ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THÈSE PRÉSENTÉE À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DU
DOCTORAT EN GÉNIE
Ph. D.

PAR
Jocelyn BENOIT

APPROCHES DE REMPLISSAGE AUTOMATIQUE DE TROUS À L'INTÉRIEUR D'IMAGES ET DE SÉQUENCES VIDÉO

MONTRÉAL, LE 20 AVRIL 2017

PRÉSENTATION DU JURY

CETTE THÈSE A ÉTÉ ÉVALUÉE

PAR UN JURY COMPOSÉ DE :

M. Jacques A. De Guise, président du jury
Département de la production automatisée à l'École de technologie supérieure

M. Michael John McGuffin, membre du jury
Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

M. Tiberiu Popa, examinateur externe
Department of computer science and software engineering à l'Université Concordia

ELLE A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY

LE 2 MARS 2017

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

AVANT-PROPOS

Droits d'auteur

Pour l'écriture de cette thèse, il aurait souvent été souhaitable de présenter des images des autres travaux sur la synthèse de texture ou le remplissage de régions manquantes dans une séquence vidéo. Cependant, les droits d'auteur contraignent la possibilité de reproduire ces images. Tout de même, certaines images de ce document ont pu être tirées d'articles publiés dans des journaux ou des conférences scientifiques. La provenance des images est clairement indiquée dans les figures et la référence au travail est également fournie.

Voici la note de droits d'auteur pour les travaux tirés d'une publication d'ACM :
ACM COPYRIGHT NOTICE. Copyright © 1982-2016 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept, ACM Inc., fax +1 (212) 869-0481, or permissions@acm.org.

Voici la note de droits d'auteur pour les travaux tirés de Graphics Interface :
Copyright © 2016 par l'Association canadienne de l'informatique. Il est permis de citer de courts extraits et de reproduire des données ou tableaux du présent compte rendu, à condition d'en identifier clairement la source.

Finalement, les images tirées de films d'animation et de jeux vidéo proviennent de galerie d'images promotionnelles de basses résolutions visant à promouvoir le produit dans les médias.

REMERCIEMENTS

Dans un premier temps, je souhaite remercier très chaleureusement mon directeur de thèse, M. Eric Paquette, pour l'encadrement que j'aie reçu durant les dernières années. En plus de me transmettre et de m'enseigner ses connaissances en recherche, il m'a offert un encadrement très rigoureux qui m'a assurément permis de hausser la qualité de cette thèse. Je lui suis très reconnaissant pour son temps, ses commentaires, ses critiques constructives et pour les discussions que nous avons eues. Pour tout ceci, et bien plus, je lui suis très reconnaissant.

Les mots me manquent pour remercier, à sa juste valeur, ma conjointe Marie-Joëlle pour son soutien moral et psychologique qui a été un ingrédient clé au succès de ce projet malgré les aléas de la vie qui ne se veulent pas toujours faciles. Merci d'avoir toujours cru en moi. Tu es irremplaçable.

Je tiens aussi à remercier mes parents, Danielle et Jean-Pierre, qui m'ont constamment encouragé et soutenu dans mon parcours académique. Vous m'avez enseigné la vie et le travail; je vous en serai à jamais reconnaissant.

Un énorme merci à tous les autres membres de ma famille et à mes amis pour votre support inconditionnel. Je ne les nomme pas par peur d'en oublier, mais ils sauront se reconnaître.

Je tiens également à remercier le Fonds de recherche du Québec – Nature et technologie (FQRNT), l'École de technologie supérieure (ÉTS), Mokko Studio et la compagnie Genetec pour leur appui financier tout au long de mon doctorat. Sans ces contributions financières, il m'aurait été impossible de réaliser ce projet.

Finalement, j'aimerais remercier tous ceux qui ont participé à ce projet de recherche de près ou de loin. Un merci particulier à Olivier Clément pour sa contribution inestimable.

APPROCHES DE REMPLISSAGE DE RÉGIONS MANQUANTES À L'INTÉRIEUR D'IMAGES ET DE SÉQUENCES VIDÉO

Jocelyn BENOIT

RÉSUMÉ

À notre époque, les images et les séquences vidéo destinées au cinéma ou à la télévision sont fréquemment altérées durant l'étape de postproduction afin d'effectuer le remplissage de régions indésirables. Par exemple, les graffitis à caractères haineux présents dans une image sont supprimés. Pour produire un résultat de qualité, il est important que le remplissage ait une apparence réaliste et qu'il présente des signes d'usure. Les méthodes actuelles traitant de ce problème ne sont pas adaptées puisqu'elles utilisent des paramètres peu intuitifs et qu'elles traitent généralement d'un seul effet de détérioration. Le remplissage peut aussi se faire sur une séquence vidéo dans laquelle la perche de son a malencontreusement été filmée. Le remplissage de régions manquantes dans une séquence vidéo pose des défis additionnels, comme la cohérence spatio-temporelle et la grande quantité d'information à traiter, et les approches actuelles sont inadaptées. En effet, la plupart des méthodes, dont celles basées sur les champs aléatoires de Markov, ne peuvent traiter directement la haute résolution dans un délai raisonnable. De plus, les méthodes actuelles sont limitées par le type de mouvement de caméra, la taille des régions indésirables et la variation de l'intensité lumineuse.

Un objectif de cette thèse est de développer un système de remplissage qui permet la génération d'effets de détérioration basé sur une image échantillon contenant un exemple de l'effet voulu. Pour y arriver, une approche de synthèse de textures par remplissage de trous qui ne comporte aucun paramètre complexe à manipuler par l'artiste et qui permet de reproduire de nouveaux effets similaires est introduite. Un deuxième objectif est l'élaboration d'un système de remplissage de régions manquantes de séquences vidéo de haute définition. Un algorithme de synthèse de textures par remplissage de trous est adapté en tirant profit du principe de la cohérence et d'une recherche locale. De plus, le dernier volet de la thèse présente une approche de remplissage basée sur le suivi de caractéristiques invariantes permettant de compléter de très grandes régions manquantes provenant de séquences vidéo filmées avec des mouvements de caméra non-triviaux.

Les résultats obtenus à partir des différentes contributions du projet de recherche montrent un réalisme accru lors du remplissage de régions manquantes d'images et de séquences vidéo. Les différentes méthodes sont faciles d'utilisation et intuitives puisqu'elles ne possèdent aucun paramètre complexe à spécifier par l'artiste. De plus, elles s'intègrent bien dans le processus itératif de création de ce dernier. Finalement, les petits temps de calculs rendent faciles leur intégration dans le pipeline de production des studios.

Mots-clés :
Remplissage, synthèse de texture, champ aléatoire de Markov, haute définition, détérioration, usure, région manquante

HOLE FILLING APPROACHES FOR IMAGE AND VIDEO COMPLETION

Jocelyn BENOIT

ABSTRACT

In our time, images and movies for cinema and television are frequently altered during the post-production stage to replace and complete unwanted areas. For example, a hateful graffiti present in an image is deleted. To produce quality results, it is important that the completed region has a realistic appearance and shows signs of aging. State-of-the-art methods addressing this problem are not suitable since they require the artist to use complex parameters and usually handle only one aging effect. The completion can also be done on a video in which a sound boom was inadvertently filmed. The filling of missing region in a video poses additional challenges, enforcing spatio-temporal consistency and the sheer amount of information to name a few, and current approaches are inadequate. Indeed, most methods, including those based on random Markov fields, cannot directly handle high definition video within a reasonable time. Furthermore, currents methods are limited by the type of camera movement, the missing region size, and the variation of light intensity.

An objective of this thesis is to develop a framework that allows artists to generate new aging effects based on a sample image containing an example of the desired effect. To achieve this, a novel texture synthesis approach adapted to the aging context is introduced. This approach does not need the artist to specify any complex parameter. A second objective is the development of a system that completes the missing regions of high definition video. A texture synthesis algorithm using a hole-filling approach is adapted by taking advantage of a coherence principle and a local search. Furthermore, the last part of this thesis presents a video completion method based on scale-invariant features tracking. This method is able to complete very large missing regions from video footage with non-trivial camera movements.

The results obtained from the different contributions of this research project show increased realism during the completion of missing areas of images and video sequences. The different methods are easy to use and intuitive because no complex parameter needs to be specified by the artist. Moreover, they are suitable for the iterative process of the artist. Finally, these methods fit well in the studio production pipeline since they require small computation times.

Keywords :
Completion, texture synthesis, random Markov field, high definition, aging, missing region

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

3D	Trois dimensions
ANN	*Approximate nearest neighbor*
BP	*Belief propagation*
HD	Haute définition
HSV	Teinte saturation valeur (*hue saturation value*)
kd-tree	*k-dimensional tree*
LI	Liste d'index
LIF	List d'index au niveau de resolution le plus fin
MRF	*Markov random field*
MSS	*Multiscale structural similarity*
PCA	Analyse en composantes principales
PSNR	*Peak signal-to-noise ratio*
RANSAC	*Random sample consensus*
RVB	Rouge vert bleu
SSD	Somme des distances carrées (*sum of the squared distances*)
SSIM	*Structural similarity*
SURF	*Speeded up robust features*

INTRODUCTION

Au cours des dernières années, l'évolution des différentes techniques dans le domaine de l'infographie a considérablement changé la façon de produire une séquence vidéo destinée au cinéma ou à la télévision. À l'origine, il y avait peu de méthodes et d'outils qui permettaient d'éditer ou d'altérer les séquences recueillies lors des séances de tournage; ce qui était filmé se retrouvait bien souvent intégralement dans le produit final. Il était donc important de s'assurer que tous les éléments désirés dans la séquence finale étaient présents lors du tournage et qu'inversement, aucun objet non désiré n'était filmé par mégarde. Il s'agissait d'un travail de moine qui laissait peu de place à l'erreur.

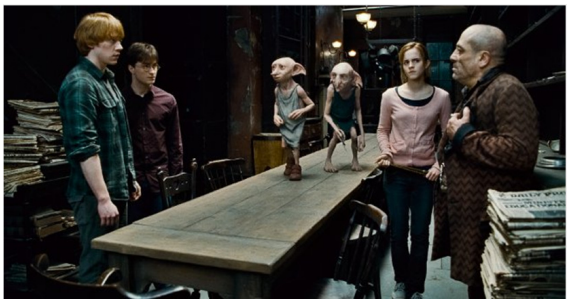
De nos jours, l'évolution des différentes méthodes en infographie nous permet de corriger ces erreurs en faisant la retouche et l'édition des séquences vidéo recueillies lors des séances de tournage et en altérant le contenu de celles-ci d'une façon suffisamment réaliste qu'il est bien souvent impossible pour l'audience de percevoir ce qui a été modifié. La figure 1 montre des exemples de films retouchés par ordinateur. Les avancées en infographie sont telles qu'il y a maintenant des productions cinématographiques qui sont créées entièrement à partir d'images de synthèse. La figure 2 illustre quelques exemples de films complètement créées par ordinateur.

Malgré tous les avancements réalisés jusqu'à présent, il y a tout de même une augmentation des besoins d'avoir des méthodes et des outils permettant la retouche de séquences vidéo qui sont encore plus efficaces. Plusieurs raisons peuvent être énumérées pour expliquer ce phénomène. Premièrement, le coût et la disponibilité des acteurs, des lieux de tournage, du matériel et de l'équipe technique font en sorte que le temps alloué pour les tournages est de plus en plus limité. Cette contrainte augmente considérablement les chances qu'un oubli ou qu'une erreur soit présent dans une séquence vidéo tournée nécessitant donc d'être retouchée par la suite. Elle a également comme impact de rendre l'équipe de tournage plus dépendante de facteurs qui sont hors de son contrôle.

Rogue One (2016)
© Lucasfilm

The Arrival (2016)
© 21 Laps Entertainment

Iron Man 2 (2010)
© Paramount Pictures



Harry Potter and the Deathly Hallows : Part 1 (2010)
© Warner Bros. Entertainment Inc.

Figure 1 Exemples de films avec des retouches réalistes.
Adaptée de *Rogue One* (©Lucasfilm), de *The Arrival* (©21 Laps Entertainment), de *Iron Man 2* (©Paramount Pictures) et de *Harry Potter and the Deatlhy Hallows : Part 1* (©Warner Bros. Entertainment Inc.)

Prenons l'exemple d'une équipe de tournage qui a besoin de monopoliser une autoroute pour filmer une scène de poursuite automobile qui se déroule sous la pluie. Si, lors de la seule journée disponible pour tourner cette scène, le ciel est sans aucun nuage et qu'il y a un soleil de plomb, l'équipe de tournage est contrainte de filmer la scène malgré tout. Par conséquent, cette séquence vidéo devra nécessairement être retouchée à posteriori afin d'ajouter la pluie et ainsi respecter le scénario original. Bref, le temps limité alloué au tournage fait en sorte qu'un plus grand nombre de séquences vidéo doivent être retouchées.

Deuxièmement, il arrive fréquemment que certaines scènes requièrent des éléments qui sont simplement impossibles à filmer. Prenons l'exemple d'une scène qui se déroule dans une navette spatiale. Pour que le résultat soit réaliste, il est primordial que les acteurs se déplacent

selon les règles de l'apesanteur. Or, il est physiquement impossible de demander aux acteurs de simplement *léviter*; il faut donc avoir recours à des câbles et à des harnais pour simuler le déplacement des acteurs dans la navette (voir figure 3).

Trolls (2016)
© DreamWorks Animation

Zootopia (2016)
© Walt Disney Pictures

The Secret Life of Pets (2016)
© Universal Pictures

Toy Story 3 (2010)
© Pixar Animation Studios

Figure 2 Exemples de films d'animation.
Adaptée de *Trolls* (©Dream Works Animation), de *Zootopia* (©Walt Disney Pictures), de *The Secret Life of Pets* (©Universal Pictures) et de *Toy Story 3* (©Pixar Animation Studios)

Évidemment, la séquence vidéo filmée devra par la suite être retouchée afin d'enlever toutes les traces des câbles et des harnais de façon à obtenir une scène finale réaliste. Le film *Avatar* (James Cameron, 2009), qui contient une grande quantité de bestioles qui n'existent pas, est un autre exemple de scène impossible à tourner. Bref, la nécessité d'avoir recours à des trucages et le désir d'incorporer des éléments irréels ou imaginaires sont d'autres raisons pour lesquelles un plus grand nombre de séquences vidéo doivent être retouchées.

Figure 3 Séquence vidéo avec un objet indésirable.
Adaptée de *Race to Mars* (© Galafilm et Discovery Channel Canada)

Généralement, l'objectif principal de la retouche ou de l'édition de séquences vidéo est qu'elle doit être imperceptible. L'audience ne doit pas être en mesure de repérer les éléments qui ont été retouchés ou altérés. Elle doit croire en l'illusion que la séquence vidéo est réelle et qu'elle a été filmée de cette façon, sans altération. Un des critères important à respecter est donc que la correction apportée doit être **réaliste**; un élément qui n'est pas jugé réel par l'audience est rapidement remarqué. Le réalisme est composé de plusieurs critères. L'objet est-il de la bonne forme? De la bonne couleur? Présente-t-il des signes d'usure et de détérioration comme de la rouille, des égratignures ou des bosses? Est-il suffisamment détaillé? Présente-t-il les signes d'un objet qui a évolué dans cet environnement depuis des mois ou des années? Cette thèse ne traitera pas de tous ces critères, mais elle s'attardera, entre autres, à la présence d'usure et de détérioration.

Dans le monde réel, tous les objets interagissent avec l'environnement qui les entoure. Cette interaction a bien souvent pour effet de modifier l'apparence de l'objet en question. Ce phénomène se définit comme de l'usure ou de la détérioration. Il existe plusieurs types

d'usure; par exemple l'apparition de rouille lorsqu'une surface de fer se corrode en présence d'oxygène et d'eau, la formation de vert-de-gris sur une surface de cuivre ou de bronze, l'ajout d'égratignures lorsque deux surfaces entrent en contact ou l'ajout de traces d'érosion lorsque l'eau ou le vent altère un objet, pour ne nommer que ces types d'usures. La grande majorité des objets du monde réel présente un ou plusieurs effets de détérioration à différents degrés tels qu'illustrés sur la figure 4. Bref, pour modifier une séquence vidéo en y ajoutant un objet de synthèse qui soit le plus réaliste possible, il est nécessaire de pouvoir lui appliquer différents phénomènes d'usure.

Figure 4 Exemples réels d'effets de détérioration.

Plusieurs travaux ont été réalisés jusqu'à présent pour comprendre, analyser et modéliser différents effets de détérioration. Ces travaux ont tous un point en commun : ils sont mal adaptés au processus de création dans un contexte de production 3D. Premièrement, plusieurs approches utilisent des paramètres physiques complexes afin de contrôler le résultat attendu de l'effet de détérioration. Or, en postproduction, ce sont des artistes qui sont appelés à utiliser ces méthodes. Puisqu'ils n'ont généralement pas les connaissances physiques et

mathématiques nécessaires pour comprendre les différents paramètres, les artistes délaissent généralement ces méthodes puisqu'il est difficile et fastidieux d'obtenir le résultat désiré.

Deuxièmement, les méthodes actuelles ne cadrent pas bien avec le processus itératif de création des artistes à l'intérieur des studios de production. Tout d'abord, l'artiste réalise une première ébauche du travail. L'artiste lui-même ou d'autres membres de l'équipe inspectent le résultat et indiquent une liste de modifications et de correctifs à effectuer au travail. L'artiste retourne à sa *planche à dessin* et il applique les améliorations nécessaires. La boucle entre la présentation et l'amélioration du travail se poursuit jusqu'à ce que le résultat soit satisfaisant. La figure 5 présente ce processus itératif de création. Or, les méthodes actuelles ne permettent généralement pas de modifier l'effet de détérioration préalablement obtenu; il est uniquement possible de créer un autre effet totalement nouveau.

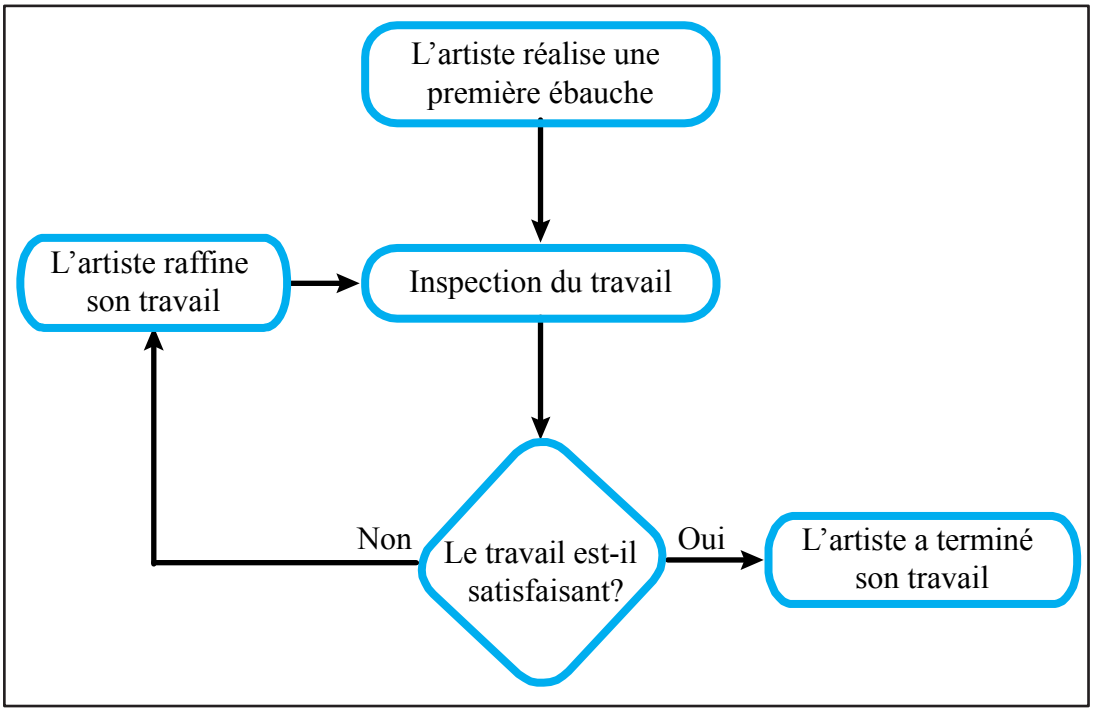


Figure 5 Processus itératif de création.

Troisièmement, la majorité des méthodes actuelles ne permettent de synthétiser un seul effet de détérioration. Or, il est fréquent de retrouver un objet de synthèse avec plusieurs effets de détérioration distincts. Un bon exemple est celui d'une voiture qui peut présenter de la

rouille, des égratignures et des bosses. Actuellement, l'artiste doit utiliser trois méthodes différentes afin d'obtenir le résultat désiré, ce qui demande une charge de travail considérable. Finalement, les méthodes actuelles n'offrent pas un contrôle adéquat sur le résultat. Il est parfois difficile de positionner l'effet de détérioration sur l'objet ou d'indiquer le degré de détérioration. Les artistes choisissent donc de créer *manuellement* les différents effets de détérioration ce qui demande temps et argent.

Un des objectifs de cette thèse est donc la conception et l'élaboration d'une méthode de détérioration d'objets de synthèse adaptée au processus itératif de création dans un contexte de production, qui tient compte des connaissances de l'artiste, qui offre un contrôle adéquat sur les résultats attendus et qui permet de créer plusieurs effets d'usure différents. Cette méthode facilitera donc la retouche et l'édition de séquences vidéo en permettant d'insérer des objets de synthèse plus réalistes.

En plus de l'ajout d'objets de synthèse, un autre aspect important à considérer pour la retouche et l'édition de séquences vidéo est la suppression d'éléments ou d'objets indésirables. Durant un tournage, il arrive fréquemment que des éléments qui ne doivent pas se retrouver dans la séquence vidéo finale soient captés par inadvertance par la caméra (fils, micros, perches, câbles, etc.). Il est donc nécessaire de retoucher la séquence vidéo afin de remplacer ces éléments indésirables par des éléments cohérents avec le reste de la séquence vidéo de façon à ce que l'audience ne soit pas en mesure de percevoir la suppression de l'objet. En pratique, ce type de retouche est principalement réalisé de façon *manuelle* par un artiste. Pour les studios de production, il serait préférable que cette tâche puisse être automatisée puisqu'il s'agit d'un travail jugé non créatif pour l'artiste. Par conséquent, le talent créatif de l'artiste pourrait être mieux exploité si celui-ci était affecté à une autre tâche. Les approches automatiques actuelles ne sont cependant pas adéquates pour les studios de production puisqu'elles sont uniquement capables de traiter des séquences vidéo de petites résolutions. Or, puisque la haute définition est maintenant très courante tant pour le domaine du cinéma que pour celui de la télévision, les studios de production sont contraints de demander à un artiste d'effectuer la retouche *manuellement*. De plus, les approches actuelles

ne sont généralement pas en mesure de compléter des séquences vidéo qui présentent des mouvements de caméra non-triviaux tels que la rotation, la mise à l'échelle et le déplacement sur un plan.

Un deuxième objectif de cette thèse est donc de développer une approche automatique permettant de supprimer un élément indésirable dans une séquence vidéo en adressant particulièrement le problème des séquences vidéo de hautes résolutions. Par conséquent, il sera possible pour les studios de production d'affecter les artistes à des tâches jugées plus créatives. Le troisième objectif de cette thèse est de développer une technique de complétion vidéo qui est en mesure de traiter les séquences vidéo présentant des mouvements de caméras non-triviaux et des changements d'intensité.

En somme, autant la création d'une approche de simulation d'effets de détérioration pour les objets de synthèse que la conception de méthodes pour la suppression d'éléments ou d'objets indésirables dans une séquence vidéo de haute résolution permettront d'améliorer les techniques et les outils pour faire la retouche et l'édition de séquences vidéo.

L'atteinte des objectifs mentionnés précédemment a mené à la publication des trois articles scientifiques (révisés par les pairs) suivant : Clément, Benoit et Paquette (2007), Benoit et Paquette (2015) et Benoit et Paquette (2016).

CHAPITRE 1

REVUE DE LA LITTÉRATURE

De façon à mieux comprendre et cerner la problématique de la retouche et de l'édition de séquences vidéo, il est primordial de débuter par une revue des différents ouvrages antérieurs reliés à ce domaine. Ainsi, le présent chapitre traite de l'état de l'art pour la simulation et la synthèse d'effets de détérioration réalistes, de la synthèse de textures et du remplissage d'une région manquante autant pour une image que pour une séquence vidéo. Une analyse objective des avantages et des limitations des travaux antérieurs permet de définir plus facilement la problématique. Une fois celle-ci cernée, ce chapitre conclu avec la liste des objectifs de cette thèse.

1.1 Simulations d'effets de détérioration

Lorsqu'un objet de synthèse est ajouté à une séquence vidéo, il est primordial que cette insertion soit transparente pour l'audience. Par conséquent, il est fondamental que l'objet soit le plus réaliste possible de façon à leurrer les spectateurs. Tel quel mentionné préalablement, un facteur important de réalisme est la présence d'usure ou d'effets de détérioration. Étant donné l'importance de ce facteur, plusieurs méthodes existent pour simuler et synthétiser différents effets de détérioration. De par leur fonctionnement, ces méthodes peuvent être divisées en trois classes : les simulations basées sur la physique, les simulations basées sur des paramètres empiriques et les simulations basées sur une image échantillon. Ces trois classes sont analysées dans les sections suivantes.

1.1.1 Simulations basées sur la physique

La majorité des effets de détérioration retrouvés sur un objet peut être définie par un ensemble de règles et de lois appartenant au domaine de la physique : les traces laissées par l'écoulement d'un fluide, la rouille, la patine, l'érosion, etc. Il est donc normal qu'un grand

nombre d'approches de détérioration se basent sur l'étude du phénomène physique dans le but de pouvoir le recréer et le reproduire. Une première méthode de cette classe a été développée par Dorsey, Pederson et Hanrahan (1996) pour simuler les changements d'apparence d'une surface résultant de l'écoulement d'un liquide sur celle-ci. Cette technique simule le phénomène à l'aide d'un système de particules dans lequel chaque particule représente une goutte d'eau. Différents paramètres, comme la gravité, la friction, le vent et la porosité, régissent le déplacement des particules sur la surface. La réaction chimique entre l'eau et la surface se contrôle par le taux d'absorption de cette dernière ainsi qu'avec son taux de solubilité et de sédimentation. Cette technique permet de simuler des régions délavées ou tachées par l'écoulement de l'eau. La figure 1.1 présente des résultats obtenus par cette méthode.

Une autre méthode de Dorsey et Perderson (1996) simule quant à elle la création de patine sur des surfaces métalliques comme celles en bronze ou en cuivre. La patine est une mince couche créée au fil du temps par le contact entre une surface métallique et son environnement. Un exemple commun de patine est le vert-de-gris. Cette méthode représente une surface par une superposition de plusieurs couches. Chaque couche se forme au contact de l'atmosphère en capturant différentes particules contenues dans celle-ci. La variation du temps d'exposition et du type de particules permettent de créer plusieurs effets différents de patine. La figure 1.2 montre un résultat obtenu avec cette méthode. Une autre méthode, celle de Chang et Shih (2001), permet également de simuler l'accumulation de patine sur des surfaces métalliques. Cette approche s'attarde plus spécifiquement aux surfaces métalliques enfouies dans le sol. En plus du temps d'exposition, les propriétés géométriques de l'objet et les caractéristiques du sol sont aussi prises en considération lors de la simulation. La même année, Mérillou *et al.* (2001a) présente une approche afin de générer de la rouille.

Figure 1.1 Résultats obtenus par la simulation d'écoulement d'eau.
Tirée de Dorsey, Pederson et Hanrahan (1996, p. 420, p. 420)

Ce ne sont pas seulement les phénomènes sur les surfaces métalliques qui ont été étudiés; plusieurs travaux portent aussi sur l'usure et la détérioration des objets en roche ou en pierre. Par exemple, la méthode de Dorsey *et al.* (1999) introduit le concept d'une structure de données en *dalles* qui permet de suivre l'évolution dans le temps du taux d'humidité. Lorsque le taux d'humidité d'une surface de pierre est suffisamment élevé, les minéraux en surface se dissolvent permettant leurs déplacements. La structure en *dalles* permet de suivre la dissolution, le transport et la recristallisation des minéraux en fonction du taux d'humidité pour ainsi simuler le changement de géométrie et d'apparence d'objets en pierre. La figure 1.2 montre des résultats obtenus avec la méthode de Dorsey *et al.* (1999).

Figure 1.2 Résultat obtenu avec la simulation de platine.
Adaptée de Dorsey et Pederson (1996, p. 395)

Il existe un très grand éventail de méthodes d'effets de détérioration basées sur la simulation physique. Par exemple, le travail de Iben et O'Brien (2009) simule la formation de fissures sur des surfaces en définissant un champ de tenseur de stress et en faisant évoluer ce dernier dans le temps en fonction de différents paramètres précisés par l'utilisateur. Glondu, Marchal et Dumont (2013) s'attaquent au même problème en proposant une méthode d'initialisation des fractures fondée sur une analyse modale et un algorithme de propagation basé sur une fonction d'énergie globale. L'approche de Mérillou *et al.* (2010) simule quant à elle le changement d'apparence des édifices urbains en fonction de différentes zones de pollution définies selon une vraie classification des types de pollution retrouvés dans l'air. De son côté, le travail de Bézin *et al.* (2014) simule des événements géomorphologiques tels que l'érosion, le transport de sédiments et leur dépôt en introduisant le concept de *grille généralisée* qui modélise différentes couches géologiques. Le tableau 1.1 énumère plusieurs méthodes et les regroupe selon la classification présentée dans le travail de Mérillou et Ghazanfarpour (2008). Il est d'ailleurs conseillé de se référer à cet article pour avoir plus de détails sur celles-ci.

Figure 1.3 Résultats obtenus avec la simulation du vieillissement de la pierre.
Adaptée de Dorsey *et al.* (1999, p. 233)

Bien que chacune des méthodes simule un effet de détérioration différent et que leurs fonctionnements soient dans bien des cas très différents, il existe tout de même de grandes similitudes, particulièrement au sujet de leurs limitations.

Premièrement, chaque technique permet de reproduire un seul effet de détérioration. Or, la majorité des objets du monde réel est affectée par plusieurs effets de détérioration en même temps. Par exemple, une voiture peut présenter de la rouille, des égratignures et des bosses. Les artistes sont donc contraints d'utiliser et de maîtriser plusieurs techniques différentes pour obtenir le résultat désiré.

Deuxièmement, puisque ces techniques se basent sur des modèles physiques complexes, elles nécessitent généralement des temps de calculs très longs (voir tableau 1.1) qui rendent difficiles leur intégration dans le processus itératif de création. En effet, l'artiste doit généralement modifier à plusieurs reprises son travail durant son processus créatif.

Tableau 1.1 Recensement d'approches basées sur la simulation physique
Données partiellement tirées de Lu *et al.(2007, p.3)*

	Effet	**Paramètres**	**Performance**
Dorsey et Hanraham (1996)	Patine	Accessibilité, inclinaison et orientation de la surface	20 minutes à 3 heures
Dorsey *et al. (1996)*	Écoulement	Rugosité et absorptivité des matériaux; adhésion et solubilité des dépôts; masse, position et vitesse des particules d'eau	3 heures
Chang et Shih (2003)	Rouille	Propriétés géométriques de l'objet, courants marins, teneur en sel dans	Données non disponibles
Chang et Shih (2001)	Patine	Propriétés géométriques de l'objet, gravité, humidité du sol	Données non disponibles
Dorsey *et al.* (1999)	Érosion, décoloration	Teneur en sel, saturation maximale, taux de perméabilité, pression de densité de l'eau, etc.	3 à 24 heures
Mérillou *et al.* (2001a)	Rouille, patine	Facteur d'imperfection, rugosité, porosité et épaisseur du matériau	Données non disponibles
Mérillou *et al. (2001b)*	Égratignures	Caractéristiques des surfaces, caractéristiques des égratignures	Données non disponibles

Conséquemment, le temps qui lui est nécessaire pour compléter son travail est drastiquement augmenté s'il doit attendre plusieurs heures afin de créer les différents effets de détérioration à chacun des cycles d'amélioration.

Troisièmement, la majorité de ces méthodes est contrôlée par un ensemble de paramètres physiques complexes (voir tableau 1.1) limitant ainsi leur utilisation dans un contexte réel de production. En effet, les artistes, c'est-à-dire les personnes les plus susceptibles d'utiliser ces méthodes, n'ont généralement pas un bagage suffisant de connaissances sur la physique. Par conséquent, il leur est difficile d'obtenir l'usure désirée puisqu'ils n'arrivent pas à déterminer la valeur des différents paramètres.

Finalement, comme leurs noms l'indiquent, les méthodes de création d'effets de détérioration basées sur la simulation physique requièrent d'avoir une connaissance très pointue du

phénomène physique en question. Or, lorsqu'il y a incompréhension du phénomène physique en soit, il est impossible de créer un modèle de simulation fidèle à la réalité. Pour régler ce problème, certains auteurs se sont tournés vers des modèles de simulation basés sur des paramètres empiriques. Autrement dit, ces modèles représentent les effets de détérioration en se basant sur l'expérience et l'observation, sans nécessairement suivre les principes physiques. Cette catégorie de méthodes est étudiée à la section suivante.

1.1.2 Simulations basées sur des paramètres empiriques

Les méthodes de création d'effets de détérioration qui utilisent des simulations basées sur la physique ont deux limitations majeures : elles nécessitent d'avoir une compréhension très pointue du phénomène physique et elles sont contrôlées par des paramètres physiques complexes. Afin de pallier ces problèmes, certains auteurs se sont tournés vers des simulations basées sur des paramètres empiriques. Puisque celles-ci se basent principalement sur l'expérience et l'observation du phénomène, elles ne requièrent pas une compréhension scientifique des effets de détérioration à reproduire. De plus, ces méthodes sont contrôlées par des paramètres plus intuitifs pour les artistes.

Le travail de Hsu et Wong (1995) est un premier exemple de méthode de cette classe. Celui-ci présente une approche qui prédit la quantité de poussière s'accumulant sur les surfaces d'un objet en fonction des propriétés des surfaces ainsi que sur la géométrie globale de l'objet. Dans un premier temps, la quantité *normale* d'accumulation de poussière est déterminée en tenant compte de l'inclinaison et du type de chaque surface. Par la suite, la quantité *normale* de poussière est modifiée en fonction de plusieurs facteurs externes comme l'accessibilité de la surface, le vent et le contact de la surface avec un autre objet. L'épaisseur de poussière finale est ensuite emmagasinée dans une texture qui peut être appliquée à l'objet. L'utilisation de cette approche est relativement simple puisque Hsu et Wong définissent des *sources de poussière*; une analogie aux sources de lumière utilisées couramment par les artistes.

En utilisant une approche par *sources* similaire à celle de Hsu et Wong (1995), le travail de Wong, Ng et Heng (1997) permet de créer des différentes effets. Cette méthode permet de placer des *sources de tendance* qui influencent la probabilité qu'une surface développe de la patine ou tout autre effet. Par conséquent, cette méthode offre un meilleur contrôle sur le positionnement de l'effet que celui retrouvé dans les travaux de Dorsey, Pederson et Hanrahan (1996) et de Dorsey et Perderson (1996). Cependant, les résultats présentés dans l'article ne sont pas assez réalistes pour être utilisés dans un contexte réel de production.

D'autres auteurs, Paquette, Poulin et Drettakis (2001), ont proposé une méthode également basée sur la simulation empirique pour reproduire les effets de détérioration de surface causés par des impacts. La méthode propose à l'utilisateur de sélectionner plusieurs objets dynamiques ou outils et de spécifier certaines de leurs propriétés, comme la trajectoire. Ces outils seront utilisés pour créer une série d'impacts. La méthode débute ensuite un processus itératif durant lequel chaque impact *compacte* la géométrie de l'objet, représentée par un maillage statique, en déplaçant ses différents sommets. Cette géométrie modifiée est par la suite combinée avec de nouvelles valeurs de normales afin d'effectuer le rendu de l'objet détérioré. La figure 1.4 montre des résultats obtenus avec cette méthode.

Paquette, Poulin et Drettakis (2002) se sont également intéressés à la simulation de craques et d'écaillures que l'on retrouvent sur certaines surfaces peinturées. Leur méthode détermine le positionnement initial et la propagation des craques. Pour y arriver, la méthode utilise une grille à deux dimensions contenant différentes propriétés de la surface comme le stress de tension, le stress d'adhésion et la résistance. Lorsque le stress de tension est plus élevé que la résistance de la peinture, une craque est créée. La méthode tient aussi compte de la perte d'adhésion de la couche de peinture située à proximité des craques pour simuler l'écaillement de la peinture. Une fois la simulation terminée, l'information sur les craques, comme leurs positionnements et leur tailles, est utilisée pour créer une nouvelle surface qui est ajoutée sur la surface initiale afin de pouvoir faire le rendu de l'écaillement et du retroussement de la peinture. Puisqu'il s'agit d'une simulation empirique, les paramètres utilisés sont assez

intuitifs pour l'utilisateur, mais leurs impacts sur le résultat final ne semblent pas être facilement prévisibles.

