Désoccultation d'images basée patches pour la synthèse de vues virtuelles

Pierre Buyssens, Maxime Daisy, David Tschumperlé, Olivier Lézoray

Désoccultation d'images basée patches pour la synthèse de vues virtuelles

Pierre Buyssens, Maxime Daisy, David Tschumperlé, Olivier Lézoray *
Université de Normandie, UNICAEN, ENSICAEN, GREYC UMR CNRS 6072
6 Bd. du maréchal Juin, 14050, CAEN
{prénom.nom}@unicaen.fr

Résumé

Dans ces travaux, nous proposons une approche d'inpainting basée sur les patches et la profondeur pour la désocculation des trous qui apparaissent lors de la synthèse de vues virtuelles de scènes RGB-D. L'information de profondeur est ajoutée à chaque étape clé de l'algorithme classique basé patch [3] afin de guider la synthèse des structures et des textures manquantes. Ces contributions aboutissent à un algorithme efficace en comparaison des approches de l'état de l'art (à la fois en terme de qualité visuelle et calculatoire), tout en ne requérant qu'un seul paramètre additionnel (facile à ajuster).

Mots Clef

Désoccultation à l'aide de la profondeur, Inpainting basé patch, synthèse de vues virtuelles RGB-D.

Abstract

In this paper we propose a depth-aided patch based inpainting method to perform the disocclusion of holes that appear when synthesizing virtual views from RGB-D scenes. Depth information is added to each key step of the classical patch-based algorithm from [3] to guide the synthesis of missing structures and textures. These contributions result in a new inpainting method which is efficient compared to state-of-the-art approaches (both in visual quality and computational burden), while requiring only a single easy-to-adjust additional parameter.

Keywords

Depth-Aided Disocclusion, Patch-based Inpainting, RGB-D Virtual View Synthesis.

1 Introduction et contexte

Les techniques 3DTV et le rendu indépendant d'un point de vue (*Free-Viewpoint Rendering* - FVR) sont devenues des technologies clés qui ont vu l'émergence de nouvelles expériences multimédias telles que le cinéma 3D, l'affichage 3D, la diffusion de vidéos 3D ...La tchenique de rendu d'image plus profondeur (*Depth Image Based Rendering* - DIBR) est ainsi devenue un élément majeur pour la synthèse de vues virtuelles, et consiste, en plus du rendu classique de l'image couleur, à récupérer la carte de profondeur attenante. Déformer l'image en fonction de la carte de profondeur (*warping*) afin de rendre la scène d'un nouveau point de vue conduit à la synthèse de vues virtuelles [7]. Le problème majeur qui se pose alors est celui des zones occultées : les zones de l'arrière-plan (*background* - BG) qui étaient cachées par des objets au premier-plan (*foreground* - FG) dans la vue originale doivent être rendues dans la vue synthétisée (Fig. 5, colonne de gauche). Dans de tels cas, les informations couleurs ainsi que l'information de profondeur sont manquantes et doivent etre remplies. Le remplissage de ces trous est connu sous le nom de *désocculation* et est un cas particulier du problème plus général d'*inpainting*.

Les méthodes de désoccultation proposées dans la littérature peuvent être séparées en deux groupes : les premières réalisent la désoccultation de l'image couleur et de la carte de profondeur en même temps [14], tandis que les secondes restaurent d'abord la carte de profondeur, puis l'utilise pour guider la restauration de l'image couleur. En particulier, les méthodes proposées dans [8], [4], [19] restaurent l'image couleur en supposant la carte de profondeur déjà restaurée. Comme souligné dans [19], la restauration de la carte de profondeur n'est pas une tâche aisée, mais est possible à l'aide d'algorithmes dédiés : restaurer dans un premier temps la carte de profondeur pour ensuite l'utiliser comme guide lors de la restauration de l'image couleur permet plus de souplesse pour la création d'algorithmes dédiés.

Dans cet article, nous considérons aussi que la carte de profondeur a déjà été restaurée. Nos travaux reposent sur l'algorithme de restauration basé sur les patches proposé dans [3]. Nous proposons de revisiter chaque étape clé de cet algorithme en y introduisant les informations de profondeur de manière naturelle et intuitive. Ces modifications ne font intervenir qu'un seul paramètre additionnel λ (comparé à l'algorithme classique [3]) facile à a juster en fonction des données. Ce paramètre de seuil λ discrimine les pixels adjacents selon leur profondeur respective selon qu'ils appartiennent au même objet ou non. En particulier, deux pixels adjacents p et q appartiennent au même objet (FG ou BG) si $|\text{depth}(p) - \text{depth}(q)| < \lambda$.

2 État de l'art

Dans cette section, nous procédons à l'état de l'art des méthodes proposées dans la littérature traitant de la désoccul-

*Ces recherches sont financées par le projet Action 3DS

tation de l'image couleur en supposant la carte de profondeur déjà restaurée. À noter que nous ne traitons ici que le cas de la synthèse de vue par translation horizontale. De récents travaux traitent spécifiquement du problème de la synthèse de vue par translation du point de vue en profondeur (zoom de la scène) [12, 11] mais ils sont clairement en dehors du sujet.

