

Synthèse de champs nuageux sous contraintes physiques

Pierrick CHATILLON^{1,2} Yann GOUSSEAU² Sidonie LEFEBVRE¹

¹ONERA : DOTA LMA2S, ONERA, Université Paris Saclay F-91123 Palaiseau, France

²LTCI ,Télécom Paris, IP Paris, 19 place Marguerite Perey - 91120 Palaiseau, France

Résumé – Nous proposons dans cet article une architecture de réseau de neurone et une forme de supervision permettant de spécifier des paramètres physiques en entrée d’un réseau générateur et d’obtenir en sortie des images texturées de nuages respectant les statistiques correspondantes. Ce modèle paramétrique contrôle la pente spectrale et la distribution des niveaux de gris des images. La capacité du modèle à générer des images sous contraintes est testée sur des images de télédétection dans l’infrarouge et des images de contenu en eau simulées.

Abstract – In this paper, we propose an architecture and a supervision allowing to specify physical parameters as input to a generator network, and to obtain as output textured cloud images matching the corresponding statistics. This parametric model controls the decay of the Fourier module and the gray level distribution of the images. The capacity of the model to generate images under constraints is tested on infrared remote sensing images and on simulated water content images.

1 Introduction

L’évaluation des performances des capteurs optiques nécessite de disposer de scènes de référence et de prendre en compte la variabilité des objets d’intérêt et du fond. Pouvoir générer rapidement des images reproduisant les niveaux radiométriques et la texture des fonds naturels réels permet de compléter les bases de données expérimentales ou issues de codes de simulation coûteux. Nous nous intéressons ici plus particulièrement aux nuages, pour différentes applications : prédiction de couverture nuageuse afin de décider de déclencher ou non une prise de vue, et donc optimiser la planification des constellations de satellites agiles pour l’observation de la Terre, mais aussi détection d’anomalies, des drones sur fond de ciel nuageux par exemple. Pour ce faire, nous utiliserons des modèles génératifs convolutionnels, produisant des images de fonds texturés de manière rapide et contrôlée.

Les simulations d’écoulements turbulents donnent des champs nuageux réalistes, mais sont très coûteuses. Des modèles stochastiques basés sur un cadre fractal ou de Fourier, qui exploitent les propriétés d’invariance d’échelle observées dans les nuages réels ont également été proposés [10]. Les cascades infiniment divisibles [1] donnent également un cadre de super-résolution stochastique préservant la pente spectrale et la moyenne locale d’une image.

Nous souhaitons donc générer des images de nuages caractérisées tant par leur texture que par leur couverture nuageuse et leur comportement fractal, que nous associons à la décroissance en puissance du module de Fourier $\|\hat{f}(\xi)\| \sim \frac{1}{|\xi|^p}$ [9]. Des modèles non paramétriques permettent la synthèse de champs gaussiens avec un spectre prescrit ([3]), mais sans contrôle fin de la texture [4].

De nombreuses méthodes de synthèse de texture récentes s’appuient sur les travaux de Gatys et al. [4], certaines tentent notamment d’ajouter de l’information fréquentielle à la synthèse de manière non-paramétrique [5]. D’autre part, de multiples travaux ont été menés sur le spectre d’images générées par réseaux profonds, et sur la difficulté de le contrôler (par exemple [2]).

Contraindre les statistiques d’une image est également possible par post-traitement, avec par exemple des contraintes d’histogramme ou des manipulations du module de Fourier, mais ce type d’approche a le défaut de modifier la texture de l’image et nécessite une optimisation pour chaque image et caractéristiques statistiques désirées.

Pour résoudre ces problèmes, nous proposons une approche par apprentissage profond de type *feed forward*, paramétrique, permettant la synthèse rapide de champs nuageux respectant les propriétés physiques attendues.