Figure 1.4 Résultats obtenus avec la simulation d'impact.
Adaptée de Paquette, Poulin et Drettakis (2001, p.180)

Quelques années plus tard, Chen *et al.* (2005) ont présenté une méthode de simulation qui permet de traiter une grande variété d'effets de détérioration qu'ils ont nommée *le tracé de particules γ-ton*. Cette méthode émet une grande quantité de particules *γ-ton* à l'intérieur de la scène, d'une façon similaire à ce qui est utilisé dans le tracé de photons (Jensen, 1996), et modélise le phénomène d'usure en se basant sur l'information de transport des différentes particules *γ-ton.* L'utilisateur peut choisir différents types de particules ce qui lui permet de simuler plusieurs effets de détérioration différents et de les combiner ensemble. Ces effets ne sont cependant pas automatiquement créés puisqu'ils doivent être représentés par des textures fournies par l'utilisateur. La figure 1.5 montre des résultats obtenus avec cette

méthode. Kider, Raja et Badler (2011) présentent quant à eux une approche pour simuler le vieillissement biologique et la pourriture des fruits basée sur l'interaction de plusieurs processus biologiques comme la création de fongus ou la perte de nutriments.

Il existe un très grand nombre de méthodes d'effets de détérioration basées sur la simulation empirique. Le tableau 1.2 présente plusieurs de ces méthodes et les regroupe selon la classification présentée dans le travail de Mérillou et Ghazanfarpour (2008). Tel que mentionné précédemment, il est conseillé de se référer à cet article pour avoir plus de détails sur celles-ci.

Figure 1.5 Résultats obtenus avec l'approche de particules *γ-ton.*
Adaptée de Chen *et al. (2005, p. 1133)*

Même si le fonctionnement et les objectifs des différentes méthodes basées sur la simulation empirique sont très différents, plusieurs similitudes peuvent être observées, particulièrement au sujet de leurs limitations. Premièrement, bien que les paramètres qu'elles utilisent soient plus intuitifs pour un artiste que ceux des méthodes basées sur une simulation physique, il

n'en demeure pas moins que ces paramètres ne sont pas adaptés au processus de création en trois dimensions (3D) (voir tableau 1.2). En effet, de manière générale, ces méthodes ne donnent pas un contrôle assez précis sur l'apparence finale du résultat. Par exemple, il est parfois impossible pour un artiste de positionner l'effet d'usure à un endroit précis sur le modèle puisque la technique détermine automatiquement cette position. Or, il est primordial pour un artiste d'avoir une assez grande flexibilité sur l'apparence finale de façon à satisfaire les exigences de sa création artistique. Deuxièmement, la majorité des approches empiriques sont limitées à la simulation d'un seul effet de détérioration. Les artistes doivent donc apprendre à utiliser et à maîtriser plusieurs méthodes différentes de façon à pouvoir combiner plusieurs phénomènes d'usure différents. Cet apprentissage demande beaucoup de temps et, plus souvent qu'autrement, les artistes vont laisser de côté ces approches automatiques pour plutôt créer les différents effets manuellement.

Tableau 1.2 Recensement d'approches basées sur la simulation empirique
Données partiellement tirées de Lu *et al.(2007, p. 3)*

	Effet	**Paramètres**	**Performance**
Hsu et Wong (1995)	Poussière	Inclinaison et viscosité des surfaces, accessibilité, vent; collision, source de poussière	Données non disponibles
Wong *et al.* (1997)	Général	Source de tendance, exposition des surfaces, accessibilité	Données non disponibles
Paquette *et al.* (2001)	Impact	Type d'outils, localisation des impacts	2 heures
Paquette *et al.* (2002)	Écaill.	Stress de tension, stress d'adhésion, résistance	6 à 78 minutes
Chen *et al.* (2005)	Général	Propriétés des particules *γ-ton* , propriété des surfaces	30 à 93 minutes

1.1.3 Synthèse à partir d'une image de référence

Les méthodes de simulation d'effets de détérioration basées sur la physique ou sur des paramètres empiriques sont limitées puisqu'elles permettent de représenter qu'un seul phénomène d'usure à la fois. En effet, pour ce type de méthode, il est important de bien comprendre le phénomène, soit par l'observation ou par l'étude des règles physiques, afin de pouvoir le représenter et le modéliser. Par conséquent, l'artiste est contraint d'utiliser plusieurs méthodes avec des fonctionnements et des paramètres différents pour créer un objet de synthèse avec des traces d'usure diverses. Or, cette façon de faire ne tient pas compte de la démarche créative de l'artiste. En effet, ce dernier se base généralement sur une image de référence pour créer et modéliser un objet de synthèse. L'artiste va donc chercher à reproduire le plus fidèlement possible ce qu'il voit dans l'image de référence dans le but d'avoir un résultat réaliste. Cependant, qu'est-ce que l'artiste peut faire lorsqu'aucune technique de simulation disponible lui permet de recréer un effet de détérioration présent dans l'image? Qu'est-ce que l'artiste peut faire lorsque les techniques disponibles ne lui permettent pas de recréer l'effet d'usure tel que vu dans l'image de référence? L'artiste n'a d'autre choix que de créer manuellement l'effet en question. C'est dans le but de tenir compte davantage de la démarche créative de l'artiste qu'une troisième classe de méthodes de synthèse d'effet de détérioration a été créée : la synthèse à partir d'une image de référence. Comme le nom l'indique, cette classe d'approches se base sur une image de référence contenant des exemples d'effets d'usure pour en synthétiser de nouveaux.

Gu *et al.* (2006) ont présenté une méthode appartenant à cette classe permettant de faire l'acquisition des images de référence, d'analyser l'évolution du phénomène de détérioration en fonction du temps et de synthétiser de nouveaux effets. La première étape consiste à placer un objet qui présente un phénomène d'usure au centre d'un mécanisme sophistiqué de capture d'images, tel qu'illustré sur la figure 1.6.

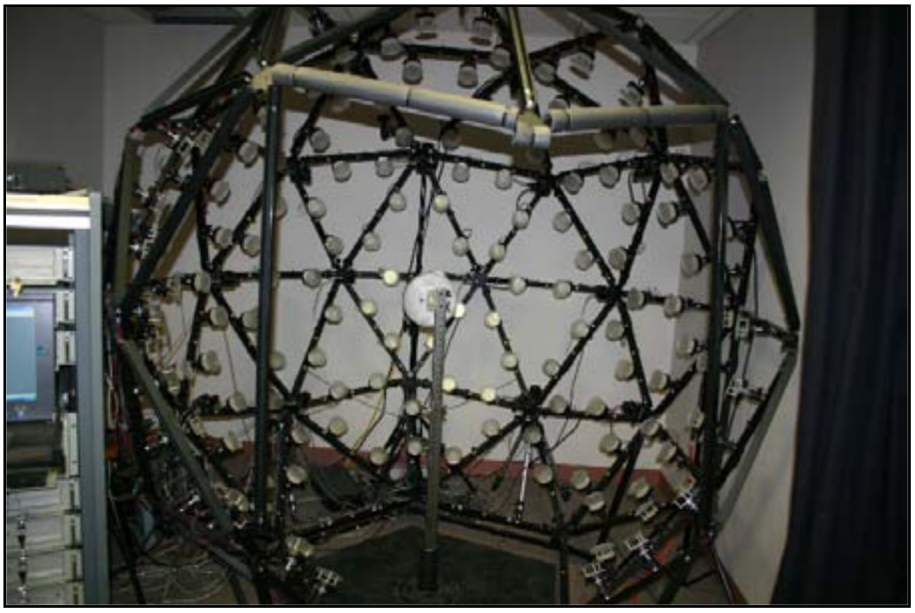


Figure 1.6 Mécanisme sophistiqué de capture d'images.
Tirée de Gu *et al.* (2006, p. 764)

Une série d'images prises à intervalle de temps régulier permet de représenter l'évolution de l'effet de détérioration sur un objet en fonction du temps. Par la suite, la méthode utilise cette banque d'images pour construire une fonction spatio-temporelle qui représente le changement d'apparence d'un objet. En se basant sur cette fonction, cette méthode est capable de générer de nouveaux effets d'usure à l'aide d'une approche non paramétrique de synthèse de texture. La synthèse de texture sera plus détaillée à la section 1.2. Cette approche permet donc de simuler plusieurs effets de détériorations, contrairement aux approches de simulation basées sur la physique ou sur les paramètres empiriques. Cependant, les effets de détérioration ne peuvent pas tous être reproduits. En effet, il est nécessaire que le phénomène évolue relativement rapidement dans le temps afin de pouvoir prendre une série d'images de référence à l'aide du mécanisme illustré à la figure 1.6. Il est donc impossible de synthétiser des phénomènes qui se déroulent sur plusieurs années comme l'érosion de la pierre. De plus, si un artiste désire créer un effet d'usure qui n'est pas présent dans la banque de données, il doit nécessairement avoir accès à un mécanisme de capture d'images qui est relativement complexe et dispendieux. Les résultats obtenus avec cette approche sont cependant très réalistes comme le montre la figure 1.7.

Figure 1.7 Résultats obtenus selon le degré de détérioration.
Adaptée de Gu *et al.* (2006, p. 771)

Wang *et al.* (2006) se sont également intéressés à l'évolution d'effets de détérioration en fonction du temps. Contrairement au travail de Gu *et al.* (2006) qui capture les différents degrés de détérioration à l'aide de plusieurs images, Wang *et al.* (2006) identifient les différents degrés de détérioration à l'aide d'une seule image. Leur raisonnement se base sur l'hypothèse qu'il est possible de trouver sur un seul objet plusieurs degrés du même phénomène de détérioration. Par exemple, une voiture présentant des traces de rouilles peut montrer des régions très, peu et non rouillée, et ce, de façon simultanée. Cette méthode identifie les régions d'une image qui possèdent des traces d'usure et leur attribue un degré de détérioration. Lors de la synthèse de l'effet d'usure, la technique se basera sur l'une ou l'autre des régions identifiées selon le degré de détérioration désiré par l'artiste. La figure 1.8 montre des résultats obtenus à l'aide de cette technique. Cette approche possède cependant une limitation majeure : l'image de référence doit absolument montrer plusieurs degrés de

détérioration du même phénomène. Or, tous les phénomènes ne peuvent pas remplir ce critère.

Figure 1.8 Résultats de la fonction de changement d'apparence.
Adaptée de Wang *et al.* (2006, p754)

De leur côté, Bosch *et al.* (2011) ont en partie repris le pipeline présenté dans cette thèse de façon à simuler le changement d'apparence d'édifices en fonction de différentes conditions météorologiques. En effet, l'approche se base sur des images échantillons de façon à extraire les données relatives au changement d'apparences (lavement et tache) suite à l'écoulement de la pluie sur les édifices. Ces données sont par la suite utilisées pour guider une simulation d'écoulement sur des scènes synthétiques. Bien que la méthode présente des résultats de bonne qualité, elle est malheureusement limitée à un seul effet de détérioration. Récemment, Bosch *et al.* (2011) ont également amélioré le pipeline présenté dans cette thèse en automatisant la génération de plusieurs masques cibles contenant différents degrés de détérioration.

L'approche de synthèse d'effets de détérioration présentée dans cette thèse appartient également à cette classe puisqu'elle se base sur une image de référence. En effet, l'utilisation d'une image de référence permet d'offrir à l'artiste un processus qui est beaucoup plus intuitif et adapté à ses connaissances. De plus, le fait de se baser sur une image de référence permet à l'artiste d'avoir déjà une idée approximative du résultat final sans avoir à effectuer des simulations. Cependant, contrairement aux méthodes présentées, l'approche proposée ne nécessite pas de mécanisme complexe de captation puisqu'une simple image de référence est suffisante. De plus, l'approche proposée permet également de représenter des phénomènes qui n'évoluent pas au fil du temps comme tes taches ou de la peinture. Maintenant, afin de comprendre comment l'effet est recréé, il est nécessaire d'expliquer le fonctionnement des algorithmes de synthèse de textures. La section 1.2 couvrira cette notion.

1.2 Synthèse de textures

Tel qu'indiqué à la section 1.1, plusieurs méthodes de synthèse d'effets de détérioration se basent sur la synthèse de texture pour générer de nouveaux phénomènes d'usure. De plus, la synthèse de texture est le fondement de certains algorithmes de complétion d'images et de complétion de séquences vidéo. Il est donc important de comprendre son fonctionnement avant d'approfondir ces sujets. Cette section débute par une brève mise en contexte de l'utilisation de la synthèse de texture et se poursuit avec une revue des principaux travaux reliés à ce domaine.

Un des facteurs influençant le réalisme des images générées par ordinateur est le niveau de détails des différentes surfaces d'un modèle 3D. Une façon d'augmenter ce niveau de détails est de subdiviser les surfaces en plusieurs polygones. Cependant, il est souvent impossible de subdiviser suffisamment la surface lorsque le niveau de détails désiré est très fin. Dans ce cas, il est possible d'appliquer une image synthétique ou réelle sur une surface, c'est-à-dire faire appliquer une texture (Heckbert, 1986). L'image utilisée est alors appelée texture et elle permet de modifier plusieurs propriétés de la surface comme la couleur, la réflexion ou la transparence. Cette texture peut être obtenue de diverses façons, en numérisant une

photographie par exemple. Bien souvent, la texture initiale est cependant trop petite et il est nécessaire de la répéter plusieurs fois pour couvrir toute la surface, ce qui peut occasionner des artéfacts visibles aux frontières et ainsi diminuer le réalisme. C'est à ce moment qu'intervient la synthèse de texture. Son but est de créer une nouvelle texture, en se basant sur un échantillon, qui est perçue comme le prolongement naturel de la texture de référence. Puisqu'elle est créée de toute pièce, la nouvelle texture peut avoir une taille plus grande et ainsi éliminer les artéfacts de répétition.

Bien que plusieurs types de modèles différents aient été proposés au fil des années, les modèles qui se basent sur les *Markov random field* (MRF) se sont démarqués par la qualité de leurs résultats. Cette section se concentrera donc spécifiquement sur ce type d'approches.

Les modèles basés sur les MRF sont fondés sur deux hypothèses : la synthèse de la nouvelle texture utilise un processus aléatoire qui est *local* et *stationnaire*. Le processus est *local* si chaque pixel peut être caractérisé par un petit ensemble des pixels voisins, appelé fenêtre, et que cette caractérisation est indépendante du reste de l'image. Le processus est *stationnaire* si cette fenêtre est la même pour tous les pixels. En pratique, ces deux hypothèses se traduisent de la façon suivante : la fenêtre de chacun des pixels de la texture de sortie doit être similaire à au moins une fenêtre de la texture de référence. Le principe derrière les méthodes basées sur les MRF est que la similarité de toutes les fenêtres locales garantit une similarité pour la texture dans son ensemble. Un des premiers travaux qui appliquent ce concept est celui de Efros et Leung (1999) qui présente une méthode de synthèse de textures automatique dans laquelle le seul paramètre est la taille du voisinage. Cette méthode est une pionnière dans le domaine de la synthèse de textures et elle présente des résultats d'une qualité étonnante. Cependant, l'utilisation d'une recherche exhaustive pour la sélection de la fenêtre la plus similaire rend cette technique peu performante.

Plusieurs travaux ont depuis proposé des façons d'augmenter la performance de l'approche de Efros et Leung (1999). Par exemple, Wei et Levoy (2000) proposent d'utiliser une structure de recherche en arbre tandis que Lefebvre et Hoppe (2006) optent plutôt pour

réduire la dimensionnalité de la fenêtre du voisinage. Dans le but de diminuer le nombre de recherches, Ashikhmin (2001) propose d'utiliser le principe de *cohérence*. Ce principe se base sur l'hypothèse que deux pixels voisins dans la texture de référence ont de fortes chances d'être également voisins dans la texture de sortie. Tong *et al.* (2002) poussent cette hypothèse encore plus loin et proposent le concept de *k-coherence*. Certaines méthodes (Efros et Freeman, 2001; Kwatra *et al.*, 2003; Liang *et al.*, 2001) utilisent quant à elles une approche par pièce (*patch-based*) contrairement à l'approche pixel par pixel classique pour diminuer le nombre de recherches.

Ashikhmin (2001) innove également en offrant à l'artiste un contrôle sur l'apparence de la texture de sortie. Alors que la méthode présentée par Efros et Leung (1999) crée une texture de sortie totalement aléatoire, Ashikhmin (2001) permet à l'artiste de dessiner sommairement le résultat attendu et la méthode se base sur cette information pour diriger l'apparence de la texture de sortie. Un exemple de résultat obtenu par la méthode de Ashikhmin (2001) est présenté sur la figure 1.9.

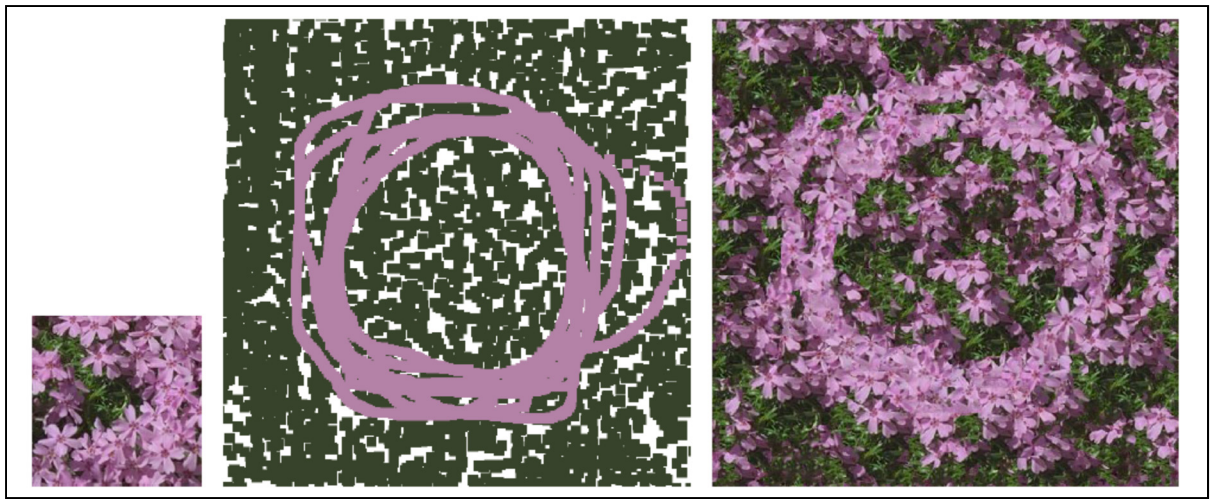


Figure 1.9 Résultat obtenu avec la méthode de Ashikhmin *et al.*
Tirée de Ashikhmin *et al.* (2001, p. 217)

Hertzmann *et al.* (2001) proposent également des améliorations majeures au travail de Efros et Leung (1999). Par exemple, Hertzmann *et al.* (2001) utilisent un processus multi-résolution pour créer la texture de sortie. L'information sur la structure globale de la texture échantillon est conservée dans les résolutions grossières, tandis que l'information sur les

détails plus subtils est contenue dans les résolutions plus fines. Par conséquent, la taille du voisinage est toujours fixe et l'artiste n'a pas besoin de la spécifier. De plus, la méthode donne à l'utilisateur un plus grand contrôle sur le résultat final puisque celui-ci peut dessiner approximativement ce qu'il souhaite obtenir.

1.3 Remplissage de régions dans une image

La synthèse de textures peut être utilisée dans plusieurs situations, telles que le remplissage de régions dans une image. Que ce soit pour réparer des égratignures sur de vieilles photos, pour enlever un élément indésirable (fil, perche, micro, etc.) dans une image ou pour enlever du texte imprimé sur une photo, il est souvent nécessaire de supprimer certaines régions d'une image et de remplacer celles-ci par quelque chose de plausible de façon à ce qu'une personne ne soit pas capable de percevoir que l'image ait été altérée. La figure 1.10 montre un exemple d'une image avec une région indésirable (micro) qui devait être enlevée et corrigée.

Figure 1.10 Exemple d'image nécessitant une retouche.
Tirée de Bertalmio *et al.* (2000, p. 424)

Bertalmio *et al.* (2000) proposent une solution à ce problème basée sur la synthèse de textures. Leur approche consiste à propager itérativement l'information sur la couleur des pixels autour de la région indésirable vers les pixels à l'intérieur de celle-ci. La figure 1.10 montre un résultat obtenu à l'aide de cette approche. La méthode de Bertalmio *et al.* (2000) produit des résultats de bonne qualité lorsque la région à remplacer est petite, mais elle présente des artéfacts visuels lorsque la région à compléter devient trop grande. Criminisi, Perez et Toyama (2003) proposent une méthode similaire à celle de Bertalmio *et al.* (2000) qui est cependant capable de corriger de plus grandes régions. Ils y arrivent en établissant un ordre de priorité selon lequel les pixels de la région manquante doivent être traités. Les pixels présents sur la continuité d'un contour très prononcé et ceux ayant un niveau de confiance élevé sont traités en premier. Le niveau de confiance d'un pixel se base sur la similarité entre la fenêtre du pixel et la fenêtre la plus ressemblante dans l'échantillon. Criminisi, Perez et Toyama (2004) présentent une version plus détaillée de cette méthode. Drori, Cohen-Or et Yeshurun (2003) s'attaquent également au remplissage de grandes régions en proposant une méthode multi-résolutions qui permet de recréer des structures avec différents niveaux de détails. Cette méthode utilise également une notion de confiance semblable à celle de Criminisi, Perez et Toyama (2003). La figure 1.11 montre des résultats obtenus par la méthode de Drori, Cohen-Or et Yeshurun (2003). Cette figure met aussi en évidence que cette méthode est plus apte que celle de Bertalmio *et al.* (2000) pour le remplissage de grandes régions manquantes.

Figure 1.11 Résultats obtenus avec la méthode de Drori, Cohen-Or et Yeshurun.
Adaptée de Drori, Cohen-Or et Yeshurun (2003, p. 312)

L'ensemble de ces méthodes se bute cependant au même problème. Lorsque la région à compléter cache une structure complète, il n'y a aucun moyen de recréer cette structure. Sun *et al.* (2005) s'attaquent à cette problématique. Ils proposent une nouvelle approche pour la complétion d'image : la propagation de structure. Cette approche demande à l'artiste d'indiquer grossièrement l'information manquante sur les structures en prolongeant certaines lignes et courbes provenant de la partie connue de l'image. La méthode tient ensuite compte de cette nouvelle information et effectue une première passe durant laquelle elle recrée la structure manquante indiquée pas l'artiste. Pour y arriver, la méthode débute la complétion en remplaçant les pièces à l'intérieur de la région manquante se trouvant à proximité des lignes et courbes tracées par l'utilisateur en cherchant uniquement des pièces de remplacement situées près des lignes et courbes de la région connue. Une deuxième passe est ensuite effectuée en utilisant un algorithme de synthèse de texture sur les régions restantes. La figure 1.12 présente des résultats obtenus par cette méthode.

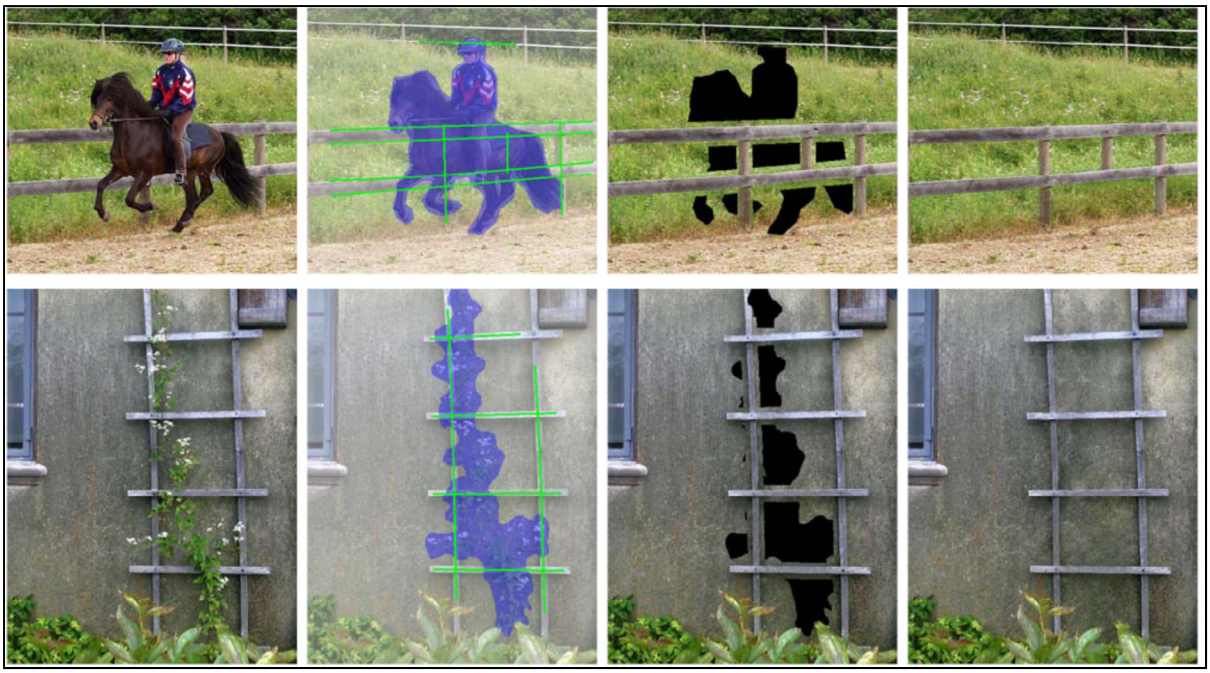


Figure 1.12 Résultats obtenus par la méthode de Sun *et al.*
Adaptée de Sun *et al.* (2005, p. 867)

Des travaux plus récents de Hays et Efros (2008) couvrent également le remplissage de régions manquantes dans une image. Leur approche remplit la région manquante en s'appuyant sur une base de données contenant des millions d'images existantes. Finalement, Ting, Chen et al. (2007) ont aussi présenté une méthode de remplissage de régions manquantes dans une image se basant sur les MRF et sur une technique de *Belief Propagation* (BP).

Ce recencement des méthodes de complétion d'images ne se veut pas exhaustif. Pour une revue plus complète des différentes méthode de complétion d'images, veuillez vous référer aux travaux de Wei *et al.* (2009) et Chunxiao *et al.* (2011).

1.4 Remplissage de régions dans une séquence vidéo

Les travaux portant sur la synthèse de texture et le remplissage de régions manquantes à l'intérieur d'images ont inspiré un certain nombre d'approches traitant le remplissage de régions manquantes dans une séquence vidéo. En effet, comme dans le cas d'une image, il est fréquent qu'une séquence vidéo nécessite d'être retouchée suite à sa captation pour supprimer des éléments indésirables (micros, perches, fils, etc.). Il est donc nécessaire de pouvoir éliminer ces régions indésirables et de les compléter d'une façon transparente pour le spectateur; ce dernier ne doit pas être en mesure de percevoir la modification. Cette section présente différentes méthodes de remplissage de régions manquantes pour une séquence vidéo et les divise en trois classes : traitement image par image, utilisation d'un arrière-plan fixe et minimisation d'une fonction d'énergie globale.

1.4.1 Traitement image par image

Bertalmio, Bertozzi et Sapiro (2001) ont fait partie d'un des premiers groupes de chercheurs à tenter de résoudre la problématique du remplissage d'une région manquante dans une séquence vidéo. Dans leur article, les auteurs présentent une méthode automatique de remplissage fondée sur une approche d'*inpainting*. Cette dernière se base sur les équations

qui régissent la dynamique des fluides pour propager les lignes d'isophotes contenues à l'extérieur de la région à remplacer vers l'intérieur de celle-ci. Leur hypothèse consiste à considérer l'intensité d'une image comme une fonction de courant pour un fluide bidimensionnel incompressible. Le Laplacien de l'intensité de l'image joue le rôle du tourbillon d'un fluide puisqu'il est transporté vers l'intérieur de la région manquante selon un vecteur défini par la fonction de courant. Cet algorithme est conçu pour propager les isophotes tout en harmonisant les vecteurs gradients aux frontières de la région à remplir. Bertalmio, Bertozzi et Sapiro (2001) fondent leur modèle sur les équations de Navier-Strokes pour l'écoulement de fluides dynamiques. Bien que la qualité des résultats obtenus pour une image fixe soit suffisante, cette approche crée des artéfacts visuels considérables lors du remplissage de régions manquantes dans une séquence vidéo. En effet, puisque le remplissage d'une image de la séquence vidéo est indépendant des images précédentes et suivantes, rien ne garantit que la cohérence temporelle sera conservée. Par conséquent, la variation de l'intensité des pixels d'une image à l'autre crée un effet de pétillement facilement perceptible par un spectateur. En raison de ce problème, le traitement image par image pour le remplissage d'une région manquante dans une séquence vidéo a rapidement été délaissé au profit de méthodes qui considèrent la séquence vidéo dans son ensemble afin de mieux conserver la cohérence temporelle. D'autre part, il est souvent difficile d'appliquer directement une méthode de complétion d'image au problème de la complétion vidéo puisque cette dernière se bute au défi additionnel d'avoir à traiter une très grande quantité d'informations ayant un impact autant sur l'espace mémoire nécessaire que sur le temps de recherche.

1.4.2 Utilisation d'un arrière-plan fixe (ou mosaïque)

Un des désavantages d'utiliser le traitement image par image est qu'il ne porte aucune considération à la variation de l'intensité d'un pixel d'une image à l'autre ce qui peut introduire un effet de pétillement facilement perceptible par un spectateur. Cette contrainte a eu comme impact d'orienter les recherches vers des solutions qui considèrent la séquence vidéo dans son ensemble pour compléter ses régions manquantes. En ce sens, un groupe

d'approches de complétion vidéo débute par la création d'une grande mosaïque qui représente l'arrière-plan contenu dans la séquence vidéo et qui est ensuite utilisée pour remplir les régions manquantes. Cette section présente quelques méthodes qui utilisent le concept de mosaïque de différentes façons.

Patwardhan, Sapiro et Bertalmio (2005) font parties des premiers à présenter une solution de cette catégorie et ils s'intéressent au cas spécifique de la séquence vidéo captée avec une caméra fixe. Pour un objet indésirable en mouvement et qui cache un arrière-plan statique, Patwardhan, Sapiro et Bertalmio (2005) utilisent un algorithme de synthèse de textures basé sur le travail d'Efros et Leung (1999) pour compléter la partie manquante de l'arrière-plan. Jia *et al.* (2004), Kokaram (2004) et Kokaram, Collis et Robinson (2005) présentent également des méthodes similaires pour le remplissage de trou dans une séquence vidéo qui sont basées sur l'utilisation d'une mosaïque.

Les méthodes de remplissage de régions manquantes dans une séquence vidéo précédemment mentionnées qui font l'utilisation d'un arrière-plan fixe n'emploient qu'une seule mosaïque. Cette mosaïque est une image de référence qui représente l'arrière-plan et est utilisée pour compléter les régions où l'objet indésirable cache l'arrière-plan. Or, le fait de n'avoir qu'une seule mosaïque ne permet pas de traiter une séquence vidéo captée à l'aide d'une caméra en mouvement puisque l'arrière-plan de telles séquences n'est pas fixe. Patwardhan, Sapiro et Bertalmio (2007) proposent une solution à ce problème. Plutôt que de n'avoir qu'une seule mosaïque pour une séquence vidéo complète, leur méthode propose d'utiliser trois mosaïques différentes. Ces dernières permettent de suivre, dans une certaine mesure, l'évolution de l'arrière-plan tout au long de la séquence vidéo sans augmenter considérablement le temps de calcul nécessaire à sa complétion. La méthode de Patwardhan, Sapiro et Bertalmio (2007) est efficace et montre de bons résultats. Cependant, la méthode éprouve des difficultés lorsque des objets en mouvement sont à proximité ou à l'intérieur d'une grande région manquante. De plus, la méthode fonctionne uniquement lorsque le mouvement de la caméra est très simple et lent, comme dans le cas d'une légère translation, puisque toutes les variations de l'arrière-plan de la séquence vidéo doivent pouvoir être présentées en seulement trois images.

D'autres auteurs poussent le concept encore plus loin en considérant une séquence vidéo comme une superposition de plusieurs couches ou segments. Zhang, Xiao et Shah (2005) font partie des premiers à présenter une solution de cette catégorie. En considérant une séquence vidéo d'entrée, le but de leur méthode est de pouvoir y enlever un objet et de remplir la région laissée vacante par de l'information sur la texture et la couleur qui est raisonnablement ressemblante au reste de la séquence. Leur méthode se divise en trois étapes distinctes. Premièrement, un algorithme de segmentation divise la séquence vidéo d'entrée en plusieurs couches qui contiennent chacune un objet ayant un mouvement précis et établit l'ordre selon lequel les différentes couches se superposent pour créer la séquence vidéo d'entrée. Deuxièmement, l'objet indésirable est enlevé simplement en supprimant la couche qui le contient. Cette suppression a comme effet de laisser des *trous* dans les couches avec un ordre d'importance inférieur. Un modèle de compensation du mouvement détermine le déplacement des objets d'une image à l'autre et remplit les *trous* de chacune des couches en se basant sur cette information. Finalement, les différentes couches sont recomposées pour créer la séquence vidéo de sortie.

Tel que vu précédemment, la méthode de Patwardhan, Sapiro et Bertalmio (2005) peut corriger, à l'aide d'une mosaïque fixe, un objet indésirable se déplaçant devant un arrière-plan fixe. De plus, leur méthode peut compléter une région manquante dans laquelle un objet en mouvement est partiellement caché par l'objet indésirable qui est enlevé. La méthode priorise les pixels manquants en donnant une importance plus élevée aux pixels appartenant à l'objet en mouvement. Ces pixels sont complétés en copiant des pièces de l'objet en mouvement provenant des images précédentes et suivantes qui ne sont pas cachées par l'objet indésirable. Cette étape est basée sur le travail de Criminisi *et al.* (2004). Une fois l'objet en mouvement complété, la couleur des pixels manquants restants est déterminée comme lors du remplissage de l'arrière-plan fixe. Plus récemment, Ebdelli, Guillemot *et al.* (2012) ont proposé une méthode de complétion vidéo qui se base sur les travaux de Patwardhan, Sapiro et Bertalmio (2005). Dans celle-ci, les pixels manquants sont estimés en utilisant une combinaison linéaire des K fenêtres (*patchs*) les plus similaires en utilisant une

technique de *neighbor embedding*. Cette approche diminue le temps nécessaire pour compléter une séquence vidéo, mais les résultats montrent plusieurs artefacts visuels.

Jia *et al.* (2006) présentent également une méthode permettant de faire le remplissage d'une région manquante lorsque des objets en mouvement sont partiellement cachés par l'élément indésirable. Premièrement, l'artiste identifie les différentes couches de la séquence vidéo ayant des mouvements similaires à l'aide d'un mécanisme de cadres clés et spécifie l'élément indésirable. La méthode sépare par la suite l'information de la couleur de celle de l'illumination. L'information manquante de l'arrière-plan est synthétisée à l'aide d'une approche de synthèse d'images. Puis, l'information manquante pour les éléments en mouvement est déduite à partir de l'alignement spatio-temporel de plusieurs échantillons pris à différents niveaux de résolution. En séparant l'information de la couleur et celle de l'illumination, cette méthode est beaucoup plus efficace pour les séquences vidéo avec des changements d'illumination.

Contrairement aux travaux de Patwardhan, Sapiro et Bertalmio (2005) et Jia *et al.* (2006) qui traite uniquement la situation où un objet en mouvement est *partiellement* caché, Koochari et Soryani (2010) s'attaquent de leur côté aux situations lorsque l'objet en mouvement est *complètement* caché par l'objet indésirable. L'information de l'arrière-plan et de l'avant-plan est divisée en deux couches et les régions manquantes de l'arrière-plan sont complétées en utilisant la méthode de Criminisi, Perez et Toyama (2004). Par la suite, la méthode de Koochari et Soryani (2010) utilise un processus itératif pour créer une image référence statique contenant tout le cycle d'un objet en mouvement. La figure 1.13 montre un exemple d'une telle image référence statique. Par la suite, la méthode utilise cette image référence pour compléter l'avant-plan. La figure 1.14 présente des résultats obtenus à l'aide de cette méthode.

Shen *et al.* (2006) s'attaquent quant à eux aux séquences vidéo captées à l'aide d'une caméra fixe ou en mouvement. Le mouvement de la caméra ajoute un niveau de complexité puisqu'il introduit des effets de distorsion dans la séquence vidéo. La méthode de Shen *et al.* (2006)

sépare les objets en mouvement de l'arrière-plan, puis elle construit une variété de volumes spatio-temporels contenant chacun des objets en mouvement dans la séquence vidéo. Ces volumes sont par la suite projetés sur un plan 2D et ces mosaïques sont ensuite utilisées pour remplir les régions manquantes. La méthode de Shen *et al.* (2006) présente des résultats acceptables, mais la présence d'artefacts visuels et l'incapacité à remplir de grandes régions manquantes sont deux de ses limitations majeures. De plus, les mouvements de caméra sont limités à des rotations et mises à l'échelle simples.