2.1 Notations et définitions

Une image couleur est considérée comme une fonction $I : \mathcal{I} \rightarrow \mathbb{R}^n$ où $\mathcal{I}$ définit le support de l'image, et $n =$ pour les images couleur usuelles. De façon similaire, une carte de profondeur est considérée comme une fonction $J : \mathcal{J} \rightarrow \mathbb{R}$ où $\mathcal{J}$ est le support de la la carte de profondeur. Dans la suite, une image RGB-D est considérée comme une paire (I, J) partageant le même support $\mathcal{I} = \mathcal{J}$. L'image couleur I contient un ensemble de trous $\Omega = \{\Omega_1, \ldots, \Omega_N\}$. Ω représente le masque de I qui doit être restauré (i.e., les pixels inconnus devant être re-synthétisés), et $\delta\Omega$ est le contour du masque.

Étant donné que toutes les méthodes de l'état de l'art utilisent des patches pour la désoccultation de l'image couleur, nous introduisons ici la notion de patch. Un patch Ψ_p centré sur le pixel p est considéré comme une fonction $\Psi_p : \mathcal{N}_p \rightarrow \mathbb{R}^n$ où $\mathcal{N}_p \in \mathcal{I}$ est le support carré du patch Psi_p et n est la dimension de l'image ($n = 3$ pour un patch issu d'une image couleur). À noter que ce patch peut être masqué (i.e., certains de ses pixels sont inconnus). Dans la suite, $|\mathcal{N}_p|$ représente la taille du support d'un patch (i.e., le nombre de pixels), tandis que $|\Psi_p|$ représente le nombre de pixels *connus* du patch Psi_p. Dans le cas d'un patch incomplet (i.e., qui contient des pixels inconnus), on a alors $|\Psi_p| < |\mathcal{N}_p|$. Dans la suite, $\Psi_{\hat{p}}$ désigne le patch minimisant la métrique :

$$\Psi_{\hat{p}} \;=\; \left\{ \Psi_q | \underset{q|\mathcal{N}_q \cap (\mathcal{I}-\Omega)}{\arg\min}\; d(\Psi_p, \Psi_q) \right\}$$

La distance d la plus utilisée afin de comparer deux patches est la somme des différences au carré (*Sum of Square Differences* ou SSD) :

$$d_{SSD}(\Psi_p, \Psi_q) \;=\; \sum_{v \in (\mathcal{N}_p \cap (\mathcal{I}-\Omega))} \|\Psi_p(v) - \Psi_q(v+p-q)\|^2 \tag{1}$$

Finalement, nous désignons par $\mathcal{W} : \mathcal{I} \rightarrow \mathcal{I}$ le processus de transformation (*warping*) qui transforme une image couleur *originale* I_o en une image couleur *synthétisée* $I_s = \mathcal{W}(I_o)$. Cette fonction transforme également (et de la même manière) la carte de profondeur *originale* J_o en une carte de profondeur *synthétisée* $J_s = \mathcal{W}(J_o)$. Dans la suite, nous supposons que la carte de profondeur synthétisée J_s est entièrement connue (i.e., elle a déjà été restaurée à l'aide d'un algorithme dédié). À noter que, tout au long de cet article, la synthèse d'une nouvelle vue est effectuée à l'aide des équations de transformations 3D standards [17].

2.2 Désoccultation d'images couleur dans la littérature

Dans cette section, nous passons en revue les méthodes proposées dans l'état de l'art pour la désoccultation d'images couleur. Étant donné que la plupart des méthodes reposent sur l'algorithme de Criminisi *et al.* [3], algorithme pionnier d'inpainting basé sur les patches, nous commençons par en décrire les principales étapes. À noter que nous ne parlerons pas ici des méthodes variationnelles dédiées à la restauration [1] étant donné que celles-ci sont largement ignorées dans le domaine de la désoccultation, la majorit'e des méthodes proposées dans l'état de l'art reposant sur [3].

Squelette de l'algorithme d'inpainting basé sur les patches. En 2004, un algorithme d'inpainting majeur a été proposé par Criminisi *et al.* dans [3]. Basé sur les travaux traitant de la synthèse de texture [6], cette approche gloutonne propose d'utiliser une notion de priorité pour guider l'ordre de remplissage, et consiste principalement en l'itération des 4 étapes suivantes :

1. Un terme de priorité est assigné à chaque pixel $p \in \delta\Omega$ où $\delta\Omega$ est le bord *extérieur* du masque Ω, et est calculé par
$$P(p) \;=\; C(p) \times D(p) \tag{2}$$
où $C(p)$ et $D(p)$ sont respectivement le terme de *confiance* et le terme de *données*. Le premier reflète le nombre de données fiables (connues) dans $\mathcal{N}_p$, tandis que le second est basé sur un gradient local dans $\mathcal{N}_p$ et reflète les structures qui entrent dans le masque. Ils sont définis dans [3] par :
$$C(p) \;=\; \frac{\sum_{q \in (\mathcal{N}_p \cap (\mathcal{I}-\Omega))} C(q)}{|\mathcal{N}_p|} \tag{3}$$
$$D(p) \;=\; \frac{\nabla I_p^{\perp} \cdot \mathbf{n}_p}{\alpha} \tag{4}$$
où α est un facteur de normalisation (qui peut en fait être ignoré), $\mathbf{n}_p$ est le vecteur unitaire orthogonal au bord du masque $\delta\Omega$ en p, et $\perp$ désigne l'opérateur orthogonal. Le pixel $t \in \delta\Omega$ ayant la priorité maximale est choisi comme pixel cible (*target*).
2. Soit le patch cible Ψ_t centré en t, la deuxième étape consiste à trouver dans $\bar{\Omega}$ le patch $\Psi_{\hat{t}}$ qui minimise la SSD sur la partie connue de Ψ_t (Eq. 1).
3. La troisième étape consiste en la recopie des pixels de $\Psi_{\hat{t}}$ autour de t dans Ω :
$$\Psi_t(q) \;=\; \Psi_{\hat{t}}(p) \mid q - t = p - \hat{t}, \forall q \in \mathcal{N}_t \cap \Omega \tag{5}$$
4. La dernière étape consiste en la mise à jour du contour $\delta\Omega$ ainsi que des termes de *données* et de *confiance*.

