

Presentation d'Article: Wasserstein Generative Models for Patch-based Texture Synthesis

Yoann POUPART

Systèmes Complexes
ENS de Lyon
Machine Learning

3 Janvier 2022

Plan

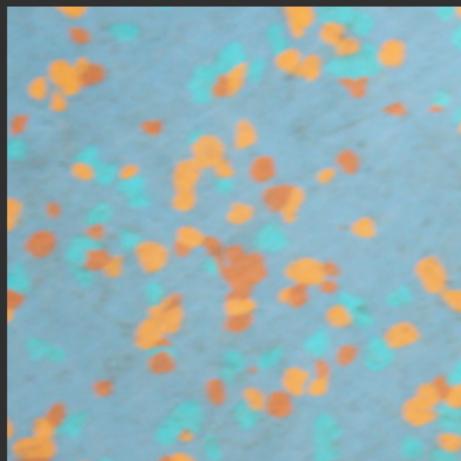
1 Motivations

2 État-de-l'art et Contribution

3 Optimisation

4 Génération d'images

5 Reproduction et application



Synthèse de texture

- Meilleur compréhension des textures
- Applications dans les jeux vidéo

Synthèse de texture

- Meilleur compréhension des textures
- Applications dans les jeux vidéo
 - Remplace un travail de graphiste répétitif

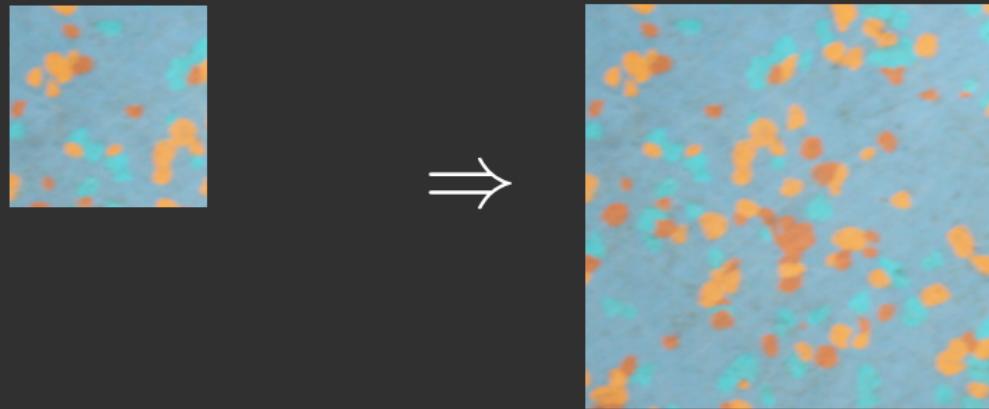
Synthèse de texture

- Meilleur compréhension des textures
- Applications dans les jeux vidéo
 - Remplace un travail de graphiste répétitif



Synthèse de texture

- Meilleur compréhension des textures
- Applications dans les jeux vidéo
 - ↪ Remplace un travail de graphiste répétitif



Remplissage

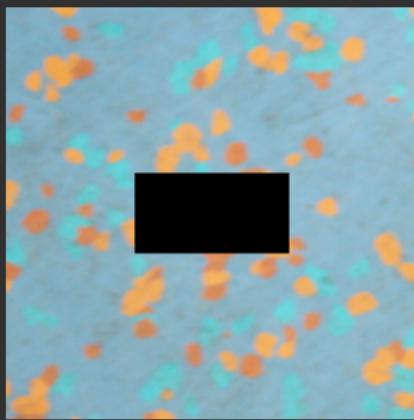
- Réparation de données manquantes
- Filtrage et remplacement

Remplissage

- Réparation de données manquantes
- Filtrage et remplacement
 - ↪ Édition d'images

Remplissage

- Réparation de données manquantes
- Filtrage et remplacement
 - ↪ Édition d'images