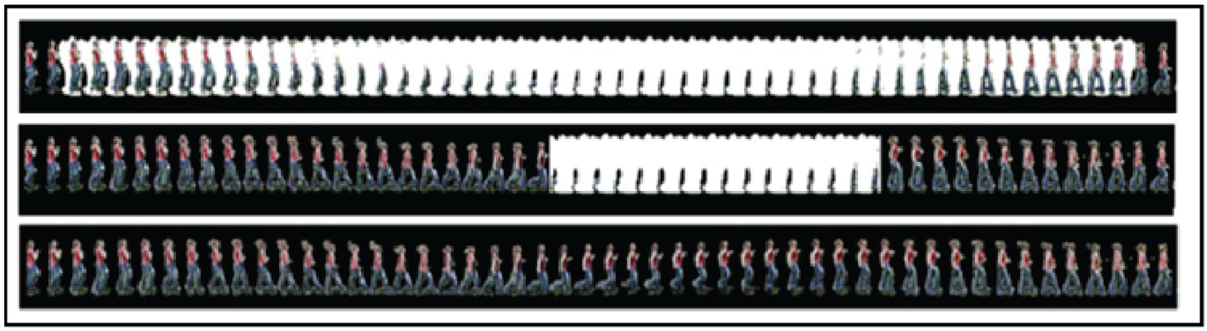


Figure 1.13 Itérations de l'image référence statique pour un objet en mouvement.
Adaptée de Koochari et Soryani (2010, p. 274)

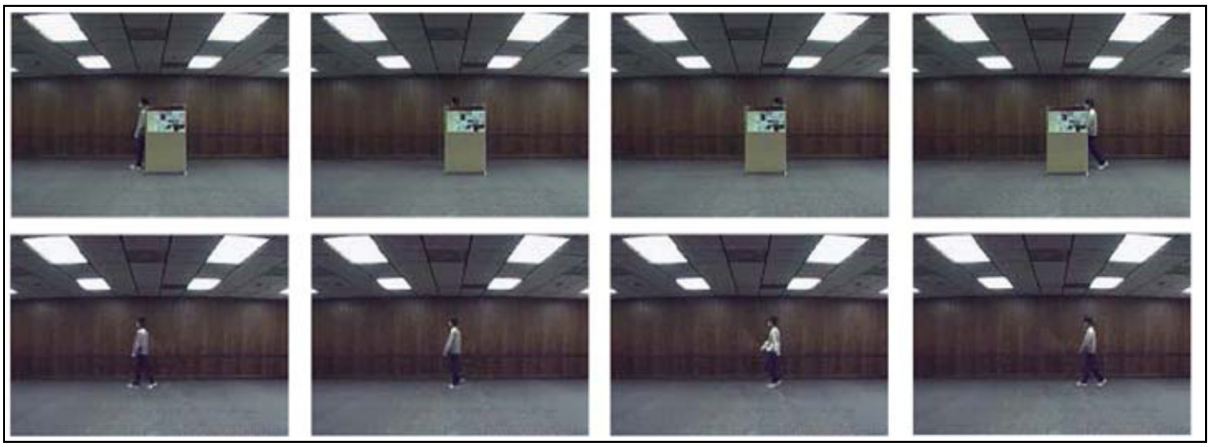


Figure 1.14 Résultats obtenus par la méthode de Koochari et Soryani.
Tirée de Koochari et Soryani (2010, p. 275)

De leur côté, Venkatesh, Cheung et Zhao (2009) séparent également la séquence vidéo en avant-plan (objet dynamique) et en arrière-plan afin de compléter les régions manquantes de chaque plan de façon indépendante. L'arrière plan, qui se doit d'être statique, est complété à

l'aide d'une image mosaïque et d'un algorithme de complétion d'image. Les objets en avant-plan sont quant à eux peints en utilisant des *modèles d'objets* qui minimisent une fonction de coût. Cette approche est cependant limitée par la durée et la résolution des séquences qu'elle est en mesure de traiter et n'est pas robuste aux mouvements de caméra, ce qui restreint son utilisation.

Bien que les méthodes basées sur l'utilisation d'une ou d'un petit nombre de mosaïques fixes donnent de bons résultats dans certains cas, elles possèdent des inconvénients majeurs qui limitent leur utilisation. Premièrement, puisqu'elles utilisent un nombre limité de mosaïques statiques pour compléter l'arrière-plan, les séquences vidéo retouchées ne peuvent pas contenir de mouvements dans l'arrière-plan (aucun mouvement ou de très légers mouvements). Par conséquent, ces méthodes sont, pour la plupart, limitées à des séquences vidéo captées à l'aide d'une caméra fixe. Or, puisqu'un grand pourcentage des séquences vidéo réelles est capté à l'aide d'une caméra en mouvement, les méthodes basées sur l'utilisation d'une ou d'un petit nombre de mosaïques fixes ne sont pas assez versatiles pour être utilisées dans un contexte réel de production. Deuxièmement, l'utilisation d'une ou d'un petit nombre de mosaïques fixes implique que l'intensité doit être constante durant toute la séquence vidéo sans quoi des artéfacts seront visibles dans les régions complétées. Par conséquent, ces méthodes ne sont pas en mesure de compléter un grand ensemble de séquences vidéo. Finalement, ces méthodes ne donnent pas de bons résultats lorsque les régions à remplir sont trop grandes puisque la qualité du résultat du processus de remplissage de trous est dépendante de la profondeur de la région manquante.

1.4.3 Minimisation d'une fonction d'énergie globale

Puisque les méthodes de remplissage de régions manquantes basées sur l'utilisation d'une ou d'un petit nombre de mosaïques fixes présentent plusieurs limitations majeures, les chercheurs se sont également intéressés à des méthodes qui traitent la séquence vidéo dans son ensemble sans avoir à la segmenter en avant-plan et en arrière-plan ou en couches.

Wexler, Scechtman et Irani (2004) sont les premiers à présenter une méthode de remplissage *spatio-temporelle* qui permet de traiter des séquences vidéo statiques et dynamiques. Les auteurs se basent sur la méthode de synthèse de texture présentée par Efros et Leung (1999) et utilisent un algorithme d'échantillonnage non paramétrique qui leur permet de traiter l'information dynamique et statique d'une séquence vidéo de façon simultanée. Les régions manquantes de la séquence vidéo sont remplies en copiant des pièces (*patch*) spatio-temporelles similaires trouvées dans le reste de la séquence vidéo. Ces pièces sont choisies de façon à renforcir la cohérence spatio-temporelle globale de la séquence vidéo. Cette cohérence spatio-temporelle globale est vue comme un problème d'optimisation globale avec une fonction objective bien définie. Celle-ci s'assure que le remplissage respecte deux critères : (1) chaque pièce copiée vers la région manquante est similaire à au moins une pièce de la région connue de la séquence vidéo et (2) toutes les pièces copiées vers la région manquante sont cohérentes entre elles autant spatialement que temporellement. Bien que les résultats obtenus représentent une percée majeure dans ce domaine, la qualité de la complétion n'est pas suffisante dans un contexte réel de production. En effet, la région corrigée présente un flou assez prononcé qui crée un effet de fantôme facilement perçu par le spectateur. Ce flou résulte de la technique de recherche de la pièce la plus similaire qui fait une moyenne des meilleures pièces trouvées. De plus, la technique effectue des recherches exhaustives dans l'ensemble de l'information contenue dans la séquence vidéo qui demande beaucoup de temps et de mémoire. Par conséquent, la méthode peut uniquement traiter des séquences vidéo de basse résolution alors que la plupart des émissions de télévision et des productions cinématographiques sont filmées en haute résolution. Wexler, Scechtman et Irani (2007) ont raffiné leur méthode sans toutefois régler les problèmes énoncés précédemment. La méthode de remplissage de trous présentée au chapitre 3 de cette thèse se base sur le même raisonnement que les travaux de Wexler, Schchtman et Irani (2004; 2007) tout en éliminant ou en atténuant certaines de leurs limitations. Entre autres, la méthode présentée au chapitre 3 propose une technique de remplissage qui ne nécessite pas une structure de recherche gourmande en espace mémoire tout en diminuant le temps de recherche. De plus, la méthode présentée permet aussi d'éliminer l'effet de fantôme généralement associé aux résultats de Wexler, Schchtman et Irani (2004; 2007)

Shih *et al.* (2006) proposent également un algorithme se basant sur un système de remplissage par pièces. La technique utilisée pour évaluer la similarité de deux pièces est cependant différente de celle de Wexler, Schchtman et Irani (2004; 2007). En plus de l'information sur la couleur, elle tient également compte de la propriété des contours. Leur hypothèse est que l'œil humain est sensible aux fortes variations d'intensité comme dans le cas d'un contour. Par conséquent, Shih *et al.* (2006) priorisent la complétion des régions sur lesquelles des contours prononcés sont présents. De plus, Shih *et al.* (2006) proposent également une méthode pour effectuer le transfert d'objets d'une séquence vidéo vers une autre. La figure 1.15 montre un exemple de transfert alors que les personnages d'une séquence vidéo sont transférés vers une séquence vidéo existante.

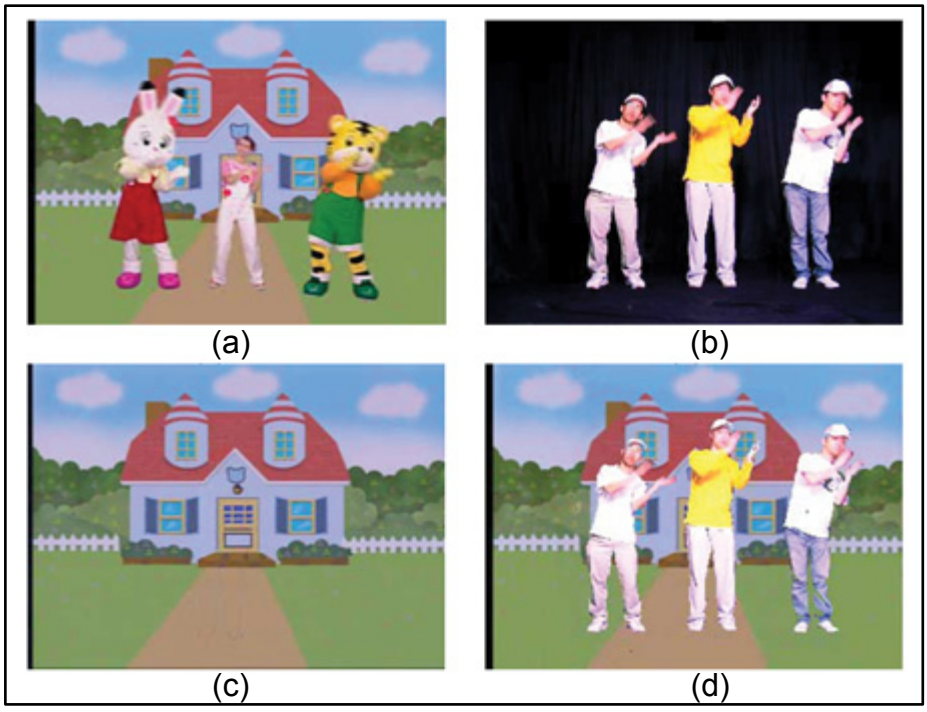


Figure 1.15 Transfert d'objets d'une séquence vidéo vers une autre.
(a) séquence originale; (b) séquence source du nouvel avant-plan;
(c) mosaïque de l'arrière-plan et (d) surimpression finale.
Adaptée de Shih *et al.* (2006).

D'autres travaux présentent également des méthodes de remplissage qui utilisent une fonction d'énergie globale, mais qui n'utilisent pas uniquement l'information sur la couleur pour déterminer la similarité de deux pièces. Shiratori *et al.* (2006) et Xu et al. (2015) utilisent les champs de mouvement, Cheung, Frey et Jojic (2005) se fient sur des *epitomes*, Xiao *et al.* (2008) tiennent compte des champs de mouvement et de l'information sur la couleur, tandis que Mosleh, Bouguila et Ben Hamza (2012) exploitent les régularités anisotropes avec les bandelettes. Cependant, ces méthodes requièrent des temps de calculs très longs, même pour des séquences vidéo de petites résolutions.

Certains auteurs se sont spécifiquement attardés à l'optimisation du temps de recherche pour les méthodes de complétion vidéo. Par exemple, Zarif, Faye et Rohaya (2013) ont diminué le temps nécessaire pour les recherches en proposant d'utiliser une méthode basée sur une *matrice de cooccurrence des niveaux de gris,* mais cette approche est limitée au cas très précis d'un objet manquant statique dans une séquence vidéo avec caméra fixe. De leur côté, Newson *et al.* (2013) ont accéléré les travaux de Wexler, Schchtman et Irani (2004; 2007) en adaptant la méthode *PatchMatch,* présentée par Barnes, Schechtman *et al.* (2010), en lui ajoutant la notion de temporalité (domaine 3D) de façon à diminuer le temps de recherche de la pièce la plus similaire. Cette méthode demeure cependant sensible aux changements d'intensité et est limitée à des séquences vidéo montrant des mouvements de caméra plus simples et lents que ceux qui nous intéressent au chapitre 4. Les mêmes auteurs (Newson et al., 2014) ont amélioré leur approche pour s'attaquer au problème de la complétion de régions contenant des textures dynamiques (vagues, feuilles d'arbre, etc.) en modifiant la métrique de distance des pièce afin d'identifier celles-ci. Les auteurs proposent également l'utilisation d'une pyramide multi-résolution contenant l'information sur les textures. Cette information sur les textures est complétée, au même titre que celle portant sur la couleur, à chaque niveau de résolution par la minimisation d'une fonction d'énergie globale. Cette approche offre d'excellents résultats, mais elle nécessite beaucoup d'espace mémoire, entre autre pour la pyramide multi-résolution contenant l'information sur les textures, ce qui rend difficile son utilisation sur des séquences vidéo HD.

Herling et Broll (2014) proposent quant à eux une solution qui permet d'obtenir une méthode de complétion vidéo en temps réel qui a évidemment comme objectif de réduire le temps de recherche. Pour y arriver, les auteurs utilisent uniquement l'image à corriger et la précédente afin de compléter la région manquante. Par conséquent, cette méthode est limitée à des séquences vidéo relativement simples dans lesquelles l'information manquante se trouve toujours dans ces deux images.

Pour régler le problème des longs temps de calcul, Ebdelli, Le Meur et Guillemot (2015) ont proposé une amélioration de leur précédente approche (Ebdelli, Guillemot et Le Meur, 2012). Pour compléter une région manquante dans une image de la séquence, cette nouvelle approche considère uniquement les vingt images précédentes et les vingt images suivantes. Par conséquent, la recherche des pièces manquantes se fait plus rapidement puisque la région valide est beaucoup plus petite. Il s'agit cependant d'une contrainte majeure puisque l'information pour compléter une image doit impérativement se trouver dans cet intervalle de quarante images, sans quoi le remplissage ne sera pas en mesure d'obtenir un bon résultat.

La majorité des méthodes de remplissages basées sur la minimisation d'une fonction d'énergie globale donnent des résultats d'une qualité acceptable. Cependant, plusieurs limitations majeures les empêchent d'être utilisées dans un contexte réel de production. Premièrement, la structure de recherche utilisée pour trouver la pièce la plus similaire demande beaucoup de mémoire. Par conséquent, la résolution des séquences vidéo qui peuvent être traitées est très basse. Or, les industries de la télévision et du cinéma utilisent de plus en plus des séquences vidéo en haute définition (HD) qui ont une résolution beaucoup plus élevée. De plus, le temps nécessaire pour compléter une séquence vidéo est très long, même pour une séquence à basse résolution, ce qui rend l'utilisation d'une telle méthode moins intéressante. Aussi, ces méthodes sont sensibles aux variations d'intensité dans la séquence vidéo. Finalement, les méthodes actuelles ne sont pas en mesure de compléter de grandes régions manquantes. En effet, plusieurs artefacts visibles sont présents dans les régions complétées. Pour toutes ces raisons, les méthodes actuelles minimisant une fonction

d'énergie globale ne sont pas adéquates pour une utilisation dans un contexte réel de production.

En terminant, il est important de mentionner l'approche de Granados, Tompkin *et al.* (2012) qui se base sur la méthode de complétion d'image de Pritch *et al.* (2009). Bien qu'elle se catégorise comme une méthode de complétion vidéo qui minimise une fonction d'énergie globale, elle ne présente pas les avantages et les inconvénients généralement associés à cette catégorie. Cette méthode est en mesure de compléter des séquences vidéo avec une résolution plus grande que celle des travaux antérieurs et présente des mouvements de caméra non-triviaux. Cependant, elle nécessite de longs temps de calcul (jusqu'à 90 heures pour certaines séquences vidéo) et l'assistance active de l'utilisateur pour réduire l'espace de recherche. Ce faisant, cette méthode est peu adaptée au pipeline des studios de production.

1.5 Objectifs

Précédemment, plusieurs problèmes ont été répertoriés, tant pour la simulation d'effets de détérioration que pour le remplissage de régions manquantes dans une séquence vidéo, qui empêchent ou limitent l'utilisation des travaux antérieurs dans un contexte réel de production. Afin de solutionner ces problématiques, cette thèse propose trois approches de remplissage automatique de trous à l'intérieur d'images et de séquences vidéo qui améliorent les techniques MRF classiques. Ces approches visent trois objectifs principaux. Premièrement, concevoir une méthode de simulation d'effets de détérioration basée sur une image échantillon qui améliore le réalisme des objets de synthèse. Deuxièmement, définir une approche de remplissage de régions manquantes de séquences vidéo haute définition basée sur une recherche locale qui ne nécessite pas de structure de recherche volumineuse. Finalement, élaborer une méthode de remplissage de régions manquantes pour des séquences vidéo présentant des changements d'intensités et des mouvements de caméra non-triviaux en se basant sur une approche de suivi des caractéristiques invariantes. Cette section détaille ces objectifs et explique de quelle façon ce projet de recherche peut être utilisé dans un contexte réel de production.

1.5.1 Génération automatique d'effets de détérioration

Tel que mentionné précédemment, la création d'effets de détérioration pour des objets de synthèse est primordiale afin d'augmenter leur réalisme. Le premier objectif de cette thèse est donc de concevoir une méthode de simulation d'effets de détérioration basée sur une image échantillon adaptée aux artistes et au pipeline de production. Les paramètres de la méthode proposée doivent donc être simples et intuitifs pour l'artiste et ne doivent pas nécessiter de connaissances scientifiques pointues. De plus, la méthode doit permettre à l'artiste d'avoir un contrôle suffisant sur le positionnement et la forme de l'effet de détérioration qui sera créé sur un objet de synthèse 3D. La méthode devra également permettre de simuler plusieurs effets de détérioration comme la rouille, la patine et la corrosion de façon à ce que l'artiste n'ait pas besoin de recourir à plusieurs outils différents pour créer les différents effets de détérioration d'un même objet. Pour y arriver, l'approche proposée utilise une image échantillon d'un effet de détérioration de référence. L'acquisition de cette image échantillon ne doit pas nécessiter un mécanisme complexe de capture d'image. Finalement, la méthode proposée doit permettre d'avoir des corrections rapides à un résultat déjà obtenu afin de respecter le processus itératif de création dans un contexte de production. La méthode proposée est détaillée au chapitre 2.

1.5.2 Remplissage de vidéo à l'aide d'une recherche locale

Le deuxième objectif de cette thèse est de concevoir une méthode de remplissage de régions manquantes dans une séquence vidéo de haute définition. Pour que cette méthode soit utile dans un contexte réel de production, elle doit être en mesure de remplir de grandes régions. De plus, puisque les séquences vidéo HD sont de plus en plus utilisées à la télévision et au cinéma, la méthode doit être capable de traiter des séquences vidéo de hautes résolutions. Il est donc impératif de résoudre les problèmes liés à l'espace mémoire que nécessite la structure de données ainsi que les longs temps pour effectuer une recherche. La méthode proposée est détaillée au chapitre 3.

1.5.3 Remplissage de vidéo à l'aide des caractéristiques invariantes

Le troisième objectif de cette thèse est de concevoir une méthode de remplissage de régions manquantes d'une séquence vidéo de haute définition présentant des mouvements de caméra non-triviaux (mise à l'échelle, rotation, déplacement sur un plan, roulement etc.). De plus, la méthode doit être robuste face aux changements d'intensité présents dans la séquence vidéo et doit être en mesure de compléter de très grandes régions manquantes. La méthode proposée est détaillée au chapitre 4.

Ensemble, la conception d'une méthode de simulation d'effets de détérioration basée sur une image échantillon, la définition d'une méthode de remplissage de séquences vidéo basée sur une recherche locale ainsi que la conception d'une méthode de remplissage de séquences vidéo basée sur une approche de suivi des caractéristiques invariantes fournissent une contribution significative et ciblée aux problèmes de synthèse et de remplissage de régions manquantes à l'intérieur de textures et de séquences vidéo.

CHAPITRE 2

GÉNÉRATION AUTOMATIQUE D'EFFETS DE DÉTÉRIORATION

Tel que mentionné précédemment, le premier objectif de cette thèse est de concevoir une approche de simulation d'effets de détérioration basée sur une image échantillon qui est adaptée aux artistes et au pipeline de production[1]. Pour atteindre cet objectif, l'approche proposée doit impérativement présenter une interface simple et conviviale qui ne requière pas de connaissances scientifiques avancées, permettre la synthèse de plusieurs effets de détérioration, offrir un contrôle suffisant sur le résultat final et s'exécuter dans un court délai afin de permettre le raffinement interactif du résultat. Par conséquent, une approche de simulation basée sur une image échantillon, ou image de référence, est préconisée vis-à-vis d'une simulation basée sur la physique ou sur des paramètres empiriques. Ce chapitre présente l'approche proposée en détaillant étape par étape son fonctionnement, montre des exemples de résultats et discute des avantages et des limitations de la méthode proposée par rapport à l'état de l'art.

2.1 Présentation générale de l'approche proposée

L'objectif principal du système de simulation d'effets de détérioration proposé est d'offrir à l'artiste un mécanisme d'édition de textures contrôlant la création et la modification des effets d'usure. L'artiste applique par la suite ces textures sur des objets de synthèse et il est ainsi en mesure d'augmenter leur niveau de réalisme. Pour un artiste, la démarche la plus intuitive pour créer de l'usure consiste à se fier à une image de référence, comme une photographie, afin de créer un nouvel effet de détérioration semblable, sans toutefois être identique. Comme le montre la figure 2.1, le système se base sur cette démarche intuitive

---

[1] Le contenu du chapitre 2 a été publié dans le cadre d'une conférence internationale : Clément, Benoit et Paquette (2007). Cet article a obtenu le prix du « Best Paper Award ».

pour établir les étapes clés que l'artiste doit suivre lors de son utilisation. Dans un premier temps, l'artiste fournit une image source, généralement une photographie, bien qu'une image de synthèse puisse également être utilisée, dans laquelle se trouve un exemple d'effet de détérioration qu'il cherche à recréer. Par la suite, il fournit au système une autre image, le masque cible, qui indique l'endroit dans l'image source où l'on veut ajouter le nouvel effet de détérioration. Ce masque cible est une image binaire qui identifie la forme, ou le patron, de l'effet de détérioration attendu. Ce masque peut être créé en utilisant une application de traitement d'image externe, tel que *Adobe Photoshop*, ou à l'aide d'un outil offert par le système proposé. Avec l'image source et le masque cible, le processus d'édition d'effets de détérioration produit l'image de reproduction, c'est-à-dire l'image source où l'effet de détérioration initial a été identifié, supprimé et rempli et sur laquelle un nouvel effet de détérioration a été ajouté à l'endroit indiqué par le masque cible.

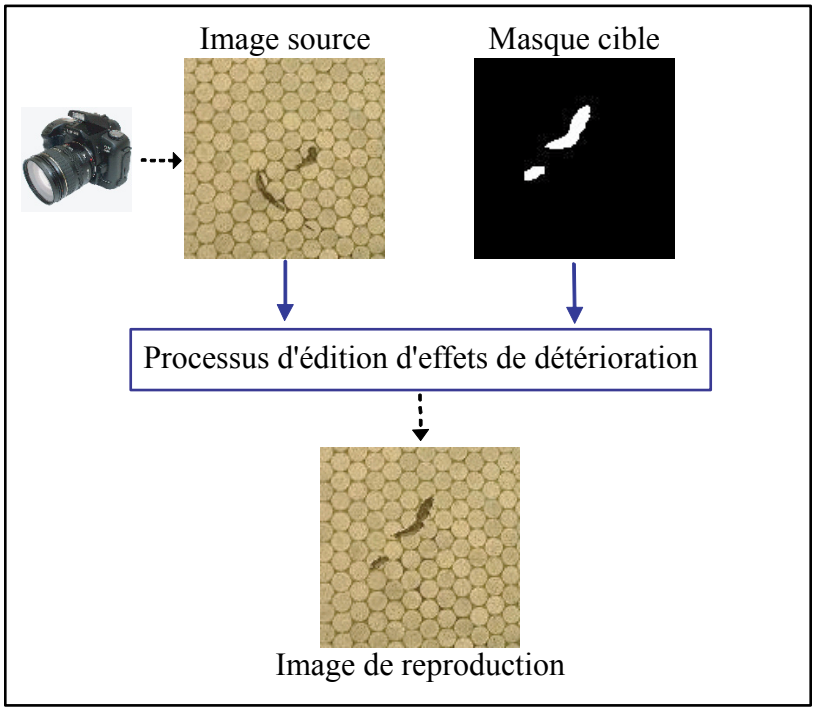


Figure 2.1 Présentation générale du système de simulation d'effets de détérioration.

Comme le montre la figure 2.2, le processus d'édition d'effets de détérioration est divisé en trois étapes : la segmentation, l'élimination (ou remplissage) et la reproduction. Le but de

l'étape de segmentation est d'identifier les régions de l'image source où l'on retrouve les différents effets de détérioration. Le résultat de cette étape est la création du masque source. Le masque source, au même titre que le masque cible, est une image binaire qui identifie la localisation des effets de détérioration. Quant à elle, l'étape d'élimination a comme objectif de nettoyer l'image source en remplissant les régions laissées manquantes par la suppression des effets de détérioration identifiés par le masque source. Pour y parvenir, une méthode de synthèse de texture avec une approche par remplissage de trous est utilisée. Le résultat de cette étape est l'image nettoyée. Finalement, l'étape de reproduction permet de synthétiser de nouveaux effets de détérioration, semblables à ceux présents dans l'image source, positionnés aux endroits spécifiés par le masque cible. Le processus d'édition proposé se démarque des travaux antérieurs parce qu'il est adapté au pipeline de production. En effet, la démarche artistique de création des artistes est un processus itératif qui demande beaucoup de cycles de raffinement et d'amélioration pour le même travail. Or, le processus d'édition d'effets de détérioration proposé est divisé de telle façon que les étapes de segmentation et d'élimination ne sont effectuées qu'une seule fois. Durant les itérations de raffinement, l'artiste peut refaire uniquement l'étape de reproduction pour modifier l'allure des effets de détérioration ajoutés. Ceci permet donc à l'artiste de minimiser le temps et l'effort nécessaires pour les cycles de révision du travail. Les étapes de segmentation, d'élimination et de reproduction sont détaillées respectivement dans les sections 2.2, 2.3 et 2.4.

Mes contributions personnelles à l'article de Clément, Benoit et Paquette (2007) présenté dans ce chapitre ont été concentrées principalement, mais pas uniquement, aux étapes d'élimination et de reproduction puisqu'elles traitent plus spécifiquement de remplissage de régions manquantes et de synthèse de texture.

2.2 Étape de segmentation

Tel qu'expliqué précédemment, l'étape de segmentation consiste à identifier les régions de l'image source dans lesquelles on retrouve les effets de détérioration et à créer le masque source correspondant. Cette étape est cruciale; la précision du masque source a un impact

direct sur la qualité des résultats obtenus durant les étapes suivantes d'élimination et de reproduction. Par exemple, si certains effets de détérioration ne sont pas totalement identifiés par le masque source, l'étape d'élimination laissera des artéfacts indésirables dans l'image nettoyée.

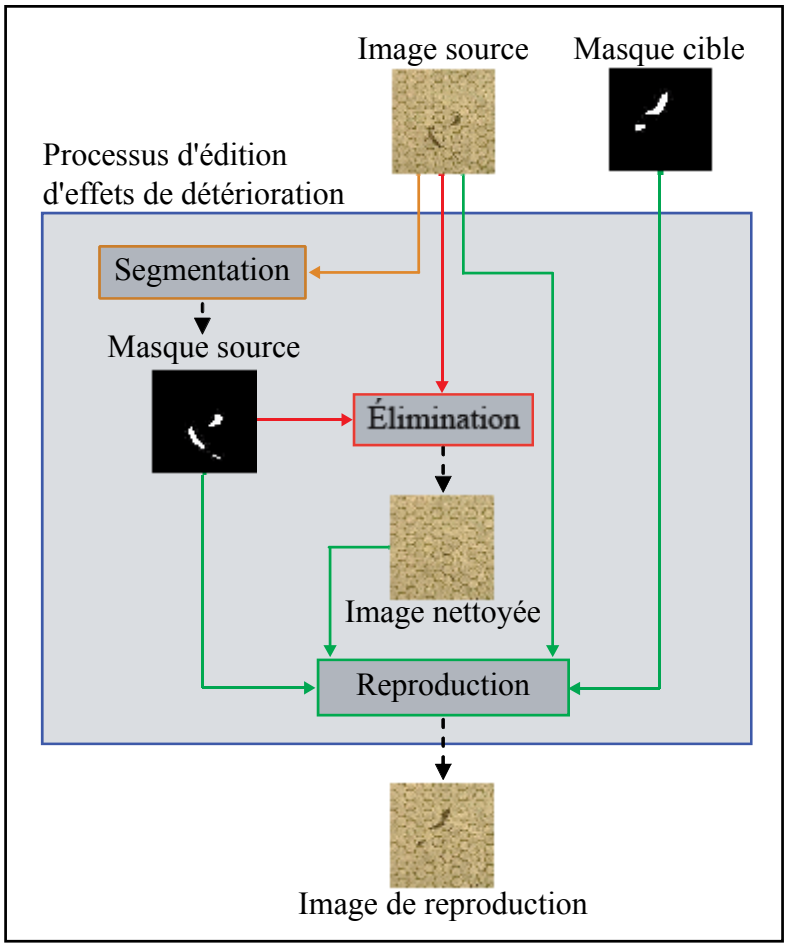


Figure 2.2 Processus d'édition d'effets de détérioration.

La création du masque source peut être réalisée à l'aide de logiciels externes de traitement et d'édition d'images tel qu'*Adobe Photoshop*. Le système d'édition d'effets de détérioration proposé contient cependant un ensemble complet d'outils de segmentation permettant à l'artiste de créer le masque source. Bien que la majorité des outils proposés soient inclus dans les différents logiciels commerciaux existants, leur présence dans le système d'édition rend ce dernier plus adapté au pipeline de production utilisé par les artistes que les travaux

antérieurs. En effet, en minimisant le nombre de logiciels et de systèmes distincts que doit utiliser un artiste, on réduit les pertes de temps pour passer d'un logiciel à l'autre et on augmente sa productivité.

La figure 2.3 montre l'interface de l'outil de segmentation interactif inclu dans le système d'édition d'effets de détérioration. Afin de créer le masque source, l'outil de segmentation permet à l'artiste d'identifier les régions détériorées de l'image source à l'aide de plusieurs techniques. L'interface fournit à l'artiste quatre techniques distinctes de segmentation : le seuillage sur le niveau de gris, le seuillage sur la couleur en utilisant les modèles HSV et RVB et une technique de segmentation par coups de pinceau. Ces techniques sont détaillées dans les sections 2.2.1 et 2.2.2. De plus, l'artiste peut combiner les résultats de différentes techniques de segmentation en utilisant des opérateurs ensemblistes comme l'union, la différence et l'intersection. L'artiste peut également altérer le masque source avec les outils morphologiques de dilatation et d'érosion. Lorsque requis, l'artiste peut aussi faire l'édition manuelle du masque source en utilisant le pinceau et l'efface.

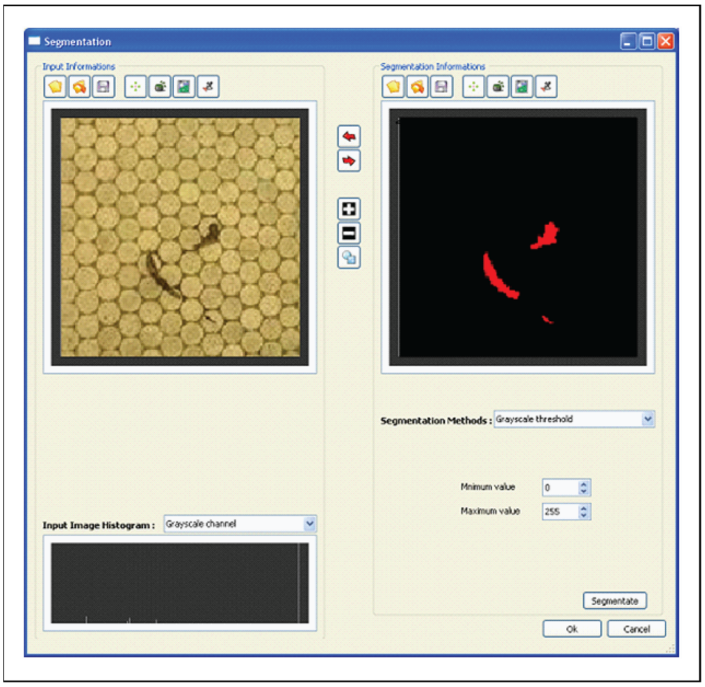


Figure 2.3 Interface de l'outil de segmentation interactif.

2.2.1 Segmentation par seuillage

Le seuillage est une technique de traitement d'image qui permet de convertir une image en niveaux de gris ou en couleurs vers une image binaire. Si l'intensité d'un pixel de l'image source est plus petite que la valeur du seuil, la couleur du pixel correspondant dans le masque source est fixée à noir (0). Inversement, si l'intensité du pixel est égale ou plus grande que la valeur du seuil, la couleur du pixel correspondant dans le masque source est fixée à blanc (255). Plutôt que d'utiliser un seul seuil, l'outil de segmentation interactif propose l'utilisation de deux seuils : $\text{seuil}_{\text{MIN}}$ et $\text{seuil}_{\text{MAX}}$. Ceci permet à l'artiste de sélectionner un intervalle d'intensité tel que décrit par l'équation (2.1).

$$\begin{matrix} blanc\ (255) : \text{seuil}_{\text{MIN}} \leq I \leq \text{seuil}_{\text{MAX}} \\ noir\ (0) : autrement \end{matrix} \tag{2.1}$$

Pour aider l'artiste à déterminer la valeur des seuils, l'outil de segmentation présente l'histogramme de l'image source qui montre la distribution des valeurs d'intensité. De plus, puisque cette opération est instantanée, l'artiste peut utiliser une approche par essais et erreurs pour raffiner la valeur des seuils selon les résultats de segmentation obtenus. La figure 2.4 montre un exemple de segmentation obtenu avec la technique de seuillage sur le niveau de gris.

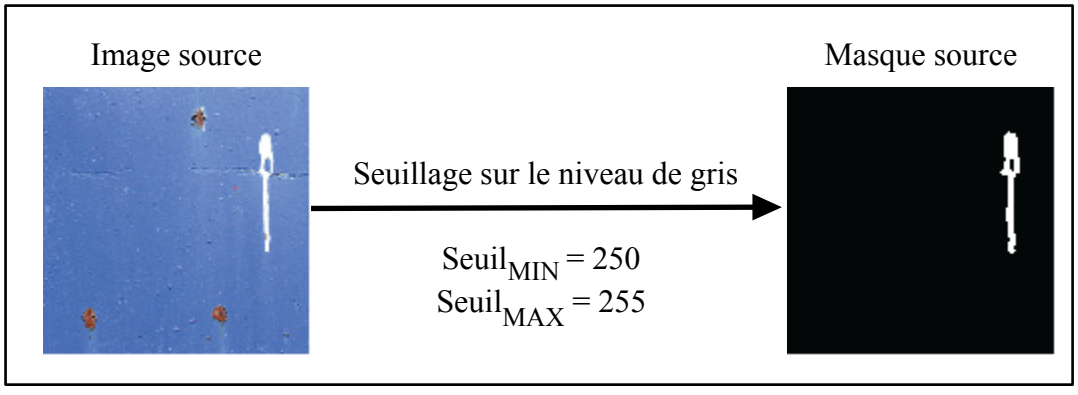


Figure 2.4 Segmentation par seuillage sur le niveau de gris.