Notre paramétrisation physique consiste en 3 scalaires : la pente spectrale p , qui est un paramètre de régularité fractale, et deux paramètres d’une loi log-normale régissant la distribution des niveaux de gris des images, couramment utilisés pour les images de nuages [7]. Notre méthode évite les optimisations post-synthèse et permet de générer à la volée des images respectant conjointement des contraintes potentiellement antagonistes. La structure de notre architecture multi-échelle rend possible la manipulation de chaque paramètre indépendamment des autres, et le paramètre de pente est incorporé dans l’architecture par une technique inspirée des algorithmes de bruit fractal.

2 Méthode

Comme expliqué au paragraphe précédent, nous souhaitons générer des champs de nuages en contrôlant l’histogramme et la pente spectrale. Les images sont synthétisées grâce à une approche *feed forward*, à l’aide d’un générateur G d’architecture de type StyleGAN, et d’une supervision de la texture. Le réseau générateur G prend en entrée les paramètres d’histogramme et de pente désirés. Un point clé de l’approche, expliqué en section 2.4, est que l’architecture multi-échelle du générateur permet de prendre en compte le paramètre de régularité p via une pondération en puissance de l’échelle de bruits, à la manière des bruits fractals de Perlin [8].

2.1 Contraintes physiques

2.1.1 Contrôle de la pente spectrale

Nous souhaitons contrôler la décroissance du spectre azimuté, calculé en moyennant le module de la transformée de Fourier discrète $\|\hat{I}\|$ d'une image I sur des cercles de rayon r :

$$AI_{cercles}(I)_r = \frac{1}{N_r} \int_0^{2\pi} \|\hat{I}(r \cos(\phi), r \sin(\phi))\| d\phi \quad (1)$$

où N_r est le facteur de normalisation. Pour contrôler cette décroissance de manière moins contraignante, nous nous intéressons à cette même quantité calculée sur des couronnes de largeur k :

$$AI(I)_r = \frac{1}{N_{r,k}} \int_{kr}^{k(r+1)} \int_0^{2\pi} \|\hat{I}(r' \cos(\phi), r' \sin(\phi))\| d\phi dr' \quad (2)$$

Le comportement souhaité de AI_r est : $AI_r \sim \frac{1}{r^p}$.

Ainsi, nous évaluons p sur une image par régression linéaire de $\log(AI_r)$ contre $\log(r)$. De façon à ne porter attention qu'aux hautes fréquences, où ce comportement en puissance se manifeste, nous appliquons la régression à partir d'un rayon r_0 de cassure, fixé à $\frac{1}{10}$ du rayon maximal d'une image. Ainsi, le profil idéal de décroissance en puissance p à partir de r_0 d'un spectre azimuté $AI(I)$ s'écrit :

$$f_p(AI) = \begin{cases} AI_{r_0} \cdot \left(\frac{r}{r_0}\right)^{-p} & \text{si } r \geq r_0 \\ AI_r & \text{si } r < r_0 \end{cases} \quad (3)$$

Pour superviser la génération d'images de profil azimuté de pente p à partir de r_0 , nous utiliserons la fonction de perte suivante :

$$\mathcal{L}_{spe}(I, p) = \|AI(I) - f_p(AI(I))\|^2 \quad (4)$$

2.1.2 Contrôle de l'histogramme

Comme expliqué précédemment, nous modélisons les niveaux de gris des images par une loi log-normale, de densité de probabilité :

$$f(x, s, \lambda) = \frac{1}{sx\sqrt{2\pi}} \exp\left(-\frac{\log^2\left(\frac{x}{\lambda}\right)}{2s^2}\right) \quad (5)$$

où s et λ correspondent respectivement aux paramètres de forme et d'échelle.

En fournissant des paramètres de loi s, λ à notre générateur nous voulons que les niveaux de gris des images générées suivent une loi log-normale de paramètres s, λ .