À noter que de nombreuses modifications ont été apportées à cet algorithme initial dans des tentatives pour l'améliorer à des fins d'inpainting générique. Une revue sur le sujet peut être trouvée dans [9].

Désoccultation d'image couleur. Dans cette section, nous nous concentrons sur les modifications apportées à l'algorithme d'inpainting basé sur les patches pour la désoccultation d'images couleur. Deux angles d'attaque ont principalement été étudiés afin d'ajouter l'information de profondeur à l'algorithme initial : (a) modifications du terme de priorité (étape 1 de l'algorithme de Criminisi *et al.* détaillé plus haut), et (b) modifications du processus de recherche du meilleur patch (étape 2).

(a) Modifications du terme de priorité : Les auteurs de [4] proposent d'ajouter un troisième terme multiplicatif au terme de priorité $P(p) = C(p) \times D(p) \times L(p)$ où $L(p)$ est un terme de régularité de profondeur, défini comme l'inverse de la variance de profondeur du patch Ψ_p^d centré en p :

$$L(p) = \frac{|\mathcal{N}_p|}{|\mathcal{N}_p| + \sum_{q \in (\mathcal{N}_p \cap (\mathcal{I}-\Omega))} \left(\Psi_p^d(q) - \overline{\Psi_p^d}\right)^2} \quad (6)$$

où Ψ_p^d est le patch défini sur la carte de profondeur J_s centré en p, et $\overline{\Psi_p^d}$ est la profondeur moyenne de Ψ_p^d. Ce terme additionnel favorise les pixels p qui sont dans des régions homogènes ($L(p) \simeq 1$), et laisse les pixels se trouvant près de la bordure d'un objet ($L(p) \ll 1$) pour la fin de l'inpainting.

De façon similaire, un terme additionnel au terme de priorité est proposé dans [10] :

$$P(p) = C(p) \times D(p) \times \left(1 - \frac{\overline{\Psi_p^d}}{z_{max}}\right) \quad (7)$$

où $\overline{\Psi_p^d}$ est la profondeur moyenne des pixels de Ψ_p^d et où z_{max} désigne le maximum global des profondeurs de la carte de profondeur.

Dans [13], le terme de priorité est inchangé, mais l'ensemble des pixels $p \in \delta\Omega$ est discrétisé entre pixels de l'avant-plan et pixels de l'arrière-plan, et seuls ceux appartenant à l'arrière-plan recoivent une priorité effective (les autres ayant une priorité de 0).

De manière similaire, les auteurs de [8] proposent de mettre à 0 la priorité des pixels se trouvant sur le bord droit du trou si la caméra a bougé de droite à gauche. L'idée derrière cette astuce est que, dans ce cas, les trous apparaissent à gauche des objets du premier-plan, et en mettant à 0 leur priorité, cela évite à l'inpainting de commencer du côté des objets du premier-plan. En plus de ce schéma ne donnant des priorités qu'à un seul côté du trou, les auteurs de [8] proposent également d'utiliser des tenseurs de structure 3D [5] pour le calcul du terme de données en y incluant l'information de profondeur. Ce terme de données repose ainsi à la fois sur des caractéristiques couleur et structurelle.

Les auteurs de [18] proposent de modifier l'ordre de remplissage de sorte que l'algorithme commence à partir de l'arrière-plan tout en favorisant la continuation des structures qui entrent dans le masque :

$$P(p) = F(p) \cdot D(p) \cdot M(p) \cdot \frac{|\Psi_p|}{|\mathcal{N}_p|} \quad (8)$$

où $F(p)$ est une fonction binaire telle que $F(p) = 0$ si p appartient au premier-plan, et $F(p) = 1$ si p appartient à l'arrière-plan, et $M(p)$ est un terme additionnel calculé en p par :

$$M(p) = \frac{\sum_{r \in \mathcal{N}_p \cap \Omega} \sum_{q \in (\mathcal{N}_p \cap (\mathcal{I}-\Omega))} e^{-\frac{(J(r)-J(q))^2}{2\sigma}}}{|\Psi_p| \left(|\mathcal{N}_p| - |\Psi_p|\right)}$$

Finalement, les auteurs de [19] proposent de calculer le terme de priorité par :

$$P(p) = C(p)^\alpha \cdot D(p)^\beta \cdot E(p)^\gamma \quad (9)$$

où $C(p)$ et $D(p)$ sont les termes de confiance et de données inchangés (Eq. 3), $E(p)$ est l'inverse de la carte de disparité, et $\{\alpha, \beta, \gamma\}$ sont des hyperparamètres de pondération fixés par les auteurs. Au delà de ces hyperparamètres, l'idée derrière le terme additionnel $E(p)$ est de donner une priorité supérieure au pixels de l'arrière-plan (de faibles valeurs de disparité signifient de grandes valeurs de $E(p)$) par rapport à ceux du premier-plan.