L'outil de segmentation présente également une technique de seuillage similaire qui se base quant à elle sur l'information du triplet rouge-vert-bleu (RVB) des pixels de l'image source. Avec l'aide de celle-ci, l'artiste peut rapidement identifier un effet de détérioration qui présente une couleur particulière. L'équation (2.2) peut se substituer à l'équation (2.1) de façon à tenir compte de l'information sur la couleur.

$$\begin{aligned} blanc\ (255) : & \begin{cases} \text{seuil}_{\text{R}_{\text{MIN}}} \leq R \leq \text{seuil}_{\text{R}_{\text{MAX}}} & \text{ET} \\ \text{seuil}_{\text{V}_{\text{MIN}}} \leq V \leq \text{seuil}_{\text{V}_{\text{MAX}}} & \text{ET} \\ \text{seuil}_{\text{B}_{\text{MIN}}} \leq B \leq \text{seuil}_{\text{B}_{\text{MAX}}} \end{cases} \\ noir\ (0) : & autrement \end{aligned} \tag{2.2}$$

Tout comme pour le seuillage sur le niveau de gris, l'outil de segmentation montre l'histogramme des couleurs RVB afin de faciliter la sélection des différents seuils. Le modèle de couleur RVB n'est cependant pas très intuitif pour l'artiste; la représentation des couleurs dans le modèle RVB n'est pas faite d'une façon naturelle au système de vision humain. Par exemple, il n'est pas évident de déterminer les seuils RVB requis pour segmenter les pixels avec une couleur dans les teintes de rouge et d'orangée. Par contre, le modèle de couleur qui se base sur la teinte, la saturation et la luminance (HSV) est beaucoup plus approprié puisqu'il représente la couleur d'une façon plus naturelle au système de vision humain. C'est pourquoi l'outil de segmentation offre également la possibilité de faire un seuillage en utilisant le modèle de couleurs HSV, tel que décrit dans l'équation (2.3).

$$\begin{aligned} blanc\ (255) : & \begin{cases} \text{seuil}_{\text{H}_{\text{MIN}}} \leq H \leq \text{seuil}_{\text{H}_{\text{MAX}}} & \text{ET} \\ \text{seuil}_{\text{S}_{\text{MIN}}} \leq S \leq \text{seuil}_{\text{S}_{\text{MAX}}} & \text{ET} \\ \text{seuil}_{\text{V}_{\text{MIN}}} \leq V \leq \text{seuil}_{\text{V}_{\text{MAX}}} \end{cases} \\ noir\ (0) : & autrement \end{aligned} \tag{2.3}$$

Bien que les techniques de segmentation par seuillage soient relativement simples, elles permettent tout de même à l'artiste de créer rapidement et intuitivement le masque source dans certains cas. Cependant, la sélection des différents seuils peut être une tâche qui n'est pas triviale, mais l'utilisation des histogrammes et d'une approche par essais et erreurs permet à l'artiste d'obtenir rapidement de bons résultats avec peu d'effort. Bref, les

techniques de segmentation par seuillage donnent de bons résultats et ce rapidement, mais elles sont limitées à des cas relativement simples.

2.2.2 Segmentation avec la technique par coups de pinceau

Tel que mentionné précédemment, les techniques de segmentation par seuillage donnent de bons résultats, mais elles sont uniquement applicables dans certaines situations précises. Il est donc nécessaire d'offrir une technique de segmentation qui peut traiter un éventail de cas plus grand tout en restant facile d'utilisation. L'outil de segmentation interactif offre donc une technique de segmentation par *coups de pinceau* inspirée du travail de Lischinski *et al.* (2006) où une nouvelle fonction d'énergie est définie. Avec cette technique, l'artiste doit simplement tracer des coups de pinceau approximatifs sur les effets de détérioration contenu dans l'image source et le système effectue automatiquement la segmentation des régions correspondantes. La méthode attribue à chaque pixel de l'image une étiquette *détériorée* ou une étiquette *non détériorée* de façon à minimiser une fonction d'énergie. La figure 2.5 présente la succession des étapes qui permettent la segmentation automatique par coups de pinceau.

Tel qu'illustré sur la figure 2.5, le coup de pinceau peut être dessiné rapidement et n'a pas besoin d'identifier précisément tous les pixels de l'effet de détérioration. Chaque pixel q identifié par le coup de pinceau de l'artiste est assigné à l'ensemble CP ($p \in CP$). L'ensemble CP constitue l'échantillon sur lequel se base l'algorithme de segmentation automatique pour assigner les étiquettes. Pour chaque pixel p de l'image source, l'algorithme désire attribuer une des deux étiquettes *détériorée* ou *non détériorée*.

Ce problème de vision peut être formulé de façon naturelle par la minimisation d'une fonction d'énergie. En effet, pour chaque pixel de l'image, l'algorithme désire assigner une étiquette *eti* qui minimise la fonction d'énergie présentée par l'équation (2.4).

$$E(eti) = E_{couleurs}(eti) + E_{contours}(eti) \quad (2.4)$$

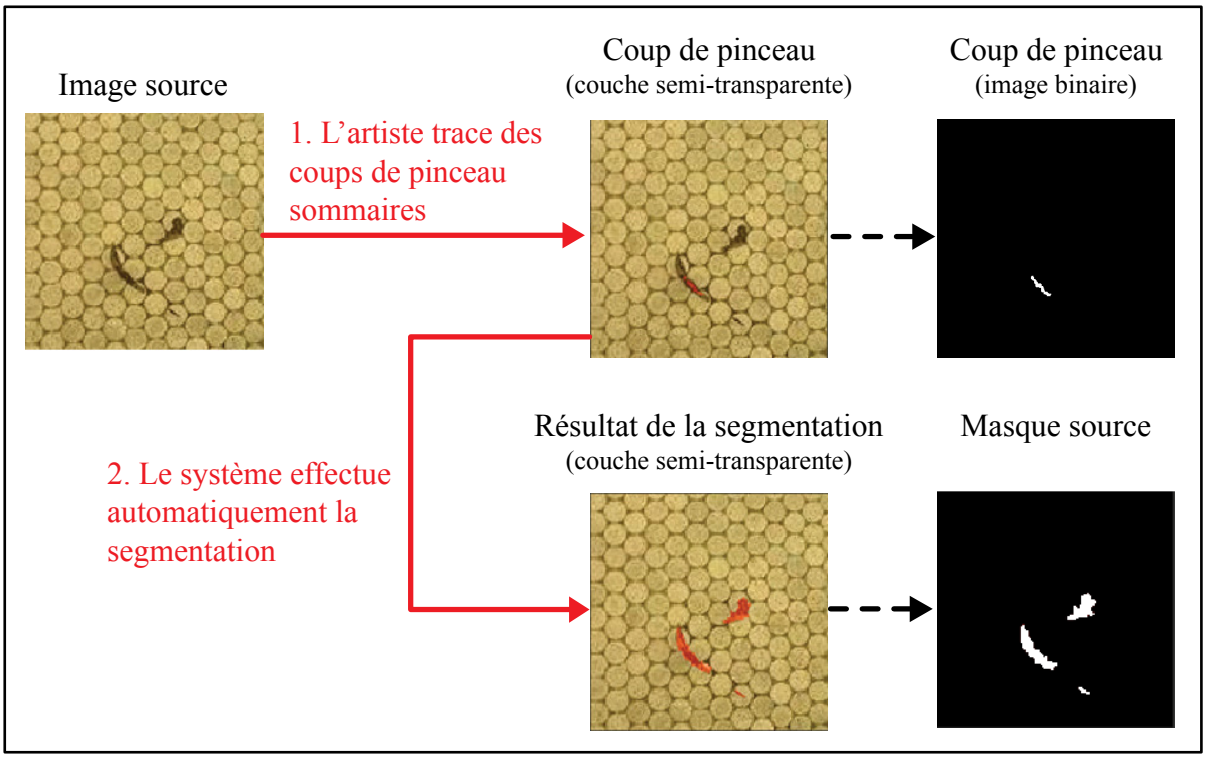


Figure 2.5 Étapes de la segmentation par coups de pinceau.

Le terme $E_{couleurs}$ de la fonction mesure la ressemblance entre la couleur du pixel *p* et celles des pixels contenus dans l'échantillon *CP*. Cela permet aux pixels avec une couleur correspondante à celles retrouvées sous les coups de pinceau de l'artiste d'avoir plus de chance d'obtenir l'étiquette *détériorée*.

Le terme $E_{contours}$ de l'équation (2.4) s'assure quant à lui que tous les pixels voisins appartenant à une même région obtiennent la même étiquette en se basant sur les contours trouvés dans l'image. Dans un premier temps, une détection de contours est effectuée dans l'image source à l'aide de l'algorithme de Canny. Puis, le poids énergétique d'un pixel augmente lorsque l'étiquette de ce dernier est différente de celle d'un pixel voisin et qu'aucun contour n'est présent à cet endroit. Ceci favorise la création des frontières entre les régions avec des étiquettes différentes en fonction des contours trouvés dans l'image source.

Pour assigner une étiquette *détériorée* ou *non détériorée* à chacun des pixels de l'image source de façon à minimiser la fonction d'énergie présentée par l'équation (2.4), la technique de segmentation automatique utilise une librairie externe créée par Boykov, Veksler et Zabih (2001). Puisqu'il ne s'agit pas d'une contribution de cette thèse, cette technique n'est pas détaillée. Nous vous invitons à consulter leur article pour avoir plus d'information à ce sujet.

Cette technique de segmentation basée sur les coups de pinceau est un ajout important à l'outil de segmentation interactif proposé. En effet, elle permet à l'artiste de rapidement pouvoir identifier les différents effets de détérioration contenus dans l'image source avec un minimum d'effort. De plus, cette technique est très simple et conviviale puisque l'artiste n'a pas besoin d'utiliser des paramètres physiques ou empiriques complexes; il doit simplement tracer quelques coups de pinceau çà et là. Aussi, puisque les résultats sont obtenus de façon interactive, l'artiste n'est pas ralentit dans son processus créatif. Cette technique est également très flexible et peut être utilisée dans un large éventail de situations. Durant nos expérimentations, la technique de segmentation par coups de pinceau est celle qui a été le plus souvent utilisée et qui a donné les meilleurs résultats.

2.2.3 Combinaison de techniques et édition manuelle

Bien que la technique de segmentation par coups de pinceau donne des résultats de très bonne qualité dans la majorité des cas, il demeure néanmoins des situations où elle n'est pas en mesure d'identifier correctement tous les effets de détérioration contenus dans l'image source. Ces situations surviennent généralement lorsque plusieurs effets de détérioration distincts se trouvent dans une même image. Par conséquent, il est primordial que l'outil de segmentation interactif permette de combiner les résultats de plusieurs techniques de segmentation différentes de façon à obtenir un résultat satisfaisant pour ces cas. L'outil de segmentation interactif offre donc à l'artiste plusieurs opérateurs ensemblistes (union, différence et intersection) qui lui permettent de combiner les résultats de plusieurs techniques de segmentation.

Toutefois, dans certaines situations, aucune technique de segmentation automatique n'est assez précise pour les besoins de l'artiste. Or, tel que mentionné préalablement, il est primordial pour l'artiste de fournir un masque source qui identifie correctement tous les effets de détérioration de l'image source, sans quoi les résultats des étapes d'élimination et de reproduction seront de moins bonne qualité. Pour pallier ces situations problématiques, le système interactif de segmentation offre à l'artiste une série d'outils pour faire l'édition manuelle du masque source. En effet, l'artiste dispose des outils classiques de dessin, comme le crayon et l'efface, pour modifier manuellement l'image de segmentation. Afin de faciliter l'utilisation de ces outils, le système permet à l'artiste de dessiner directement sur l'image source et affiche le masque cible par-dessus celle-ci sous la forme d'une couche rouge semi-transparente. L'artiste peut donc voir rapidement si le masque cible correspond bien aux différents effets de détérioration dans l'image. De plus, le système de segmentation interactif propose aussi des opérateurs morphologiques, comme l'érosion et la dilatation, pour altérer l'allure du masque source. Puisque les techniques automatiques de segmentation éprouvent à l'occasion des difficultés à la frontière des régions détériorées et non détériorées, les opérateurs morphologiques sont particulièrement intéressants pour élargir ou rétrécir les régions détériorées identifiées. En somme, le système interactif de segmentation offre des outils pour l'édition manuelle du masque source, tel que le crayon, l'efface et les opérateurs morphologiques, qui donnent une grande flexibilité et une meilleure précision pour l'identification des régions détériorées.

2.3 Étape d'élimination

Suite à l'étape de segmentation qui permet d'identifier les effets de détérioration dans l'image source et de créer le masque source correspondant, l'étape d'élimination permet quant à elle de supprimer toutes les traces des effets de détérioration de l'image source. Cette étape produit l'image nettoyée, soit l'image source dans laquelle toutes les régions détériorées ont été enlevées et remplacées à l'aide d'un algorithme de synthèse de texture par remplissage de trou. Cette image nettoyée est utilisée à l'étape de reproduction dans le but d'y ajouter de nouveaux effets d'usure. Contrairement à l'étape de segmentation qui

nécessite l'interaction de l'artiste, l'étape d'élimination est totalement automatique. Cette section détaille le fonctionnement du processus d'élimination.

2.3.1 Présentation générale de l'algorithme

Durant l'étape d'élimination, le système d'édition d'effets de détérioration utilise le résultat de l'étape de segmentation pour identifier et supprimer les régions détériorées dans l'image source. Pour y parvenir, le système propose une nouvelle approche de synthèse de texture adaptée des travaux d'Efros et Leung (1999) et de Hertzmann *et al.* (2001) qui se basent sur le concept des *Markov random field* (MRF) expliqué à la section 1.2. La nouvelle approche propose d'utiliser la synthèse de texture basée sur les MRF en y ajoutant des contraintes spécifiques au contexte de l'élimination d'effets de détérioration. En effet, contrairement aux techniques actuelles, l'approche proposée tient compte du caractère détérioré ou non détérioré du pixel et de son voisinage lors de la recherche du meilleur candidat. Tel qu'illustré sur la figure 2.6, l'algorithme traite un à un les pixels identifiés par le masque source qui appartiennent à la région détériorée et cherche pour chacun d'eux un pixel de remplacement dans la région non détériorée de l'image source. Cette recherche se base sur la couleur des pixels voisins afin de trouver le meilleur candidat.

Entrées : Image source, masque source

Image nettoyée ⟵ image source
Région à remplir ⟵ obtenir région à remplir (masque source)
Bordure ⟵ obtenir pixels en bordure (région à remplir)
Tant que *la région à remplir n'est pas vide* **faire**
    Pour chaque *pixel P dans Bordure* **faire**
        R ⟵ trouver meilleur candidat (P, image source, masque source)
        Image nettoyée (P) ⟵ image source (R)
        Région à remplir (P) ⟵ rempli
    Fin
    Bordure ⟵ obtenir pixels en bordure (Région à remplir)
Fin

Sortie : Image nettoyée

Figure 2.6 Pseudo-code de l'algorithme d'élimination.

Évidemment, puisque la recherche du meilleur candidat de remplacement se base sur la couleur des pixels contenus dans le voisinage du pixel à remplacer, l'ordre utilisé pour le remplissage a un impact direct sur la qualité des résultats. Tel qu'illustré sur la figure 2.6, l'algorithme proposé utilise un ordre de remplissage basé sur le remplissage de trous. Les détails de cette approche sont exposés à la section 2.3.2. Un autre élément de l'algorithme qui mérite une attention particulière est la technique utilisée pour sélectionner le candidat le plus ressemblant. Quelle métrique doit être utilisée pour mesurer la similitude entre deux voisinages? Étant donné le grand nombre de candidats possibles, peut-on minimiser le temps nécessaire pour effectuer une recherche afin d'obtenir des résultats dans un délai acceptable pour l'artiste? La section 2.3.3 répond à ces questions et détaille le mécanisme de sélection du meilleur candidat.

2.3.2 Ordre de remplissage

Les algorithmes de synthèse de texture basés sur les MRF prennent le voisinage du pixel à remplacer et cherchent le voisinage le plus similaire dans le reste de l'image ou de l'échantillon. Par conséquent, le contenu du voisinage du pixel à remplacer est d'une importance capitale. Prenons l'exemple illustré sur la figure 2.7, le voisinage du pixel de gauche fait majoritairement partie de la région non détériorée de l'image source tandis que celui du pixel de droite fait majoritairement partie de la région détériorée. Par conséquent, le résultat de la recherche du voisinage le plus similaire de l'exemple 1 sera beaucoup plus pertinent que celui de l'exemple 2.

Afin de maximiser la pertinence du voisinage du pixel à remplacer, il faut donc maximiser le nombre de pixels non détériorés du voisinage. Puisque le remplacement du pixel courant a un impact sur la pertinence du voisinage des pixels à proximité, il est donc important de s'attarder à l'ordre de remplissage. L'ordre de remplissage standard utilisé dans les travaux d'Efros et Leung (1999) et de Hertzmann *et al.* (2001) consiste à utiliser un voisinage en forme de « L » combiné avec une approche ligne par ligne. Pour la synthèse d'une image

complète, cette approche est très efficace et donne des résultats de bonne qualité. En effet, l'approche ligne par ligne assure que tous les pixels contenus dans le voisinage sont pertinents en tout temps puisque tous les pixels en haut et à gauche du pixel à remplacer ont nécessairement déjà été traités.

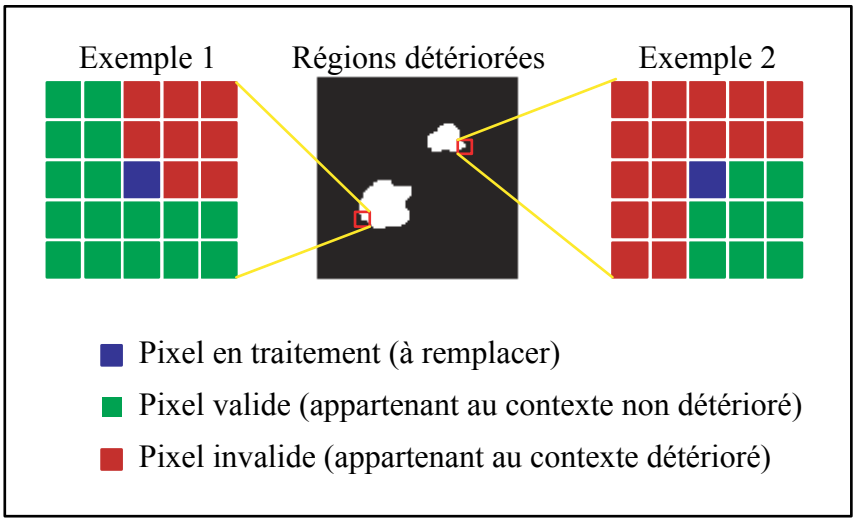


Figure 2.7 Pertinence du voisinage du pixel à remplacer.

Cependant, lors de la synthèse partielle d'une portion de l'image ou d'un trou, comme dans le cas qui nous intéresse, la technique ligne par ligne mène à des discontinuités visibles en bas à droite de la région complétée. La figure 2.8 illustre une telle situation. Ceci est attribuable au fait que la recherche du meilleur candidat de remplacement se base uniquement sur le voisinage situé en haut ou à gauche du pixel à remplacer.

Il est donc nécessaire de trouver une alternative à l'ordre de remplissage ligne par ligne pour éliminer les discontinuités visibles. L'alternative proposée consiste à traiter prioritairement tous les pixels en bordure de la région détériorée. Logiquement, ces pixels sont les plus susceptibles d'avoir le plus grand nombre de pixels non détériorés dans leur voisinage et de trouver un voisinage similaire dans l'image source. L'ordre de remplissage consiste donc à remplir itérativement tous les pixels en bordure de la région détériorée. De cette façon, il est possible de maximiser le nombre de pixels non détériorés pour le voisinage de chaque pixel à

remplir. La figure 2.9 montre l'exemple d'une itération de l'approche par remplissage de trous proposée.

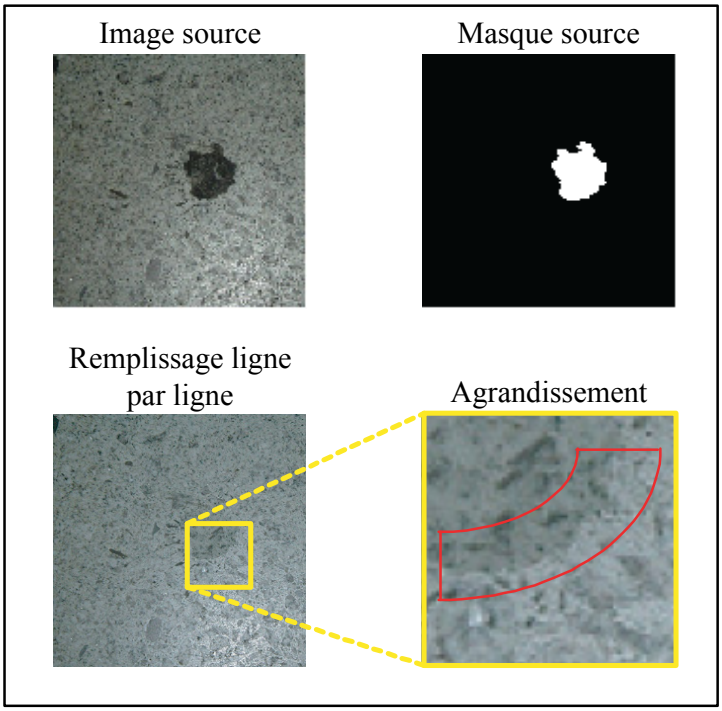


Figure 2.8 Discontinuité causée par le remplissage ligne par ligne.

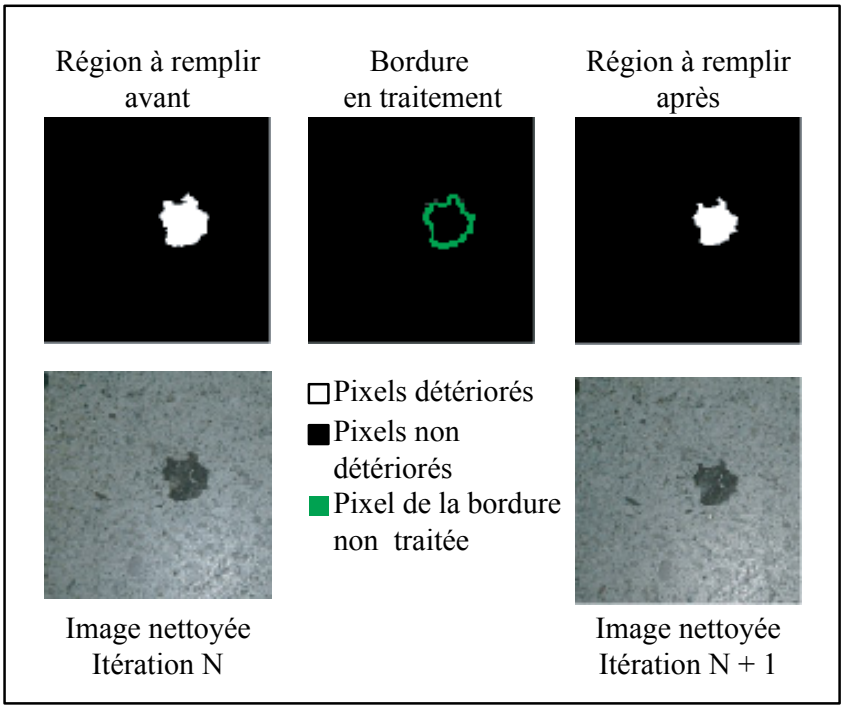


Figure 2.9 Une itération de l'approche par remplissage de trou.

L'utilisation de cette approche entraîne cependant un problème: contrairement au voisinage de l'approche ligne par ligne qui est constant pour tous les pixels, celui de l'approche par remplissage de trou est variable en fonction de la position du pixel à remplacer dans la région détériorée. En effet, pour un pixel situé en bas à gauche de la région détériorée, le voisinage en bas à gauche du pixel est utilisé tandis que pour un pixel situé en bas à droite, le voisinage en bas à droite du pixel est utilisé. Par conséquent, il est impossible d'avoir un voisinage unique et constant pour tous les pixels. Pour pallier ce problème, notre approche propose quatre voisinages distincts qui permettent de traiter les différents cas possibles. Lors du remplissage, le voisinage avec le plus grand nombre de pixels non détériorés pour un pixel à remplir est utilisé pour la recherche de son remplaçant. La figure 2.10 présente les quatre fenêtres proposées et illustre la distribution de leur utilisation pour un exemple donné.

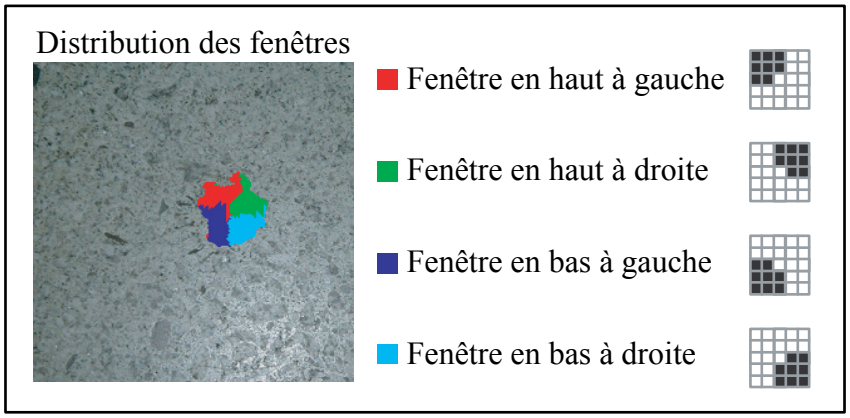


Figure 2.10 Distribution de l'utilisation des quatre fenêtres proposées.

L'utilisation de l'approche par remplissage de trou et la superposition partielle des quatre fenêtres proposées permettent de réduire considérablement le problème de discontinuité observé avec la technique ligne par ligne. La figure 2.11 compare les résultats obtenus avec la méthode ligne par ligne et celle par remplissage de trou.

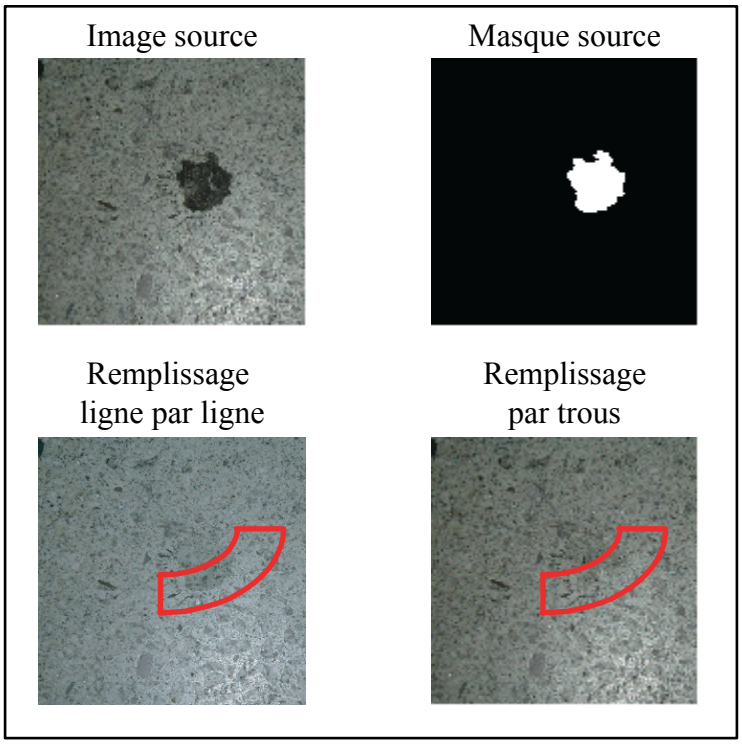


Figure 2.11 Comparaison des méthodes
ligne par ligne et remplissage de trou.

2.3.3 Sélection du meilleur candidat

Tel qu'expliqué à la section 2.3.2, l'ordre selon lequel les différents pixels manquants sont complétés a un impact direct sur la qualité des résultats obtenus. En effet, l'utilisation d'une approche de synthèse de texture par remplissage de trou permet de maximiser le nombre de pixels non-détériorés contenus dans le voisinage de chaque pixel à remplacer. Une fois cet ordre défini, il est également important de s'attarder à la méthode utilisée pour la sélection du meilleur candidat puisqu'elle influence aussi la qualité des résultats obtenus à l'étape d'élimination. Pour chaque pixel à remplacer, la méthode parcours la région non-détériorée de l'image source à la recherche du pixel avec le voisinage le plus similaire. La similarité entre deux voisinages se calcule à l'aide d'une sommation des distances euclidiennes sur les couleurs RVB de chaque pixel contenu dans ces voisinages. L'équation (2.5) détaille la métrique utilisée pour mesurer la similitude entre deux voisinages :

$$similitude(S,R) = \sum_{\mathrm{i,j} \in \mathrm{W}} \big(\mathrm{D(i,j)}\big)$$

$$\text{avec } D(i,j) = [S_w(i,j) - R_w(i,j)]^2 \tag{2.5}$$

où W représente la fenêtre du voisinage, S_w représente le voisinage autour du pixel à remplacer dans l'image nettoyée et R_w symbolise le voisinage autour du pixel candidat dans l'image source.

Une recherche force brute du voisinage le plus similaire dans la région non-détériorée de l'image source assure de trouver le meilleur candidat possible. De plus, elle permet une grande flexibilité au sujet de la forme et la taille de la fenêtre du voisinage puisque ces deux éléments peuvent être variables d'un pixel à l'autre. Cependant, la recherche force brute du meilleur candidat nécessite un délai de traitement beaucoup trop long. En effet, en utilisant cette approche pour la phase d'élimination, le délai de traitement se calcule généralement en heures. Or, ce long délai ne cadre pas bien dans le processus itératif de création des artistes. Bref, il est nécessaire de trouver une technique alternative pour la recherche du voisinage le plus similaire qui permet d'obtenir les résultats dans un contexte interactif.

Pour solutionner ce problème, l'approche proposée utilise une méthode développée par Arya *et al.* (1998) permettant d'effectuer une recherche approximative de plus proche voisin. Les auteurs offrent une librairie, *Library for an Approximate Nearest Neighbor Searching* (ANN), qui supporte différents types de structures de données et d'algorithmes pour effectuer la recherche des plus proches voisins pour des vecteurs à plusieurs dimensions. La librairie propose trois métriques sur lesquelles l'algorithme de recherche peut se baser soient L_1, L_2 et L_∞. La librairie offre également différentes structures de recherche dont le *k-dimensional tree* (*kd-tree*) pour accélérer la recherche. Cette structure est utilisée par la technique proposée puisqu'il a été démontré que le *kd-tree* est la structure la plus efficace pour les recherches approximatives (Barnes et al., 2009; Kumar, Zhang et Nayar, 2008). De façon à respecter

l'équation (2.5), la métrique L_2 est utilisée et le vecteur de recherche contient les valeurs RVB de tous les pixels non-détériorés contenus dans le voisinage du pixel à remplacer.

L'utilisation de la méthode développée par Arya *et al.* (1998) nécessite cependant une étape d'initialisation pendant laquelle les différentes structures de recherche (*kd-tree)* sont créées et initialisées. Tel que mentionné à la section 2.3.2, l'algorithme d'élimination utilise quatre fenêtres distinctes pour maximiser le contexte non-détérioré du voisinage pour le pixel à remplacer (voir figure 2.10). Il est donc nécessaire de créer, durant la phase d'initialisation, une structure *kd-tree* pour chacune des quatre fenêtres possibles. Par la suite, pour chaque pixel à remplacer, l'algorithme d'élimination identifie la fenêtre optimale du voisinage, construit le vecteur de recherche correspondant et envoie cette information à la librairie ANN qui fera la sélection du meilleur candidat possible. L'utilisation de la libraire ANN permet de réduire de façon considérable le temps associé à la recherche des meilleurs candidats permettant ainsi d'obtenir des résultats dans un délai plus adapté au processus itératif de création d'un artiste.

2.4 Étape de reproduction

Suite à l'étape de segmentation, qui permet d'identifier les effets de détérioration dans l'image source et de créer le masque source correspondant, et après l'étape d'élimination, qui permet de supprimer ces effets pour créer l'image nettoyée, l'étape de reproduction permet quant à elle d'ajouter de nouveaux phénomènes d'usure. Cette étape produit l'image de reproduction, soit l'image nettoyée dans laquelle de nouveaux effets de détérioration ont été ajoutés aux endroits indiqués par le masque cible en utilisant une technique de synthèse de texture avec contraintes basée sur une approche de remplissage de trou. Comme pour l'étape d'élimination, l'étape de reproduction est entièrement automatique et ne nécessite aucune interaction avec l'artiste. Cette section détaille le fonctionnement du processus de reproduction des effets de détérioration.

2.4.1 Approche de synthèse de détérioration

Durant l'étape de reproduction, le système d'édition d'effets de détérioration utilise le résultat des étapes de segmentation et d'élimination pour ajouter de nouvelles régions détériorées dans l'image nettoyée. Ces nouvelles régions détériorées ressemblent à celles contenues dans l'image source sans pour autant y être identiques. Pour y parvenir, le système propose un algorithme similaire à celui utilisé pendant l'étape d'élimination en y ajoutant cependant des contraintes spécifiques au contexte de la reproduction d'effets de détérioration. En effet, la méthode proposée tient compte du caractère détérioré ou non-détérioré du pixel et de son voisinage lors de la recherche du meilleur candidat de remplacement. Tel qu'illustré sur la figure 2.12, l'étape de reproduction consiste à considérer tous les pixels de l'image nettoyée identifiés par le masque cible et à trouver un candidat de remplacement dans la partie détériorée de l'image source. La façon de sélectionner le meilleur candidat diffère cependant de celle de l'étape d'élimination et elle est détaillée à la section 2.4.2.

Entrées : Image source, masque source, image nettoyée, masque cible

Image de reproduction ⟵ image nettoyée
Région à remplir ⟵ obtenir région à remplir (masque cible)
Bordure ⟵ obtenir pixels en bordure (région à remplir)
Tant que *la région à remplir n'est pas vide* **faire**
    Pour chaque *pixel P dans Bordure* **faire**
        R ⟵ trouver meilleur candidat (P, image source, masque source,
                                        image de reproduction, masque cible)
        Image de reproduction (P) ⟵ image source (R)
        Région à remplir (P) ⟵ rempli
    Fin
    Bordure ⟵ obtenir pixels en bordure (Région à remplir)
Fin

Sortie : Image de reproduction

Figure 2.12 Pseudo-code de l'algorithme de reproduction.

2.4.2 Sélection du meilleur candidat

Comme dans la phase d'élimination, la sélection du meilleur candidat se base sur l'information des couleurs RVB des pixels appartenant au voisinage du pixel à remplacer (voir équation 2.5). Cependant, la sélection du meilleur candidat se fie également au contexte détérioré et non-détérioré du pixel à remplacer et de son voisinage. De cette façon, l'approche de synthèse de texture par remplissage de trou utilisée permet de mieux recréer les transitions entre les régions détériorées et non-détériorées. La sélection du meilleur candidat de remplacement se base donc autant sur le contexte de la couleur que sur le contexte détérioré ou non-détérioré du voisinage tel qu'illustré sur la figure 2.13.

Il est donc nécessaire de modifier l'équation 2.5 de façon à ce que la métrique qui mesure la similitude entre deux voisinages tiennent également compte du contexte détérioré et non-détérioré.

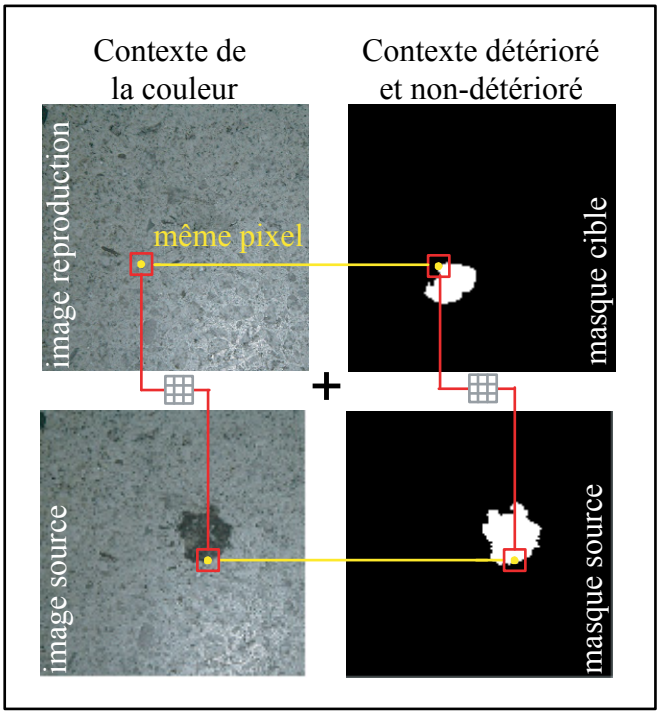


Figure 2.13 Représentation visuelle du nouvel ensemble de caractéristiques.

$$similitude\_reproduction(S,R) = \sum_{\mathrm{i,j} \in \mathrm{W}} \big(\mathrm{D(i,j)}\big)$$
$$\text{avec } D(i,j) = (1-\alpha)[S_w(i,j) - R_w(i,j)]^2 + (\alpha)[S_m(i,j) - R_m(i,j)]^2 \quad (2.6)$$

où W représente la fenêtre du voisinage, S_w symbolise le voisinage autour du pixel à remplacer dans l'image de reproduction, R_w représente le voisinage autour du pixel candidat dans l'image source, S_m symbolise le contexte détérioré et non-détérioré dans le masque cible et R_m représente le contexte détérioré et non-détérioré dans le masque source. Quant à lui, le paramètre α ($0 \leq \alpha \leq 1$) permet de pondérer et contrôler l'impact de la couleur RVB et du contexte détérioré et non-détérioré dans la recherche du meilleur candidat. Lorsque la valeur du paramètre α tend vers 0, l'algorithme de sélection favorise les candidats avec des couleurs RVB similaires et lorsque cette valeur tend vers 1, l'algorithme favorise les candidats avec un contexte détérioré et non-détérioré similaire. Ce paramètre offre un meilleur contrôle sur les résultats obtenus et ne nécessite pas de connaissances scientifiques étendues de la part de l'artiste. Pour la majorité des résultats présentés dans cette thèse, une valeur de 0,5 a été utilisée pour le paramètre α.