En utilisant la méthode de la transformée inverse, nous obtenons N réalisations $\{l_i^{s,\lambda}, 1 \leq i \leq N\}$ d'une loi lognormale de paramètres s, λ , où N est le nombre de pixels de l'image I dont on souhaite contraindre l'histogramme. Nous pénalisons la distance de transport entre les histogrammes de l'image I et de $l^{s,\lambda}$, par différence terme à terme des valeurs triées de ces deux distributions discrètes.

$$\mathcal{L}_{hist}(I, s, \lambda) = \|\text{tri}(I) - \text{tri}(l^{s,\lambda})\|^2 \quad (6)$$

Cette approche de prescription d'histogramme permet d'imposer la distribution log-normale paramétrée par s, λ et est ainsi plus précise qu'une méthode de prescription des paramètres ne permettant qu'un contrôle des moments des distributions.

2.2 Contrôle de la texture

L'idée ici est de contrôler l'apparence visuelle des images générées grâce à un terme de texture. Nous supposons que nous disposons d'une ou plusieurs images exemples. Pour contrôler la texture, nous choisissons les classiques matrices de Gram calculées sur les *features* issus d'un réseau convolutif profond, en l'occurrence VGG 19, selon la méthodologie introduite dans les travaux de [4]. En plus des supervisions paramétriques introduites aux paragraphes précédents, notre réseau sera entraîné avec une fonction de perte \mathcal{L}_{style} égale à une distance entre les matrices de Gram d'images générées et exemples. Remarquons que, dans le cas d'une base de données, cette approche ne permet de reproduire qu'un comportement moyen et non de synthétiser une variété de textures.

2.3 Supervision

Notre réseau générateur G prend en entrée les paramètres physiques p, s, λ et doit produire une image avec un profil spectral de puissance p et un histogramme suivant une loi log-normale de paramètres s et λ . De plus G doit renvoyer une image du style de notre base d'entraînement. En conséquence, en notant $\phi = (p, s, \lambda)$ les paramètres physiques, et $\phi_I = (p_I, s_I, \lambda_I)$ les paramètres évalués sur une image I , notre supervision s'écrit :

$$\mathcal{L} = \mathbb{E}_{I \sim p_{data}} \left[\mathcal{L}_{hist}(G(\phi_I), s_I, \lambda_I) \right] \quad (7)$$

$$+ \mathcal{L}_{spe}(G(\phi_I), p_I) + \mathcal{L}_{style}(G(\phi_I), I) \\ + \mathbb{E}_{\phi \sim p_\phi} \left[\mathcal{L}_{hist}(G(\phi), s, \lambda) + \mathcal{L}_{spe}(G(\phi), p) \right] \quad (8)$$

La première partie de la supervision 7 peut être interprétée comme une reconstruction : la pente spectrale et la couverture de l'histogramme d'une image étant des facettes importantes du style d'une image, nous voulons superviser l'aspect visuel des images générées avec des vérités terrain de même caractéristiques physiques. C'est pourquoi nous évaluons ϕ_I sur I , pour ensuite comparer les styles de $G(\phi_I)$ et de I , en s'assurant que $\phi_{G(\phi_I)} \approx \phi_I$ grâce à \mathcal{L}_{spe} et \mathcal{L}_{hist} .

La seconde partie 8 s'interprète comme un terme de généralisation et de désenchevêtrement des paramètres physiques. En effet, l'évaluation de ϕ_I produit des paramètres physiques corrélés, et nous voulons pouvoir contrôler chacun de ces paramètres de façon indépendante. Nous demandons ainsi à $G(\phi)$ de respecter les statistiques décrites par ϕ , pour $\phi \sim p_\phi$, avec $p_\phi = p_{(p,s,\lambda)} = p_p \cdot p_s \cdot p_\lambda$. Cette supervision permet de contrôler un paramètre physique indépendamment des autres, et de contraindre les caractéristiques physiques des images générées au delà du domaine observé dans les données.

