Transfert de style d'images par mise en correspondance multi-échelle et contrainte de patchs

Benjamin SAMUTH, David TSCHUMPERLÉ, Julien RABIN

Normandie Univ., UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France
{Benjamin.Samuth, David.Tschumperle, Julien.Rabin}@unicaen.fr

Résumé – Depuis l'avènement des réseaux de neurones convolutifs, les algorithmes de transfert de style entre images se sont considérablement améliorés. Cependant, ces méthodes requièrent souvent un temps d'apprentissage relativement long. Pour cette raison, des approches de traitement d'images sans apprentissage, utilisant des algorithmes à « patchs », ont récemment été proposées, cherchant à rivaliser esthétiquement avec les méthodes neuronales. Nous proposons ici une avancée dans cette direction, en introduisant une nouvelle méthode de transfert de style qui utilise une version contrainte et multi-échelle de l'algorithme de mise en correspondance de patchs *PatchMatch*, de manière à privilégier un échantillonnage uniforme des patchs de caractéristiques du style à différentes résolutions. Notre méthode permet par ailleurs d'associer avantageusement les paradigmes des méthodes à patchs et des réseaux de neurones en combinant la projection de patchs de couleurs selon la métrique de l'espace perceptuel défini par un réseau de neurones.

Abstract – With the advent of convolutional neural networks, algorithms for artistic style transfer between images have steadily improved considerably. However, these methods either requires long offline training or online optimization. This is why non-learning image processing approaches recently strove to propose patch-based algorithms able to aesthetically compete with neural methods. This paper goes one step further in this direction by introducing a new patch-based method for style transfer, using a constrained multi-scale version of the fast approximate nearest-neighbor algorithm *PatchMatch*, enforcing uniform sampling of style feature-patch at different resolutions. Our method also aims at mixing the patch-based and neural paradigms by enabling the embedding of color patches using the metric of the feature space defined by a neural network.

1 Introduction

Le transfert de style, ou stylisation d'images, est une technique récente de traitement d'images cherchant à améliorer la valeur esthétique des images en empruntant les caractéristiques stylistiques d'une image de *style* et en l'appliquant sur une image dite de *contenu* (Fig. 1). Une approche classique du transfert de style est de considérer le problème comme une spécialisation de la synthèse non-supervisée de texture par patchs [1], ou encore de transfert de texture [2]. L'idée est de préserver localement la cohérence d'un style (vu comme une texture à synthétiser), tout en le faisant correspondre à la structure spatiale d'une image de contenu. Dans la littérature, plusieurs variations de ce principe ont été proposées : [3] utilise une partition adaptative du contenu afin de recoller des patchs de style les plus larges possible dans des régions avec le moins de contenu géométrique ; [4] propose de synthétiser une image stylisée en utilisant la méthode d'optimisation de texture de [5], combinée avec un masque permettant de préserver le contenu pertinent ; [6] introduit un algorithme multi-échelle pour le problème de transport optimal semi-discret, et l'applique dans un espace de patchs, avec des applications en synthèse de texture et en transfert de style. Par ailleurs, la méthode que nous élaborons dans cet article tire plus largement son inspiration de plusieurs algorithmes de mise en correspondance basés patchs tels que [7] pour l'*inpainting*, [8] pour la génération de textures, ou encore [9] pour la génération d'images à partir d'un exemple.
Notre méthode est fondée sur la résolution d'un problème d'assignement « optimal » de patchs, tel qu'étudié dans [11], afin de conserver la diversité de la distribution des patchs de l'image de style. Nous nous inspirons également du travail pionnier de [12] qui introduit une classe de méthodes performantes de transfert de style utilisant des réseaux neuronaux profonds, tels

FIGURE 1 – Notre méthode de transfert de style s'adapte à différentes caractéristiques Φ. La contrainte d'occurrence que nous proposons permet à l'image de contenu de recevoir l'esthétique globale de celle de style. Plus de résultats sont consultables sur [10].