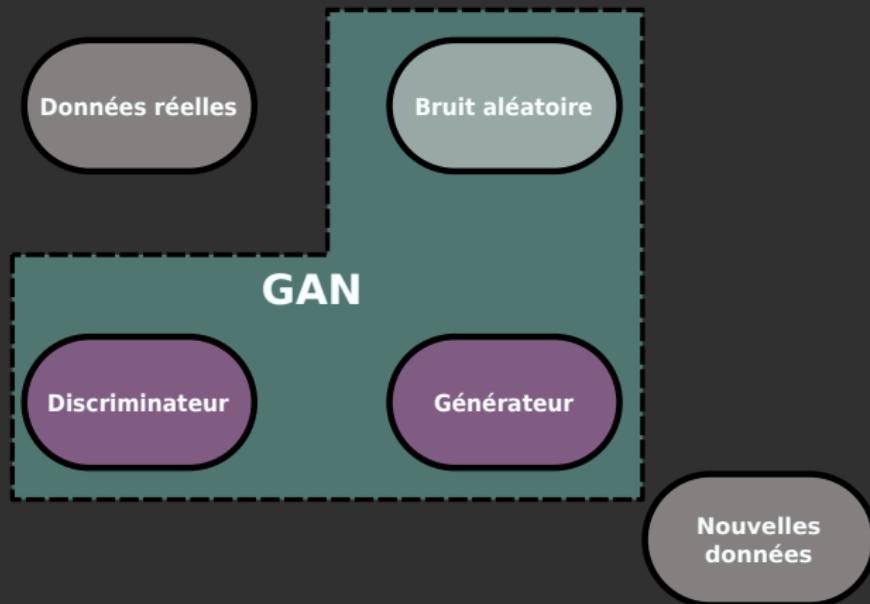
Remplissage

- Réparation de données manquantes
- Filtrage et remplacement
 - ↪ Édition d'images



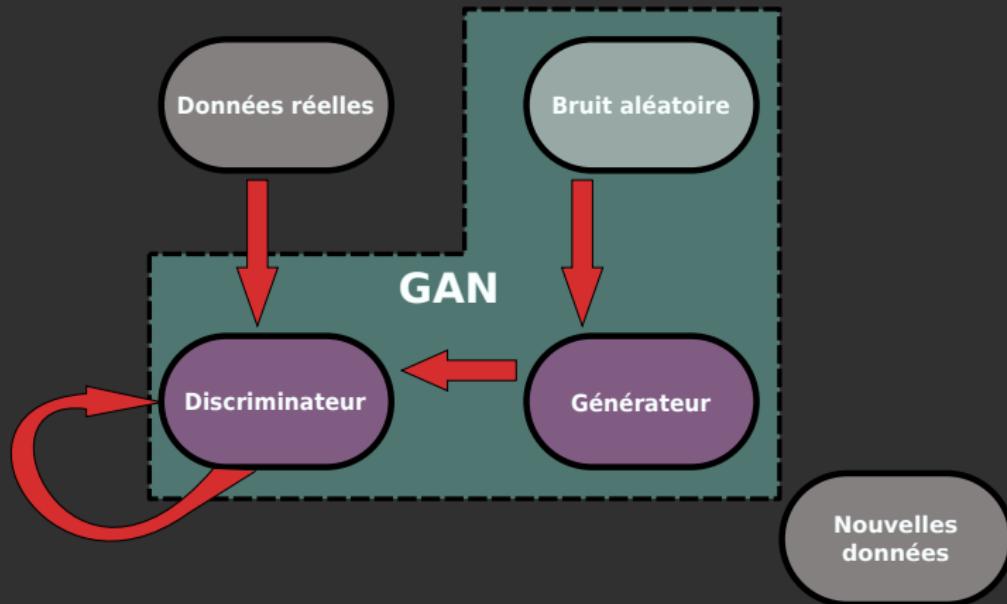
Réseaux génératifs

- Principe d'un GAN (Goodfellow et al. 2014)