(b) Modifications du processus de recherche du meilleur patch : Au delà des modifications liées au terme de priorité, la plupart des méthodes de la littérature proposent également de modifier la façon dont le meilleur patch est cherché/trouvé (étape 2 de l'algorithme de Criminisi *et al.* résumé à la section 2.2).

Les auteurs de [4] et [8] incorporent la carte de profondeur au calcul de la SSD en tant que quatrième canal. Tandis que les auteurs de [4] pondèrent cet ajout à l'aide d'un paramètre de poids additionnel, les auteurs de [8] considèrent la carte de profondeur comme ayant autant de poids que chacun des 3 canaux de l'image couleur. À noter que ces derniers récupèrent les 5 meilleurs patches et réalisent une combinaison de ces patches pondérés par leurs SSD respectives.

Ce schéma de recherche du meilleur patch favorise la sélection de patches ayant une profondeur similaire au patch cible Ψ_t sans pour autant éviter la sélection d'un patch du premier-plan.

Un autre schéma consiste à restreindre la recherche du meilleur patch aux patches ayant une profondeur inférieure. Ce schéma est employé par les auteurs de [19] et [13] avec un paramètre de tolérance ϵ fixé par les auteurs. En restreignant la recherche du patch candidat aux patches ayant une profondeur moyenne inférieure (ou égale), il devient impossible de procéder à la désoccultation du l'arrière-plan avec des données issues du premier-plan, ce qui est une propriété désirée.

Finalement, les auteurs de [18] formulent le processus de matching de patch comme une combinaison multiplicative

entre deux termes de similarité. Le premier, calculé sur la partie non masquée de Ψ_t, est lié à la fois à la similarité entre couleur et profondeur, tandis que le second, calculé sur la partie masquée de Ψ_t n'est lié qu'à la similarité entre profondeurs.

3 Algorithme d'inpainting revisité

Dans cette section, nous détaillons la méthode que nous proposons pour la désoccultation de l'image couleur I_s. En particulier, nous proposons de revisiter chacune des 3 étapes principales de l'algorithme de Criminisi *et a.*[3] détaillé plus haut.

3.1 Terme de priorité prenant en compte la profondeur

Lors de précédents travaux [2], nous avons montré que le terme de priorité, et tout particulièrement le terme de données, est un facteur très sensible. Le modifier en ajoutant des termes supplémentaires liés à la profondeur, comme ce qui est fait par la majorité des travaux de l'état de l'art, tend à diminuer sa robustesse. De plus, plusieurs approches de l'état de l'art introduisent des termes de pondération qui sont, en pratique, difficiles à régler.

Dans cet article, nous proposons d'éviter l'ajout de termes supplémentaires au terme de priorité initial $P(p) = C(p) \times D(p)$. Comme détaillé ci-dessous, nous n'introduisons pas la notion de profondeur dans le terme de *données* $D(p)$, mais nous introduisons celle-ci dans le terme de *confiance* de manière naturelle et intuitive.

Premièrement, nous choisissons le terme de données précédemment proposé dans [2] fondé sur des tenseurs de structures :

$$D(p) = \|\mathbf{G}_p \, \overrightarrow{n_p}\| \tag{10}$$

où $\overrightarrow{n_p}$ est le vecteur unitaire orthogonal au contour du masque en p, $\mathbf{G}_p$ est la moyenne pondérée des tenseurs de structure estimés sur la partie non masquée du patch cible Ψ_p :

$$\mathbf{G}_p = \sum_{q \in (\mathcal{N}_p \cap (\mathcal{I}_s - \Omega))} w_q \, \overrightarrow{\nabla I_q} \overrightarrow{\nabla I_q}^T$$

et w est une fonction Gaussienne 2D normalisée centrée en p. Ce terme de *données*, comme montré dans [2], donne plus de robustesse au terme de priorité que les différentes versions du terme de données précédemment proposées dans la littérature. Ce terme de données ne contient donc pas d'informations liées à la profondeur.

Deuxièmement, nous faisons une constatation simple : le terme de *confiance* est supposé compter l'information *fiable* (pixels connus) autour d'un pixel p, de sorte que ces mêmes pixels vont être utilisés pour l'étape de recherche du meilleur patch. À ce stade, il paraît donc contre-intuitif de compter des pixels appartenant au foreground comme *fiables* pour inpainter des pixels de l'arrière-plan. Nous proposons donc de définir comme *fiables* les pixels se trouvant à une profondeur similaire à la profondeur du pixel cible :

$$C(p) = \frac{1}{|\mathcal{N}_p|} \sum_{\substack{q \in (\mathcal{N}_p \cap (\mathcal{I} - \Omega)) \\ |J_s(p) - J_s(q)| < \lambda}} C(q) \tag{11}$$

avec $C(p) = 1, \forall p \in \bar{\Omega}$.

De sorte que ce terme de priorité puisse fonctionner correctement (et en particulier le terme de confiance que nous proposons ici), une légère modification est faite concernant le calcul du contour du masque : au lieu de considérer le contour *extérieur* du masque, nous considérons ici le contour *intérieur*. Cette légère modification fait que, pour un contour du masque se trouvant à la frontière entre l'arrière-plan et le premier-plan, les pixels du contour se trouveront alors dans l'arrière-plan. Les pixels voisins appartenant au premier-plan ne seront alors pas pris en compte grâce au terme de confiance proposé ci-dessus.