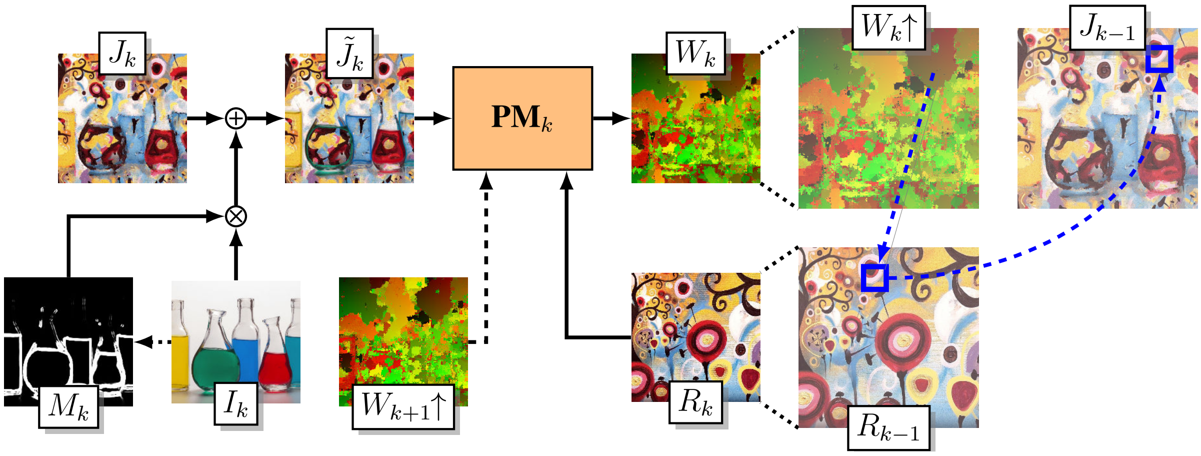


FIGURE 2 – Schéma de l'algorithme à une échelle intermédiaire k. Les étapes qui s'enchaînent sont, de gauche à droite, celles de l'injection de détails, de recherche des patchs plus proches voisins, du changement d'échelle de la carte de correspondance et la synthèse par agrégation de patchs. Nous n'injectons pas de détails à la première échelle $k = N$ et utilisons directement l'image d'origine à basse résolution. De la même manière, l'agrandissement d'échelle n'est pas nécessaire à la dernière résolution $k = 0$.

que VGG [13]. De tels réseaux sont en effet capables d'encoder à la fois le contenu géométrique à petite échelle et le style global, voire sémantique, d'une image. En optimisant les pixels de l'image d'entrée et préservant certaines caractéristiques (*features*) du réseau, [12] réussit à synthétiser des images stylisées convaincantes. De nombreuses variantes ont découlé de ces travaux fondateurs, améliorant les quelques points faibles de la technique initiale. Notons que la phase d'apprentissage requise par ces types de réseaux est souvent fastidieuse et nécessite en pratique une large base de données d'images.

L'algorithme rapide de transfert de style que nous présentons ici définit un schéma multi-échelle utilisant une contrainte d'occurrences originale pour la recherche des patchs les plus proches (Sec. 2). Notre méthode est également capable d'utiliser des caractéristiques de réseaux (Sec. 3) comme présentée dans la Figure 1, en permettant donc l'élaboration de modèles hybrides, géométriquement mieux explicables.

2 Description de la méthode proposée

Soit $I : \Omega_I \to \mathbb{R}^3$ l'image (couleur) de contenu (*input*) et R l'image de style (*référence*). On nomme Φ la méthode d'extraction de patchs de caractéristiques, telle qu'appliquée sur I, on ait $\Phi(I) : \Omega_I \to \mathbb{R}^{c \times \sigma^2}$ avec c la dimension des caractéristiques et $\sigma \times \sigma$ la taille des patchs. Soit F l'ensemble des cartes de correspondance $W : \Omega_I \to \Omega_R$, qui associent la coordonnée d'un patch de I à celle d'un patch de R. Au cœur de notre méthode se pose la question de trouver une correspondance optimale $W \in F$ entre $\Phi(I)$ et $\Phi(R)$. Le nouveau contenu est ensuite agrégé en l'image J par le calcul d'une moyenne spatiale des patchs de $\Phi(R) \circ W$ qui se superposent. Nous cherchons à calculer une carte des plus proches voisins (*Nearest Neighbor Field*, ou NNF) W^*, définie comme :

$$W^* \in \underset{W \in F}{\operatorname{argmin}} \sum_{p \in \Omega_I} ||\Phi(I)(p) - \Phi(R) \circ W(p)||^2 \qquad (1)$$

Les NNFs sont des éléments clés de certaines méthodes de traitements d'images (*inpainting* [7] ou synthèse de texture [2] par exemple). Cependant, leur obtention par la minimisation exacte de (1) souffre de plusieurs limitations auxquelles nous nous attarderons pour le transfert de style. Par souci de simplicité, nous omettrons ici l'opérateur de patchs de caractéristiques Φ. Son rôle sera étudié plus en détails dans la partie 3.

