MASTER 2 Imagine 2023-2024 MAURIN Christina
HAI927I Projet Image COQUERON Solal

Christal

Compte rendu n°2

Générateurs de bruit artificiel

Dans un premier temps, nous avons décidé qu'il serait plus judicieux de générer du bruit artificiel dans le but de pouvoir bien observer si nos algorithmes de filtrage étaient *visiblement* convenables pour chaque type de bruit.

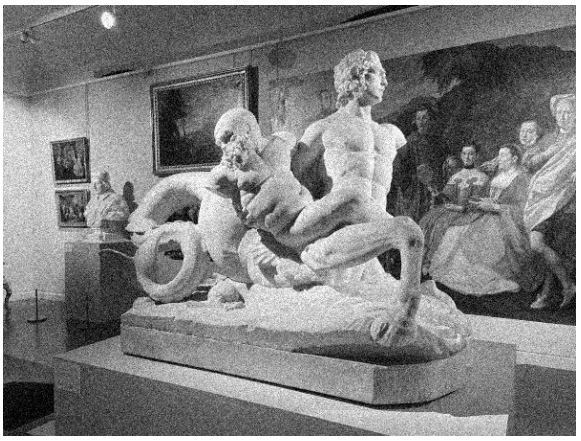
La photographie amateure nous a permis de constater que certains types de bruits étaient plus fréquents que les autres. C'est pourquoi nous avons sélectionné 4 types que l'on souhaite accentuer avec nos générateurs :

- Le bruit poivre et sel
- Le bruit gaussien
- Le bruit chromatique
- Le bruit périodique

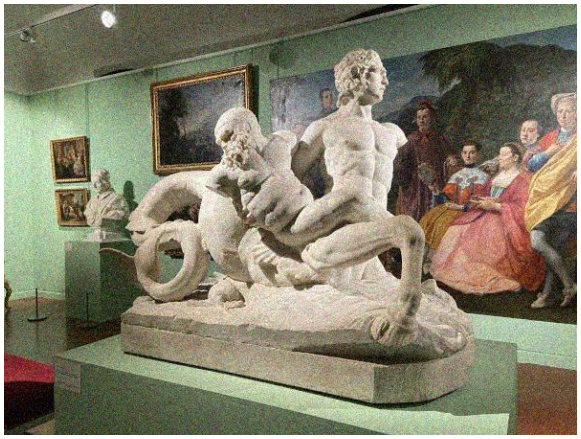
Nous avons utilisé la librairie de Pillow (anciennement PIL). Pour le moment nous n'avons utilisé qu'une image qui n'a pas de bruit spécifique. C'est une photographie personnelle prise au Musée Fesch d'Ajaccio. Elle est intéressante car possédant des variations de couleurs, de luminosités et de textures.

MAURIN Christina
COQUERON Solal

Bruit poivre et sel (densité = 0.01)



Bruit gaussien (écart-type = 20)



Bruit chromatique (écart-type = 20)

Bruit périodique (bande = 50 période = 100)

Tous nos générateurs se trouvent dans le dossier “gen_noise” du git. Les images résultantes sont dans “imgoutnoised”

Expérimentations de filtrage

Nous avons ensuite essayé de minimiser les bruits produits en leur appliquant un filtre ou une combinaison de filtres. Le but étant de comprendre les changements produits par ces filtres selon le bruit testé.

Dans notre code, nous avons utilisé les librairies Pillow (anciennement PIL) et OpenCV. Lorsque l'on utilise les filtres, certains paramètres sont à spécifier:

Filtre moyenneur :

```
Utilisation : python f_moyenneur.py le_chemin.jpg radius
```

Ce filtre remplace la valeur du pixel courant par la moyenne des pixels voisins (radius)

Filtre médian :

```
Utilisation : python f_median.py le_chemin.jpg
```

Ce filtre remplace la valeur du pixel courant par la médiane des pixels voisins. (le radius est configuré à 3)

Filtre bilatéral :

```
Utilisation : python f_bilateral.py le_chemin.jpg diamètre var_couleurs
var_spatiales
```

Ce filtre utilise ces trois paramètres pour atténuer le bruit en prenant en compte la distance spatiale entre les pixels tout en préservant les couleurs.

Filtre gaussien :

```
Utilisation : python f_gaussien.py le_chemin.jpg radius
```

Ce filtre remplace la valeur du pixel courant par la moyenne des pixels voisins.