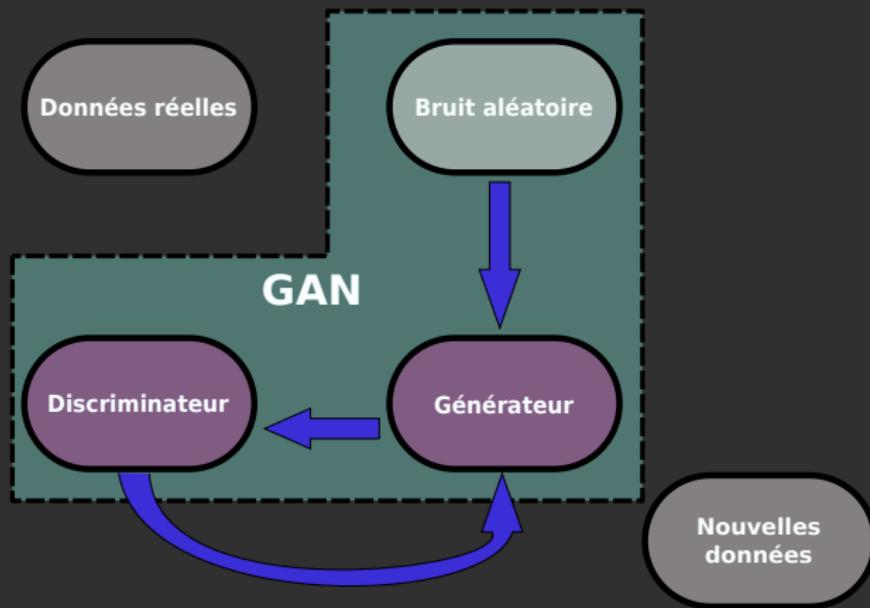
Réseaux génératifs

- Boucle d'apprentissage pour le discriminateur



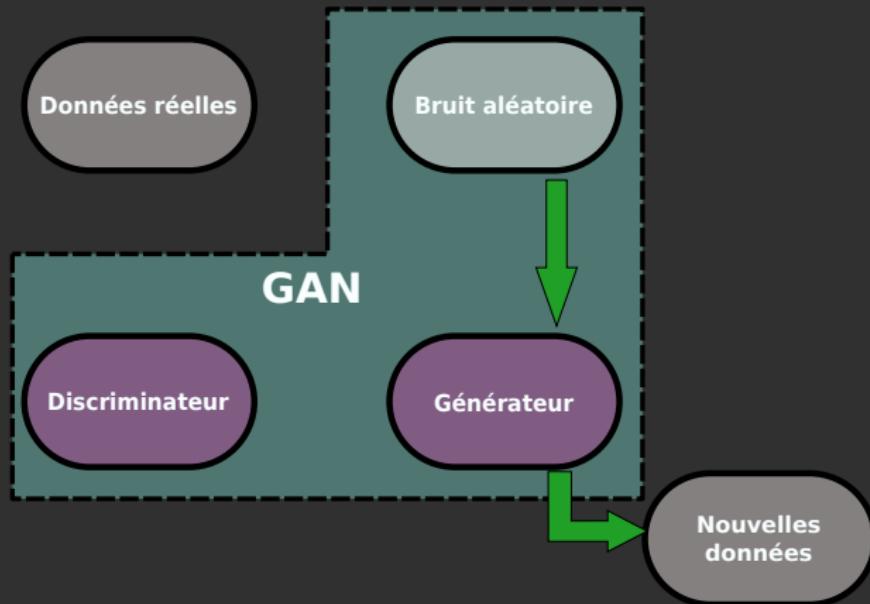
Réseaux génératifs

- Boucle d'apprentissage pour le générateur



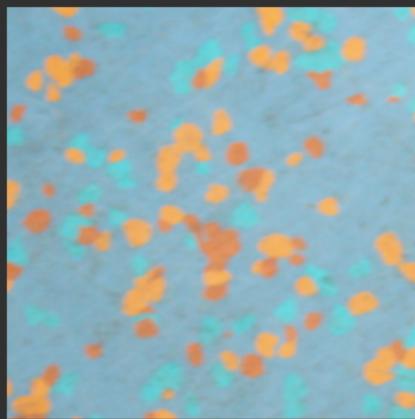
Réseaux génératifs

- Boucle de production de nouvelles données



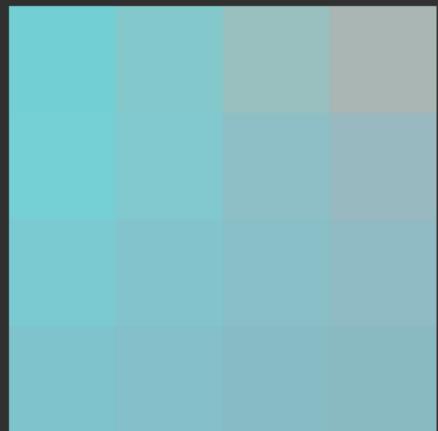
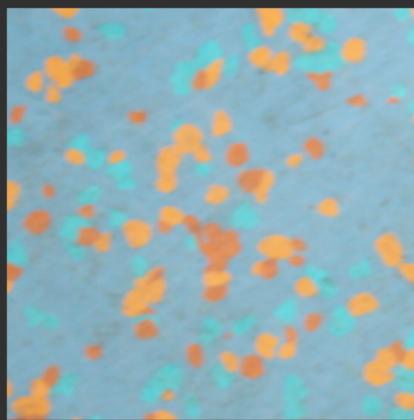
Décomposition en sous-images

- Formulation à l'échelle du pixel
- Décomposition en sous-images (4×4)



Décomposition en sous-images

- Formulation à l'échelle du pixel
- Décomposition en sous-images (4×4)