Présentation de l'algorithme. Afin de promouvoir la création de régions de patchs, c'est-à-dire des agglomérations de patchs spatialement continues au sein d'une carte de correspondance, nous construisons une représentation multi-échelle $\{I_k\}$ de I sur $N + 1$ différentes résolutions ($k \in \{0, \dots N\}$), avec un rapport d'échelle de r. D'une manière semblable à [7, 9], en utilisant un algorithme allant de basse à haute résolution, un patch capture la géométrie globale du contenu aux plus grossières échelles ainsi que les détails du style aux plus fines. Le cœur du transfert de style par patch à l'échelle k peut alors être exprimé comme la recherche des plus proches voisins, la mise à l'échelle de la carte de correspondance et l'agrégation des patchs, comme décrit en Fig. 2. Cette méthode ne requiert par ailleurs aucun apprentissage sur l'image de style.

À la sortie du module *PatchMatch*, le NNF courant est mis à l'échelle suivante (plus fine), par une méthode d'agrandissement conçue spécialement pour les cartes de correspondances, car les interpolations usuelles ne sont pas pertinentes pour traiter ce genre de cartes. Nous agrandissons directement le NNF plutôt que l'image agrégée car cela amènerait par la suite à une comparaison entre patchs de différentes résolutions (l'agrandissement direct de l'image en cours de synthèse aurait tendance à sur-lisser celle-ci), avec un effet négatif sur la recherche de patchs aux résolutions supérieures. Nous calculons donc le NNF agrandi de la façon suivante :

$$W{\uparrow^r}(p) = rW\left(\left\lfloor \frac{p}{r} \right\rfloor\right) + p - r\left\lfloor \frac{p}{r} \right\rfloor$$

Les opérations sont appliquées à chaque dimension spatiale de p. En pratique, W est un tableau 2D de coordonnées dans Ω_R. Le NNF résultant est alors donné comme initialisation du *PatchMatch* de l'échelle suivante, de manière à ce que la cohérence des régions de patchs de la carte calculée jusqu'ici

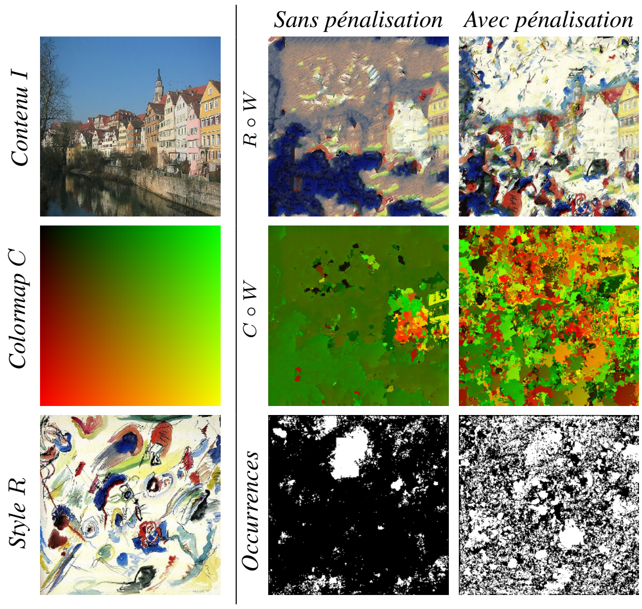


FIGURE 3 – Comparaison du dernier NNF avec/sans pénalisation des occurrences. L'ajout de la pénalisation améliore très clairement la diversité des patchs utilisés. (1ère et 2ème lignes) La couleur d'un pixel indique quel patch est agrégé de l'image (style ou *colormap*) à cette coordonnée. (3ème ligne) Un pixel blanc indique que le patch de style à cette position est utilisée au moins une fois.

persiste au travers des différentes échelles de synthèse. La toute première initialisation est choisie comme une permutation aléatoire de la carte identité afin de garantir une première recherche équilibrée en terme d'occurrences de patchs.