2.5 Combinaisons d'effets de détérioration

En plus d'ajouter de nouvelles régions détériorées, la phase de reproduction permet également à l'artiste de combiner plusieurs phénomènes d'usure provenant de différentes images sources. En effet, le système d'édition offre à l'artiste la possibilité d'appliquer plusieurs itérations de la phase de reproduction sur une même texture à partir de plusieurs images sources et de leur masque source correspondant. Cette approche permet à l'artiste de transférer un effet de détérioration vers une autre image tel qu'illustré sur les figures 2.14 et 2.15. Cette caractéristique est possible puisque l'algorithme de synthèse de texture par remplissage de trou permet au système de synthétiser uniquement des régions spécifiques de l'image de sortie. De plus, la phase de reproduction ne doit pas nécessairement partir de l'image nettoyée. Par conséquent, l'image de reproduction peut être construite itérativement à partir de plusieurs images sources.

2.6 Résultats

Cette section présente des résultats obtenus à l'aide du système d'édition d'effets de détérioration. Les figures 2.16 à 2.19 montrent des résultats obtenus sur une grande variété de textures (bois, céramique, ciment, etc.) et reproduisant plusieurs effets de détérioration tels que la rouille, les égratignures, les taches de peinture ou les taches d'huile. Les résultats des étapes d'élimination et de reproduction sont très réalistes autant pour les textures stochastiques que structurées. Ce grand éventail de textures et d'effets de détérioration démontre bien la flexibilité du système proposé. Les images présentées sur la figure 2.20 mettent en évidence la flexibilité de l'approche proposée qui fonctionne bien sur une grande variété de patrons pour le masque cible. En effet, à partir de l'image source, il est possible d'éliminer et de reproduire des régions détériorées de différentes formes et de traiter des masques cible et source qui se chevauchent.

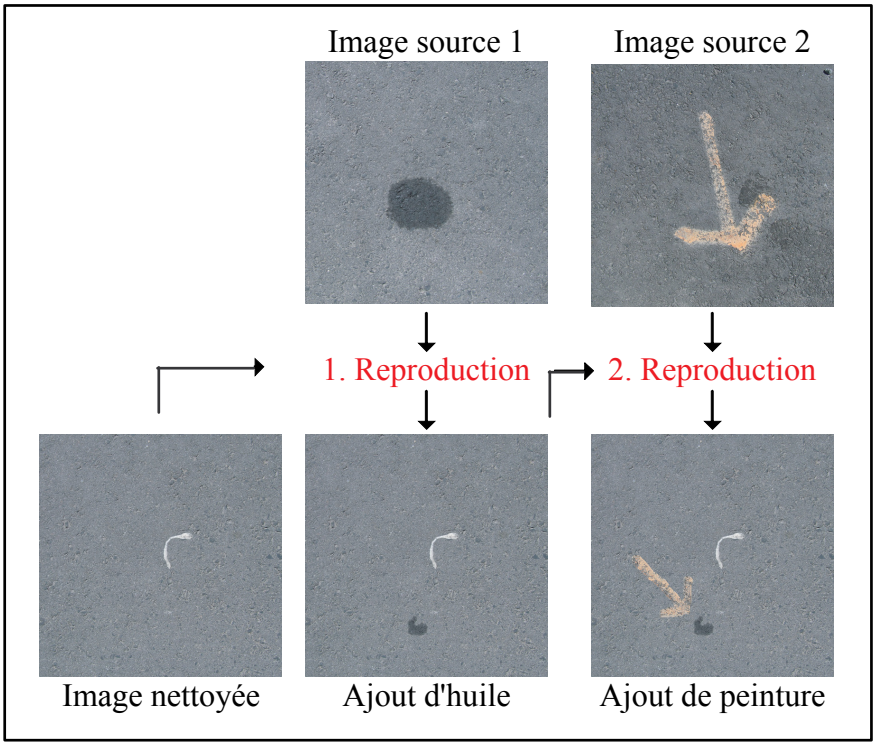


Figure 2.14 Combinaison d'effets de détérioration sur de l'asphalte.

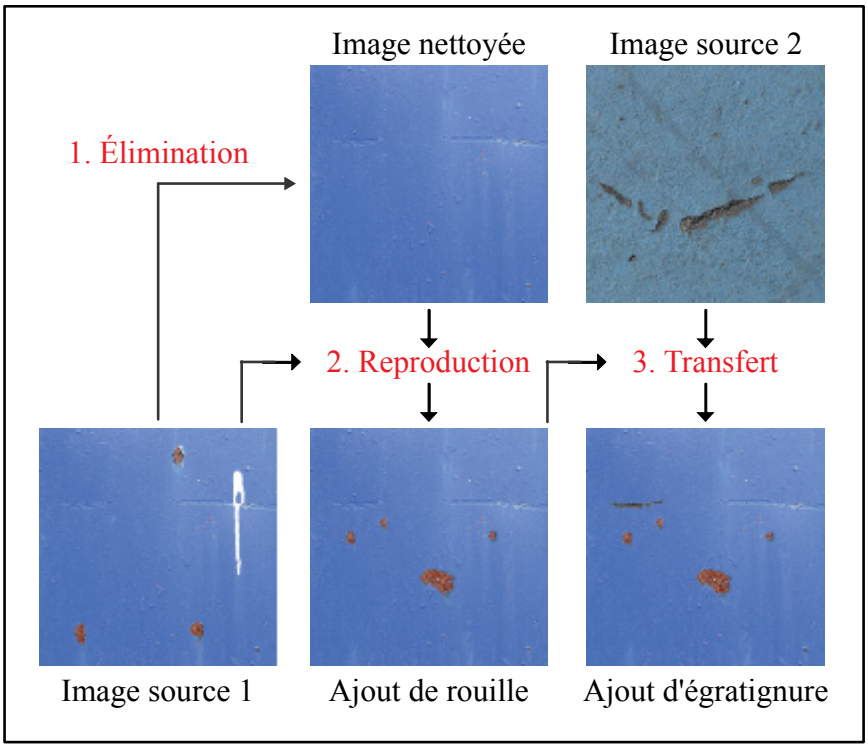


Figure 2.15 Combinaison et transfert d’effets de détérioration sur du métal.

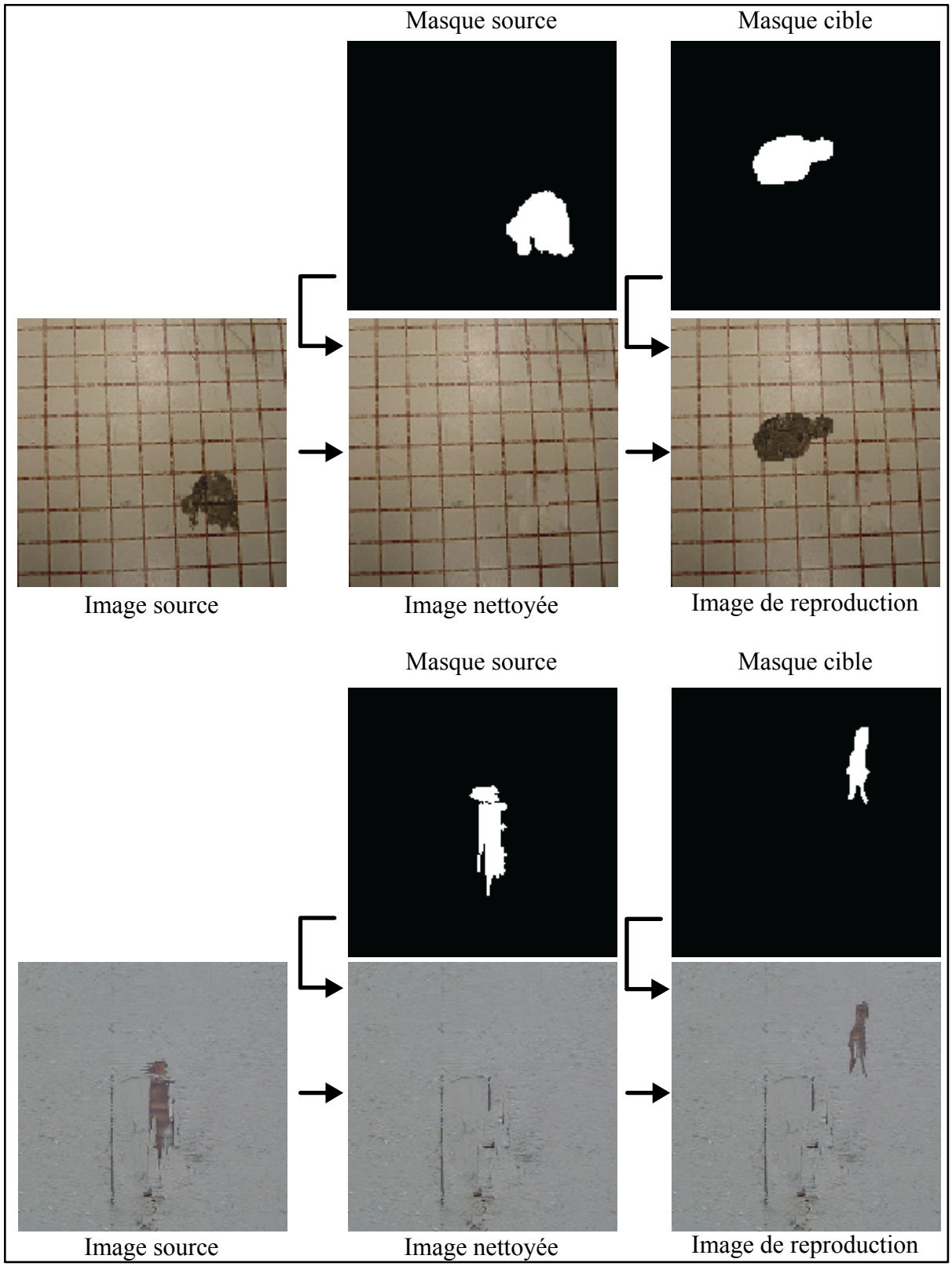


Figure 2.16 Résultats sur de la céramique et du bois.

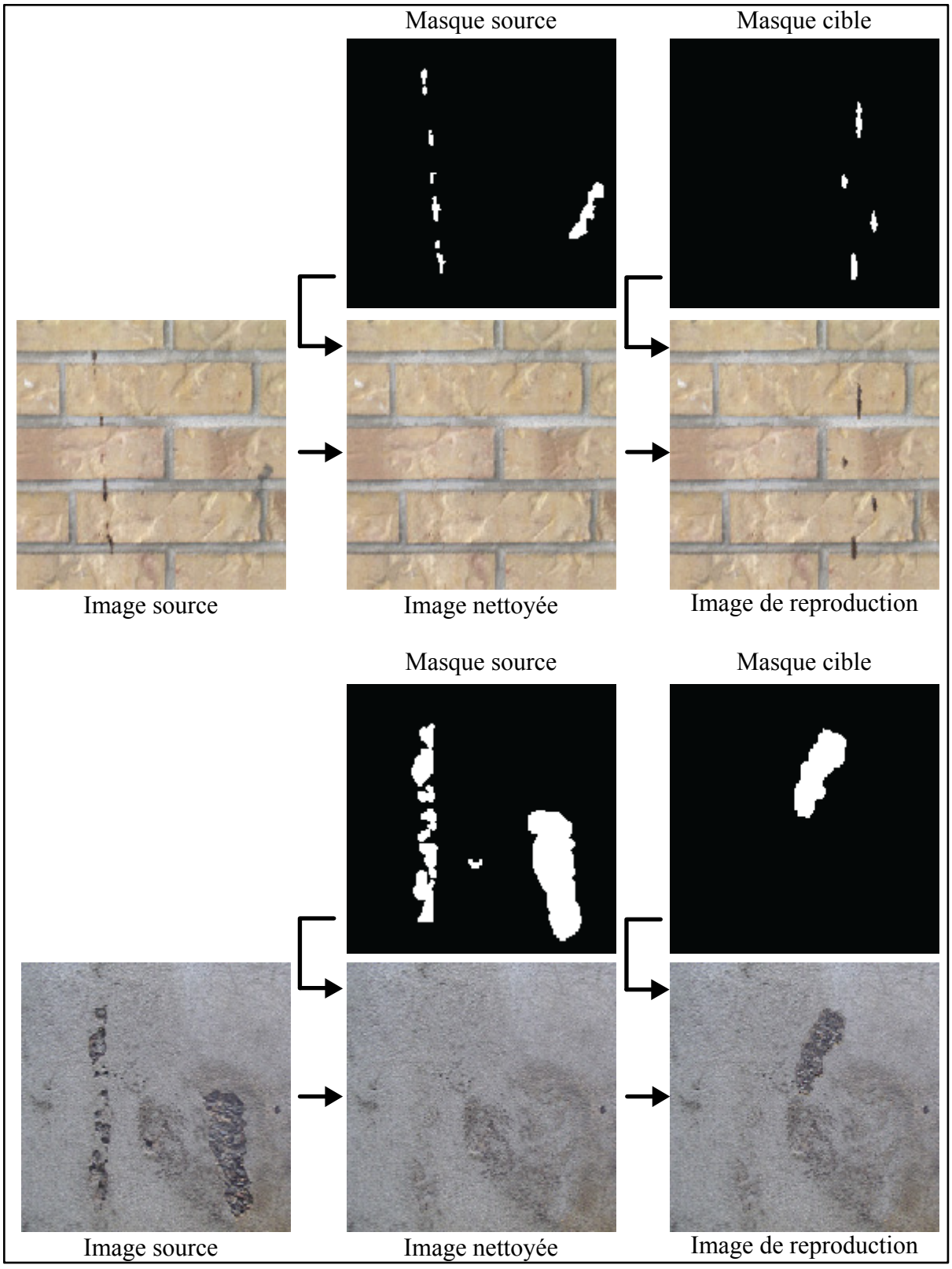


Figure 2.17 Résultats sur de la brique et du ciment.

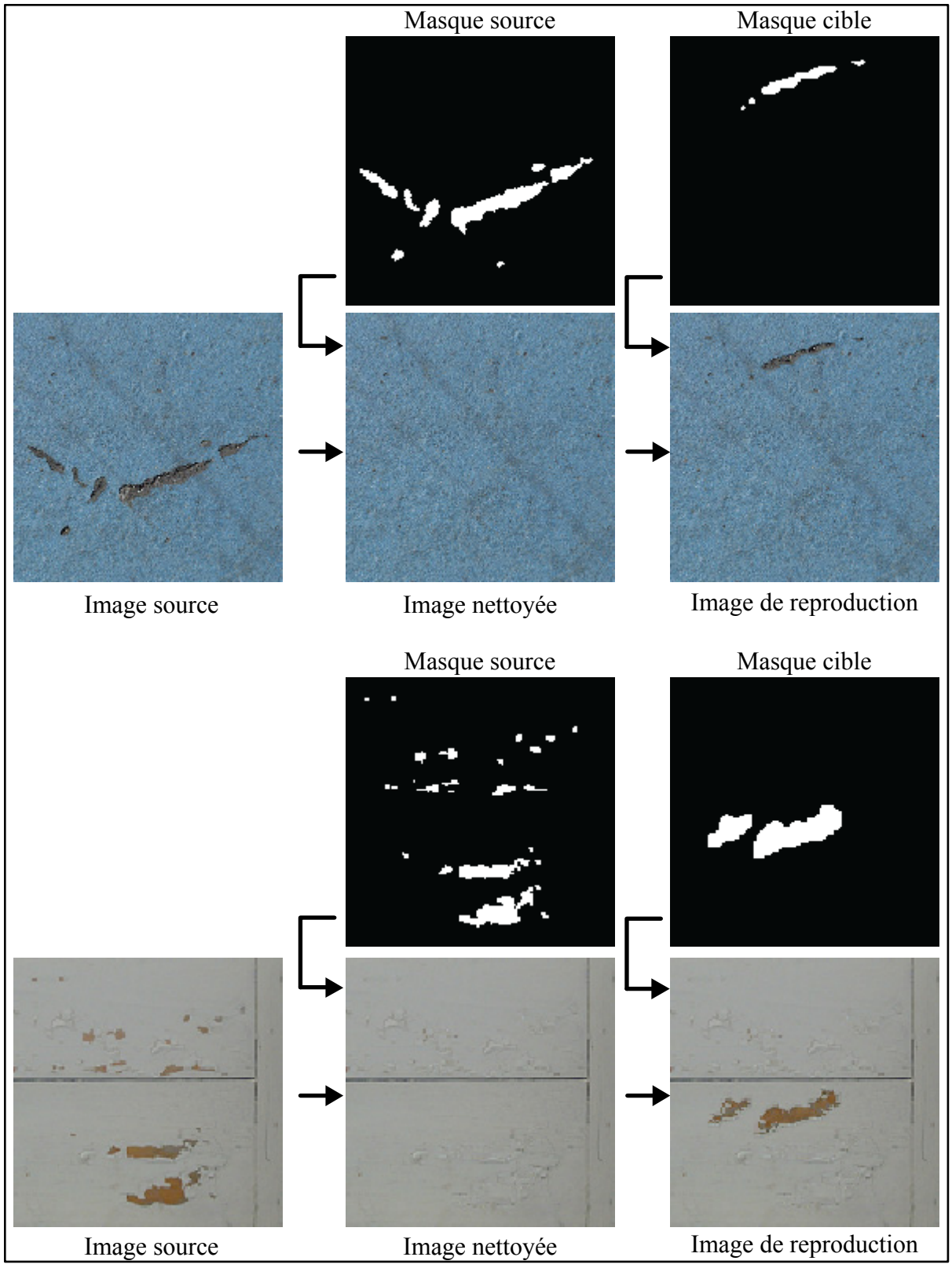


Figure 2.18 Résultats sur du ciment et du bois.

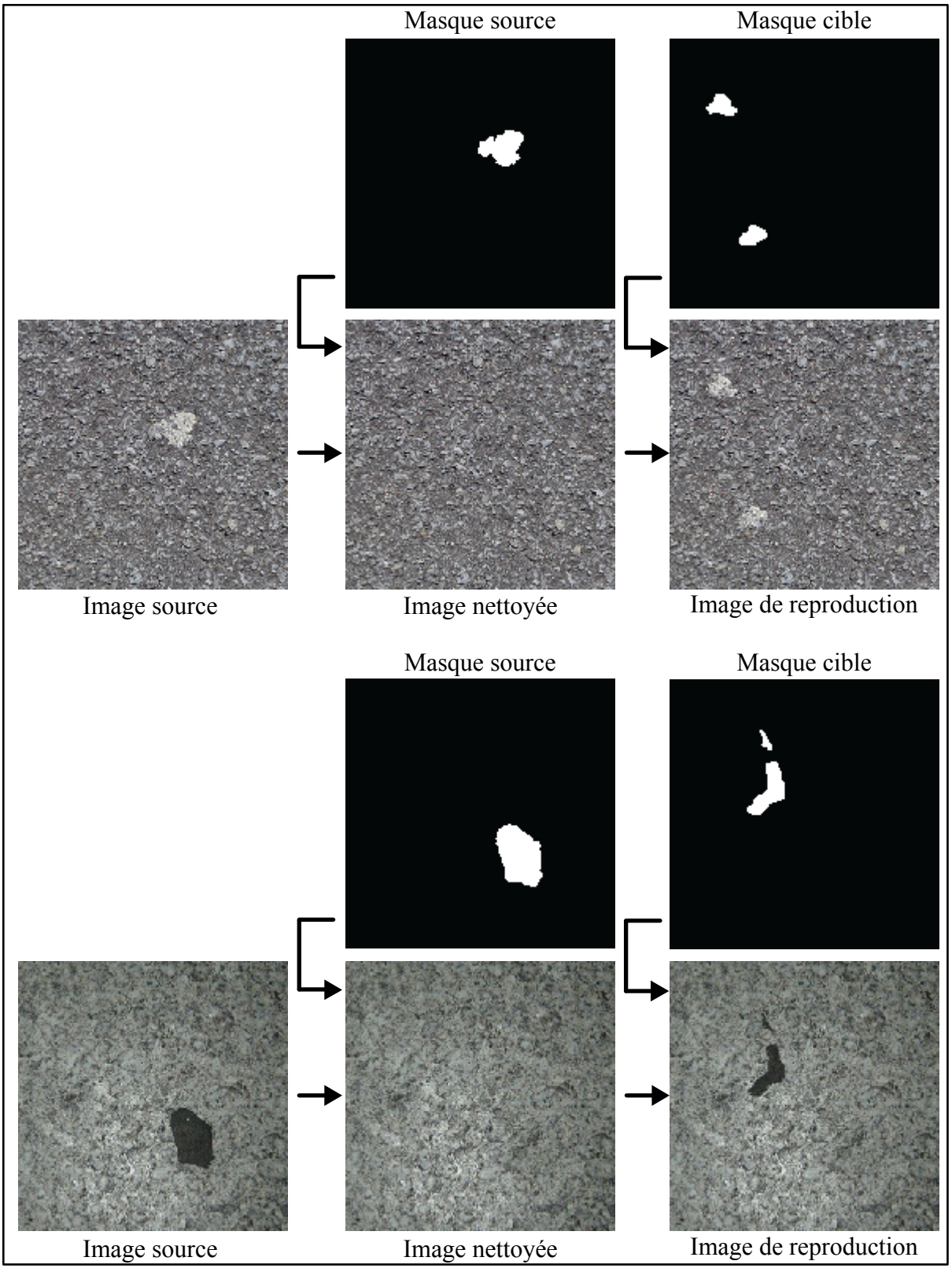


Figure 2.19 Résultats sur du gravier et du marbre.

Figure 2.20 Résultats de la synthèse avec différents patrons.

La figure 2.21 propose un sommaire des temps requis pour produire les résultats présentés dans ce chapitre. Les différentes barres de couleurs représentent le temps moyen pour chaque opération tandis que les lignes noires indiquent la dispersion des données recueillies. Le temps pendant lequel l'artiste doit interagir avec le système se calcule en minutes; de une à six minutes pour la création du masque source et de une à quatre minutes pour générer le masque cible. La figure 2.21 montre également que le temps requis au système d'édition pour la phase d'élimination se situe entre 4 et 65 secondes et que celui de la phase de reproduction est en bas d'une seconde. Les différents temps ont été obtenus sur un ordinateur possédant trois giga-octets de mémoire vive et un processeur de 3.2 gigahertz.

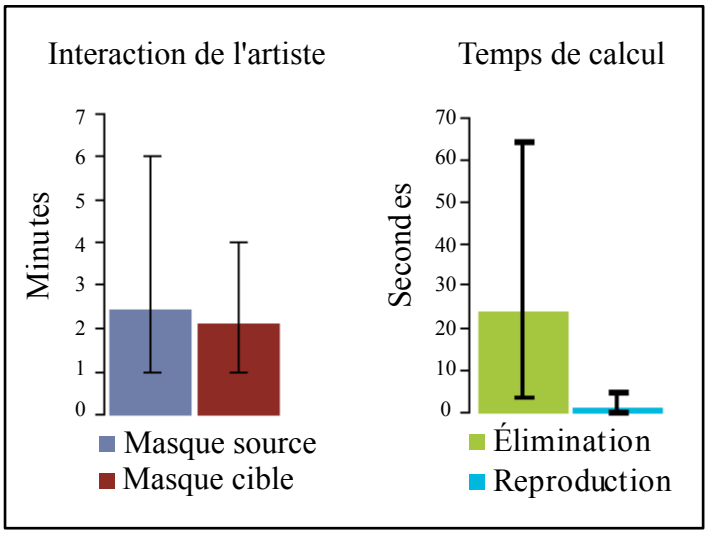


Figure 2.21 Temps requis pour l'obtention des résultats.

2.7 Discussion

Cette section fait l'analyse et l'interprétation des résultats obtenus à l'aide du système d'édition d'effets de détérioration proposé et discute de ses principales caractéristiques. La section 2.7.1 explique dans un premier temps les avantages du système proposé par rapport à l'état de l'art tandis que la section 2.7.2 discute des limitations et des travaux futurs. Rappelons que mes contributions personnelles à l'article de Clément, Benoit et Paquette (2007) présenté dans ce chapitre se sont concentrées principalement, mais pas uniquement, aux étapes d'élimination et de reproduction puisqu'elles traitent plus spécifiquement de remplissage de régions manquantes et de synthèse de texture.

Il faut aussi noter que, depuis la parution de l'article en 2007, d'autres auteurs ont repris le pipeline novateur introduit par l'approche présentée dans ce chapitre. D'ailleurs, le travail de Bosch *et al.* (2011) mentionne que celle-ci a fait la démonstration que ce nouveau pipeline fonctionnait. Bosch *et al.* (2011) ont d'ailleurs repris en partie ce pipeline en automatisant la génération du masque source. L'approche proposée s'y compare avantageusement puisqu'elle permet la simulation de plusieurs types d'effets, qu'elle permet la combinaison d'effets et qu'elle offre un contrôle sur l'apparence finale contrairement à Bosch *et al.*

(2011). De son côté, le travail de Bellini, Kleiman et Cohen-Or (2016) a réutilisé le même pipeline en automatisant la génération d'un ensemble de masques cibles contenant différents degrés de dégradation. Cette approche présente des résultats pour plusieurs types d'effets, offre un bon contrôle sur l'apparence finale et ne nécessitent pas de paramètres complexes. Elle n'est cependant pas en mesure de combiner plusieurs effets et permet uniquement le transfert de l'allure (forme) de l'effet, mais pas de son pattern (couleur).

2.7.1 Avantages

Le système d'édition d'effets de détérioration proposé possède plusieurs avantages par rapport à l'état de l'art qui le rend plus adapté à l'artiste et à son processus itératif de création. Premièrement, la technique proposée est très simple d'utilisation pour un artiste. Une grande majorité des techniques de détérioration (Chang et Shih, 2001; 2003; Dorsey *et al*., 1999; Dorsey et Hanrahan, 1996; Dorsey, Pedersen et Hanrahan, 1996; Paquette, Poulin et Drettakis, 2001; 2002; Wong, Ng et Heng, 1997) nécessitent que l'artiste manipule différents paramètres scientifiques étendus. Ces paramètres sont une barrière pour l'artiste qui préfère souvent laisser la technique automatique de côté et réaliser le travail manuellement. Or, le système proposé est très intuitif pour un artiste puisqu'il doit simplement fournir une image référence, identifier les effets de détérioration qu'elle contient et indiquer la localisation des nouveaux effets de détérioration. En somme, la technique proposée est plus intuitive et simple d'utilisation que l'état de l'art puisqu'elle ne nécessite pas la connaissance de paramètres scientifiques pointus.

Deuxièmement, les techniques actuelles ne permettent généralement pas de modifier l'effet de détérioration préalablement obtenu; il est uniquement possible de créer un nouvel effet totalement différent. Or, les artistes ont souvent à corriger leur travail en fonction des commentaires obtenus suite à son inspection (voir la figure 5). Quant à lui, le système proposé offre une grande flexibilité à l'artiste qui peut ainsi corriger et retravailler certaines portions précises de la texture de façon itérative. En effet, l'artiste peut modifier le masque cible en fonction des correctifs nécessaires et recommencer la phase de reproduction pour

obtenir rapidement un résultat corrigé. Tel qu'illustré sur la figure 2.21, le temps requis pour la phase de reproduction est en moyenne d'une seconde, ce qui permet à l'artiste d'apporter les corrections voulues de façon interactive.

Troisièmement, la majorité des techniques actuelles permettent uniquement de synthétiser un seul effet de détérioration. Par conséquent, l'artiste doit manipuler un système différent pour chaque effet de détérioration désiré diminuant ainsi son efficacité. Les résultats présentés à la section 2.6 démontrent que le système d'édition proposé permet de synthétiser une vaste gamme d'effets de détérioration différents. L'artiste doit simplement fournir une image de référence de l'effet d'usure voulu. De plus, la technique proposée ne nécessite pas un mécanisme complexe et coûteux de capture pour obtenir l'image de référence contrairement au travail de Gu *et al.* (2006). Le système d'édition ne contraint pas non plus l'artiste à fournir une image de référence contenant l'effet de détérioration à plusieurs stades de son évolution comme pour Wang *et al.* (2006) ou Bellini, Kleiman et Cohen-Or (2016).

Finalement, le système d'édition d'effets de détérioration proposé offre un contrôle adéquat sur les résultats obtenus. En effet, le choix d'un algorithme par remplissage de trou permet à l'artiste d'éditer une région spécifique de la texture sans avoir à la générer au complet. De plus, l'utilisation d'un masque cible pour positionner les effets d'usure offre un contrôle très précis sur leur localisation. Ceci est également un avantage comparativement aux approches de synthèse de texture (Efros et Leung, 1999; Hertzmann et al., 2001) qui génèrent l'ensemble de la texture. Tel qu'illustré à la section 2.3.2, la méthode proposée utilise quatre fenêtres de recherche selon le contexte détérioré et non détérioré ce qui permet de réduire considérablement le problème de discontinuité observé avec la technique ligne par ligne qu'utilise généralement les approches de synthèse de texture. Le tableau 2.1 synthétise la comparaison de l'approche proposée avec les travaux récents selon les différents critères présentés dans cette section.

Tableau 2.1 Comparaison de l'approche proposée avec l'état de l'art

Travaux	Plusieurs effets?	Transfert?	Combinaison?	Contrôle apparence finale?	Paramètres complexes?
Clément, Benoit et Paquette (2007)	Oui	Oui	Oui	Oui	Non
Glondu, Marchal et Dumont (2013)	Non	Oui	Non	Non	Oui
Bellini, Kleiman et Cohen-Or (2016)	Oui	Non	Non	Oui	Non
Iben et O'Brien (2009)	Non	Oui	Non	Non	Oui
Xue, Dorsey et Rushmeier (2011)	Non	Non	Non	Non	Oui
Bosch *et al.* (2011)	Non	Oui	Non	Non	Non
Mérillou et al. (2010)	Non	Non	Non	Non	Oui
Kider, Raja et Badler (2011)	Non	Non	Non	Non	Oui
Bézin *et al*. (2014)	Non	Non	Non	Non	Oui
Endo *et al.* (2010)	Non	Non	Non	Non	Oui

2.7.2 Limitations

Bien que le système d'édition de texture permette de synthétiser plusieurs types d'usure d'une façon réaliste à l'intérieur de courts délais, il demeure néanmoins quelques limitations liées à son utilisation. Premièrement, il faut se souvenir que la technique proposée travaille uniquement avec des textures 2D. Par conséquent, le processus de synthèse d'effets de détérioration traite des phénomènes d'usure qui modifient l'apparence d'une surface uniquement. Tous les phénomènes qui déforment ou fracturent la géométrie 3D d'un objet, comme le plastique qui fond au soleil et une bouteille de vitre qui se brise suite à un impact, ne sont pas considérés. Bref, le système d'édition traite uniquement des phénomènes qui changent l'apparence d'une surface et non sa géométrie.

Deuxièmement, l'approche actuelle divise la texture en régions et assigne à chacune d'elles un état détérioré ou non-détérioré. Cette division binaire a comme impact l'impossibilité de spécifier au système si une région est plus ou moins détériorée. Par conséquent, ceci implique que l'image de référence doit impérativement contenir l'effet d'usure au stade de dégradation désiré.

Finalement, l'artiste doit porter une attention particulière lorsqu'il sélectionne ou photographie son image de référence. En effet, comme pour les autres techniques d'acquisition de texture basée sur les photographies, l'artiste doit soigneusement prendre en considération les sources de lumière lorsqu'il prend ses photos de référence afin d'éviter la formation d'effets spéculaires qui fausseraient les résultats. De plus, l'artiste doit également faire attention de ne pas avoir de distorsion dans l'image de référence; l'utilisation de photographies de surfaces planes est généralement préférable.

CHAPITRE 3

REMPLISSAGE VIDÉO À L'AIDE D'UNE RECHERCHE LOCALE

Tel que mentionné à la section 1.5, le deuxième objectif de cette thèse est de concevoir une technique de remplissage automatique et efficace de régions manquantes dans une séquence vidéo adaptée aux artistes, aux studios de production et aux pipelines de production[2]. Pour atteindre cet objectif, la technique proposée doit nécessairement être simple d'utilisation pour l'artiste, traiter aussi bien les séquences vidéo réelles capturées à l'aide d'une caméra que celles synthétiques créées par ordinateur, traiter les séquences vidéo de haute résolution et s'exécuter dans un délai assez court afin que la technique s'intègre bien dans le pipeline de production. Comme au chapitre 2, une approche de synthèse de texture basée sur les champs aléatoires de Markov sera dérivée, mais sera cette fois-ci appliquée au domaine de la retouche de séquences vidéo en portant une attention particulière aux fenêtres de recherche, à l'ordre de remplissage et à l'optimisation des temps de recherche. Afin de pouvoir compléter des séquences vidéo HD, une méthode novatrice de complétion basée sur une recherche locale et sur le principe de la cohérence sera détaillée. Ce chapitre présente l'approche proposée en détaillant étape par étape son fonctionnement, montre des exemples de résultats et discute des avantages et des limitations de la méthode proposée par rapport à l'état de l'art.

3.1 Présentation générale de l'approche proposée

L'objectif principal du système de retouches de séquences vidéo proposé est d'offrir à l'artiste une technique d'édition permettant de faire la suppression d'objets indésirables tels qu'un micro ou une perche de son. Une fois ces objets supprimés et remplacés par du contenu perceptiblement plausible, l'artiste est libre d'appliquer d'autres traitements à la séquence vidéo afin d'ajouter différents effets visuels. Pour lui, la suppression d'objets ou de

---

[2] Le contenu du chapitre 3 a été publié dans une revue scientifique internationale : Benoit et Paquette (2015).

régions indésirables contenus dans une séquence vidéo est une tâche répétitive et machinale qui ne requière que peu de talent artistique. Il serait donc préférable pour les studios de production d'automatiser le plus possible cette tâche permettant ainsi aux artistes d'être affectés à d'autres travaux plus créatifs. Idéalement, l'artiste n'aurait qu'à fournir la séquence vidéo originale et à identifier l'objet indésirable pour que le système d'édition soit en mesure de la corriger.

Comme le montre la figure 3.1, le système proposé se base sur cette démarche intuitive pour établir les étapes clés que l'artiste doit suivre lors de son utilisation. Premièrement, l'artiste fournit une séquence vidéo source, généralement une séquence vidéo réelle captée avec une caméra (bien qu'une séquence synthétique créée par ordinateur puisse également être utilisée) dans laquelle se trouve un objet ou une région indésirable qui nécessite d'être enlevée. Deuxièmement, il fournit au système une autre séquence vidéo, le masque source, qui indique les endroits où se trouve l'objet indésirable dans la séquence vidéo source.

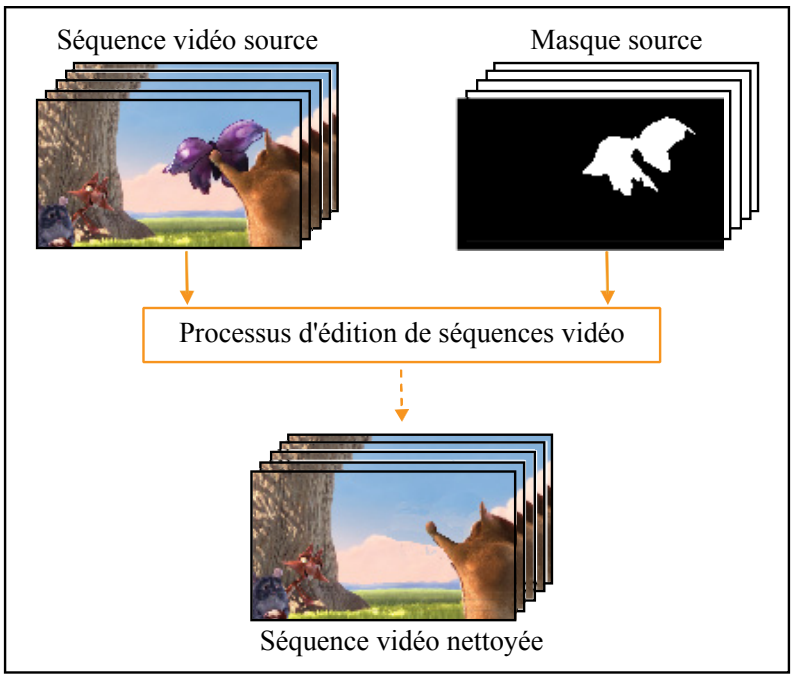


Figure 3.1 Système de remplissage basé sur une recherche locale.