La figure 1 montre les avantages de ce terme de confiance et détaille le calcul des termes de priorité pour tous les pixels appartenant au contour d'un trou. Étant donné que les pixels du contour se trouvant près du premier-plan (la tasse) se trouvent dans l'arrière-plan, leur terme de confiance est égal à 0 (Figure 1, quatrième image). Ces pixels ont dont un terme de priorité nul, et sont donc considérés par l'algorithme à la toute fin du processus.

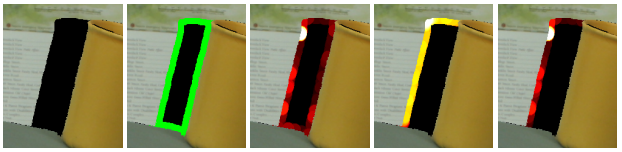

FIGURE 1 – Illustration de l'effet du terme de confiance proposée. De gauche à droite : masque à inpainter, bords intérieurs du masque (en vert), et en couleur chaudes, les termes de *données*, *confiance*, et les termes de priorité finaux. Étant donné que les pixels se trouvant sur le bord de la tasse ont un terme de *confiance* nul, leur priorité est également nulle, et le processus d'inpainting commence par l'arrière-plan.

3.2 Recherche du meilleur patch prenant en compte la profondeur

La stratégie de recherche du meilleur patch est composée de 3 étapes :

— Étant donné que les données de l'image couleur synthétisée I_s sont calculées par transformation à partie de l'image originale I_o, nous avons fait l'hypothèse (raisonnable) que la plupart des données de I_s peuvent se retrouver dans I_o. Partant de cette constatation, nous contraignons la recherche du meilleur patch (Eq. 1) aux patches de I_o. Grâce à la carte d'offsets (i.e., carte de disparités) utilisée lors du warping initial, les fenêtres de recherche peuvent facilement être calculées dans I_o en prenant l'inverse des offsets.

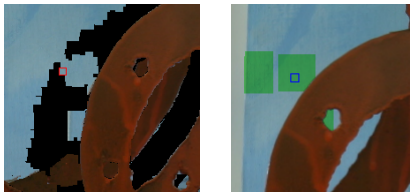

FIGURE 2 – Illustration de la méthode de recherche proposée. Gauche : trou à inpainter (en noir), avec le patch cible en rouge. Droite : espace de recherche (en vert) dans la vue originale I_o, avec le meilleur patch trouvé représenté par le carré bleu. Voir texte pour plus de détails.

— Seuls les patches ayant une profondeur égale à la profondeur de t ($\pm\lambda$) sont les patches candidats (étape similaire à celle utilisée dans [13, 19]) :

$$\Psi_{\hat{t}} = \left\{ \Psi_p \in I_o \middle| \underset{|J_o(p)-J_s(t)|<\lambda}{\arg\min} d_{SSD}(\Psi_t, \Psi_p) \right\} \tag{12}$$

— La zone de recherche dans I_o est *divisée* en de nombreuses sous-fenêtres selon la méthode précédemment proposée dans [2]. En bref, cette méthode propose d'utiliser les sources précédentes (i.e., les emplacements des meilleurs patches précédemment recopiés) comme zones de recherche prioritaire pour la recherche courante du meilleur patch (plus de détails dans [2]).

De nombreux avantages découlent de ce schéma de recherche : 1) Grâce à la méthode multi-fenêtrée, la cohérence globale de la partie reconstruite est mieux préservée qu'avec un schéma de recherche classique fondé sur une seule fenêtre centrée en t. 2) Étant donné que les patches sont cherchés uniquement dans I_o, tous les patches candidats sont complets (i.e., aucun d'entre eux n'a de pixels manquants). 3) Étant donné que les patches candidats ont une profondeur similaire t, tous les patches qui sont sans rapport avec le patch cible ne sont pas considérés : les patches du premier-plan sont automatiquement écartés lorsque le patch cible est un patch appartenant à l'arrière-plan. 4) Étant donné les points 1 et 3 tout juste énoncés, la zone de recherche est fortement réduite. La recherche du meilleur patch étant connue pour être le goulot d'étranglement (en termes de temps de calcul) des méthodes basées patches, la réduire a un impact non négligeable sur la charge de calcul de l'algorithme entier.

La figure 2 (droite) montre la zone de recherche (en vert) pour un patch cible appartenant à l'arrière-plan (carré rouge dans l'image de gauche). Les sous-fenêtres proviennent de patches précédemment copiés, et ne contiennent pas les anneaux qui appartiennent au premier-plan. Le meilleur patch est représenté par le carré bleu (droite).

3.3 Copie des pixels manquant prenant en compte la profondeur

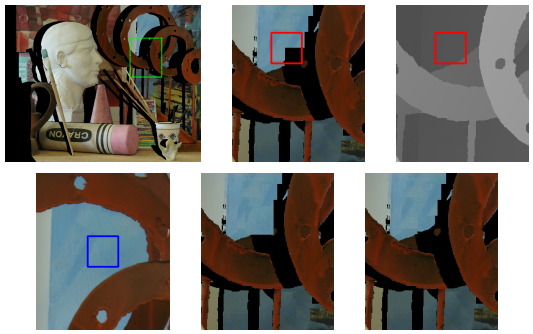

FIGURE 3 – Illustration de la recopie en fonction de la profondeur. Haut, de gauche à droite : Image à inpainter, zoom sur l'image avec le patch cible en rouge, carte de profondeur correspondante. Bas, de gauche à droite : Meilleur patch trouvé dans I_o en bleu, résultat de la copie classique, et résultat de la copie en fonction de la profondeur. Aucun pixel de couleur bleue n'est recopié dans l'anneau (étant donné que leurs profondeurs sont différentes), contrairement au schéma de copie classique (deuxième ligne, image du milieu).