Soit $k \in \{N, \dots, 0\}$, l'indice d'échelle, allant de N (l'échelle la plus grossière) à 0 (l'échelle la plus fine). Nous définissons le meilleur NNF à l'échelle k comme :

$$W_k^* \in \underset{W \in F}{\operatorname{argmin}} \sum_{p \in \Omega} ||\tilde{J}_k(p) - R_k \circ W(p)||^2 \tag{2}$$

où $\tilde{J}_k := (1 - \max(M_k, \rho_k)) \odot J_k + \max(M_k, \rho_k) \odot I_k$ est la sortie de l'agrégation de patchs J_k à laquelle est injectée une partie de l'image I_k pour pousser l'algorithme à reconstruire sa géométrie à l'échelle suivante. Cette combinaison linéaire est pondérée ($\odot$ indique une multiplication pixel-à-pixel) par un masque M_k calculé à partir des gradients de I_k comme réalisé dans [6]. Ce masque est restreint par le paramètre $\rho_k \in [0, 1]$ qui contrôle la quantité de détails géométriques de I_k injectés. ρ_k décroît aux échelles plus fines afin de préserver les larges structures de l'image de contenu I.

Pénalisation d'occurrences. Pour le calcul d'une NNF cohérente, nous nous basons sur l'algorithme rapide d'approximation de plus proches voisins de patchs *PatchMatch* [14] (notre implémentation étant optimisée par l'utilisation de GPUs). L'un des problèmes connus de l'algorithme *PatchMatch* original est sa tendance à dupliquer les mêmes régions d'origine lorsque les images de patchs à comparer n'ont pas de contenu visuellement « compatible » entre elles. Ceci arrive fréquemment dans le cadre du transfert de style, puisque les images de contenu et de style peuvent être quelconques, donc très différentes. En pratique, ces répétitions de régions de patchs engendrent des artefacts de synthèse très visibles, que l'on peut souvent remarquer avec les méthodes [3, 8, 11]. Pour éviter cet effet de répétition visuelle, nous proposons une modification de l'algorithme *PatchMatch* par l'introduction d'une contrainte de pénalisation d'occurrences, permettant d'obtenir une meilleure distribution spatiale des patchs recopiés (Fig. 3). Cette contrainte joue un rôle essentiel pour obtenir des résultats de transfert de style visuellement plus fidèles au style. Nous nous inspirons de [6], où l'assignement optimal de patchs est forcé en utilisant un transport optimal semi-discret. La solution de ce problème consiste à pénaliser chaque norme carrée de (1) avec une variable scalaire $\lambda_W(y)$, variable duale qui est optimisée par une montée de gradient stochastique.

Soit W l'estimation courante du NNF donnée par *PatchMatch* et $\nu \in \mathbb{N}^{\Omega_R}$ les contraintes d'occurrences souhaitées, correspondantes à chaque patch dans W. Alors la contrainte d'occurrences s'implémente par la mise à jour de λ_W, à chaque itération de *PatchMatch*, de la façon suivante :

$$\begin{aligned} \lambda_W(y) &\leftarrow \lambda_W(y) + \delta \partial_y \lambda_W(y) \\ \partial_y \lambda_W(y) &= \frac{|\{p \mid y = W(p)\}|}{|\Omega_I|} - \frac{\nu(y)}{|\Omega_R|} \end{aligned} \tag{3}$$

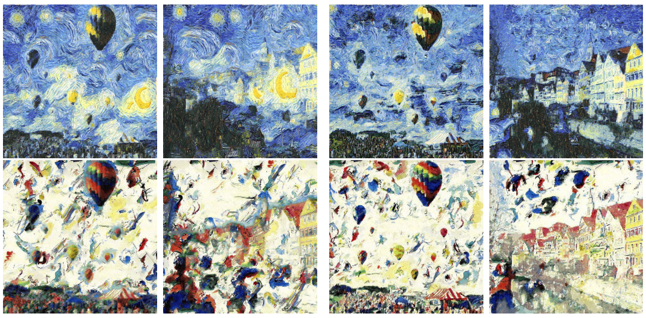
avec δ le pas de gradient. En théorie, nous devrions imposer $\nu(y) = 1$ afin de favoriser un échantillonnage le plus uniforme possible des patchs de référence, mais, en pratique, les conditions aux bords nécessitent d'autres contraintes sur certains patchs. La mesure (2) est également modifiée pour prendre en compte la variable duale λ_W optimale obtenue par (3) :