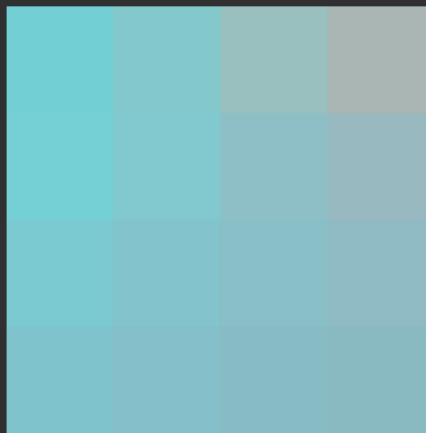


Décomposition en sous-images

- Distribution empirique (image u à n pixels) :

$$\mu_u = \frac{1}{n} \sum_{i=1}^n \delta_{P_i u}$$

Avec $P_i u$ le i-ième patch (image périodique)

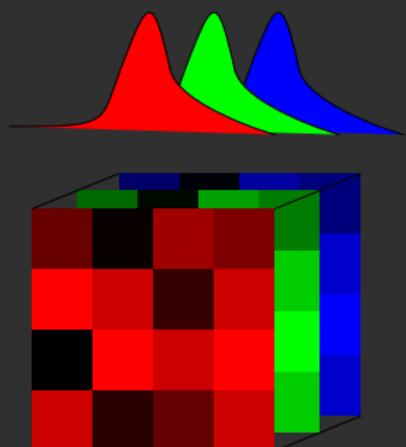
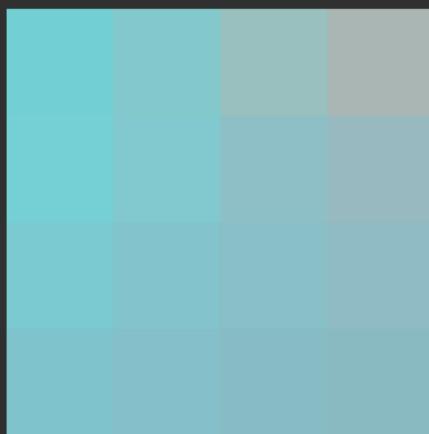


Décomposition en sous-images

- Distribution empirique (image u à n pixels) :

$$\mu_u = \frac{1}{n} \sum_{i=1}^n \delta_{P_i u}$$

Avec $P_i u$ le i-ième patch (image périodique)



Transport Optimal

■ Coût optimal de transport

Avec μ, ν deux mesures

$$\text{OT}_c(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int c(x, y) d\pi(x, y).$$

