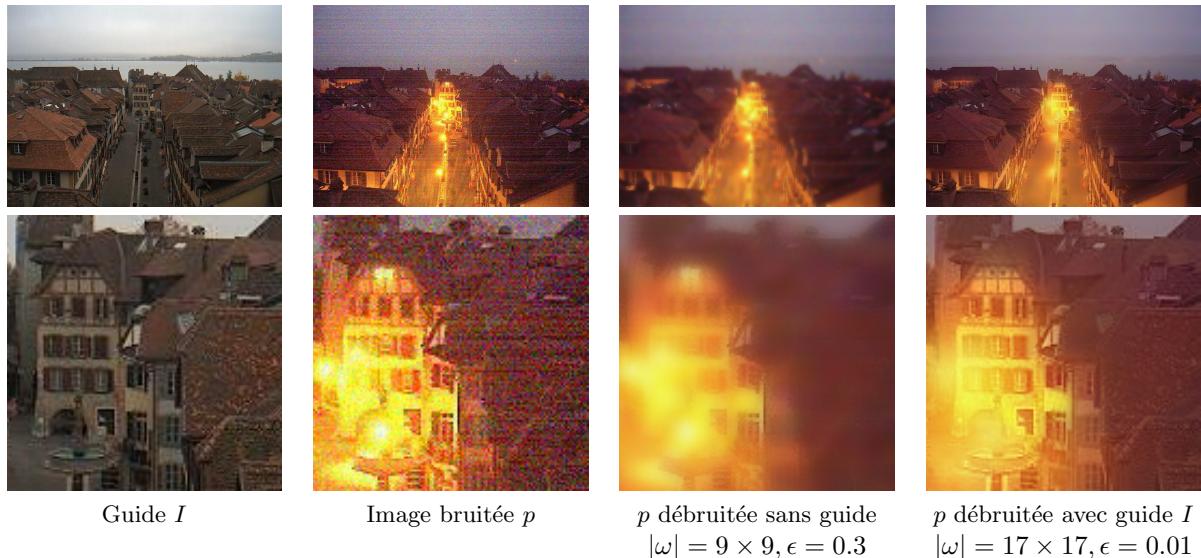


Figure 4: Débruitage d'une photo bruitée de nuit avec une photo prise le jour comme guide.