$$W_k^* = \underset{W \in F}{\operatorname{argmin}} \sum_{p \in \Omega} \frac{D_k(p)}{Var(D_k)} - \lambda_W \circ W(p) \tag{4}$$

avec $D_k(p) = ||\tilde{J}_k(p) - R_k \circ W(p)||^2$, que l'on divise par la variance afin de normaliser l'expression. Ainsi, lorsque $\lambda_W(y) > 0$, le patch y est favorisé, tandis que lorsque $\lambda_W(y) < 0$, il est pénalisé. Comme illustrée en Fig. 3, cette pénalisation d'occurrences évite efficacement les artefacts de copie durant le transfert (1ère ligne) et tend vers un échantillonnage spatialement plus homogène des patchs (3ème ligne). La taille des régions transférées (2ème ligne) décroît quand δ augmente.

Liens avec d'autres travaux. Notre approche est sensiblement différente d'autres méthodes de diversification, comme celle de la normalisation des scores de [9] puisque nous utilisons directement le nombre d'occurrences des patchs dans notre comparaison. Observons que, contrairement à [6, 11] qui requièrent un grand nombre d'itérations pour converger vers la solution optimale, notre méthode ne nécessite qu'un petit nombre d'itérations (5 à 8). Nous partageons des idées similaires à [8] dans lequel une carte d'occurrences est itérativement mise à jour pour pénaliser les distances de patchs. Leur stratégie n'est par contre pas adaptée au calcul massivement parallèle. Dans notre cas, la pénalisation ne s'applique qu'une fois les phases parallélisées de propagations ou de recherches aléatoires de patchs, propres à l'algorithme *PatchMatch*, ont été complétées.

(a) Images de contenu (b) Images de style



(c) Résultats de transfert de style avec Φ : RGB+Gradients (d) Résultats de transfert de style avec Φ : Caractéristiques VGG-16 (relu_1_1)

FIGURE 4 – Exemples de résultats de transfert de style sur des images 512×512. Notre méthode parvient à préserver l'esthétique globale du style par la recopie de larges régions de patchs. Les images ont été générées en utilisant les paramètres par défaut de la Sec. 3. Plus d'exemples sont disponibles sur [10].

3 Expériences et discussion

Pour les résultats présentés ici, nous avons fixé le nombre d'échelles à $N = 14$, le facteur d'échelle à $r = 1.3$, et la taille des patchs à $\sigma = 5$. L'injection de détails à chaque échelle est contrôlable avec le paramètre ρ_k dans l'Eq. (2), qui est par défaut choisi pour décroître linéairement de $\rho_{N-1} = 1$ (résolution la plus basse) à $\rho_0 = 0.5$ (résolution la plus fine). Ceci permet d'assurer que les parties saillantes de l'image de contenu restent toujours visibles à chacune des échelles.

Patch de caractéristiques. Le résultat du transfert de style à patchs peut être étendu à l'utilisation d'autres caractéristiques Φ que la couleur pour décrire les patchs. Notons qu'un transfert de couleur préalable entre l'image de style et de contenu permet généralement d'améliorer significativement les résultats de la synthèse, comme cela a déjà été remarqué dans les différents articles de la littérature de transfert de style à patchs. Par défaut, notre méthode (Figs. 3, 4c) utilise comme caractéristiques de patchs à la fois les canaux RGB ainsi que les gradients de l'image, concaténés pour obtenir une image à cinq canaux. En outre, nous avons étendu notre méthode pour utiliser des caractéristiques de patchs provenant de modèles de réseaux de neurones pré-entraînés. Motivés par l'approche de transfert de style neuronal de [12], nous utilisons par exemple la première couche de convolution de VGG-16 [13] ($c = 64$). Cet espace des caractéristiques « neuronales » montre des différences significatives avec les caractéristiques RGB (Fig. 4d). La plus notable d'entre elles est la capacité de la couche du réseau à intégrer le principe d'aplats de couleurs. Notons néanmoins que l'utilisation de différentes métriques des caractéristiques de patchs nécessite en pratique des réglages de paramètres différents, notamment au niveau du pas de gradient δ dans la pénalisation d'occurrences des patchs.