Une fois que le meilleur patch a été trouvé, les pixels manquants de Ψ_t sont remplis à l'aide des pixels correspondants dans $\Psi_{\hat{t}}$ (Eq. 5). À notre connaissance, toutes les méthodes proposées dans la littérature reposent sur ce schéma de recopie. Celui-ci fonctionne bien lorsque le trou à inpainter n'est entouré que d'arrière-plan et d'un seul objet appartenant au premier-plan (étant donné que le trou ne doit être restauré qu'avec des profondeurs d'arrière-plan). Dans des cas plus complexes où plusieurs objets à différentes profondeurs se chevauchent, il se peut que le trou à inpainter doive être rempli avec des profondeurs d'arrière-plan et de premier-plan *intermédiaire* (objet se situant entre le fond et l'avant-plan). Dans de tels cas, le schéma classique de copie est clairement insuffisant étant donné que les profondeurs sous-jacentes du masque ne sont pas prises en compte.
Afin de répondre à ces cas complexes, nous proposons un schéma de copie prenant en compte la profondeur sous-jacente au masque, qui ne copie les pixels de $\Psi_{\hat{t}}$ vers ceux de Ψ_t que si leurs profondeurs respectives sont proches ($\pm\lambda$) :

$$\Psi_t(q) = \Psi_{\hat{t}}(p) \left| \begin{array}{l} q - t = p - \hat{t}, \forall q \in \mathcal{N}_t \cap \Omega \\ |J_o(p) - J_s(q)| < \lambda \end{array} \right. \tag{13}$$

La figure 3 illustre les bénéfices de ce schéma de copie où le masque à inpainter contient à la fois des données d'arrière-plan mais aussi de l'anneau masqué. Grâce à la stratégie de copie proposée, aucun pixel d'arrière-plan n'est recopié dans une partie correspondant au premier-plan masqué (et vice-versa). À notre connaissance, aucune

méthode de l'état de l'art n'arrive à inpainter proprement le masque de la figure 3.

3.4 Traitement de l'aliasing

Un problème important lorsque l'on traite à la fois des images de profondeurs et de couleur, est que la couleur d'un objet peut souvent *baver* sur un autre ou sur l'arrière-plan. Alors que la frontière entre objets est bien nette dans la carte de profondeur, elle peut s'étaler sur une largeur de plusieurs pixels dans l'image couleur (Figure 4, première ligne). La couleur des pixels appartenant à cette *bande étroite* qui appartiennent à l'arrière-plan (selon la carte de profondeur) est alors composée d'un mélange de couleurs d'arrière-plan et de premier-plan (i.e., *alpha matting*). Si ce phénomène n'est pas proprement pris en compte lors du processus d'inpainting, l'algorithme va copier les couleurs de ces pixels, et des composantes couleur de l'arrière-plan (ou de premier-plan) peuvent alors être copiées à des endroits inappropriés (Figure 4, troisième ligne, image du milieu).

FIGURE 4 – Illustration du problème de la *bande étroite* et de la solution proposée. Première ligne : Zoom sur une image avec (gauche) et sans (droite) l'objet du premier-plan (enlevé en fonction de sa profondeur). On peut observer des traces importantes de la couleur de l'objet. Deuxième ligne : Superposition des bandes étroites (en vert) synthétisée (gauche) et originale (droite) sur les images couleur. Troisième ligne : une partie masquée de l'image *Art* (gauche avec le masque en rouge pour la visualisation), le résultat de l'inpainting sans (image du milieu) et avec (droite) prise en compte du processus de *bande étroite* proposé.

Afin de pallier ce problème d'*aliasing*, nous proposons de calculer d'abord la *bande étroite* extérieure pour les cartes de profondeur synthétisée et originale :

$$NB_s^+ = \mathcal{T}(\delta(J_s) - J_s, \lambda)) \quad (14)$$

$$NB_o^+ = \mathcal{T}(\delta(J_o) - J_o, \lambda)) \quad (15)$$

où $\delta(\cdot)$ et $\mathcal{T}(\cdot, \lambda)$ sont respectivement les opérateurs de dilatation et de λ-seuillage. La taille de l'élément structurant pour la dilatation dépend ici essentiellement de la résolution des images (fixé à 5 dans la suite de cet article).

La figure 4 (deuxième ligne) montre la superposition des *bandes étroites* NB_s^+ et NB_o^+ (en vert) sur les images couleur synthétisée (gauche) et originale (droite).

Ces *bandes étroites* sont incorporées au sein de l'algorithme comme suit. Étant donné un pixel cible t, deux cas distincts apparaissent :

— Si $t \notin \text{NB}_s^+$, le pixel cible t ne se trouve pas à la frontière entre l'arrière-plan et le premier-plan. Dans ce cas il faut éviter la recopie des pixels *bavant* dans l'arrière-plan. La recherche du meilleur patch $\Psi_{\hat{t}}$ est donc restreinte aux patches $\Psi_p \in I_o$ ne contenant pas de pixels de NB_o^+.

— Si $t \in \text{NB}_s^+$, le pixel cible t appartient à la frontière entre l'arrière-plan et premier-plan, et la recopie de pixels *bavant* sur l'arrière-plan est autorisée (avec la possibilité de recopier la transition douce entre objets). La recherche du meilleure patch $\Psi_{\hat{t}}$ est alors effectuée sans restriction sur NB_o^+.