Dans l'article $c(x, y) = \|x - y\|^2$

Transport Optimal

■ Coût optimal de transport

Avec μ, ν deux mesures

$$\text{OT}_c(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int c(x, y) d\pi(x, y).$$

Dans l'article $c(x, y) = \|x - y\|^2$

Formulation Semi-duale (Santambrogio, 2015) :

$$\text{OT}_c(\mu, \nu) = \max_{\varphi \in \mathcal{C}(\mathcal{Y})} \int \varphi^c(x) d\mu(x) + \int \varphi(y) d\nu(y),$$

Où $\varphi^c(x) = \min_{y \in \mathcal{Y}} [c(x, y) - \varphi(y)].$

Transport Optimal

■ Coût optimal de transport

Avec μ, ν deux mesures

$$\text{OT}_c(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int c(x, y) d\pi(x, y).$$

Dans l'article $c(x, y) = \|x - y\|^2$

Formulation Semi-duale (Santambrogio, 2015) :

$$\text{OT}_c(\mu, \nu) = \max_{\varphi \in \mathcal{C}(\mathcal{Y})} \int \varphi^c(x) d\mu(x) + \int \varphi(y) d\nu(y),$$

Où $\boxed{\varphi^c(x) = \min_{y \in \mathcal{Y}} [c(x, y) - \varphi(y)]}$.

Transport Optimal

- Minimisation de fonction :

$$w(u) = \text{OT}_c(\mu_u, \mu_v) = \max_{\varphi \in \mathbf{R}^m} f(\varphi, u),$$

Avec

$$f(\varphi, u) = \frac{1}{n} \sum_{i=1}^n \varphi^c(P_i u) + \frac{1}{m} \sum_{j=1}^m \varphi_j.$$

On veut donc résoudre :

$$\min_{u \in \mathbf{R}^n} \max_{\varphi \in \mathbf{R}^m} f(\varphi, u)$$

Transport Optimal

- Minimisation de fonction :

$$w(u) = \boxed{\text{OT}_c(\mu_u, \mu_v)} = \max_{\varphi \in \mathbf{R}^m} f(\varphi, u),$$

Avec

$$f(\varphi, u) = \frac{1}{n} \sum_{i=1}^n \varphi^c(P_i u) + \frac{1}{m} \sum_{j=1}^m \varphi_j.$$

On veut donc résoudre :

$$\min_{u \in \mathbf{R}^n} \max_{\varphi \in \mathbf{R}^m} f(\varphi, u)$$

Résultats théoriques

Formulation Semi-duale (Santambrogio, 2015) :

- f est concave en φ
- f n'est pas convexe en u

Résultats théoriques

Formulation Semi-duale (Santambrogio, 2015) :

- f est concave en φ
- f n'est pas convexe en u

Recherche du point de selle :

- Recherche alternée
- Amélioration possible
↪ Gradients ⇒ Opérateur Proximal

Optimisation

■ Minimisation

$$\nabla_u f \left(\varphi^{k+1}, u^k \right) = \frac{1}{n} \left(\sum_{i=1}^n P_i^T P_i u^k - \sum_{i=1}^n P_i^T y_{j_i^{k+1}} \right)$$

Avec

$$j_i^{k+1} = \arg \min_j \frac{1}{2} \left\| P_i u^k - y_j \right\|^2 - \varphi_j^{k+1}.$$

Optimisation

■ Minimisation

$$\nabla_u f \left(\varphi^{k+1}, u^k \right) = \frac{1}{n} \left(\sum_{i=1}^n P_i^T P_i u^k - \sum_{i=1}^n P_i^T y_{j_i^{k+1}} \right)$$

Avec

$$j_i^{k+1} = \arg \min_j \frac{1}{2} \left\| P_i u^k - y_j \right\|^2 - \varphi_j^{k+1}.$$

■ Mise à jour

$$u^{k+1} = (1 - \eta)u^k + \eta w^k$$

Avec $w^k = \frac{1}{p} \sum_{i=1}^n P_i^T y_{j_i^{k+1}}$

Optimisation

■ Minimisation

$$\nabla_u f \left(\varphi^{k+1}, u^k \right) = \frac{1}{n} \left(\sum_{i=1}^n P_i^T P_i u^k - \sum_{i=1}^n P_i^T y_{j_i^{k+1}} \right)$$