2.4 Architecture

Notre générateur G se base sur l'architecture de StyleGAN [6], modifiée pour produire des images stationnaires. Cette architecture consiste à contrôler la génération d'images en changeant les statistiques d'ordre 1 et 2 des cartes d'attributs au cours d'opérations successives d'agrandissement et de convolutions jusqu'à atteindre la résolution désirée. Les caractéristiques d'histogramme représentent une information de bas niveau facilement contrôlable par les dernières couches du réseau. Ainsi, l'information d'histogramme s et λ n'est

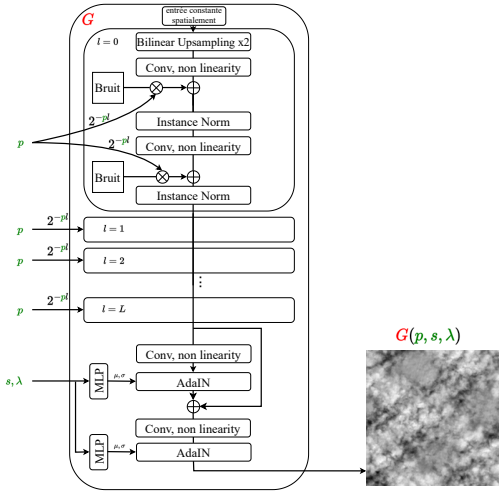


FIGURE 1 : Schéma du générateur

fournie au réseau que dans ses dernières couches et cela contribue à décorrélérer cette contrainte de celles de style et de pente spectrale.

Dans cette architecture 1, l'intégralité de l'information spatiale provient de réalisations de cartes de bruit gaussien à chaque niveau du réseau, de la résolution la plus grossière à la plus fine. La pente p est introduite dans notre générateur sous forme de pondération de la puissance de ces cartes de bruit. Toutes les échelles du réseau sont séparées par un sur-échantillonnage bilinéaire de facteur $2 \cdot 2$. À chaque échelle l du réseau, la pondération du bruit s'écrit $2^{-l \cdot p}$. Cette pondération en puissance apporte une information grossière mais correcte sur la décroissance souhaitée des hautes fréquences spatiales. De plus, cette injection d'information n'interfère pas avec le style désiré de l'image de sortie.

2.5 Entraînement

Nous entraînons notre réseau sur des images de résolution $256 \cdot 256$, sur 50000 itérations avec une taille de batch de 8 et l'optimiseur Adam avec un taux d'apprentissage de 10^{-4} .

3 Résultats

3.1 Données

Nous travaillons dans cet article sur deux jeux de données : une image de taille 4048^2 obtenue par la bande 12 (infrarouge) du satellite Sentinel-2, dans laquelle nous sélectionnons des sous-images aléatoires de taille $256 \cdot 256$ pour constituer notre base d'apprentissage, et une base de données constituée de 100 images de contenu en eau de stratocumulus non précipitant générées par le code Meso-NH du CNRM.

3.2 Résultats visuels

Notre méthode permet par son architecture stationnaire et convolutionnelle de produire des résultats d'extension arbitraire de texture par rapport à la taille des images produites à l'entraînement. Un motif périodique se remarque sur la figure d'élargissement 2 car ces artefacts de très basse fréquence ne sont pas visibles à l'échelle $256 \cdot 256$ utilisée au cours de l'entraînement pour superviser le style du générateur.



FIGURE 2 : Élargissement d'un facteur 4 dans la direction x

Nous montrons également en figure 3 (colorimétrie ajustée) des résultats de synthèse sur la base de contenu en eau de stratocumulus, conditionnellement aux statistiques physiques mesurées. Notons ici que notre architecture ne peut pas utiliser l'information spatiale contenue dans images de la base, puisqu'elle ne les perçoit que sous le prisme des trois statistiques calculées globalement sur l'image.

