



École Polytechnique de l'Université de Tours  
64, Avenue Jean Portalis  
37200 TOURS, FRANCE  
Tél. +33 (0)2 47 36 14 14  
[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr)

**Département Informatique**  
**5<sup>e</sup> année**  
**2014 - 2015**

**Rapport de validation**

**Développement d'un outil de  
traitement d'images par filtrage  
bilatéral**

**Encadrants**  
Moncef HIDANE  
[moncef.hidane@insa-cvl.fr](mailto:moncef.hidane@insa-cvl.fr)

INSA, Blois

**Étudiants**  
Natacha MARLIO-MA  
[natacha.marlio-marette@etu.univ-tours.fr](mailto:natacha.marlio-marette@etu.univ-tours.fr)

DI5 2014 - 2015

Version du 30 janvier 2015



# Table des matières

---

1	Implémentation du filtre bilatéral	4
2	Validation	5
3	Résultats	6
	Bibliographie	8

# 1. Implémentation du filtre bilatéral

---

L'implémentation du filtre bilatéral se base sur un algorithme d'un article [?] voir ci-dessous 1. Soit  $\mathbf{p}$  un pixel de l'image  $\mathbf{S}$  avec pour valeur  $I_p$ .  $\|\cdot\|$  représente une distance euclidienne et  $|\cdot|$  représente une valeur absolue. Le résultat du filtre bilatéral de l'image noté  $BF[\cdot]$ .  $G_\sigma(x)$  représente une fonction gaussienne, ici de la forme :

$$G_\sigma(x) = \frac{1}{2\pi\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (1.1)$$

---

**Algorithme 1** Implémentation naïve du filtre bilatéral

---

```
1: pour chaque pixel  $\mathbf{p}$  de  $\mathbf{S}$  faire /* Parcours de chaque pixel de l'image  $\mathbf{S}$  */
2:    $BF[I]_p = 0$  /* Initialisation */
3:    $W_p = 0$ 
4:   pour chaque pixel  $\mathbf{q}$  de  $\mathbf{S}$  faire /* */
5:      $w = G_{\sigma_s}(\|p - q\|)G_{\sigma_r}(|I_p - I_q|)$ 
6:      $BF[I]_p += wI_q$ 
7:      $W_p += w$ 
8:   fin pour
9:    $BF[I]_p = I_p/W_p$  /* Normalisation */
10: fin pour
```

---

Dans l'implémentation réalisée, la deuxième boucle `for` ne parcourt pas comme dans l'algorithme toute l'image mais seulement une fraction de celle-ci. Le parcours du pixel  $\mathbf{q}$  se fait dans un cadre de  $21 \times 21$  pixels autour du pixel courant  $\mathbf{p}$ . Si le cadre dépasse de l'image, on considère que les valeurs des pixels devant être en dehors seront nuls. Cela ne revient à ne considérer que les pixels du cadre appartenant à l'image.

Si l'image est en couleur au lieu d'être en niveaux de gris, le parcours de l'image se fait le nombre de fois qu'il y a de canaux, c'est à dire trois fois pour une image RGB. Le filtre bilatéral sera calculé pour chaque canal de l'image.

Le filtre bilatéral implémenté est dépendant de deux paramètres  $\sigma_s$  et  $\sigma_r$ .  $G_{\sigma_s}$  est une gaussienne spatiale qui permet de diminuer l'influence des pixels distants.  $G_{\sigma_r}$  est une gaussienne de gamme qui diminue l'influence d'un pixel  $\mathbf{q}$  lorsque son intensité diffère  $I_p$ .

## 2. Validation

---

La validation du filtre bilatéral s'est faite en deux étapes. La première consistait à vérifier que le filtre bilatéral reproduisait correctement le comportement d'un filtre gaussien, c'est à dire de convolution gaussienne. Une valeur grande de  $\sigma_r$  permet d'approcher une convolution gaussienne car la valeur de  $G_{\sigma_r}$  diminue et tend vers un. La convolution gaussienne ne tient compte que de la gaussienne spatiale ( $G_{\sigma_s}$ ). La seconde étape permet de vérifier que les résultats obtenus concordent avec ceux de l'implémentation du filtre bilatéral de CImg. Pour cela deux pourcentages sont calculés : un premier correspondant au pourcentage maximum de différence entre les valeurs des deux images obtenues (équation 2.1) et un deuxième correspondant au pourcentage de différence entre les deux images (équation 2.2).

$$diffPixel = \max_{i,y} \frac{\sum_c \frac{I_{fb,c} - I_{CImg,c}}{255} * 100}{c} \quad (2.1)$$

$$diffImage = \frac{\sum_x \sum_y \sum_c \frac{I_{fb,c} - I_{CImg,c}}{255} * 100}{x * y * c} \quad (2.2)$$

La comparaison se fait avec des images initiales bruitées par un bruit gaussien. On accepte un pourcentage de différence entre l'image produite avec CImg et celle produite avec l'implémentation du filtre bilatéral de 5% au maximum.

### 3. Résultats

---

Les images ci-dessous ont été obtenues avec le filtre bilatéral implémenté.