Ce mécanisme simple évite (1) la copie des mélanges de couleur lorsque celle-ci n'est pas désirée, tout en (2) l'autorisant à la frontière entre l'arrière-plan et le premier-plan. La figure 4 (troisième ligne, image de droite) montre les bénéfices de ce mécanisme : les artefacts présents (image du milieu) ont disparus. À noter que, comme les cartes de profondeur J_o et J_s sont complètement connues, les *bandes étroites* NB_s^+ et NB_o^+ ne sont pas modifiées lors de l'algorithme, et ce processus peut ainsi être implémenté à l'aide d'images intégrales pour un coût marginal.

4 Évaluation

Nous comparons notre méthode de désoccultation d'image couleur à deux méthodes représentatives de l'état de l'art [8, 19]. Nous utilisons les images de la base de données [16, 15] qui consiste en une collection de paires d'images stéréoscopiques dont les tailles varient de 1.4M à 6M pixels. La synthèse est effectuée de la vue_1 à la vue_0 pour les données de l'ensemble 2014, tandis qu'elle est effectuée de la vue_5 à la vue_1 pour le reste. Nous fixons $\lambda = 4$ pour toutes les expérimentations.

La figure 5 montre des résultats qualitatifs obtenus à l'aide de notre méthode ainsi que des extraits d'images reconstruites obtenus avec les méthodes proposées dans [8] et [19]. Afin de se comparer de manière équitable, nous utilisons les même cartes de profondeur sous-jacentes pour chaque méthode : la vérité terrain (disponible dans la base de données) pour les images *Art*, *Dolls*, *Midd2* et *Moebius* (lignes 2, 4, 5, et 6), et la carte de profondeur issue de notre algorithme dédié de désoccultation de cartes de profondeur [1] pour les images *Adirondack* et *Backpack* (lignes

1. Un article présentant un algorithme dédié à la désoccultation de cartes de profondeur est présent dans les actes de RFIA'2016.

1 et 3). Les paramètres de chaque méthode ont été ajustés manuellement pour obtenir les meilleurs résultats visuels.
Grâce aux ajustements de l'algorithme d'inpainting par patch que nous avons proposé pour la prise en compte de la profondeur, nos résultats ne montrent aucune incohérence majeure ni d'artefact. Le schéma de copie en fonction de la profondeur évite que des données de l'arrière-plan ne soient copiés sur des objets du premier-plan, même quand l'image est composée de plusieurs objets à différentes profondeurs qui se chevauchent. De plus, le terme de priorité que nous proposons est robuste et permet une bonne reconstruction à la fois des structures et des textures avec une qualité égale ou supérieure à celles obtenues avec les méthodes de l'état de l'art.
Finalement, notre méthode est (relativement) rapide en pratique (approximativement 1500 pixels/seconde sur un seul thread), et peut facilement être parallélisée (notamment la partie concernant la recherche du meilleur patch).

Références

[1] Pablo Arias, Gabriele Facciolo, Vicent Caselles, and Guillermo Sapiro. A variational framework for exemplar-based image inpainting. *International journal of computer vision*, 93(3) :319–347, 2011.

[2] Pierre Buyssens, Maxime Daisy, David Tschumperlé, and Olivier Lézoray. Exemplar-based inpainting : Technical review and new heuristics for better geometric reconstructions. *Image Processing, IEEE Transactions on*, 24(6) :1809–1824, 2015.

[3] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on*, 13(9) :1200–1212, 2004.

[4] Ismaël Daribo and Béatrice Pesquet-Popescu. Depth-aided image inpainting for novel view synthesis. In *IEEE International Workshop on Multimedia Signal Processing*, pages 167–170, 2010.

[5] Silvano Di Zenzo. A note on the gradient of a multi-image. *Computer vision, graphics, and image processing*, 33(1) :116–125, 1986.

[6] Alexei Efros, Thomas K Leung, et al. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038, 1999.

[7] Christoph Fehn. Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv. In *Electronic Imaging 2004*, pages 93–104. International Society for Optics and Photonics, 2004.

[8] Josselin Gautier, Olivier Le Meur, and Christine Guillemot. Depth-based image completion for view synthesis. In *3DTV Conference : The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pages 1–4, 2011.

[9] Christine Guillemot and Olivier Le Meur. Image inpainting : Overview and recent advances. *Signal Processing Magazine, IEEE*, 31(1) :127–144, 2014.

[10] Lingni Ma, Luat Do, and PHN de With. Depth-guided inpainting algorithm for free-viewpoint video. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1721–1724, 2012.

[11] Yu Mao, Gene Cheung, and Yusheng Ji. Image interpolation for dibr view synthesis using graph fourier transform. In *3DTV-Conference : The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*, pages 1–4. IEEE, 2014.

[12] Yu Mao, Gene Cheung, Antonio Ortega, and Yusheng Ji. Expansion hole filling in depth-image-based rendering using graph-based interpolation. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 1859–1863. IEEE, 2013.

[13] Patrick Ndjiki-Nya, Martin Koppel, Dimitar Doshkov, Haricharan Lakshman, Philipp Merkle, K Muller, and Thomas Wiegand. Depth image-based rendering with advanced texture synthesis for 3-d video. *Multimedia, IEEE Transactions on*, 13(3) :453–465, 2011.