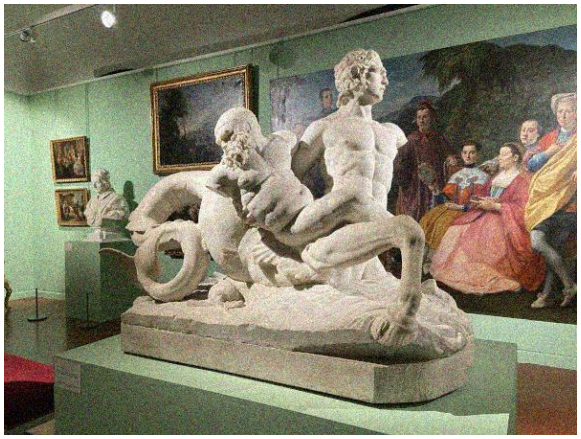
Filtre laplacien :

```
Utilisation : python f_laplacien.py le_chemin.jpg
```

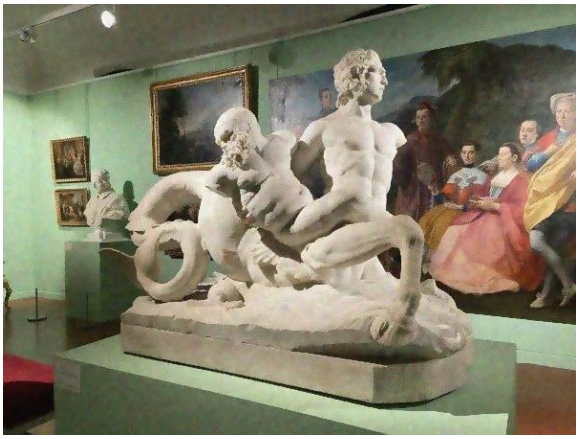
Ce filtre particulier permet de renforcer les contours de l'image qui a déjà été filtrée.

MAURIN Christina
COQUERON Solal

Ci-dessous, quelques résultats concluants concernant le bruit chromatique :



Bruit chromatique (écart-type = 20)

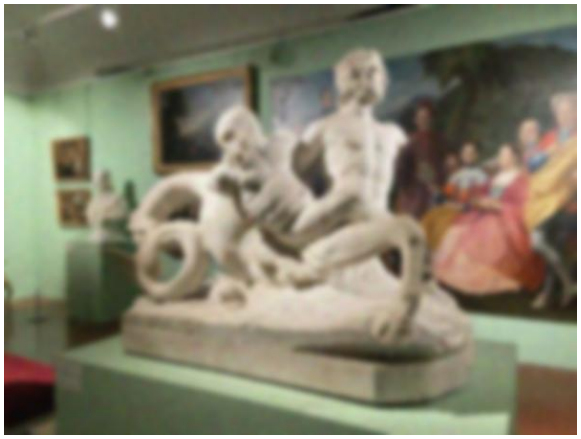


Filtre bilatéral
(diam = 9, var_c = 75, var_ s = 75)

On constate que le filtre bilatéral est plutôt efficace pour le bruit chromatique. On peut montrer également l'application des autres filtres qui ne sont guères convaincants :

Filtre médian
(radius = 3)

Filtre moyenneur
(radius = 3)



Filtre gaussien
(radius = 3)

En ce qui concerne le bruit poivre et sel, on a pu constater que le filtre médian était intéressant :

Bruit poivre et sel (densité = 0.01)

Filtre médian

Bruit poivre et sel (densité = 0.05)

Filtre médian

Nous avons laissé de côté le filtrage du bruit périodique qui est un bruit plus particulier que les autres.

Toutes les expérimentations se trouvent dans le dossier “imgoutdenoised” du git. Les filtres sont, quant à eux, dans “filters”.

Recherches sur le débruitage en photographie

Notre projet étant axé sur l’amélioration de la qualité d’une photographie à l’aide de méthodes de débruitage, il nous a semblé pertinent de concentrer nos recherches sur ce qui a déjà été fait dans ce domaine.

Nous avons donc cherché des papiers qui nous paraissaient les plus adaptés à notre thème afin de pouvoir les étudier plus en détails la semaine prochaine. Deux papiers ont attiré notre attention, le premier s’intitule *efficient poisson denoising for photography* et le second *Toward Convolutional Blind Denoising of Real Photographs.*

Le premier papier traite de la réduction du bruit dans les capteurs d’images, ils proposent une méthode avec laquelle ils estiment le nombre de photons à partir des données d’éclairage afin d’améliorer l’efficacité des méthodes de filtrages en rendant le bruit plus proche d’une distribution gaussienne additive.

Le second papier traite de l’amélioration de la réduction du bruit dans des photographies à l’aide de CNN. Ils proposent un réseau de débruitage convolutionnel (CBDNet) entraîné avec un modèle de bruit plus réaliste et des paires d’images bruitées et non bruitées. Ils intègrent également un sous-réseau d’estimation du bruit pour améliorer la précision de la réduction du bruit.

Pour la semaine prochaine

Nous allons étudier plus en détails les papiers que nous avons trouvés cette semaine afin de mieux les comprendre et d'être en mesure d'implémenter les techniques qui y sont proposées.

Nous allons aussi continuer nos recherches sur le débruitage en photographie pour trouver d'autres méthodes que nous pourrions implémenter pour notre projet.

Sources

- Efficient poisson denoising for photography;
https://www.researchgate.net/profile/Hugues-Talbot-2/publication/224115039_Efficient_Poisson_Denoising_for_Photography/links/0deec525c4e064a606000000/Efficient-Poisson-Denoising-for-Photography.pdf
- Toward Convolutional Blind Denoising of Real Photographs :
https://openaccess.thecvf.com/content_CVPR_2019/papers/Guo_Toward_Convolutional_Blind_Denoising_of_Real_Photographs_CVPR_2019_paper.pdf