(a) Représentation neuronale [12] (b) Notre méthode

FIGURE 5 – Comparaison avec la méthode neuronale [12], avec relu_X_2, pour X=1,. . .,4. Grâce à l'utilisation de patchs (même limité à la couche relu_1_1), notre transfert de style est capable d'être fidèle à l'image de style en Fig. 4b.

Comparaison. Notre algorithme est capable de synthétiser des images de transfert de style convaincantes (Fig. 4, 5) avec un temps de calcul raisonnable comparé aux approches neuronales dont l'apprentissage est chronophage, en particulier si une large base de données est nécessaire. Plus d'exemples de comparaisons sont disponibles sur [10].

Implémentation. Notre méthode utilise un algorithme *PatchMatch* parallélisé qui contient notre pénalisation d'occurrence de patchs. Le temps de calcul sur des images 512×512 est aux alentours de 5 secondes en utilisant les canaux couleurs et les gradients, et de 70 secondes en utilisant l'espace des caractéristiques du réseau de neurones VGG-16 [13].

Remerciements. Ce travail est partiellement soutenu par le projet ANR-19-CHIA-0017.

Références

[1] M. ASHIKHMIN. "Synthesizing natural textures". In : *Proceedings of the 2001 symposium on Interactive 3D graphics*. 2001

[2] A. A. EFROS et W. T. FREEMAN. "Image quilting for texture synthesis and transfer". In : *Proc. of the 28th annual conf on CG and interactive tech.* 2001

[3] O. FRIGO, N. SABATER, J. DELON et P. HELLIER. "Split and Match : Example-Based Adaptive Patch Sampling for Unsupervised Style Transfer". In : *Proc. of the IEEE Conf. on CVPR*. 2016

[4] M. ELAD et P. MILANFAR. "Style transfer via texture synthesis". In : *IEEE Transactions on Image Processing* 26.5 (2017)

[5] V. KWATRA, I. ESSA, A. BOBICK et N. KWATRA. "Texture optimization for example-based synthesis". In : *ACM SIGGRAPH 2005 Papers*. 2005

[6] A. LECLAIRE et J. RABIN. "A Stochastic Multi-layer Algorithm for Semi-Discrete Optimal Transport with Applications to Texture Synthesis and Style Transfer". In : *Journal of Mathematical Imaging and Vision* (juill. 2020)

[7] A. NEWSON, A. ALMANSA, M. FRADET, Y. GOUSSEAU et P. PÉREZ. "Video inpainting of complex scenes". In : *Siam journal on imaging sciences* 7.4 (2014)

[8] A. KASPAR, B. NEUBERT, D. LISCHINSKI, M. PAULY et J. KOPF. "Self Tuning Texture Optimization". In : *Computer Graphics Forum* 34.2 (2015)

[9] N. GRANOT, B. FEINSTEIN, A. SHOCHER, S. BAGON et M. IRANI. "Drop the gan : In defense of patches nearest neighbors as single image generative models". In : *arXiv preprint arXiv :2103.15545* (2021)

[10] B. SAMUTH. *Page web de démonstrations*. 2022. URL : https://samuth211.users.greyc.fr/2022/StyleTransfer/

[11] J. GUTIERREZ, J. RABIN, B. GALERNE et T. HURTUT. "Optimal patch assignment for statistically constrained texture synthesis". In : *International Conf. on Scale Space and Variational Methods in Computer Vision*. Springer. 2017

[12] L. A. GATYS, A. S. ECKER et M. BETHGE. "Image Style Transfer Using Convolutional Neural Networks". In : *Proc. of the IEEE Conf. on CVPR*. 2016

[13] K. SIMONYAN et A. ZISSERMAN. "Very deep convolutional networks for large-scale image recognition". In : *arXiv preprint arXiv :1409.1556* (2014)

[14] C. BARNES, E. SHECHTMAN, A. FINKELSTEIN et D. B. GOLDMAN. "PatchMatch : A randomized correspondence algorithm for structural image editing". In : *ACM Trans. Graph.* 28.3 (2009)