Avec

$$j_i^{k+1} = \arg \min_j \frac{1}{2} \left\| P_i u^k - y_j \right\|^2 - \varphi_j^{k+1}.$$

■ Mise à jour

$$u^{k+1} = (1 - \eta)u^k + \eta w^k$$

Avec $w^k = \frac{1}{p} \sum_{i=1}^n P_i^T y_{j_i^{k+1}}$

Optimisation

- Formulation pour une texture multi-echelle

$$\mathcal{L}(u) = \sum_{l=1}^L \max_{\varphi} f_{v^l} (u^l, \varphi)$$

Avec

$$f_{v^l} (u^l, \varphi) = \frac{1}{n} \sum_{j=1}^n \min_i \left[c \left(P_j u^l, P_i v^l \right) - \varphi_i \right] + \frac{1}{m} \sum_{i=1}^m \varphi_i.$$

Dont le gradient est :

$$\nabla_u \mathcal{L}(u) = \sum_{l=1}^L (D_u S_l(u))^T \nabla_u f_{v^l} (u^l, \varphi_l^\star) = \sum_{l=1}^L G_l (u, \varphi_l^\star)$$

Optimisation

- Formulation pour une texture multi-echelle

$$\mathcal{L}(u) = \sum_{l=1}^L \max_{\varphi} f_{v^l} \left(u^l, \varphi \right)$$

Avec

$$f_{v^l} \left(u^l, \varphi \right) = \frac{1}{n} \sum_{j=1}^n \min_i \left[c \left(P_j u^l, P_i v^l \right) - \varphi_i \right] + \frac{1}{m} \sum_{i=1}^m \varphi_i.$$

Dont le gradient est :

$$\nabla_u \mathcal{L}(u) = \sum_{l=1}^L (D_u S_l(u))^T \nabla_u f_{v^l} \left(u^l, \varphi_l^* \right) = \sum_{l=1}^L G_l \left(u, \varphi_l^* \right)$$

Modèles Génératifs

Principes de la méthode :

- Réseau génératif convolutif
- Réseau descriptif (pré-entraîné)
- Gradient de transport optimal

Modèles Génératifs

Principes de la méthode :

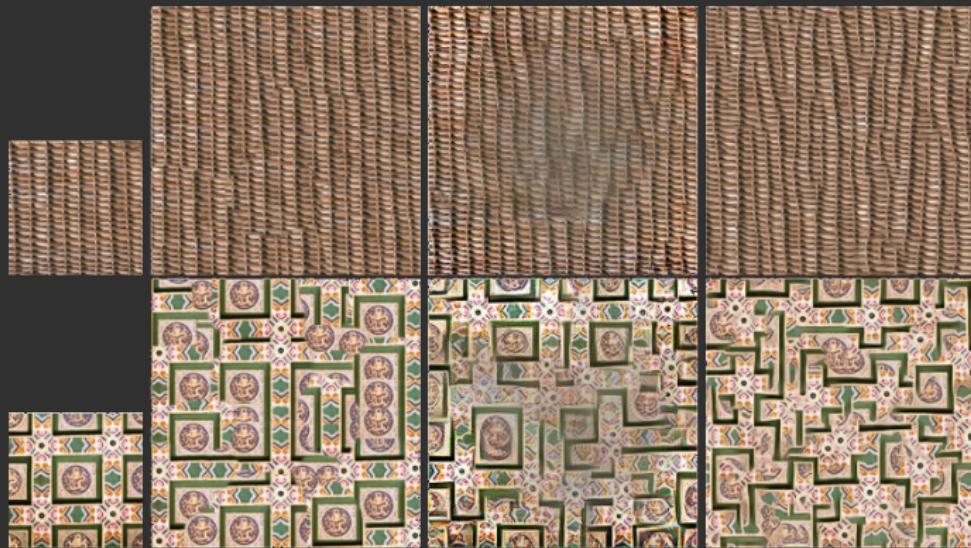
- Réseau génératif convolutif
- Réseau descriptif (pré-entraîné)
- Gradient de transport optimal