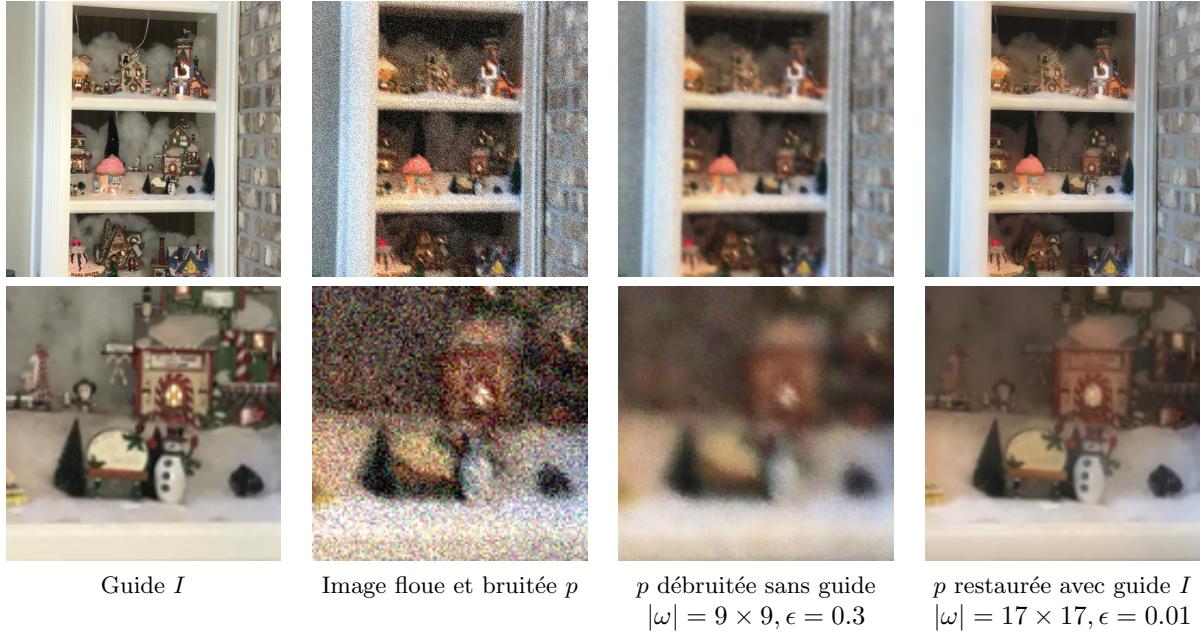


Figure 5: Débruitage et déflouage d'une photo bruitée et floue prise sans flash avec une photo prise avec flash comme guide.

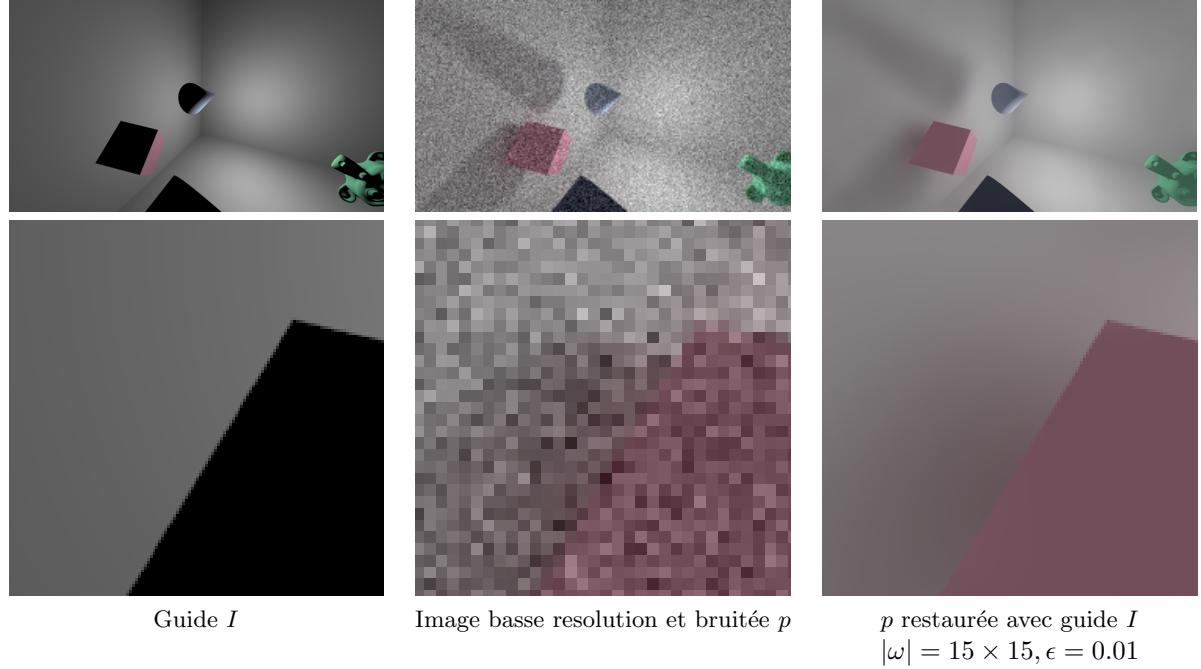
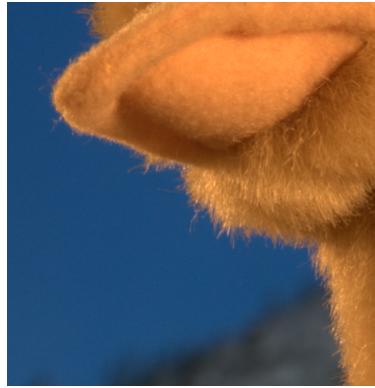


Figure 6: Débruitage et super-résolution jointe d'un rendu 3D en tracé de chemin bruitée.