Le masque source est une séquence vidéo binaire dans laquelle les pixels blancs représentent l'objet indésirable et les pixels noirs indiquent les régions à conserver. Ce masque est créé à

l'aide d'un logiciel externe de traitement vidéo tel que *Adobe After Effects*. À partir de la séquence vidéo source et du masque source, le processus d'édition produit la séquence nettoyée, c'est-à-dire la séquence vidéo source dans laquelle les régions indésirables identifiées par le masque source ont été supprimées et remplacées par du contenu perceptuellement plausible.

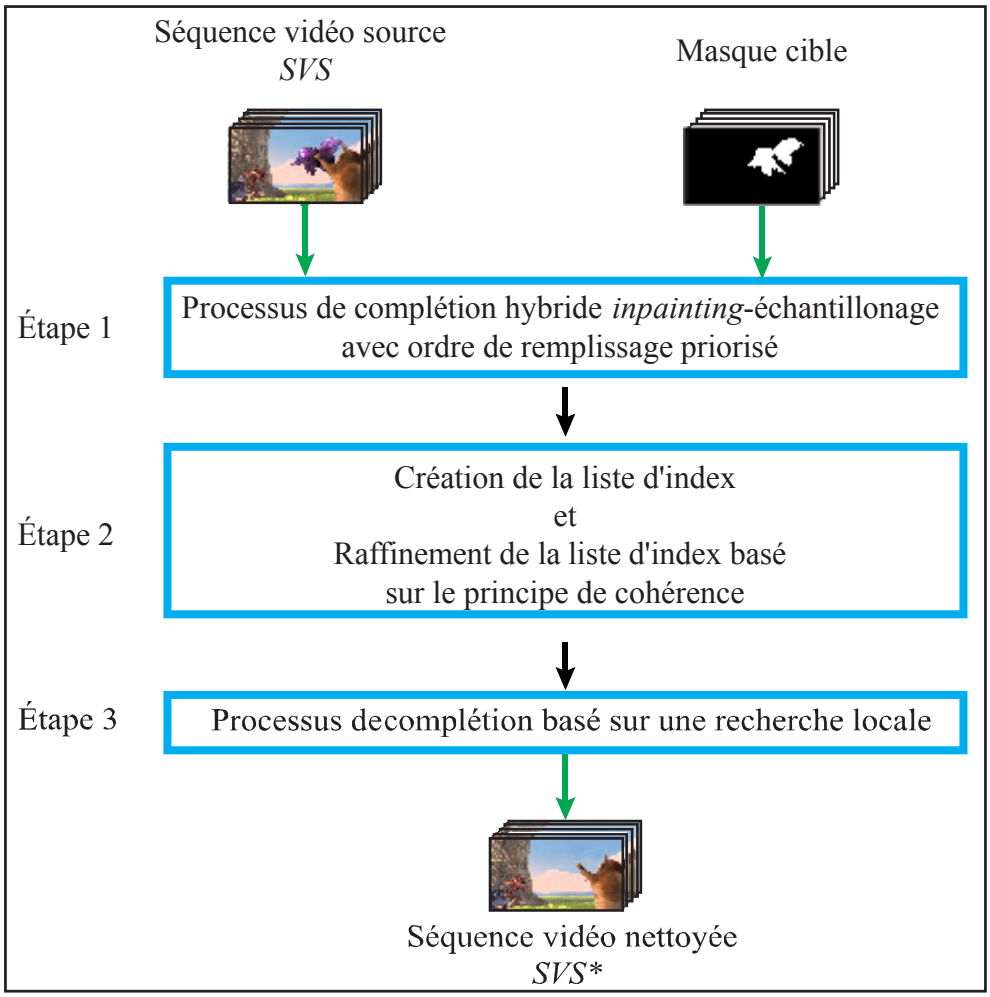


Figure 3.2 Aperçu schématique de la méthode proposée.

Tel que présenté à la figure 3.2, la séquence vidéo est tout d'abord traitée avec la méthode de complétion hybride *inpainting*-échantillonnage (voir section 3.2). Le résultat de cette étape est la séquence vidéo complétée au niveau de résolution fixe de 480 x 270. L'étape suivante de l'approche proposée consiste à la création d'une liste d'index et de son raffinement à

l'aide d'une technique basée sur le principe de la cohérence qui permet d'améliorer les résultats de recherche pour les correspondances possédant les mesures de distances les plus élevées (voir section 3.3.1). Cette étape est très rapide et permet d'améliorer de façon significative la qualité des résultats obtenus.

Finalement, la liste d'index est utilisée par le processus itératif de remplissage de séquence vidéo de haute définition pour restreindre l'espace de recherche au niveau de résolution le plus fin, permettant ainsi la complétion de séquence de haute définition à l'aide d'une recherche locale. Cette étape finale est également rapide et permet d'obtenir de bons résultats.

3.2 Processus de complétion hybride *inpainting*-échantillonnage

Le processus de complétion hybride *inpainting*-échantillonnage (voir figure 3.3) consiste à appliquer une technique itérative de remplissage successivement à différentes échelles (résolutions) de la séquence vidéo source en utilisant une pyramide spatio-temporelle. Chaque niveau de la pyramide contient le quart de la résolution spatiale du niveau supérieur et, contrairement au travail de Wexler, Shechtman et Irani (2007), maintient la résolution temporelle. Le processus de complétion débute le traitement au niveau de résolution le plus grossier de façon à recréer les plus grandes structures. Les résultats obtenus sont ensuite propagés à un niveau de résolution plus fin et ce cycle se répète jusqu'à un niveau fixe de résolution de 480 x 270 (voir section 3.2.3). La propagation des résultats vers le niveau supérieur permet d'accélérer la convergence vers la solution optimale pour ce dernier. L'algorithme de cette approche multi-résolution est illustré sur la figure 3.4.

Différentes méthodes (Wexler, Shechtman et Irani, 2004; 2007; Xiao *et al*., 2011; Xiao *et al*., 2008) font l'utilisation d'une pyramide spatio-temporelle lors du remplissage. Ceci permet de recréer aussi bien les grandes structures contenues dans la séquence source que les détails plus subtiles retrouvés aux niveaux de résolution plus fins. L'avantage principal de l'utilisation d'une pyramide spatio-temporelle réside dans le fait que la taille du voisinage

utilisée lors de la recherche du meilleur candidat reste fixe et relativement petite pour tous les niveaux. Le processus requiert donc moins de mémoire et est plus rapide aux niveaux de résolution les plus fins comparativement aux méthodes qui n'utilisent pas une technique multi-résolutions. En effet, ces dernières sont obligées d'avoir recours à une taille de voisinage beaucoup plus grande pour recréer les plus larges structures contenues dans la séquence vidéo source.

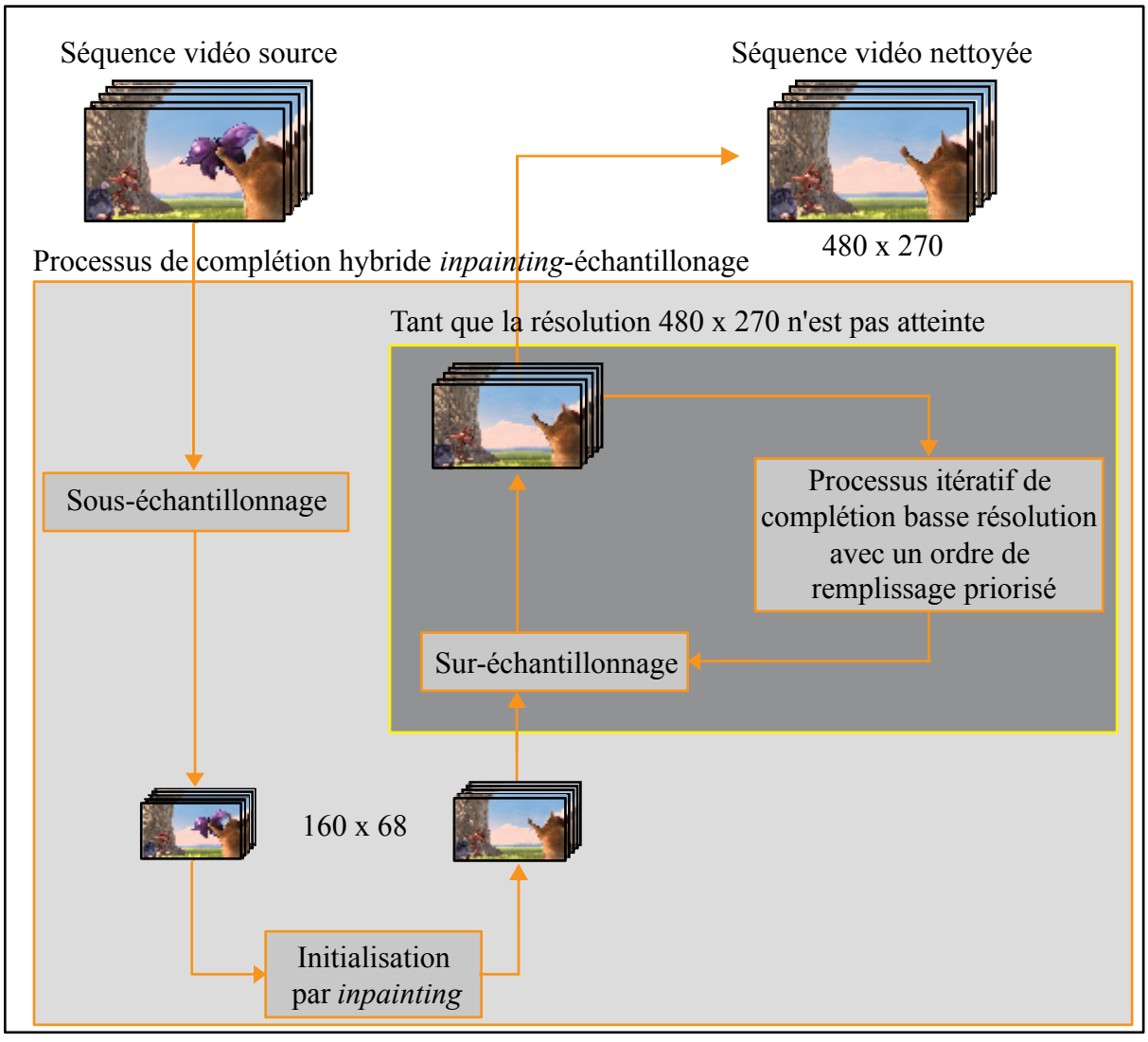


Figure 3.3 Processus de complétion hybride *inpainting*-échantillonnage.

De plus, il est utile de souligner que puisque chaque niveau de résolution contient le quart des pixels, autant pour les pixels indésirables qui doivent être remplacés que pour les pixels à conserver qui sont des candidats potentiels, le coût associé à la création et à l'utilisation d'une pyramide spatio-temporelle reste négligeable comparativement à celui lié à l'augmentation de la taille du voisinage au niveau le plus fin. D'autre part, puisque la taille du voisinage est fixe, l'artiste n'est pas dans l'obligation d'analyser la séquence vidéo source afin de déterminer cette valeur rendant ainsi le processus d'édition plus simple d'utilisation. Pour ces raisons, le processus de complétion hybride *inpainting*-échantillonage se base sur une approche qui fait l'utilisation d'une pyramide spatio-temporelle.

Entrées : Séquence vidéo source *SVS*, masque source *RR*

Nombre de niveaux ⟵ obtenir nombre de niveaux (*SVS*)
*SVS** ⟵ sous-échantillonner (*SVS*, nombre de niveaux)
*SVS** ⟵ initialiser par *inpainting* (*SVS**, *RR*)

résolution ⟵ 160 x 68
Répéter
    i ⟵ 0
    Répéter
        Pour chaque *pixel p identifié par* RR **faire**
            p' ⟵ trouver meilleur candidat (*p*, *SVS**, *RR*)
            *SVS**[*p*] ⟵ *SVS**[*p'*]
        Fin
        i ⟵ i + 1;
    Jusqu'à *ce que i = nombre_itérations_max*
    résolution ⟵ doubler (*résolution*)
    *SVS** ⟵ progager résultats au niveau supérieur (*résolution*, *RR*, *SVS**)
Jusqu'à ce que *résolution* = 480 x 270

Sortie : *SVS** (basse résolution)

Figure 3.4 Pseudo-code de l'algorithme pour la complétion hybride.

Les étapes de sous-échantillonnage, d'initialisation des régions manquantes avec une technique d'*inpainting* et de complétion par un processus itératif avec ordre de remplissage

priorisé illustrées sur les figures 3.3 et 3.4 sont détaillées dans les sections 3.2.1 à 3.2.3. Le tableau 3.1 présente un sommaire des symboles utilisés dans ce chapitre.

Tableau 3.1 Définition des symboles

Symbole	Définition
$\boldsymbol{SVS}$	Séquence vidéo source
$\boldsymbol{RR}$	Région à remplacer ($RR \subset SVS$)
$\boldsymbol{RV}$	Région valide ($SVS \setminus RR$)
$\boldsymbol{RR}^*$	Région corrigée
$\boldsymbol{SVS}^*$	Séquence vidéo corrigée
$\boldsymbol{p}$	Point spatio-temporel situé à *(x, y, t)*
$\boldsymbol{w_p}$	Cube spatio-temporel centré à p
$\boldsymbol{w_{p'}}$	Cube spatio-temporel centré à p', pièce la plus similaire à w_p
$\boldsymbol{c, c'}$	Couleurs RVB de p et p'
$\boldsymbol{\Phi(w_p^g)}$	Cube spatio-temporel au niveau de résolution le plus fin correspondant au cube spatio-temporel w_p^g à un niveau de résolution plus grossier

3.2.1 Sous-échantillonnage

Tel qu'illustré sur les figures 3.3 et 3.4, la première étape consiste à sous-échantillonner la séquence vidéo source afin d'obtenir une plus petite résolution spatiale. Pour chaque niveau, la résolution spatiale diminue de moitié autant en largeur qu'en hauteur tandis que la résolution temporelle demeure la même. Pour y parvenir, une pyramide de gaussiennes telle que décrite par le travail de Burt et Adelson (1983) est construite à partir de la séquence vidéo source. Il est également nécessaire de sous-échantillonner le masque source de façon à connaître les pixels indésirables pour les niveaux de résolution plus grossiers. Chaque pixel d'un niveau grossier de résolution est associé à une région de 2 x 2 pixels de la résolution

plus fine. Dès qu'un de ces quatre pixels est identifié par le masque source, le pixel correspondant au niveau grossier est étiqueté comme indésirable. Ce processus est exécuté de façon itérative entre les niveaux de résolution jusqu'à ce que la résolution la plus grossière soit atteinte. Un exemple de sous-échantillonnage du masque source est présenté à la figure 3.5.

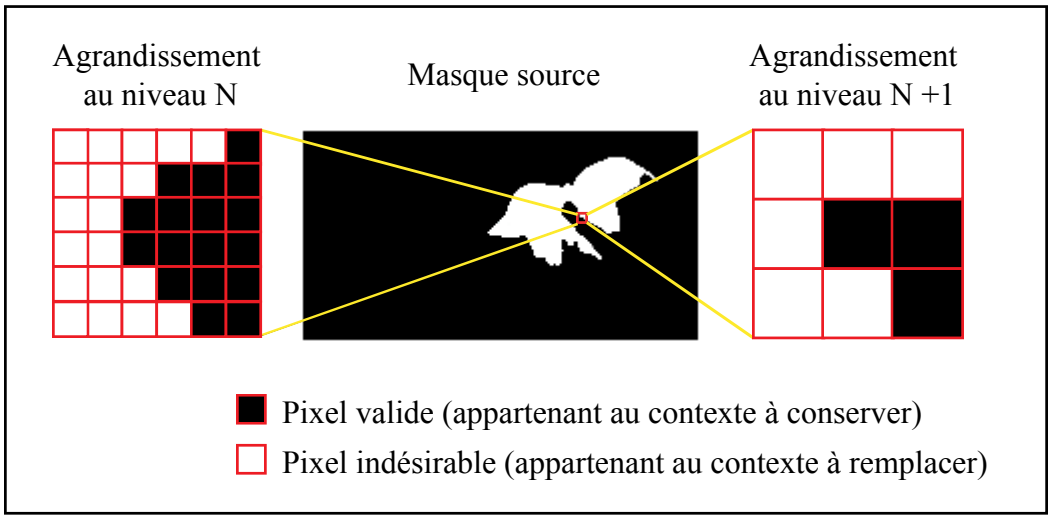


Figure 3.5 Exemple de sous-échantillonnage du masque source.

3.2.2 Initialisation des régions manquantes par *inpainting*

Suite à l'étape de sous-échantillonnage, le processus d'édition de séquence vidéo procède à l'initialisation des couleurs de chaque pixel identifié par le masque source comme indésirable. Contrairement à la technique de Wexler, Shechtman et Irani *(2007)* qui attribue une couleur aléatoire à chaque pixel, le processus d'édition proposé utilise quant à lui une technique de *image inpainting* (Bertalmio et al., 2000). Tel qu'expliqué à la section 1.3, la technique consiste à propager itérativement l'information sur la couleur des pixels à conserver autour de la région vers les pixels de la région indésirable.

Bien qu'un traitement par remplissage de trou *image par image* n'offre pas de bons résultats pour une séquence vidéo puisqu'il ne tient pas compte de l'ensemble de la séquence, cette initialisation *image par image* permet néanmoins d'accélérer la convergence par rapport à

une initialisation aléatoire telle qu'utilisée par Wexler, Shechtman et Irani *(2007)*. La figure 3.6 montre les résultats de l'initialisation des régions à remplacer selon la technique utilisée par Wexler, Shechtman et Irani *(2007)* et ceux de la technique proposée. La figure 3.6 met également en évidence que la technique d'initialisation proposée permet au processus itératif de remplissage de converger à un bon résultat beaucoup plus rapidement puisqu'il ne faut que cinq itérations pour arriver à un résultat similaire obtenu en 15 itérations lorsqu'une initialisation aléatoire est appliquée. De plus, puisque cette initialisation n'est réalisée qu'une seule fois et ce uniquement au niveau de résolution le plus grossier (120 pixels par 68 pixels sur la figure 3.6), le temps supplémentaire requis est négligeable.

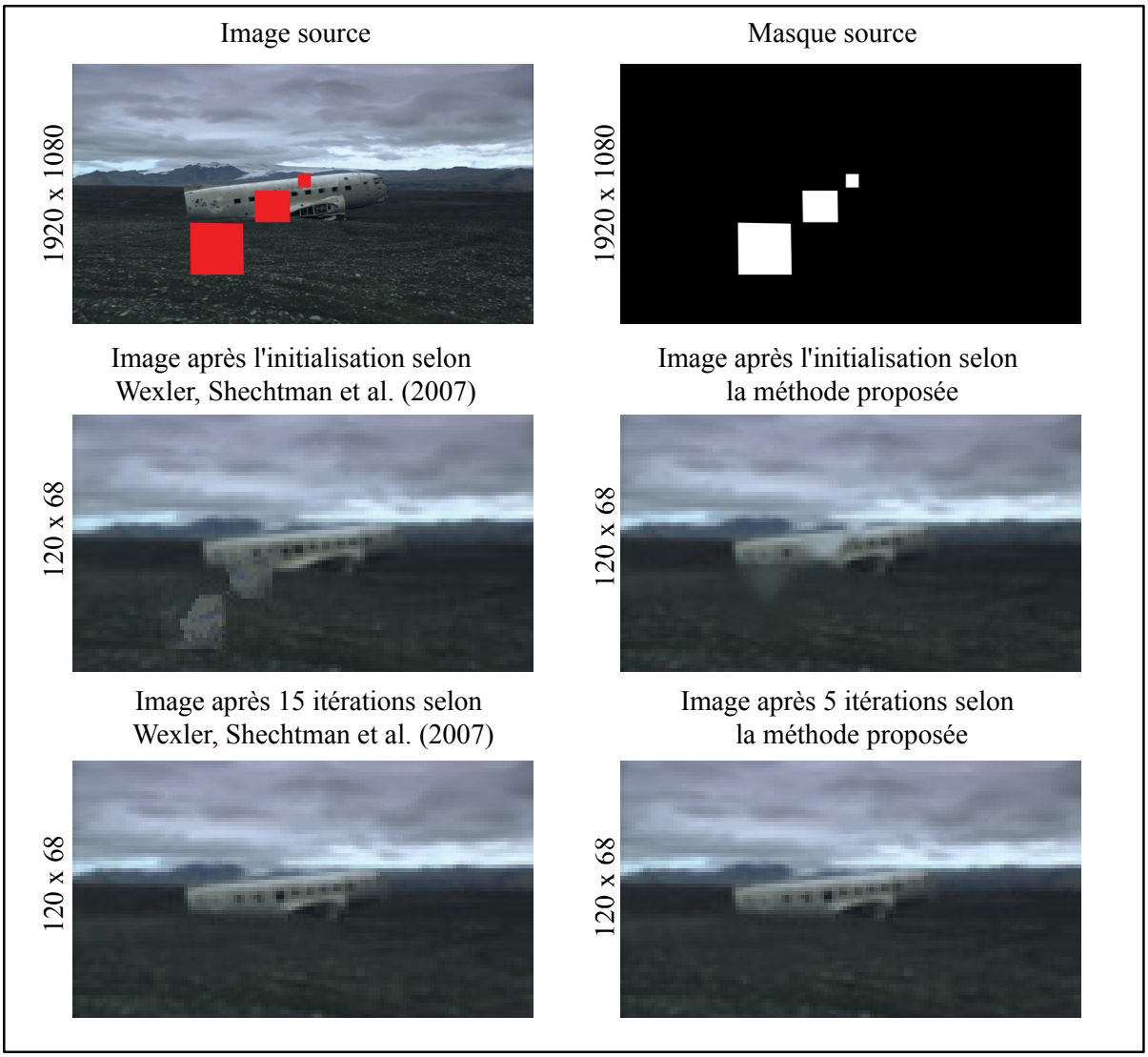


Figure 3.6 Comparaison avec l'initialisation de Wexler, Shechtman et Irani.

3.2.3 Complétion basse résolution avec ordre de remplissage priorisé

À la suite de l'étape d'initialisation de la zone à remplacer, la méthode proposée commence le processus itératif de remplissage de la séquence vidéo au niveau de résolution le plus grossier (voir figure 3.3). Ce processus suit des étapes similaires à celles des autres approches basées sur des cubes spatio-temporels. Ces étapes seront explicitées dans les prochains paragraphes, jusqu'à l'équation (3.1).

À partir de la séquence vidéo source SVS qui contient une région à remplacer RR ($RR \subset SVS$), le processus itératif de remplissage remplit la région RR de manière à ce que le résultat soit visuellement acceptable, en copiant des pièces similaires trouvées dans la région valide RV de la séquence vidéo ($RV = SVS \setminus RR$). Ce processus produit une séquence vidéo complétée SVS^*. De façon à maintenir la cohérence spatio-temporelle, il est important de ne pas considérer chaque image de la séquence vidéo de manière indépendante. Par conséquent, l'algorithme considère la séquence vidéo source comme un volume spatio-temporel où un pixel positionné aux coordonnées *(x,y)* de l'image *t* est représenté par un point spatio-temporel p = *(x, y, t)*. Dès lors, une pièce w_p peut être perçue comme un cube spatio-temporel de pixels centré au point *p*.

La complétion visuellement acceptable d'une séquence vidéo remplace la région à remplacer RR par une région complétée RR^* où tous les pixels de RR^* s'agencent bien dans la séquence vidéo SVS^*. Pour y parvenir, le processus itératif de remplissage doit satisfaire deux critères généralement définis par les techniques d'échantillonnage non-paramétrique :

1. Toutes les pièces spatio-temporelles locales de la région complétée RR^* doivent être similaires à au moins une pièce existante faisant partie de RV.
2. Toutes les pièces spatio-temporelles locales de la région complétée RR^* doivent être cohérentes entre elles.

Ainsi on peut dire que l'algorithme recherche une séquence vidéo complétée SVS^* qui minimise la fonction objective présentée à l'équation (3.1) :

$$cohérence(RR^*|RV) = \prod_{p \in RR} \min_{p' \in RV} D(w_p, w_{p'}) \tag{3.1}$$

où $D(w_p, w_{p'})$ est une métrique de similarité entre deux pièces.

Dans notre approche, la mesure de similarité entre deux pièces est évaluée à l'aide de la sommation des distances carrées (SSD) de l'information sur la couleur dans le domaine

rouge-vert-bleu (RVB) de toutes les paires de points spatio-temporels contenues dans ces pièces. De leur côté, Wexler, Shechtman et Irani (2007) ajoutent aux composantes RVB de l'information sur les dérivées spatiale et temporelle afin d'obtenir une représentation en cinq dimensions pour chaque point spatio-temporel. Selon les expérimentations effectuées, les composantes RVB suffisent à elles seules pour obtenir de bons résultats pour la plupart des séquences vidéo testées. Des problèmes surviennent lorsque l'algorithme doit reconstruire un objet en mouvement qui est partiellement ou totalement caché par la région à remplacer. Bien que la technique de Wexler, Shechtman et Irani (2007) peut parfois régler ces problèmes, elle est cependant limitée à des séquences vidéo où l'objet en mouvement qui doit être reconstruit possède un mouvement cyclique (par exemple une personne avec une démarche constante). De plus, l'algorithme présenté par Wexler, Shechtman et Irani (2007) requière plus d'espace mémoire et de temps de calcul. Pour ces raisons, la portée du processus de remplissage itératif présenté dans cette section se limite à des séquences vidéo qui ne contiennent pas d'objet en mouvement partiellement ou totalement caché par la région à remplacer et la mesure de similarité entre deux pièces tient uniquement compte des composantes RVB.

Après l'étape d'initialisation, l'algorithme applique un processus itératif de remplissage qui cherche à améliorer la cohérence globale de la région à remplacer RR. À chaque itération, l'algorithme cherche une couleur de remplacement pour tous les points spatio-temporels contenus dans RR de façon à minimiser l'équation (3.1). Contrairement aux autres techniques de remplissage de régions manquantes dans une séquence vidéo qui utilisent généralement une approche ligne par ligne (*scanline*), la technique proposée remplit la région RR en appliquant une approche de remplissage par trou en trois dimensions permettant ainsi de maximiser le nombre de points spatio-temporels de la pièce w_p qui appartiennent à la région valide RV ou qui ont déjà été traités dans l'itération courante. Cela a pour effet d'accélérer la convergence vers une solution visuellement acceptable et de diminuer les discontinuités visibles aux frontières de la région à remplacer RR.

Afin de trouver une couleur de remplacement c pour un point spatio-temporel p, l'algorithme de remplissage utilise uniquement la meilleure correspondance, c'est-à-dire la pièce $w_{p'}$ qui

minimise la métrique de similarité $D(w_p, w_{p'})$. Lorsque l'algorithme a déterminé la pièce $w_{p'}$, il copie la couleur c' du point spatio-temporel p' vers p. Les travaux antérieurs (Wexler, Shechtman et Irani, 2004; 2007; Xiao *et al*., 2011; Xiao *et al*., 2008) qui considèrent un ensemble des meilleures correspondances, et qui en font la moyenne afin de déterminer c', montrent des régions complétées RR^* qui sont floues. L'approche proposée, qui tient compte uniquement de la meilleure correspondance, ne montre pas des régions complétées RR^* qui sont floues et préserve le grain du film et le bruit contenu dans la séquence valide RV.

Le processus itératif de remplissage proposé dans cette section produit des résultats ayant une qualité équivalente à ceux obtenus par Wexler, Shechtman et Irani (2007), mais avec des temps de calcul beaucoup plus courts. La figure 3.7 montre les résultats de la complétion de la séquence vidéo « Jogging Lady » obtenus par Wexler, Shechtman et Irani (2007) et par la méthode proposée. Ces derniers mentionnent que le temps de calcul nécessaire pour chaque itération était d'environ une heure au niveau de résolution le plus fin tandis que la méthode proposée a eu besoin de moins de quatre minutes par itération.

Pour s'assurer d'obtenir une cohérence spatio-temporelle, ce processus itératif de remplissage est successivement appliqué à différentes résolutions de la séquence vidéo source (voir les figures 3.3 et 3.4). Ce processus itératif de remplissage débute avec le niveau de résolution le plus grossier (120 x 68) et propage les résultats vers les niveaux les plus fins. Puisque l'utilisation de cette méthode à des niveaux élevés de résolution (ex. 1920 x 1080) demande un trop large espace mémoire pour la création des structures de recherche telle que ANN (Arya et al., 1998) et un temps considérable pour les recherches des meilleures correspondances, le processus de complétion hybride *inpainting*-échantillonage ne peut pas être utilisé et il s'arrête à un niveau fixe de résolution (480 x 270). La suite du remplissage pour les niveaux de résolution plus élevés est réalisée avec une approche novatrice et différente des travaux antérieurs. Cette approche est présentée dans la prochaine section.

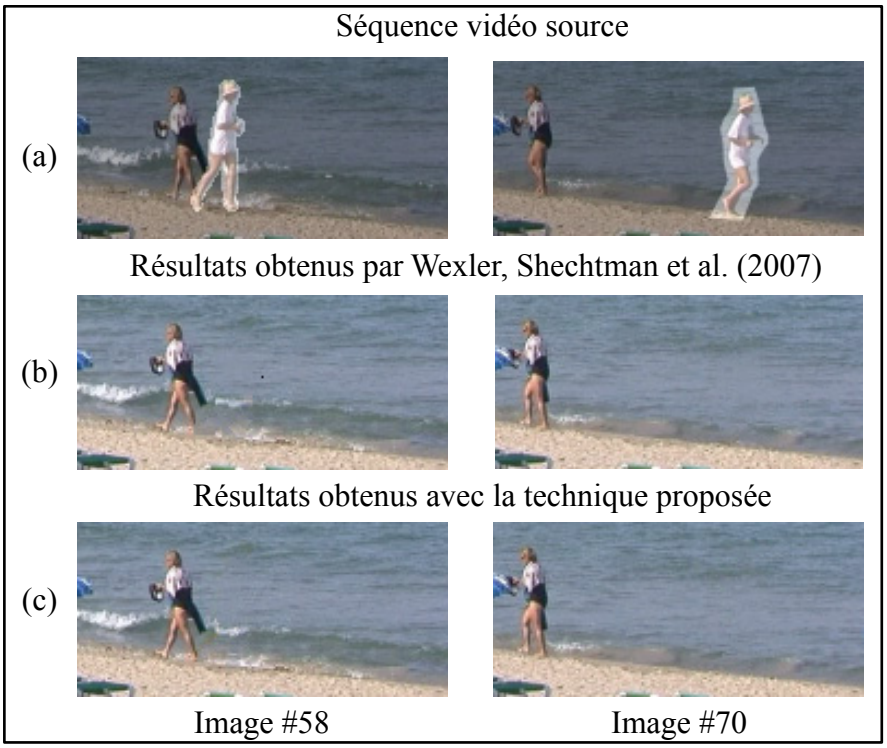


Figure 3.7 Comparaison avec les résultats
de Wexler, Shechtman et Irani.
Séquence vidéo « Jogging Lady ». (a) : Séquence vidéo source;
(b) : Résultats obtenus par Wexler, Shechtman et Irani (2007) et
(c) Résultats obtenus par la méthode proposée

3.3 Processus de complétion de haute résolution basé sur une recherche locale

La section précédente détaille une méthode de complétion hybride *inpainting*-échantillonage qui permet d'accélérer le temps nécessaire pour remplir des régions manquantes dans une séquence vidéo. Cette méthode possède cependant la même limitation que les autres méthodes basées sur un échantillonnage non-paramétrique : le temps de recherche et l'espace mémoire requis pour les structures de recherche limitent son utilisation à des séquences vidéo ayant des résolutions relativement basses. La section 3.3 présente une méthode novatrice de complétion vidéo basée sur une recherche locale qui élimine cette limitation rendant ainsi possible la complétion de séquences vidéo HD.

Afin de bien comprendre la méthode proposée pour le remplissage de régions à corriger à l'intérieur de séquences vidéo de haute définition, il faut tout d'abord se pencher sur les approches d'échantillonnage non-paramétrique et leurs limitations. Ces approches traitent itérativement toutes les pièces manquantes d'une région à remplacer et cherchent dans l'ensemble de toutes les pièces valides de la séquence source les pièces les plus similaires, c'est-à-dire les meilleures correspondances. Sans optimisation, cette recherche peut requérir beaucoup de temps : $O(m^3M^2F)$ où M représente la hauteur et la largeur de la séquence vidéo source, m la hauteur, la largeur et la profondeur de la pièce et F le nombre d'images dans la séquence vidéo source. Même avec les méthodes d'optimisation utilisées par l'état de l'art, le temps de recherche demeure tout de même un problème. De plus, les structures nécessaires aux méthodes d'optimisation nécessitent un large espace mémoire lorsque les séquences vidéo à traiter sont d'une trop grande résolution rendant ainsi inutilisables les techniques de remplissage basées sur une approche d'échantillonnage non-paramétrique dans un contexte réel de production.

Plutôt que de tenter d'accélérer les méthodes de recherche du meilleur candidat, l'approche proposée restreint plutôt l'espace de recherche aux niveaux de résolution plus fins en se basant sur l'information obtenue lors de la complétion des niveaux de résolution plus grossiers. Cet élagage de l'espace de recherche se base sur l'observation présentée à la figure 3.8. Considérons deux pièces à un niveau de résolution grossier : la pièce w_p^g qui appartient à la région à remplacer RR et la pièce $w_{p'}^g$ qui provient de la région valide RV et qui est la pièce la plus similaire à w_p^g. À un niveau de résolution plus fin, la pièce la plus similaire à la pièce correspondante w_p^f a de bonnes chances d'être trouvée près de la pièce $\Phi(w_{p'}^g)$. L'opérateur Φ prend un élément, une pièce ou un point, à un niveau de résolution plus grossier et retourne l'élément correspondant au niveau de résolution le plus fin. Par conséquent, il est possible de réutiliser l'information des recherches des pièces similaires aux niveaux de résolution grossiers afin de ne pas avoir à faire les recherches aux niveaux de résolution plus fins dans l'espace complet de RV, mais plutôt dans un sous-ensemble de RV.

La section 3.3.1 traite de la création et du raffinement de la liste d'index tandis que la section 3.3.2 explique en détail le processus itératif de remplissage basé sur une recherche locale.

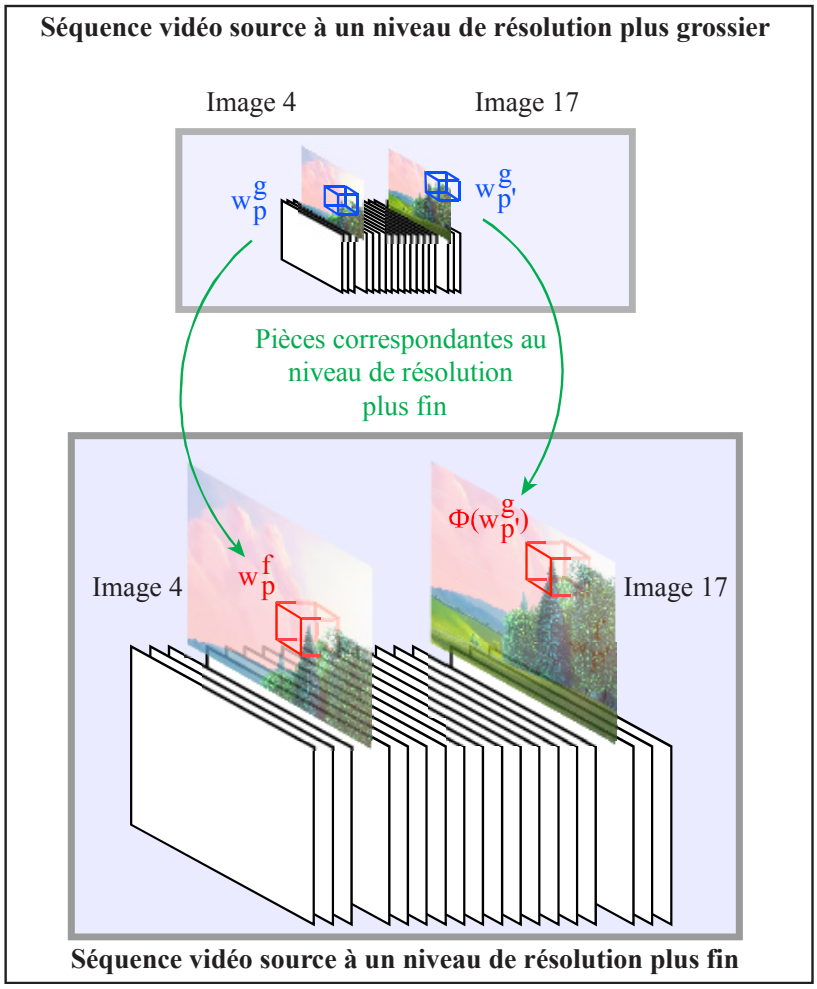


Figure 3.8 Observation menant à la réduction de l'espace de recherche.