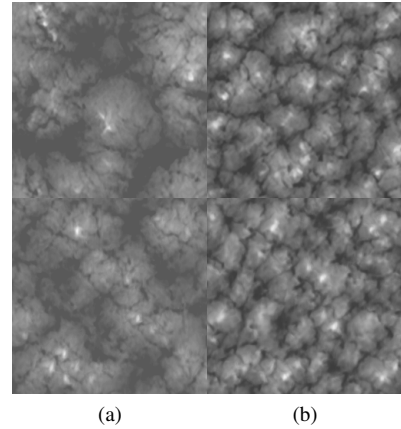


FIGURE 3 : Deux exemples de synthèse $G(p_I, s_I, \lambda_I)$ (image du bas) pour des paramètres calculés sur l'image I (en haut)

3.3 Contrôle physique

Nous explorons l'efficacité des contraintes physiques sur les données issues de l'image Sentinel-2.

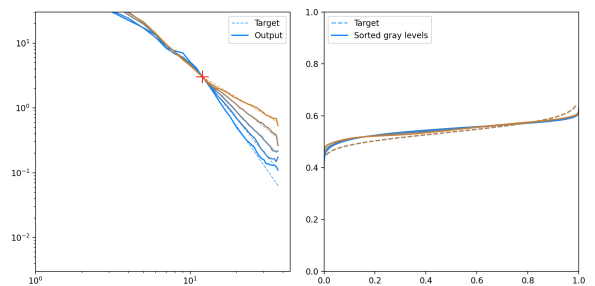


FIGURE 4 : Reconstruction d'images de pentes variables à contrainte d'histogramme fixée.

Dans les figures 4 et 5, sont affichés à gauche des profils azimutés (eq. 2) d'images (en échelle log-log), la croix correspondant au rayon de cassure r_0 . À droite sont tracés les niveaux de gris triés par ordre croissant, avec un axe des x normalisé. Dans ces deux figures, 5 expériences sont affichées (une couleur par expérience). Pour chaque expérience, le profil de spectre et d'histogramme idéal est affiché en pointillés pour des paramètres p, s, λ , et le profil calculé à partir de l'image $G(p, s, \lambda)$, en trait plein. Ces deux figures montrent qu'un

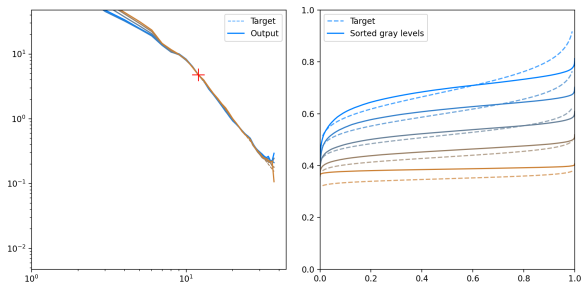


FIGURE 5 : Reconstruction d’histogrammes variables à contrainte de pente fixée

contrôle décorrélé de la pente spectrale et de l’histogramme est atteint, bien que la prescription d’histogramme manque de précision.

3.4 Influence des paramètres

Nous évaluons en figure 6 l’influence des trois paramètres p, s, λ sur les images générées en calculant $\frac{\partial G(p,s,\lambda)}{\partial p}$, $\frac{\partial G(p,s,\lambda)}{\partial s}$ et $\frac{\partial G(p,s,\lambda)}{\partial \lambda}$.