Avantages :

- Génération de taille arbitraire
- Amélioration des performances et de la stabilité
 → Nouveau cadre Wasserstein-GAN

Simulations

■ Comparaisons synthèse (Houdard et al. 2020)



Originale

Kwatra et
al., 2005

Gatys et al.,
2015

Alg. 1

Simulations

- Images à haute résolution (Houdard et al. 2020)



Originale

Kwatra et
al., 2005

Gatys et al.,
2015

Alg. 1

Simulations

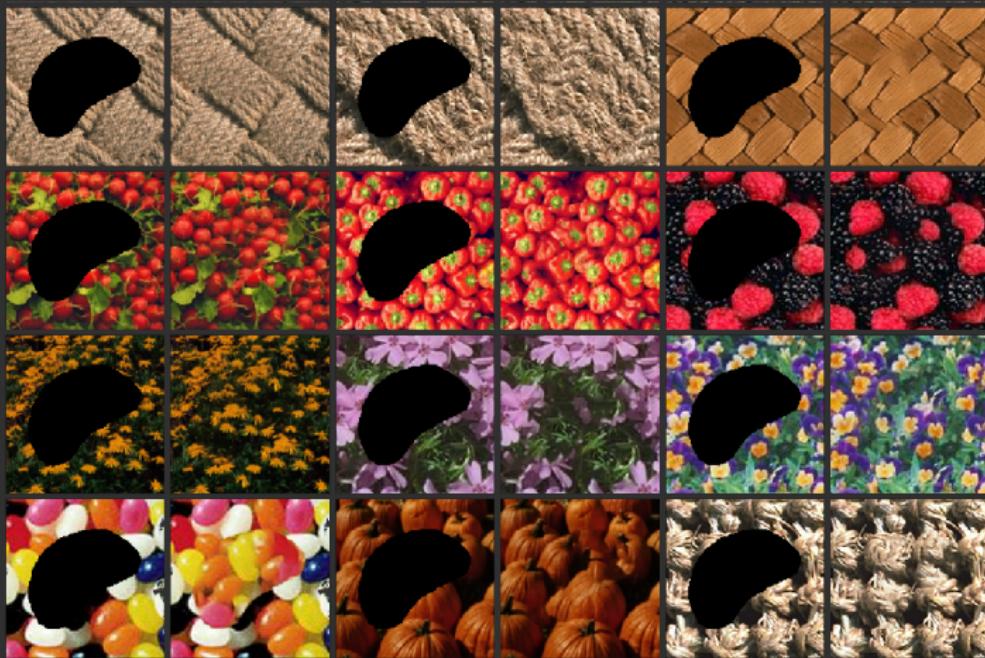
■ Comparaisons CNN (Houdard et al. 2020)



Ori	Alg. 1 Houdard et al. 2020	Alg. 2	Text. Net Ulyanov et al., 2016	SinGAN Shaham et al., 2019	PSGAN Bergmann et al., 2017	Texto Leclaire et Rabin, 2019
-----	-------------------------------	--------	-----------------------------------	-------------------------------	--------------------------------	----------------------------------

Simulations

■ Résultats remplissage (Houdard et al. 2020)



Applications

- Méthode flexible !
- Changement des pixels à optimiser



Applications

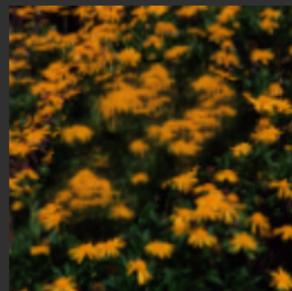
- Comparaisons remplissage
↪ Basse qualité (128×128)



Endommagée



Article



Contribution

Conclusion et références

Améliorations possible :

- Opérateur proximal
- Nombre d'échelles variable au cours de l'entraînement
- Meilleure sélection de patch (remplissage)

- Article : Houdard et al. 2020
- Code original : <https://github.com/ahoudard/wgenpatex>
- Nouvelle branche : <https://github.com/Xmaster6y/wgenpatex>

Algorithme

Algorithme 1 : Synthèse de texture multi-echelle.