3.3.1 Création de la liste d'index et raffinement basé sur la cohérence

Une fois la première étape terminée, c'est-à-dire la complétion de la séquence de basse résolution, chaque pièce w_p^g (centrée à $p_g \subset RR$) est associée avec sa pièce la plus similaire $w_{p'}^g$ (centrée à $p'_g \subset RV$) par la création d'une liste d'index LI qui contient les paires de

points spatio-temporels p_g-p'_g pour tous les points $p_g \subset RR$. Pour que ces associations soient effectuées dans un délai raisonnable, la recherche de la pièce la plus similaire $w^g_{p'}$ doit être effectuée en utilisant une technique de recherche approximative (Arya *et al*., 1998) sur des données qui sont compressées par une analyse en composantes principales. Par conséquent, la pièce $w^g_{p'}$ peut ne pas être la pièce la plus optimale contenue dans RV. Durant l'étape 3 (voir la figure 3.2), cette liste d'index permet de restreindre l'espace de recherche à un sous-ensemble seulement de la région valide RV. À titre de rappel, l'hypothèse proposée est que pour une pièce w^g_p à un niveau de résolution plus grossier et sa pièce la plus similaire $w^g_{p'}$, pour la pièce w^f_p au niveau de résolution plus fin a de bonnes chances d'être trouvée près de la pièce $\Phi(w^g_{p'})$.

Tel que mentionné précédemment, pour des raisons d'efficacité, la recherche des paires w^g_p-$w^g_{p'}$ doit être effectuée sur des données compressées en utilisant des méthodes de recherches approximatives. Bien que ces méthodes soient nécessaires pour obtenir des temps de recherche acceptables, il n'en demeure pas moins qu'elles introduisent une erreur qui entraîne que certaines paires w^g_p-$w^g_{p'}$ ne minimisent pas $D(w^g_p, w^g_{p'})$. C'est pourquoi il est nécessaire de raffiner la liste d'index LI afin d'avoir des pièces $w^g_{p'}$ plus similaires.

L'information contenue dans la liste d'index LI doit être très fiable pour que le processus itératif de remplissage puisse produire des résultats visuellement plausibles. Les pièces w^g_p et $w^g_{p'}$ doivent être très similaires pour chaque paire p_g-p'_g sinon, l'algorithme restreindra l'espace de recherche à une région où il est moins ou peu probable de trouver une pièce similaire $w^f_{p'}$. La figure 3.9 montre un exemple où l'information contenue dans LI n'est pas fiable. Plusieurs artéfacts visuels près de la fenêtre centrale ou près du contour gauche de l'édifice peuvent être observés.

(a) Image originale (b) Image HD complétée sans le raffinement de *LI* (c) Image HD complétée avec le raffinement de *LI*

Figure 3.9 Impact du raffinement de la liste d'index *LI*.

Afin de trouver des pièces $w_{p'}^{g}$ plus similaires, la technique proposée profite d'une nouvelle variante du concept de cohérence tel qu'initialement introduit par Ashikhmin (2001). Dans un premier temps, l'algorithme calcule la distance (la norme L_2 sur les composantes RVB non compressées) des pièces w_{p}^{g} et $w_{p'}^{g}$ pour chaque paire p_g-p'_g contenue dans la liste d'index LI. Par la suite, un processus itératif raffine les paires p_g-p'_g avec des distances plus élevées qu'un certain seuil. Durant la première itération, ce seuil est déterminé de façon à ce qu'environ 15 % des paires p_g-p'_g soient traitées. À la suite de chaque itération, ce seuil est réduit de 20 % de sa valeur initiale.

Pour chaque paire p_g-p'_g au-dessus du seuil, l'algorithme cherche un remplaçant p'_g qui diminue la distance entre les pièces w_{p}^{g} et $w_{p'}^{g}$. Plutôt que de choisir une approche *force brute* qui balaie la région valide RV, l'algorithme recherche plutôt à proximité des meilleures pièces $w_{p'}^{g}$ trouvées pour les voisins de p_g. La figure 3.10 en montre un exemple. L'algorithme considère les quatre voisins de p_g : en haut, à droite, en bas et à gauche respectivement nommés p_1, p_2, p_3 et p_4. À partir du voisin d'en haut (p_1), l'algorithme regarde quel était le point p'_1 préalablement associé à p_1 dans la liste d'index LI qui représentait la localisation de la pièce la plus similaire. Une fois que le point p'_1, est déterminé, l'algorithme regarde son voisin du bas, soit le point p''_1. La distance L_2 est par la

suite calculée entre les pièces centrées à p_g et p_1''. Si cette distance est plus petite que celle entre les pièces centrées à p_g et p_g', p_g' est remplacé par p_1'' dans la liste d'index p_g et la couleur RVB au point p_1'' est copiée au point p_g. Cette opération est répétée pour les points p_2, p_3 et p_4. S'il n'y a aucun bon candidat, la paire p_g-p_g' demeure inchangée et sera réévaluée à la prochaine itération.

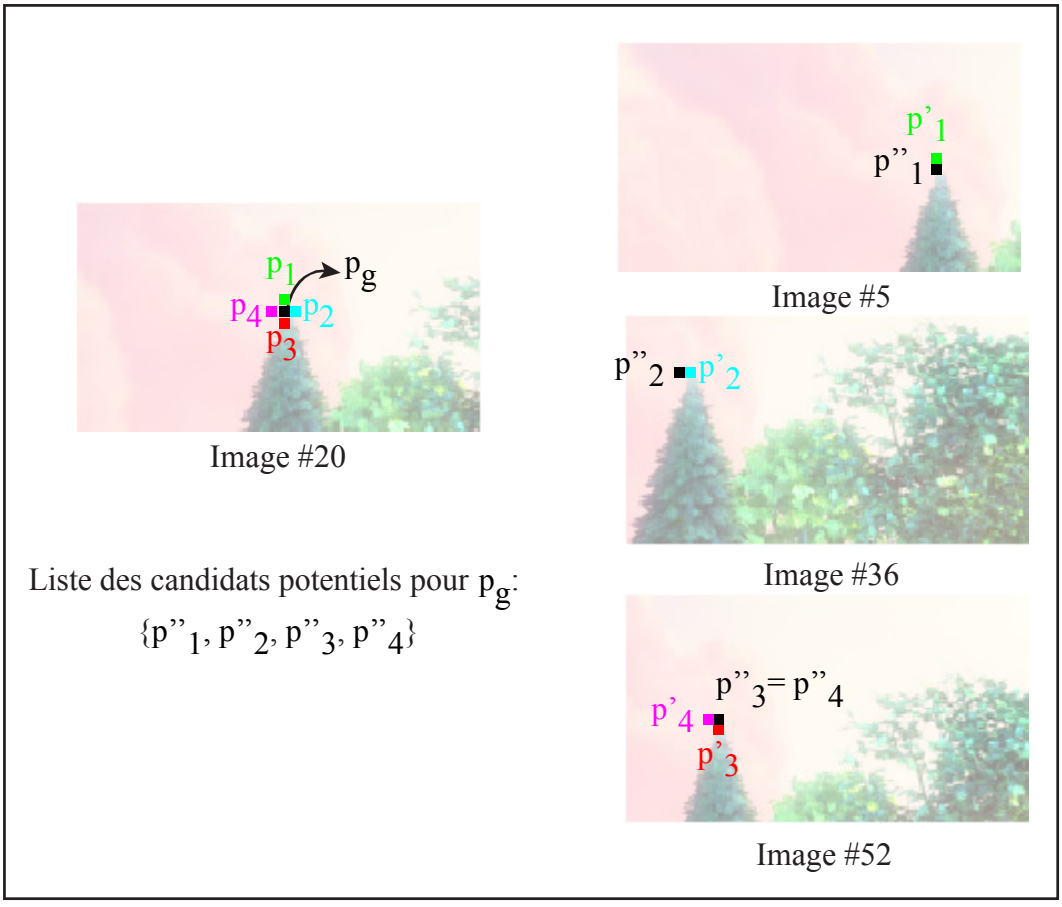


Figure 3.10 Raffinement de la liste d'index basé sur le concept de cohérence.

Lorsque l'algorithme considère le point p_1, au lieu de chercher partout autour du point p_1' qui lui est associé, l'algorithme regarde uniquement son voisin du bas (p_1''). Le raisonnement derrière ce choix est que plusieurs méthodes ont obtenu de bons résultats en copiant de large primitives tels que les *epitomes* ou les fragments. En prenant de plus larges primitives, le voisin du bas (p_1'') serait celui qui aurait été à la place de p_1. Cette décision de ne prendre

qu'un seul voisin permet de réduire l'espace de recherche à seulement quatre points (p_1'', p_2'', p_3'' et p_4''). Afin de diminuer encore plus le temps de calcul, l'algorithme commence par s'assurer que la distance L_2 des paires de voisins, p_2-p_2' par exemple, est en dessous du seuil avant même de calculer la distance L_2 entre p_g-p_2''. Ce test est très rapide puisque cette valeur est déjà emmagasinée dans la liste d'index LI. Puisqu'il n'y a qu'au maximum quatre points à vérifier, contrairement aux millions de points dans l'ensemble de la région valide RV, ce processus de raffinement est très efficace. La figure 3.11 présente un exemple d'application du processus de raffinement de la liste d'index. Elle met en évidence que l'erreur maximum diminue grandement suite à quelques itérations. Le processus de raffinement produit une amélioration significative de la qualité des résultats en ne prenant que quelques secondes.

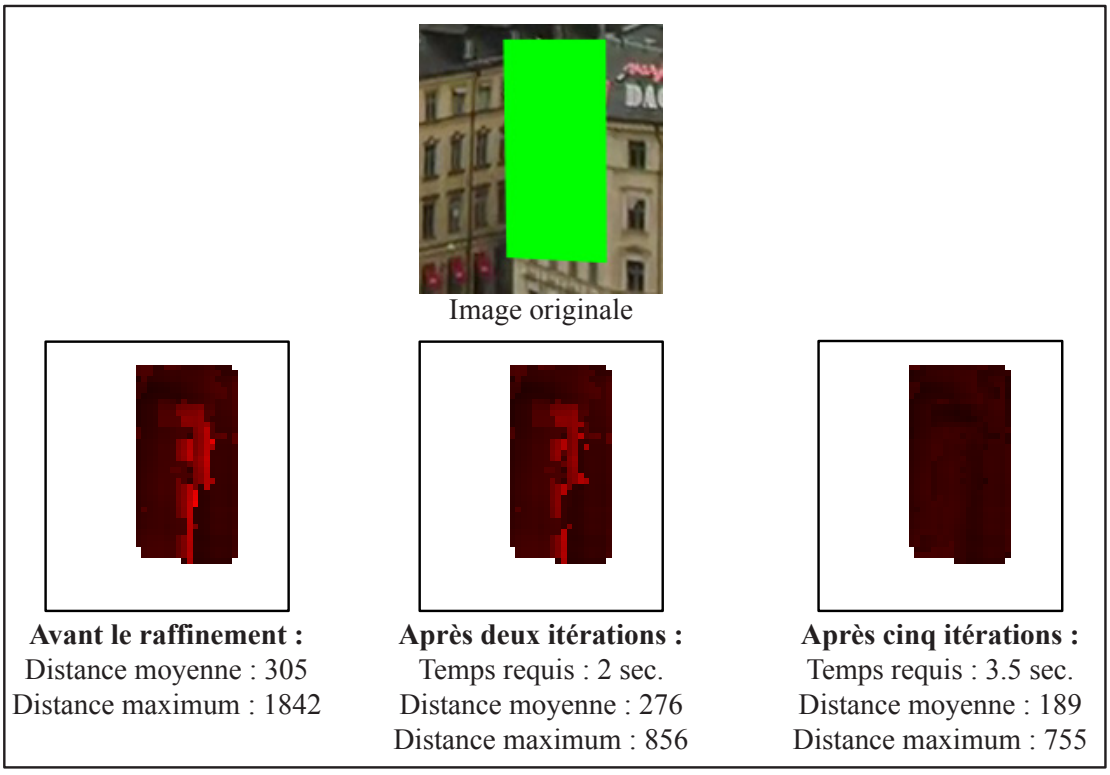


Figure 3.11 Impact du raffinement itératif de la liste d'index.

3.3.2 Processus itératif de complétion à l'aide d'une recherche locale

Avant de commencer à proprement dit le processus itératif de remplissage au niveau de résolution le plus fin, le contenu de la liste d'index *LI*, qui représente de l'information sur des points spatio-temporels à un niveau de résolution grossier, doit être mis à l'échelle afin qu'il soit convenable et utilisable au niveau de résolution plus fin. La figure 3.12 montre la démarche utilisée. Pour tous les points spatio-temporels $p_f \in RR$ au niveau de résolution le plus fin, l'algorithme regarde dans la liste d'index *LI* pour y trouver son point correspondant p_g ainsi que la localisation p'_g associée à ce dernier. Le point p'_g est par la suite mis à l'échelle et transféré vers le niveau de résolution le plus fin. Le point initial $p'_f = \Phi(p'_g)$ est donc trouvé. La paire p_f-p'_f est par la suite ajoutée à une nouvelle liste d'index *LIF* qui sera utilisée par le processus itératif de remplissage de séquence vidéo de haute résolution pour réduire l'espace de recherche des candidats de remplacement.

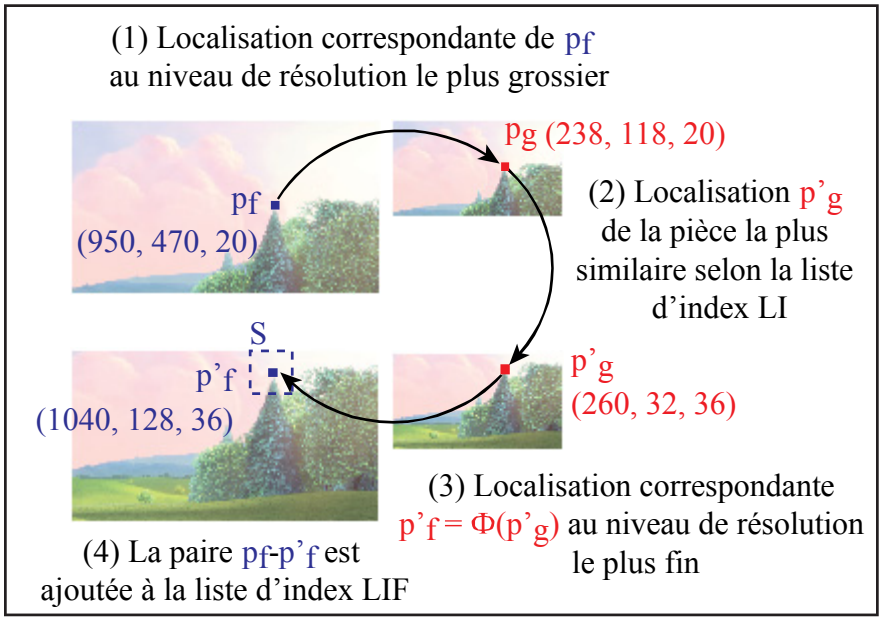


Figure 3.12 Création de la liste d'index *LIF* en se basant sur la liste d'index *LI*.

Les étapes principales du processus de remplissage de séquence vidéo de haute résolution sont similaires à celles pour la basse résolution (voir section 3.2) : en utilisant une approche

de remplissage par trou en trois dimensions, l'algorithme recherche une couleur RVB de remplacement c' pour chaque point spatio-temporel $p_f \in RR$. Cette couleur c' est déterminée en n'utilisant que la meilleure pièce correspondante $w_{p'}^{f}$. Cependant, plutôt que de parcourir tout l'espace de recherche RV en utilisant une approche *force brute* ou des structures d'optimisation coûteuses en espace mémoire, l'algorithme cherche uniquement dans une petite sous-région $S \subset RV$ déterminée par l'information contenue dans la liste d'index LIF.

Pour chaque point spatio-temporel $p_f \in RR$, l'algorithme cherche dans LIF pour trouver le point p'_f qui lui est associé. Par la suite, une petite région S est sélectionnée autour du point p'_f. Puis, l'algorithme cherche uniquement dans la sous-région S pour trouver la meilleure pièce correspondante $w_{p''}^{f}$, centrée au point $p''_f \subset S$. La couleur RVB c est ensuite remplacée par c''. Par la suite, la liste d'index LIF est actualisée en remplaçant la valeur de p'_f par celle de p''_f. À la prochaine itération du processus de remplissage, la sous-région S sera recentrée autour de cette nouvelle localisation p'_f.

Lors de la première itération, la taille de la fenêtre de la sous-région S est de 17x17 pixels. Cette taille est réduite après chaque itération (13x13, 9x9, 5x5). La figure 3.13 montre un exemple où la localisation et la taille de la sous-région S changent pendant les trois premières itérations du processus de remplissage.

Évidemment, le temps de calcul nécessaire pour effectuer les recherches est drastiquement réduit avec l'utilisation de la liste d'index LIF comparativement aux méthodes de recherche qui utilisent des structures d'optimisation comme ANN et la compression de données par l'analyse des composantes premières. En utilisant la technique proposée, moins de 1 000 pièces candidates sont comparées pour chaque point spatio-temporel p_f comparativement aux dizaines de millions de pièces potentielles contenues dans la région valide RR.

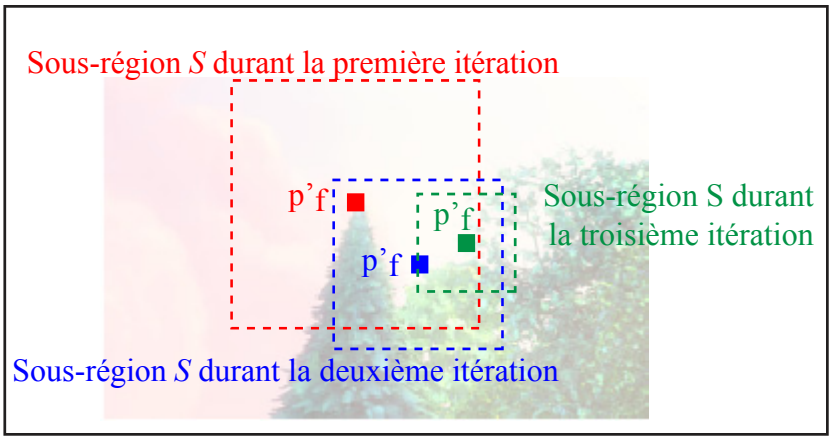


Figure 3.13 Évolution de la sous-région S du processus itératif de remplissage.

De plus, le temps de calcul nécessaire à la création et au raffinement des listes d'index *LI* et *LIF* est moindre que le temps nécessaire pour créer les structures utilisées par ANN et pour effectuer l'analyse en composantes premières. Un autre point avantageux de la technique proposée est que l'espace mémoire utilisé pour les listes d'index *LI* et *LIF* est beaucoup plus petit que celui accaparé par les structures de ANN. Finalement, le calcul de la mesure de similarité entre deux pièces ne se base pas sur des données compressées, éliminant ainsi toute possibilité d'erreur lors d'approximation.

3.4 Résultats

Cette section présente des résultats obtenus à l'aide du processus de remplissage de régions manquantes à l'intérieur de séquences vidéo. La figure 3.7 (section 3.2.3) montre les résultats de la complétion de la séquence vidéo « Jogging Lady » obtenus par Wexler, Shechtman et Irani (2007) et par le processus de complétion hybride *inpainting*-échantillonnage de séquence vidéo à basse résolution présenté à la section 3.2. Les résultats obtenus par la méthode proposée sont d'une qualité équivalente à ceux de Wexler, Shechtman et Irani (2007), mais ont été obtenus beaucoup plus rapidement.

La figure montre les résultats obtenus pour la complétion de la séquence vidéo « Station »[3] tandis que la figure 3.15 montre ceux de la séquence vidéo « Race to Mars »[4]. Le défi principal de ces deux séquences vidéo est le mouvement constant de la caméra. La séquence vidéo « Station » contient une constante mise à l'échelle de l'image tandis que la séquence vidéo « Race to Mars » contient un léger mouvement de rotation et de translation panoramique (*panning*). Ces types de séquences vidéo ne peuvent être traités avec les techniques de remplissage qui utilisent une mosaïque d'arrière-plan fixe parce que la taille et l'orientation des objets ne sont pas constantes durant toute la durée de la séquence vidéo.

La figure montre bien que la taille des régions à compléter est beaucoup plus grande que celle vue dans l'état de l'art. La figure 3.15 démontre de son côté que la méthode produit de bons résultats autant lorsque la région à compléter contient une texture stochastique que lorsqu'elle contient des éléments structurels plus précis.

La figure 3.16 montre les résultats obtenus pour la séquence vidéo « Ladle »[5]. Elle démontre que la technique fonctionne également avec des séquences vidéo générées par ordinateur. La figure 3.16 présente également une limitation de la technique proposée : lorsqu'une partie de l'information nécessaire pour compléter la région manquante ne peut être trouvée dans la région valide, l'algorithme proposé introduit des artéfacts visuels. En effet, en regardant de plus près la complétion, on peut observer que la reconstruction de la louche près de la main contient des artéfacts visibles puisqu'on ne retrouve pas la main sans qu'elle ne tienne la louche à aucun endroit dans la région valide. Par conséquent, l'algorithme n'a pas été en mesure de reconstruire l'image correctement.

---

3 Séquence *Station* (https://cs-nsl-wiki.cs.surrey.sfu.ca/wiki/Video_Library_and_Tools)
4 Séquence tirée de l'émission « Race to Mars », gracieuseté de Galafilm et Discovery Channel Canada
5 Extrait du film Sintel (https://durian.blender.org/)

Finalement, la figure 3.17 présente les résultats obtenus pour la séquence vidéo « Old town cross »[6]. Pour cette séquence, un objet de synthèse a été ajouté à une séquence dite *propre* pour par la suite être corrigé par la technique de remplissage. Il est donc possible de comparer le résultat obtenu avec la référence originale.

[6] Séquence *Old town cross* (https://cs-nsl-wiki.cs.surrey.sfu.ca/wiki/Video_Library_and_Tools)

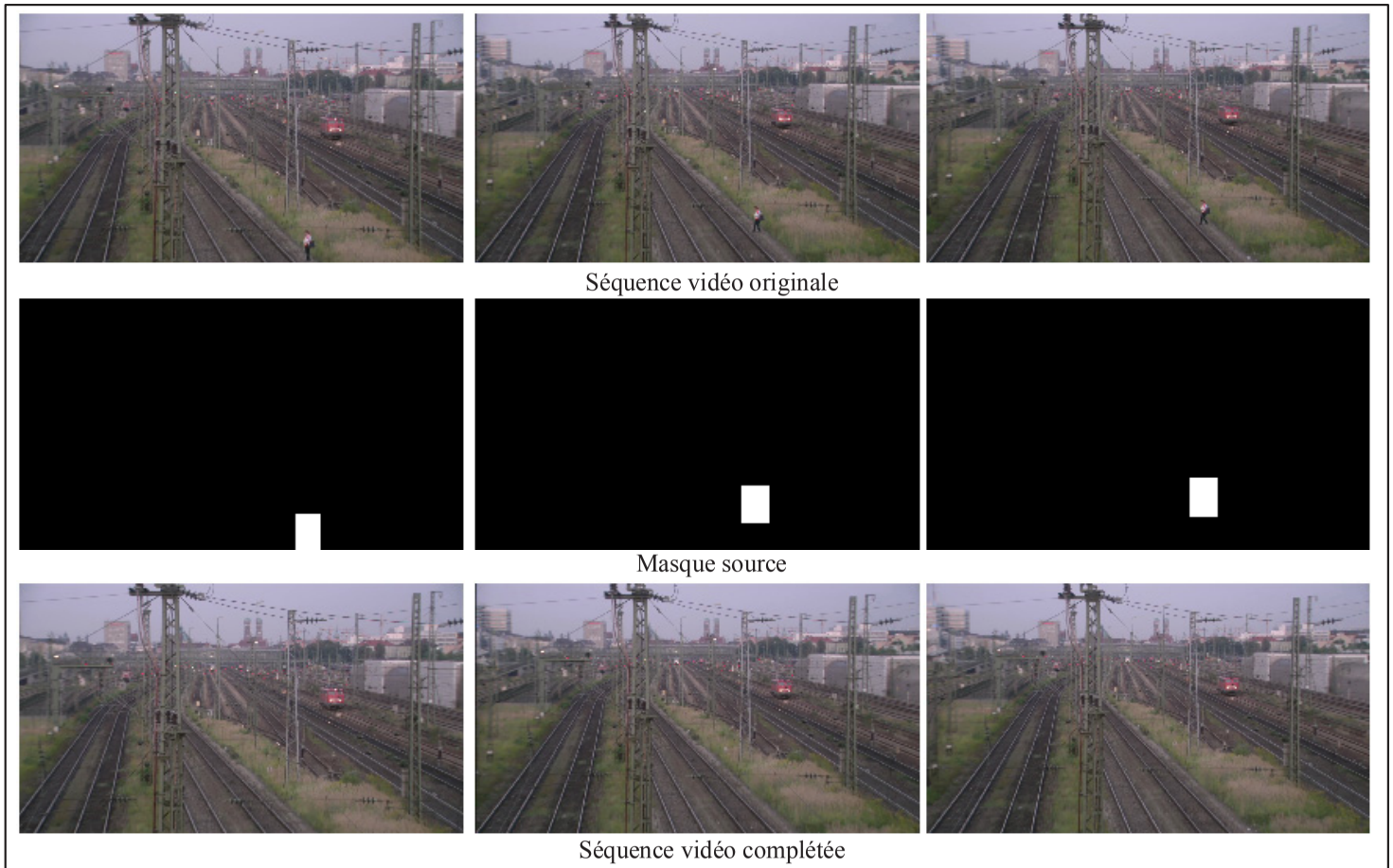


Figure 3.14 Résultats pour la séquence vidéo « Station ».

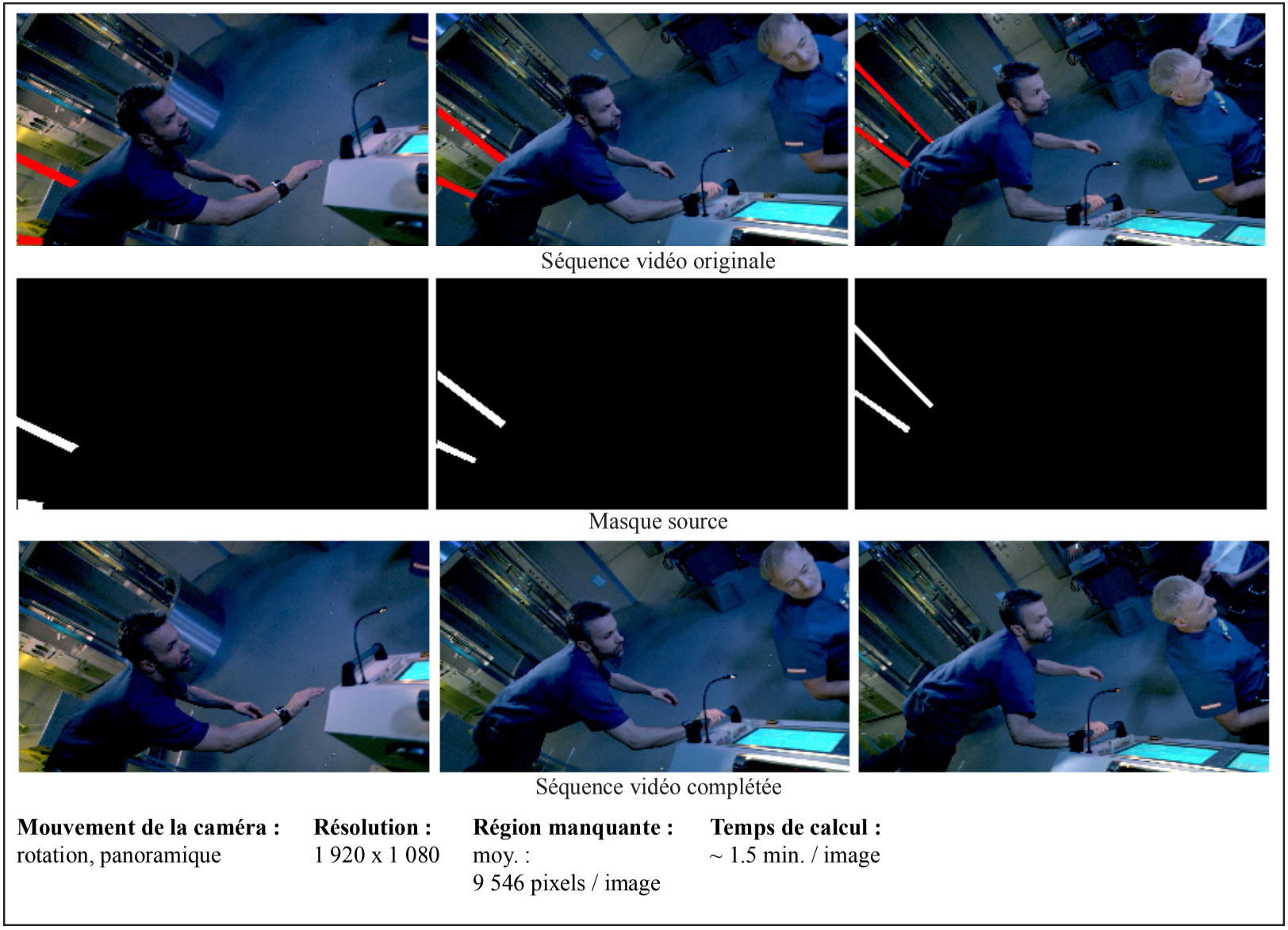


Figure 3.15 Résultats pour la séquence vidéo « Race to Mars ».

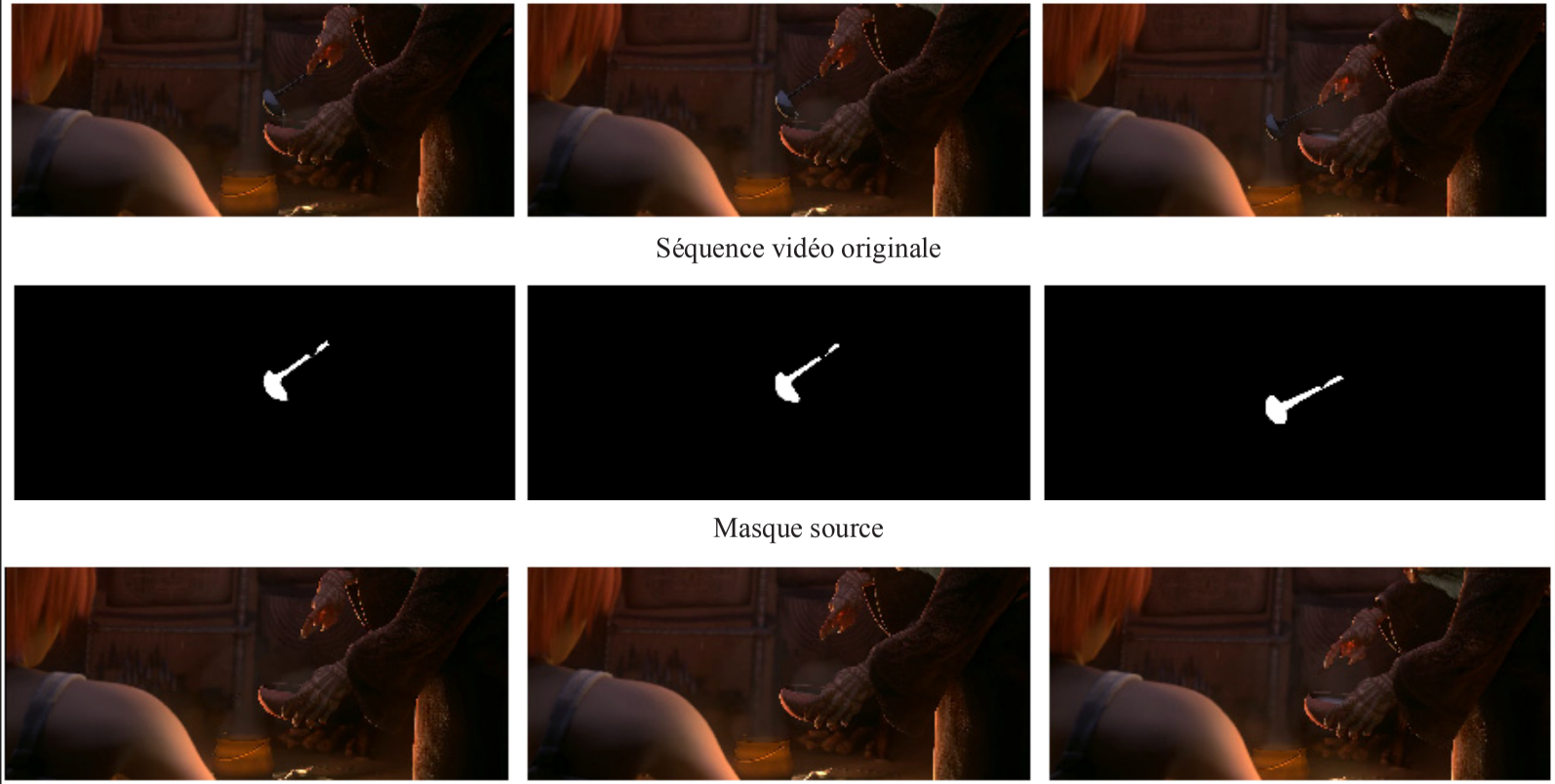
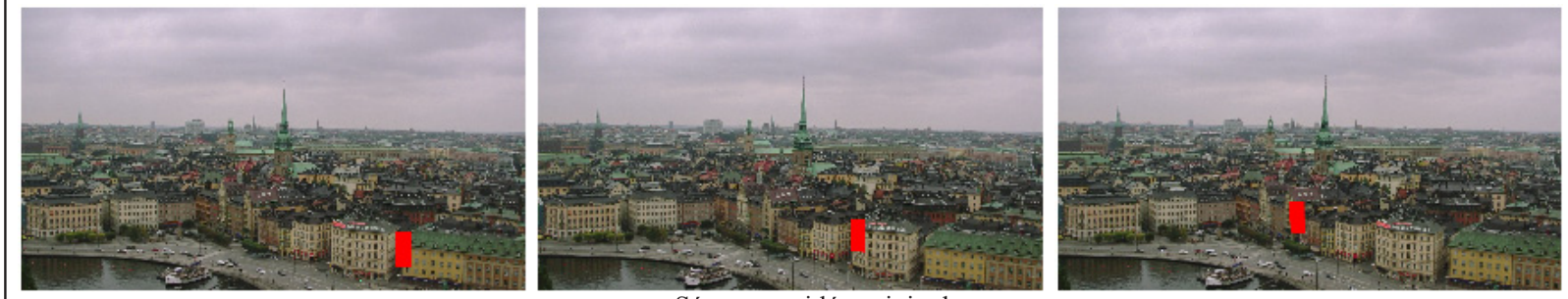


Figure 3.16 Résultats pour la séquence vidéo « Ladle ».



Séquence vidéo originale

Séquence vidéo propre

Séquence vidéo complétée

Mouvement de la caméra :	**Résolution :**	**Région manquante :**	**Temps de calcul :**
zoom avant	1 920 x 1 080	moy. : 8 456 pixels / image	~ 1.25 min. / image

Figure 3.17 Résultats pour la séquence vidéo « Old town cross ».

3.5 Discussions

Cette section analyse et interprète les résultats obtenus par le système d'édition de séquence vidéo basé sur les champs aléatoires de Markov et sur une recherche local présenté dans ce chapitre.

3.5.1 Évaluation objective de la qualité visuelle des résultats

L'évaluation quantitative et objective de la qualité visuelle des résultats obtenus est difficile dans le domaine de la complétion vidéo. Pour bien évaluer objectivement, il est nécessaire d'utiliser une métrique objective fiable et de pouvoir comparer les résultats à un ensemble de données (séquences vidéo) standardisé et couramment utilisé dans la littérature. Or, ces deux points font défaut dans le domaine de la complétion vidéo.

Premièrement, tel que souligné par Granados *et al.* (2012) et Newson *et al.* (2014), il n'existe encore aucune métrique en mesure d'évaluer objectivement ce qui représente du contenu *visuellement acceptable* dans une séquence vidéo reconstruite. Il n'est donc pas surprenant de remarquer que l'ensemble des approches récentes de complétion vidéo recensées dans le tableau 3.2 n'utilise pas de métrique objective afin d'évaluer la qualité des résultats obtenus. Il existe différentes métriques pour évaluer la similarité visuelle entre deux images, utilisées par exemple dans le domaine de la compression vidéo, telles que *Peak Signal-to-Noise Ratio* (PSNR), *Multiscale Structural Similarity* (MSS) de Wang, Simoncelli et Bovik (2003) ou *Structural SIMilarity* (SSIM) de Wang *et al.* (2004). Celles-ci sont cependant peu adaptées au domaine de la complétion vidéo. Tout d'abord, elles comparent deux images entre elles en prenant pour hypothèse qu'une des deux images représente la *référence*, c'est-à-dire l'image idéale à atteindre. Dans le cas de la compression vidéo, il s'agit de l'image avant la compression en soit. Or, pour la complétion vidéo, il n'y a pas qu'une seule image *référence* : il y a une multitude de complétions *visuellement acceptable*. De plus, ces métriques ne tiennent pas en compte la cohérence temporelle puisqu'elles comparent uniquement deux images entre elles. Or, le système de vision humain est sensible à la

cohérence spatiale. Pour ces raisons, la validité des valeurs obtenues avec ces métriques dans le cas précis de la complétion vidéo est peu fiable.