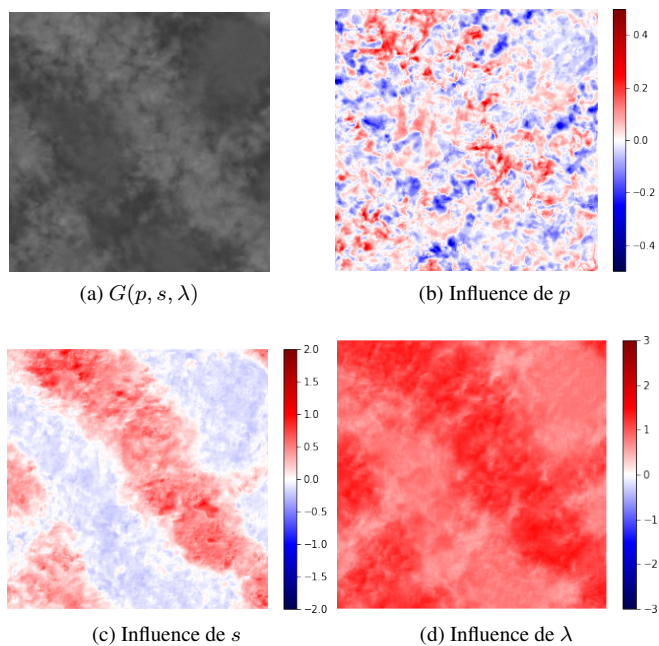


FIGURE 6 : Influence des paramètres physiques

Notons que λ contrôle la valeur moyenne, son gradient étant de même signe et régulier sur l’image ; s contrôle le contraste global de l’image, de manière régulière encore, et p contrôle la granularité de l’image avec une influence sur des structures plus fines dans l’image.

4 Conclusion et perspectives

Nous avons introduit une architecture et une supervision permettant de donner en entrée d’un réseau des paramètres physiques, et d’obtenir en sortie des images respectant les statistiques correspondantes. De plus, nous contrôlons le style des images générées. Les futurs travaux se porteront sur le

découplage du contrôle du style des images générées et du contrôle des paramètres physiques.

5 Remerciements

Ce travail est soutenu par l’Agence de l’Innovation de Défense et le projet ANR-19-CE40-0005 MISTIC.

Références

- [1] Pierre CHAINAIS, Emilie KOENIG, Véronique DELOUILLE et Jean-François HOCHEDÉZ : Virtual Super Resolution of Scale Invariant Textured Images Using Multifractal Stochastic Processes. *Journal of Mathematical Imaging and Vision*, 39(1):28–44, 2011.
- [2] Ricard DURALL, Margret KEUPER et Janis KEUPER : Watch your up-convolution : CNN based generative deep neural networks are failing to reproduce spectral distributions. *CoRR*, abs/2003.01826, 2020.
- [3] Bruno GALERNE, Yann GOUSSEAU et Jean-Michel MOREL : Random phase textures : Theory and synthesis. *IEEE Transactions on image processing*, 20(1):257–267, 2010.
- [4] Leon A. GATYS, Alexander S. ECKER et Matthias BEHTGE : Texture synthesis using convolutional neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15*, page 262–270, Cambridge, MA, USA, 2015. MIT Press.
- [5] Nicolas GONTHIER, Yann GOUSSEAU et Saïd LADJAL : High resolution neural texture synthesis with long range constraints. *J. of Mathematical Imaging and Vision*, 64: 478–492, 2022.
- [6] T. KARRAS, S. LAINE et T. AILA : A style-based generator architecture for generative adversarial networks. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 43(12):4217–4228, 2021.
- [7] Gregory M. LEWIS et Philip H. AUSTIN : An iterative method for generating scaling log-normal simulations. In *11th Conference on Cloud Physics, Univ. of British Columbia, Vancouver, BC, Canada*, 2002.
- [8] Ken PERLIN : An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.
- [9] Frédéric SZCZAP, Yahya GOUR et et AL. : A flexible three-dimensional stratocumulus, cumulus and cirrus cloud generator (3dcloud) based on drastically simplified atmospheric equations and the fourier transform framework. *Geosci. Model Dev.*, 7:1779–1801, 2014.
- [10] Victor VENEMA, Steffen MEYER et et AL. : Surrogate cloud fields generated with the iterative amplitude adapted fourier transform algorithm. *Tellus A*, 58A:104–120, 2006.