FIGURE 3.1 – Filtrage bilatéral avec  $\sigma_s=4$  et  $\sigma_r=30$



FIGURE 3.2 – Filtrage bilatéral avec  $\sigma_s=4$  et  $\sigma_r=70$



FIGURE 3.3 – Filtrage bilatéral avec  $\sigma_s=64$  et  $\sigma_r=20$

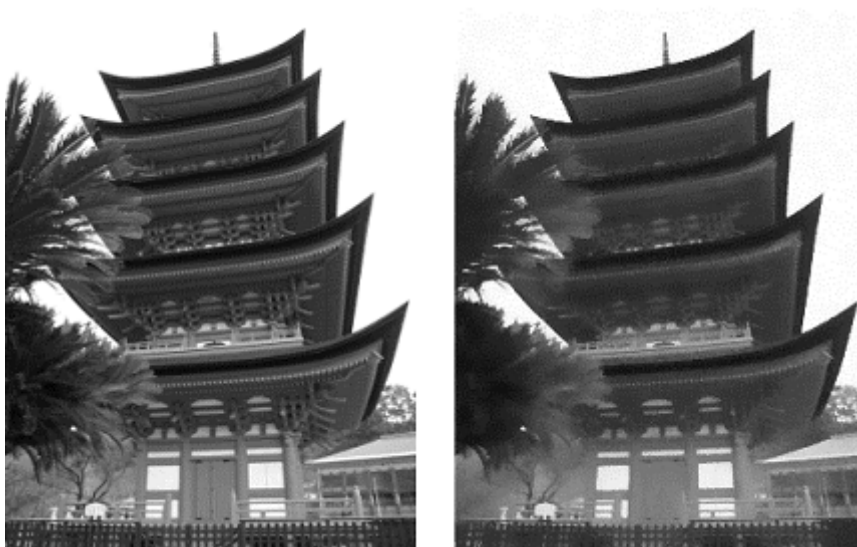


FIGURE 3.4 – Filtrage bilatéral avec  $\sigma_s=4$  et  $\sigma_r=30$



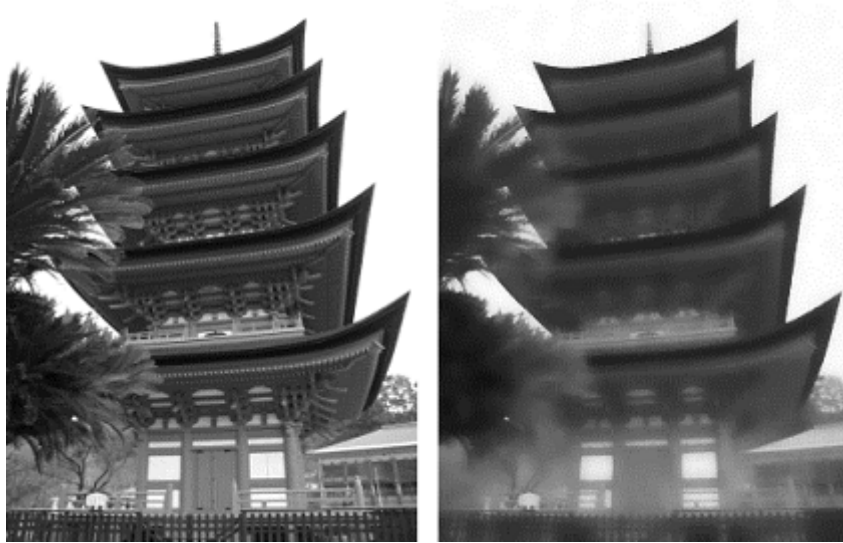


FIGURE 3.5 – Filtrage bilatéral avec  $\sigma_s=4$  et  $\sigma_r=70$

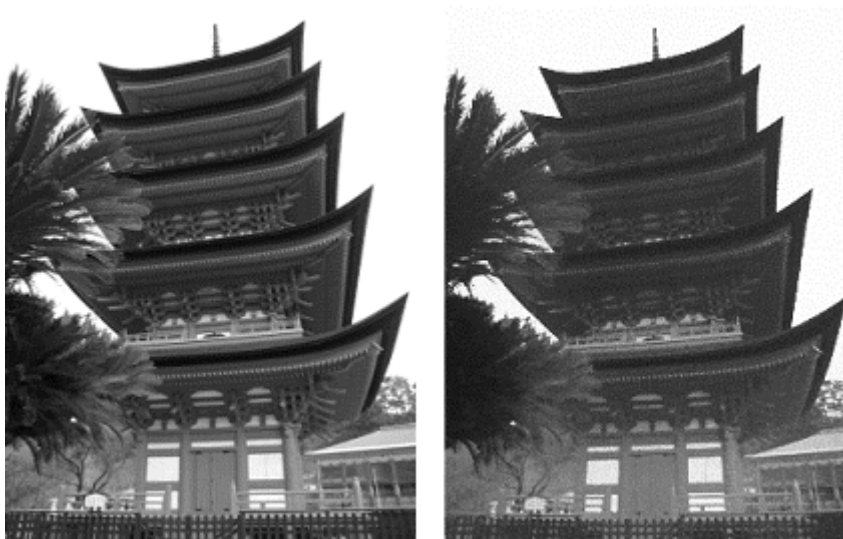


FIGURE 3.6 – Filtrage bilatéral avec  $\sigma_s=64$  et  $\sigma_r=30$





# Développement d'un outil de traitement d'images par filtrage bilatéral

---

Département Informatique

5<sup>e</sup> année

2014 - 2015

Rapport de validation

**Résumé :**

**Mots clefs :**

**Abstract:**

**Keywords:**

## **Encadrants**

Moncef HIDANE

[moncef.hidane@insa-cvl.fr](mailto:moncef.hidane@insa-cvl.fr)

## **Étudiants**

Natacha MARLIO-MA

[natacha.marlio-marette@etu.univ-tours.fr](mailto:natacha.marlio-marette@etu.univ-tours.fr)