[14] Smarti Reel, Gene Cheung, Patrick Wong, and Laurence S Dooley. Joint texture-depth pixel inpainting of disocclusion holes in virtual view synthesis. In *Signal and Information Processing Association Annual Summit and Conference*, pages 1–7, 2013.

[15] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition*, pages 31–42. Springer, 2014.

[16] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–195, 2003.

[17] Dong Tian, Po-Lin Lai, Patrick Lopez, and Cristina Gomila. View synthesis techniques for 3d video. In *SPIE Optical Engineering+ Applications*, pages 74430T–74430T. International Society for Optics and Photonics, 2009.

[18] Xuyuan Xu, Lai-Man Po, Chun-Ho Cheung, Litong Feng, Ka-Ho Ng, and Kwok-Wai Cheung. Depth-aided exemplar-based hole filling for dibr view synthesis. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 2840–2843, 2013.

[19] Soo Sung Yoon, Hosik Sohn, and Yong Man Ro. Inter–view consistent hole filling in view extrapolation for multi–view image generation. In *Image Processing (ICIP), 2014 21th IEEE International Conference on*, pages 2883–2887. 2014.

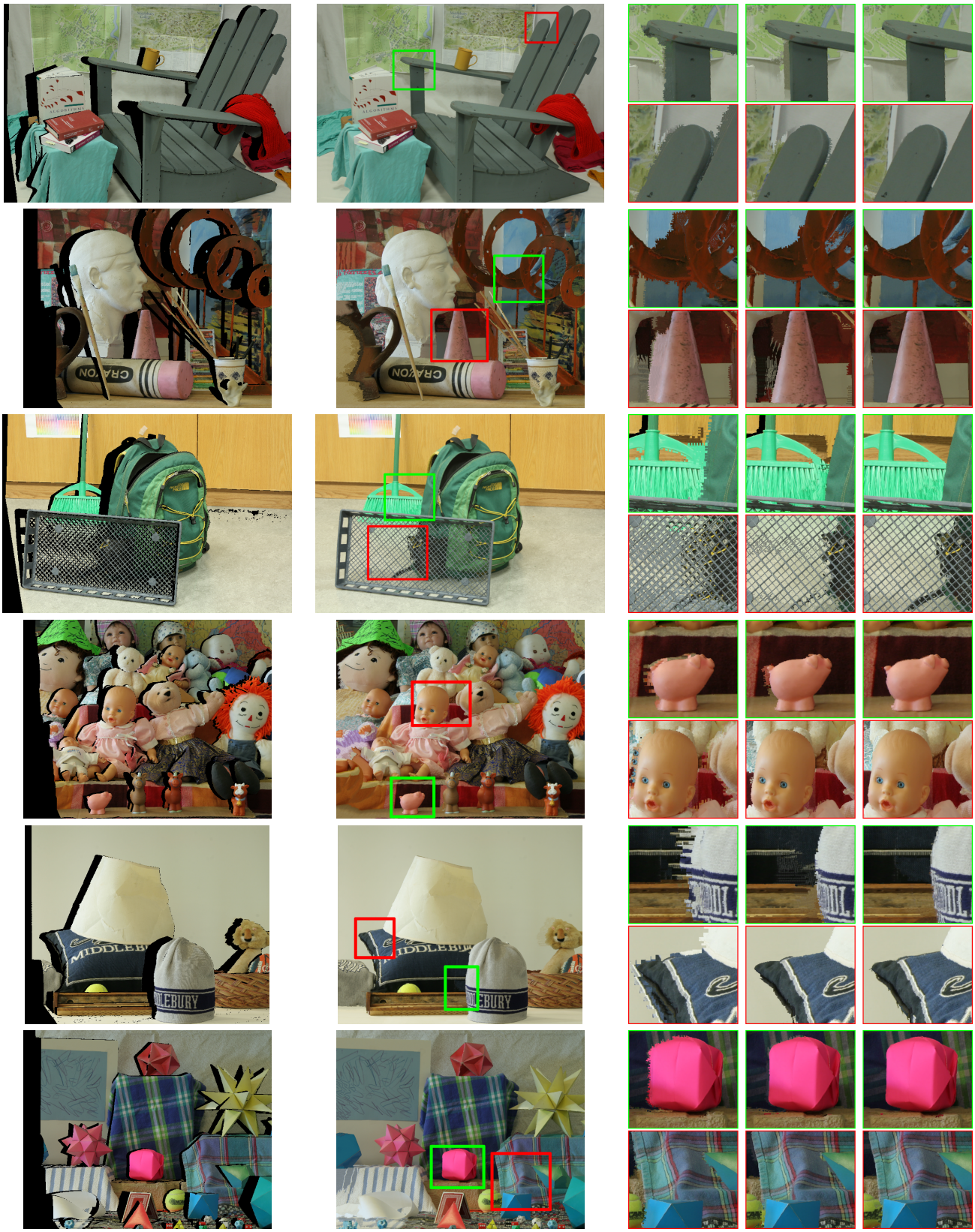

FIGURE 5 – Comparaisons des reconstructions. De gauche à droite : image masquée, notre résultat, et extraits de reconstruction avec les méthodes [8], [19], et la nôtre. Les tailles des masques sont respectivement (de haut en bas) de 401K pixels pour l'image *Art*, 837K pixels pour l'image *Backpack*, 263K pixels pour l'image *Dolls*, 157K pixels pour l'image *Midd2*, et 253K pixels pour l'image *Moebius*.