Guide I



Masque de segmentation p



p raffiné avec guide I
 $|\omega| = 59 \times 59, \epsilon = 0.002$

Figure 7: Raffinement des bordures d'une segmentation.

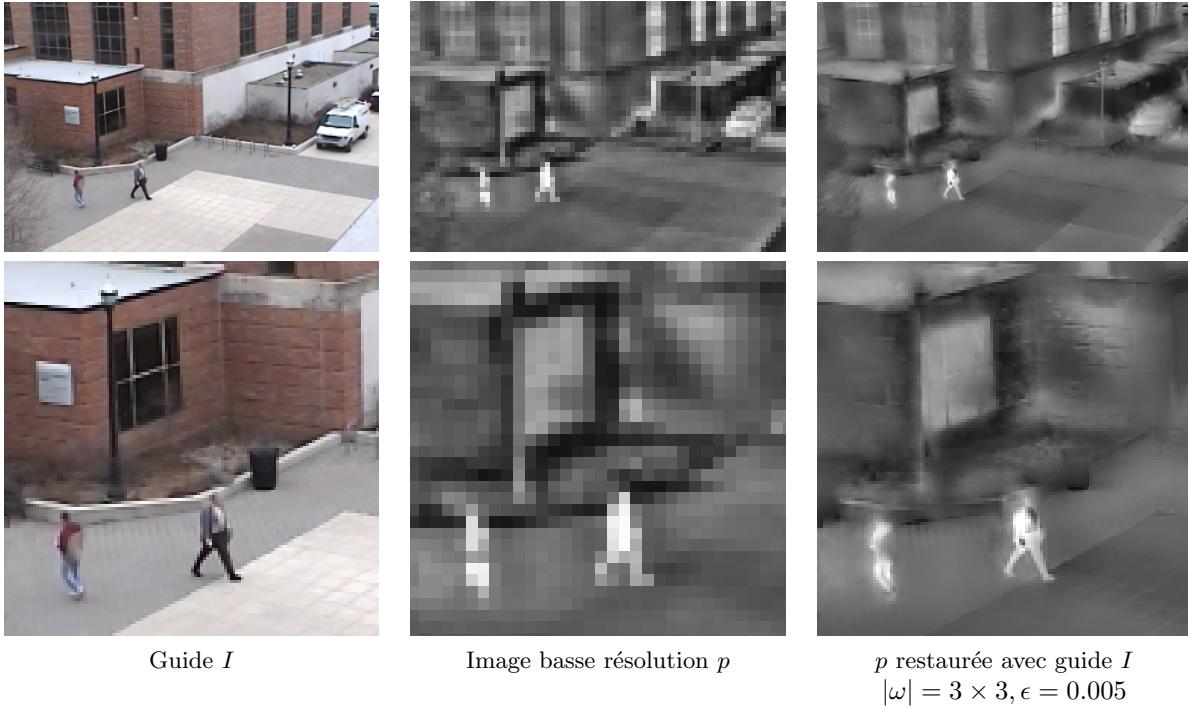


Figure 8: Super-resolution jointe d'une image infrarouge.



Figure 9: Colorisation et super-résolution jointe d'une image.

5 Conclusion

En conclusion, malgré les nombreux problèmes des filtres linéaires (les filtres linéaires causent généralement beaucoup de flou), le filtre guidé reste un filtre très versatile qui nous permet de restaurer des images de façon très efficace lorsqu'une image guide I est disponible. Ce filtre est aussi un des filtres qui préserve les contours le plus rapide grâce à la complexité $O(N)$ en temps du filtre moyen.