Entrées : Image cible v , image initiale u_0 , taux d'apprentissage η_u et η_φ , nombre d'itérations N_u et N_φ , nombre d'echelles L

Sorties : Image u

début

```

 $u := u_0$  et  $\varphi_l = 0$  pour  $l = 1 \dots L$ 
pour  $k = 1 \dots N_u$  faire
    pour  $l = 1 \dots L$  faire
        pour  $j = 1 \dots N_\varphi$  faire
             $\varphi_l := \varphi_l + \eta_\varphi g_l(\varphi_l, S_l(u))$ 
             $G_l(u, \varphi_l) := (D_u S_l(u)) T \nabla_u f_{v_l}(\varphi_l, S_l(u))$ 
        fin pour
         $u := u - \eta_u(k) \sum_{l=1}^L G_l(u, \varphi_l)$ 
    fin pour
fin pour

```

Entraînement du réseau

Algorithme 2 : Synthèse de texture par un réseau convolutif.

Entrées : Image cible v , poids initiaux θ_0 , taux d'apprentissage η_θ ,
nombre d'itérations N_u et N_φ , nombre d'echelles L

Sorties : Image u

début

```

 $u := u_0$  et  $\varphi_l = 0$  pour  $l = 1 \dots L$ 
pour  $k = 1 \dots N_u$  faire
    pour  $l = 1 \dots L$  faire
        compute  $\varphi_l$  with ASGA
        sample  $z$  from  $\xi$ 
         $G_l(u, \varphi_l) := \nabla_\theta f_{v_l}(\varphi_l, g_\theta(z)_l)$ 
    fin pour
     $\theta := \theta - \eta_\theta(k) \sum_{l=1}^L G_l(\theta)$ 

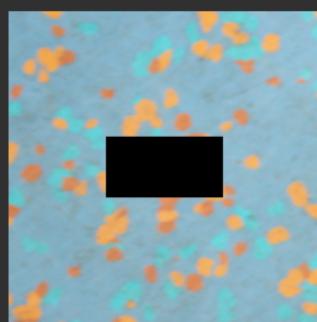
```

Applications

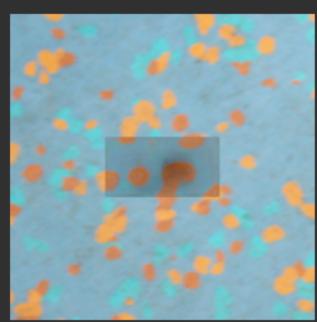
- Résultats pour le remplissage
↪ Haute qualité (256×256)



Originale



Endommagée



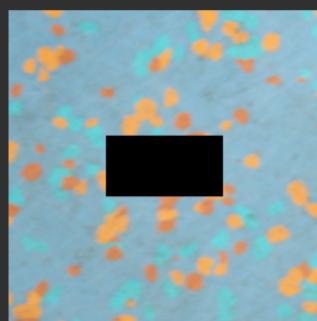
Réparée

Applications

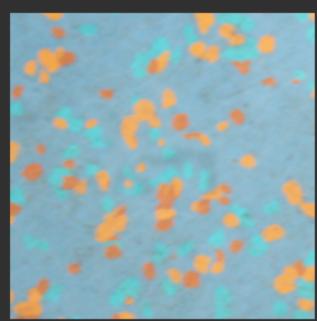
- Résultats pour le remplissage
 - ↪ Haute qualité (256×256)
 - ↪ Une échelle



Originale



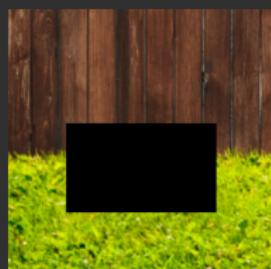
Endommagée



Réparée

Applications

- Résultats pour le remplissage
 - ↪ Texture bimodale (256×256)



Originale

Endommagée

Réparée