Deuxièmement, il n'existe pas encore un ensemble standardisé de séquences vidéo permettant de comparer entre elles les différentes approches. De plus, peu d'articles donnent accès aux séquences vidéo utilisées et complétées avec leur approche. Pour ce qui est des séquences vidéo disponibles (par exemple celles de Xiao et al. (2008)), elles présentent généralement des caractéristiques qui peuvent rendent difficile leur réutilisation. Par exemple, la résolution ou la taille des régions à compléter de certaines séquences vidéo sont trop petites pour rendre intéressante la comparaison avec l'approche présentée dans ce chapitre.

Dans le cadre de ce projet de recherche, différentes expériences ont été tenté afin de trouver une métrique ou une méthode de validation quantitative et objective de la qualité visuelle des résultats obtenues. Par exemple, une expérimentation a été effectuée avec une variation de la métrique PSNR. Le problème de la métrique PSNR classique, lors d'une utilisation dans le contexte de validation d'une complétion vidéo, réside dans le fait qu'il n'existe pas une seule image *référence* représentant la complétion idéale. Afin d'imager cette situation, considérons une séquence vidéo où un camion passe devant un piéton et l'obstrue entièrement. Si on décide d'enlever le camion avec la complétion vidéo, il est impossible de prédire avec certitude le comportement du piéton : il serait tout aussi *perceptuellement valide* que le piéton marche normalement ou qu'il décide de bondir. La variation de la métrique PSNR expérimentée se fondait sur l'hypothèse que les pixels à remplacer en bordure de la région manquante ont de plus fortes chances d'être « prévisibles »; c'est-à-dire qu'ils convergent tous vers une même image *référence*. Par conséquent, la variation de la métrique PSNR testée ne tenait pas compte de tous les pixels manquants, mais uniquement de ceux étant près de la région valide. Malheureusement, la fiabilité des validations obtenues avec cette variation de la méthode PSNR était peu convaincante.

Évidement, la définition d'une métrique fiable d'évaluation objective de la qualité visuelle de résultats obtenus lors de complétion vidéo ainsi que la création d'un ensemble standardisé de séquences vidéo seraient des contributions scientifiques importantes reliées à ce domaine de recherche. Il s'agit donc de pistes intéressantes pour des travaux futurs.

3.5.2 Avantages

Le processus d'édition de séquence vidéo proposé utilisant une approche de synthèse de textures basée sur les champs de Markov permet de résoudre plusieurs problèmes soulevés aux chapitres 1 et 2. Tout d'abord, une nouvelle approche pour l'édition de séquence vidéo de basse résolution permet de réduire le temps de calcul nécessaire afin de converger vers une solution visuellement plausible. En effet, l'ajout d'une étape d'initialisation de la couleur des pixels indésirables basée sur une approche de *image inpainting* et l'application d'une approche de remplissage par trou en trois dimensions qui maximise la pertinence du cube spatio-temporel permettent de réduire le nombre d'itérations nécessaires pour converger vers un résultat visuellement plausible.

De plus, le processus itératif de remplissage de séquence vidéo utilisant une liste d'index et une recherche locale proposé dans ce chapitre permet de corriger des séquences vidéo avec des résolutions beaucoup plus grandes que celles généralement vues dans l'état de l'art. Effectivement, l'approche novatrice proposée réduit l'espace de recherche des candidats de remplacement, contrairement aux méthodes actuelles qui tentent plutôt de minimiser le temps de recherche. Ce faisant, la méthode proposée n'est pas tenue d'utiliser des structures d'optimisation coûteuse en espace mémoire pour effectuer les recherches et rend donc possible l'édition de séquences vidéo de haute définition. En outre, le calcul de la mesure de similarité entre deux pièces ne se base pas sur des données compressées, éliminant ainsi les erreurs d'approximation lors des recherches de candidats de remplacement au niveau de résolution le fin plus. Comme le montre le tableau 3.2, l'approche proposée est la seule à traiter des séquences vidéo avec une résolution HD sans contrainte particulière. L'approche de Newson *et al.* (2014) peut traiter certaines séquences vidéo HD, mais l'utilisation de

structures de recherche pour accélérer la recherche des pièces similaires et la nécessité de conserver l'information relatives aux textures dynamiques limitent la durée des séquences vidéo à un maximum d'environ 120 cadres selon les expérimentations effectuées. Cette contrainte de durée est un désavantage majeur puisqu'elle réduit grandement l'éventail de séquences vidéo que la technique peut compléter correctement. À première vue, on peut croire qu'il suffit de segmenter une séquence en plusieurs parties de 120 cadres, d'utiliser successivement l'approche de Newson *et al.* (2014) sur celles-ci et de combiner les séquences corrigées. Malheureusement, cette façon de faire n'est pas toujours viable. En effet, la segmentation de la vidéo réduit aussi la région valide de la vidéo, donc la région où il est possible de trouver une pièce de remplacement. Par conséquent, si l'information nécessaire pour compléter une région manquante d'un cadre contenu dans l'intervalle 0 à 120 se trouve uniquement plus loin dans la séquence (ex. : dans l'intervalle des cadres 240 à 360), l'approche de Newson *et al.* (2014) n'est pas en mesure de la considérer. L'approche de Ebdelli *et al.* (2015) montre la même limitation puisqu'elle considère uniquement les 20 cadres précédents et les 20 cadres suivants pour compléter le cadre courant.

Aussi, le processus itératif de remplissage de séquence vidéo utilisant une liste d'index et une recherche locale proposé réduit le temps de recherche au niveau de résolution HD ce qui permet de compléter de plus grandes régions manquantes que l'état de l'art dans un délai acceptable. Le tableau 3.2 compare la taille des régions manquantes des approches récentes. Bien que l'approche proposée soit en mesure de compléter des régions manquantes de taille supérieure à ce que l'on retrouve dans l'état de l'art, elle présente tout de même des limitations (voir section 3.5.3).

Finalement, l'approche novatrice basée sur l'utilisation d'une liste d'index est indépendante de la mesure de similarité entre deux pièces choisies. C'est donc dire qu'elle pourrait être réutilisée avec de nouvelles mesures de similarité permettant de corriger une plus grande gamme de séquences (ex. compléter des zones indésirables dans lesquelles des objets en mouvement sont partiellement ou entièrement cachés). Le tableau 3.2 synthétise la comparaison de l'approche proposée avec les travaux récents.

Tableau 3.2 Comparaison de l'approche proposée avec l'état de l'art

Auteurs	**HD?**	**Résolution maximum**	***Nb. cadres maximum***	**Caméras complexes?**	**Grandes régions manquantes?**
Benoit et Paquette (2015)	Oui	1920 x 1080	250	Limité	Supérieures
Newson *et al.* (2014)	Limité	1120 x 754	200	Limité	Supérieures
Ebdelli *et al.* (2015)	Limité.	1440 x 1056	180	Limité	Oui
Newson *et al.* (2013)	Non	1120 x 754	200	Non	Supérieures
Daisy *et al.* (2015)	Non	960 x 544	106	Non	Non
Xu *et al.* (2015)	Non	960 x 540	93	Limité	Supérieures
Herling et Broll (2014)	Non	640 x 320	?	Limité	Non
Zarif, Faye et Rohaya (2013)	Non	640 x 480	250	Non	Non
Mosleh *et al.* (2012)	Non	320 x 240	ND	Non	Non
Vijay Venkatesh *et al.* (2009)	Non	320 x 240	140	Non	Non
Koochari et Soryani (2010)	Non	320 x 240	105	Non	Non
Xiao *et al.* (2011)	Non	320 x 130	150	Non	Non

3.5.3 Limitations

Même si le processus d'édition de séquence vidéo utilisant une approche de synthèse de texture basée sur les champs de Markov présenté dans ce chapitre permet de résoudre plusieurs problèmes, il possède tout de même certaines limitations. En premier lieu, bien qu'il permette le traitement de séquence vidéo de haute définition et qu'il soit en mesure de corriger des plus grandes régions que celles observées dans l'état de l'art, il demeure que les régions à corriger doivent être relativement petites. En fait, puisqu'une technique de remplissage de trou en trois dimensions est utilisée, il est préférable que la zone indésirable

ne soit pas très *profonde*, sans quoi il sera nécessaire d'effectuer plusieurs itérations afin d'arriver à un résultat perceptuellement acceptable. Imagé en deux dimensions, il est préférable d'avoir une zone à remplacer de 2 pixels par 32 pixels plutôt qu'une autre région de 8 pixels par 8 pixels ayant le même nombre total de pixels, mais avec une *profondeur* plus élevée. La méthode présentée à ce chapitre est donc plus adaptée pour la correction de longues et étroites régions indésirables comme les fils d'un harnais ou une perche de son. Les approches de Newson *et al.* (2014) et Ebdelli *et al.* (2015) utilisent également un processus de remplissage itératif qui limitent la profondeur des régions manquantes de la même façon. Le chapitre 4 décrit une méthode permettant de corriger des régions manquantes ayant de grandes *profondeurs*.

En deuxième lieu, puisque la correction d'une région indésirable se fonde sur la recherche du meilleur candidat en fonction uniquement de la couleur RVB des pièces contenues dans l'ensemble de la séquence vidéo, il est impératif qu'une telle pièce existe avec les mêmes valeurs de couleur et la même orientation. La technique proposée est donc sensible aux erreurs d'exposition et aux mouvements de caméras non-triviaux comme une mise à l'échelle rapide ou une rotation marquée. Le chapitre 4 s'attarde plus particulièrement à ce problème et y propose une solution. Aussi, la recherche du plus proche voisin basé uniquement sur la couleur RVB fait en sorte que l'approche ne donne pas toujours de bons résultats lorsque la région à compléter présente des textures stochastiques (ex. vagues, feuilles, etc.). En effet, on peut remarquer un effet de pétillement dans la zone corrigée puisque la cohérence temporelle n'est pas toujours respectée. L'approche de Newson *et al.* (2014) arrive à de meilleurs résultats dans cette situation puisqu'elle présente une fonction de distance pour la recherche des pièces de remplacement qui tient compte des textures dynamiques.

CHAPITRE 4

CARACTÉRIQUES INVARIANTES ET REMPLISSAGE VIDÉO

Tel que mentionné à la section 1.5, le troisième objectif de cette thèse est de concevoir une technique de remplissage automatique et efficace de régions manquantes dans une séquence vidéo adaptée aux artistes, aux studios de production et aux pipelines de production. Le chapitre 3 présente une approche de synthèse de texture basée sur les champs aléatoires de Markov appliquée au domaine de la retouche vidéo. Cette dernière propose une technique novatrice de recherche locale du meilleur candidat qui permet, entre autres choses, de traiter les séquences vidéo HD. Cependant, l'ensemble des mouvements de caméra des séquences vidéo pouvant être traitées ne contient pas des changements drastiques de mise à l'échelle ou de rotation et l'approche est sensible aux différences d'exposition que l'on retrouve fréquemment dans une séquence vidéo. De plus, les régions pouvant être corrigées demeurent relativement petites. Afin de pallier ces problèmes, ce chapitre propose une approche de correction de régions manquantes basée sur le suivi de caractéristiques invariantes[7]. Ce chapitre présente l'approche proposée en détaillant son fonctionnement étape par étape, montre des exemples de résultats et discute des avantages et des limitations de celle-ci par rapport à l'état de l'art.

4.1 Présentation générale de la méthode proposée

L'objectif principal du système de retouches de séquences vidéo proposé est d'offrir à l'artiste une technique d'édition qui permet de faire la suppression d'objets indésirables ou de remplacer une région manquante. Tel qu'énoncé à la section 3.1, l'artiste doit idéalement fournir la séquence vidéo originale et identifier la zone à corriger afin que le système d'édition soit en mesure de compléter son travail, sans avoir à spécifier différents paramètres complexes. Tel qu'illustré sur la figure 4.1, le système proposé se base sur ce principe pour

[7] Le contenu du chapitre 4 a été publié dans le cadre d'une conférence internationale : Benoit et Paquette (2016).

établir les étapes clés que l'artiste doit suivre lors de son utilisation; l'artiste n'est donc pas tenu de contrôler des paramètres complexes.

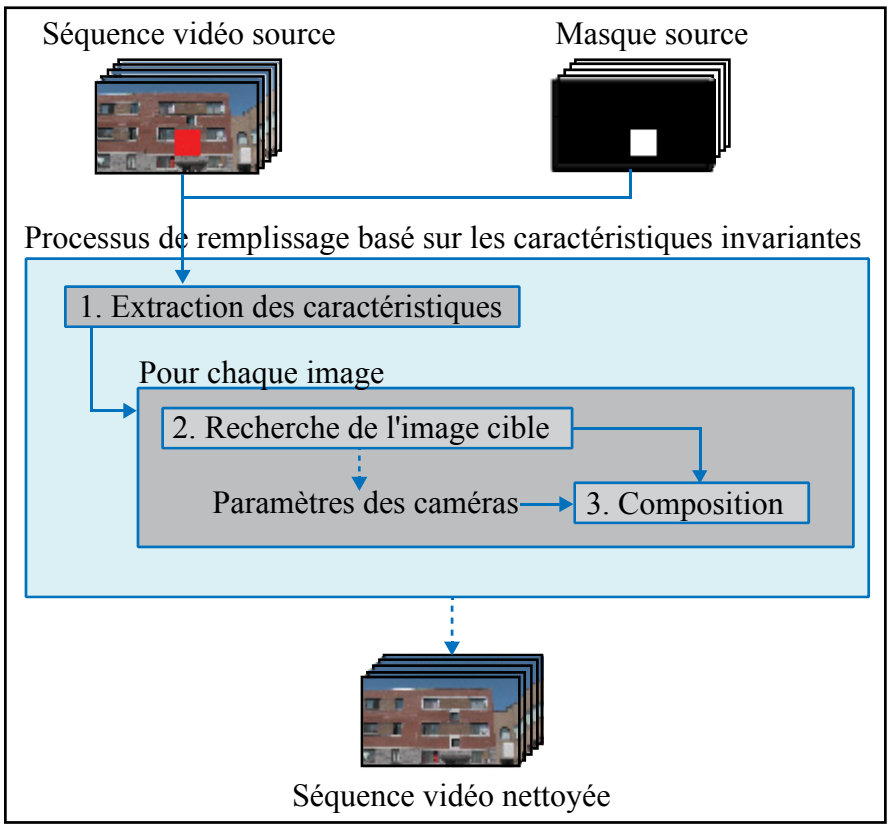


Figure 4.1 Système de remplissage basé sur les caractéristiques invariantes.

La figure 4.2 montre les étapes détaillées du processus de correction de séquences vidéo proposé dans ce chapitre. Tout d'abord, les caractéristiques invariantes SURF (Bay *et al.*, 2008) sont extraites de chacune des images contenues dans la séquence vidéo originale. Ces caractéristiques serviront à déterminer l'homographie entre deux images, c'est-à-dire la transformation linéaire entre deux plans projectifs, et les paramètres intrinsèques des caméras. Par la suite, pour chacune des images de la séquence vidéo source qui contient des régions manquantes identifiées par le masque source, la méthode associe une image cible candidate qui contient l'information nécessaire pour corriger la région manquante de l'image source. Chaque image source (qui présente une région manquante) est donc associée à une image cible (qui contient l'information nécessaire pour corriger l'image source). Afin d'accélérer cette étape, le système tire profit du concept de cohérence de façon à limiter le

nombre d'essais et d'erreurs. Le système estime ensuite les paramètres des caméras pour les paires d'images; ces derniers serviront à déformer les images de façon à les superposer.

Une fois les images cibles trouvées pour toutes les images sources, le système effectue la correction des régions manquantes ou indésirables (voir section 4.3). Tout d'abord, le système effectue le sous-échantillonnage de la séquence vidéo source et utilise la séquence sous-échantillonnée afin d'estimer les différences d'exposition. Ensuite, pour chaque paire d'images (image source et image cible) à pleine résolution, le système effectue les déformations en fonction des paramètres des caméras préalablement calculés. L'algorithme tente ensuite de minimiser les différences d'exposition avant d'effectuer la composition des images en utilisant une technique de mélange multi-bandes.

Le processus d'édition présenté dans ce chapitre est adapté au processus de création des artistes, en ce sens qu'il est simple d'utilisation et très intuitif. Il se démarque également des travaux antérieurs puisqu'il est en mesure de corriger des séquences vidéo comportant des mouvements de caméras complexes tels que des rotations rapides, des roulements et des mises à l'échelle. Il est aussi en mesure de corriger des séquences vidéo comportant des différences d'exposition. Aussi, le temps de calcul requis pour effectuer l'édition des séquences vidéo est rapide et il n'est pas dépendant du nombre d'images dans la séquence. Conséquemment, la méthode proposée est en mesure de traiter des séquences vidéo de haute définition dans un temps assez court pour qu'elle puisse être utilisée dans un pipeline de production réel. De plus, les régions manquantes corrigées sont beaucoup plus grandes que celles que l'on retrouve dans les travaux antérieurs.

La section 4.2 détaille les étapes liées à la recherche et à la déformation d'une image cible tandis que la section 4.3 précise la méthode pour corriger l'image source en fonction de l'image cible déformée.

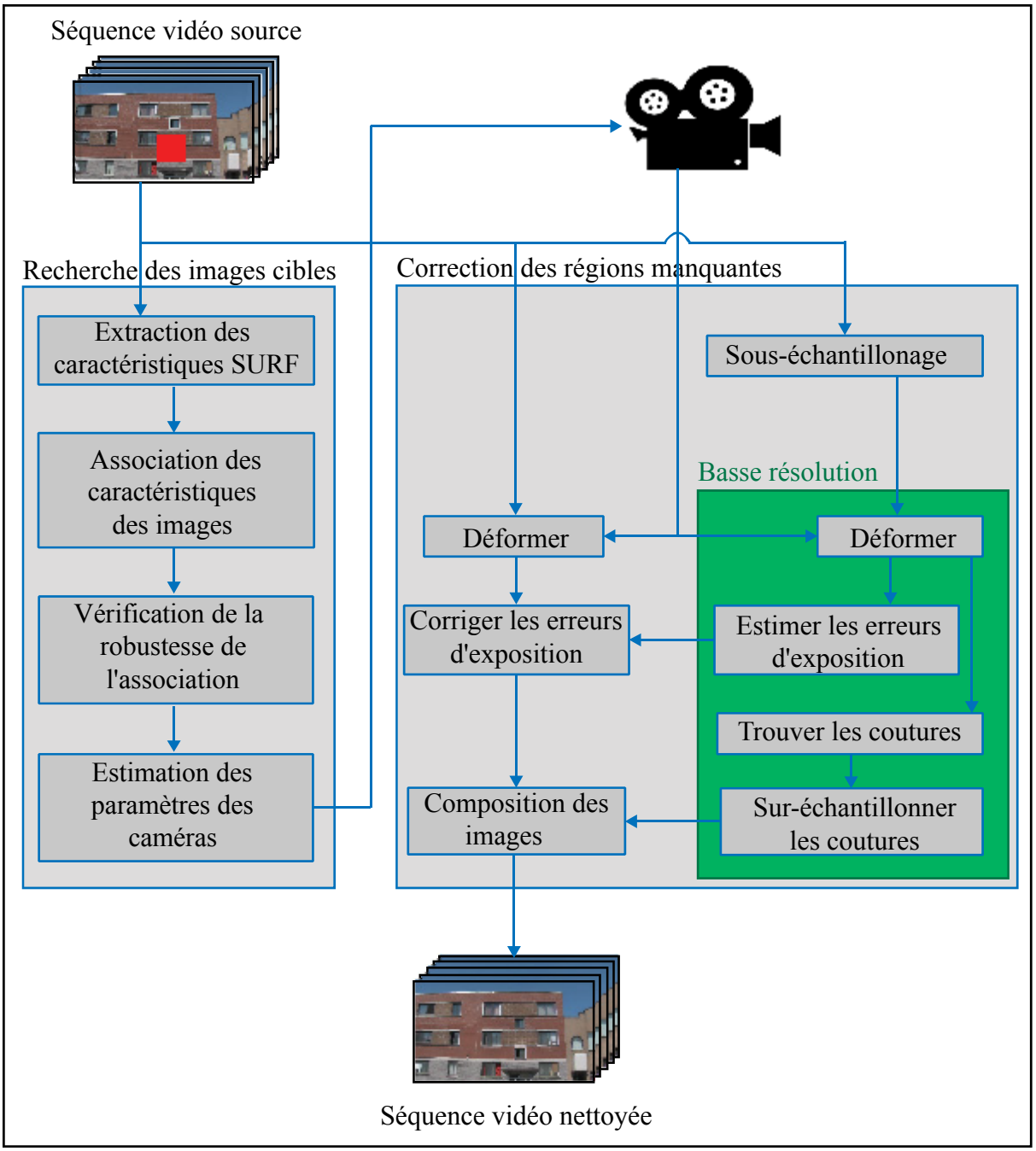


Figure 4.2 Processus d'édition de séquences vidéo.

4.2 Recherche et déformation d'une image cible

La tâche de trouver une correspondance entre deux images d'une même séquence vidéo revient fréquemment dans plusieurs domaines de la vision par ordinateur. Avec les méthodes

de suivi de caractéristiques, cette tâche se divise généralement en trois grandes étapes. Premièrement, des points d'intérêt sont identifiés à des endroits particuliers de l'image tels que les coins, les *blobs* ou les jonctions en « T ». Il est important que ces points d'intérêt, ou *détecteurs*, puissent être identifiés indépendamment des conditions d'observation (variation de l'exposition, orientation et échelle de l'image, etc.) de façon à augmenter la robustesse des correspondances. Deuxièmement, le voisinage de ces détecteurs est représenté par un vecteur de caractéristiques nommé *descripteur*. Ces descripteurs sont finalement comparés et associés pour les différentes images en calculant la distance entre ceux-ci. Cette section précise ces étapes dans la cadre d'application de la correction de régions manquantes de séquences vidéo.

4.2.1 Extraction des caractéristiques invariantes SURF

Le choix du type de caractéristiques et leur extraction a un impact important sur la qualité des résultats obtenus et sur l'éventail de séquences vidéo qu'il est possible de corriger. Une propriété cruciale des détecteurs est leur *répétabilité*, c'est-à-dire la capacité de retrouver le même point d'intérêt sous différentes conditions d'observation. De plus, le descripteur doit être robuste face aux variations d'intensité et à la présence de bruit ou de déformation. La technique présentée par Bay, Ess *et al. (2008)* nommée SURF remplit ces critères en offrant de bonnes répétabilité et robustesse. De plus, cette dernière est plus rapide que la technique SIFT présentée dans les travaux de Lowe (1999). L'implémentation SURF de la librairie OpenCV a été utilisée puisqu'elle se comparait avantageusement à d'autres implémentations (Gossow, Decker et Paulus, 2011).

Tel qu'indiqué à la figure 4.1, la première étape du processus d'édition de séquences vidéo est d'extraire les caractéristiques SURF pour l'ensemble des images. Cette extraction est réalisée une seule fois pour chaque image ce qui diminue le temps de calcul du processus de correction. La figure 4.3 montre un exemple d'extraction des caractéristiques SURF d'une image. L'image source a été convertie en noir et blanc uniquement dans le but d'augmenter la lisibilité de la figure.

Puisque les caractéristiques SURF sont invariantes aux rotations et aux mises à l'échelle, la technique proposée est capable de traiter des séquences d'images où l'orientation et le zoom varient. Ceci permet au système de se démarquer de l'état de l'art puisqu'il est en mesure de compléter des séquences vidéo avec des transformations de caméra non-triviales.

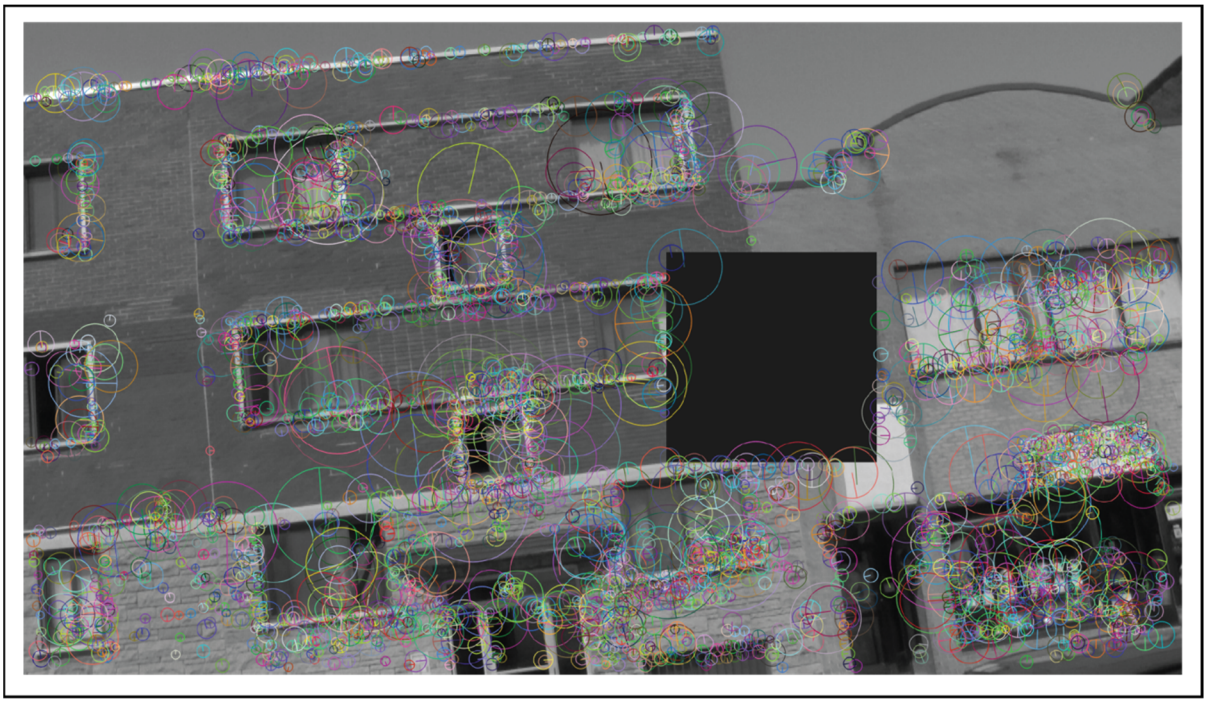


Figure 4.3 Extraction des caractéristiques.

4.2.2 Association des caractéristiques des images source et cible

Lorsque les caractéristiques invariantes et les descripteurs ont été extraits des différentes images, l'étape suivante est d'associer ces derniers afin de former des couples entre images sources et images cibles. Pour chaque image source, la technique proposée cherche une image cible qui est valide (plus d'information à ce sujet à la section 4.2.5) et où la majorité des descripteurs de l'image source y trouvent une correspondance. La correspondance entre deux caractéristiques est calculée en minimisant la distance euclidienne des vecteurs des deux descripteurs. La façon la plus simple de trouver ces correspondances est de comparer

toutes les caractéristiques entre elles. Malheureusement, cette technique requiert beaucoup trop de temps, ce qui la rend inutilisable dans ce contexte. Une meilleure approche consiste à utiliser une structure de données qui permet d'accélérer la recherche de meilleur candidat. Plusieurs de ces structures de recherche ont été développées au fil des années. Le travail de Muja et Lowe (2009) compare certaines de ces structures, présente une nouvelle technique de recherche (*priority search on hierarchical k-means tress*) et conclu que la structure kd-trees offre souvent les meilleures performances. La technique présentée par Muja et Lowe (2009) est utilisée par le système proposé pour effectuer la recherche des meilleures correspondances.

4.2.3 Estimation robuste des paramètres de l'homographie

Une fois l'association des caractéristiques de l'image source et de l'image cible terminée, le système proposé se base sur ces dernières afin d'estimer les paramètres de la transformation entre les deux images. Le système utilise la méthode RANSAC, présentée par Fischler et Bolles (1981), afin de trouver la solution la plus représentative compte tenu de l'ensemble des correspondances entre les caractéristiques. Cette technique prend successivement différents sous-ensembles aléatoires de correspondances, estime une matrice d'homographie **H** pour chacun d'eux et détermine la matrice la plus représentative de l'ensemble des données en maximisant le nombre de correspondances pertinentes (*inliers*). Avec un nombre de sous-échantillons suffisamment élevé, la probabilité d'obtenir une bonne matrice **H** est très élevée.

4.2.4 Estimation des paramètres des caméras

Puisqu'une majorité des séquences vidéos est tournée avec une caméra subissant une *rotation pure*, par l'utilisation d'un trépied par exemple, il est possible d'utiliser un modèle de mouvement simplifié spécifique pour le cas d'une rotation 3D. Ce modèle est caractérisé par cinq paramètres plutôt que les huit paramètres de l'homographie, rendant ainsi son estimation plus stable. L'utilisation de ce modèle est donc souhaitable afin de limiter les erreurs

d'alignement. L'estimation des paramètres intrinsèques et de la distance focale des caméras à partir d'homographies est déjà bien expliquée dans la littérature et excède la portée de cette thèse. Par conséquent, elle ne sera pas détaillée dans ce document. Afin d'avoir plus d'information à ce sujet, le lecteur peut se référer au travail de Hartley et Zisserman (2003).

4.2.5 Validation de l'image cible

La recherche d'une image cible valide est primordiale à la réussite de la correction d'une image source. Afin d'être jugée comme valide, l'image cible doit respecter les deux conditions suivantes :

1. La majorité des descripteurs de l'image source doit trouver une correspondance avec ceux de l'image cible et une matrice **H** doit exister (voir les sections 4.2.2 et 4.2.3).
2. Tous les pixels de l'image source identifiés comme manquants par le masque source doivent être présents dans l'image cible une fois sa déformation réalisée.

Il est important de se rappeler que les images cibles candidates proviennent de la séquence vidéo source et que, par conséquent, elles peuvent également contenir des zones à corriger identifiées par le masque source. Une fois l'image cible déformée (en fonction de la matrice **H)**, il est donc nécessaire de valider qu'il n'y a pas de chevauchement entre la zone à remplacer de l'image source (carré bleu et jaune dans la figure 4.4) et celle de l'image cible (carré rouge dans la figure 4.4). La figure 4.4 montre des cas de validation d'images cibles (63, 112 et 183) pour une même image source (0). L'image cible 63 est un exemple qui ne respecte pas la deuxième condition puisque la zone à corriger de l'image source (identifiée en bleu et jaune) chevauche une portion manquante de l'image cible candidate identifiée par le masque cible (rouge). L'image cible 183 est un autre exemple qui ne respecte pas la deuxième condition compte tenu que la zone à corriger de l'image source se trouve à l'extérieur de l'image cible. Finalement, l'image cible 112 est un exemple du respect des deux conditions citées préalablement.

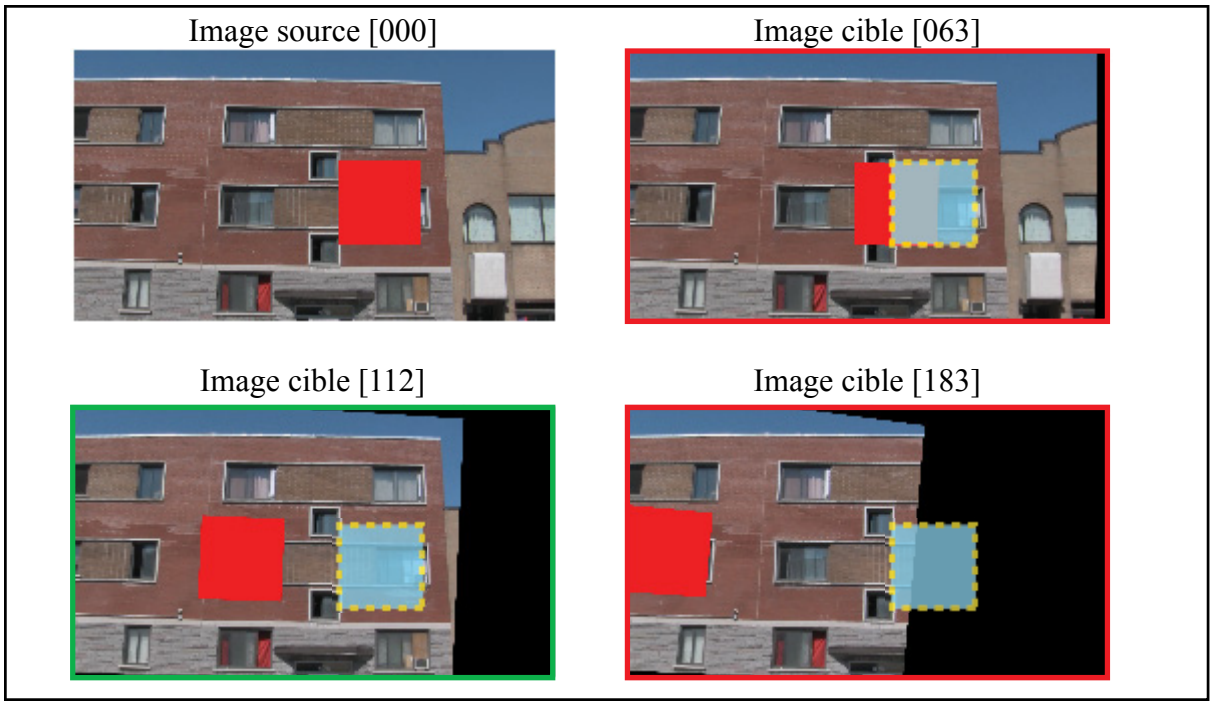


Figure 4.4 Validation de l'image cible.

4.2.6 Accélération de la recherche par une méthode de cohérence

La recherche d'une image cible pour chaque image source n'est pas triviale et représente une portion considérable du temps de calcul nécessaire à la correction d'une séquence vidéo. L'approche simple, mais peu efficace, consiste à tester, tour à tour, toutes les images cibles de la séquence vidéo source jusqu'à l'identification d'une image cible valide. Évidemment, cette façon de faire implique que plusieurs calculs inutiles devront être effectués sur les images non valides. Il est donc important de minimiser le nombre d'images cibles testées pour chaque image source afin de réduire le temps de calcul nécessaire à la correction d'une séquence vidéo.

Les images cibles à proximité de l'image source (N-2, N-1, N+1, N+2, etc.) ont généralement de meilleures chances d'obtenir un taux élevé de correspondance puisque les transformations sont assez subtiles d'une image à l'autre, excepté dans le cas d'un mouvement de caméra très brusque. Il est cependant difficile de tirer avantage de cette observation puisque, de la même

façon, la zone à corriger de l'image source est généralement présente dans les images cibles à proximité, les rendant ainsi non valides (voir la condition 2 de la section 4.2.5).

Le système proposé accélère plutôt la recherche en se basant sur l'information recueillie lors de la correction des images sources précédentes. En effet, le système commence par tester la validité des images cibles trouvées pour les trois images sources précédentes. Si une correspondance est trouvée à ce moment, la recherche se termine et l'image source est corrigée. Dans le cas contraire, le système teste la validité des images voisines de l'image source en débutant par des images rapprochées (N-1, N+1, N-2, N+2, N-4, N+4) puis en espaçant plus rapidement l'intervalle d'échantillonnage (N-8, N+8, N-16, N+16, N-32, N+32, etc.).

Cette méthode de recherche hybride permet de réduire considérablement le temps de calcul nécessaire à la recherche d'images cibles valides, rendant ainsi utilisable la méthode proposée dans le pipeline de création d'un artiste.

4.3 Correction de l'image source à partir de l'image cible

Une fois l'association entre les images sources et leur images cibles terminée, la prochaine étape consiste à corriger les régions manquantes de la séquence vidéo source. Cette étape inclut la déformation des images, l'identification des pixels qui contribueront à la composition finale et le mélange de ces pixels de manière à minimiser le nombre d'artéfacts visibles. Cette section présente les techniques utilisées par le système proposé pour réaliser ces tâches.

4.3.1 Déformation des images

Dans un premier temps, le système utilise les paramètres de la caméra, déterminés préalablement (voir section 4.2.4), afin de déformer l'image cible de façon à ce que les différentes caractéristiques des deux images soient alignées. Cette étape requiert tout d'abord la sélection d'une surface de composition (planaire, cylindrique, sphérique, etc.). Puisqu'il n'y a que deux images, l'approche intuitive est de convertir directement l'image cible vers le système de coordonnées de l'image source (composition planaire). Cependant, puisque l'image source et l'image cible peuvent être assez éloignées dans la séquence vidéo source, la sélection d'une surface de composition planaire peut fortement déformer (étirer) l'image cible. Ceci est d'autant plus vrai pour les pixels situés en bordure de l'image. Évidemment, ces pixels *étirés* engendreront des artéfacts visibles lors de la composition finale. Par conséquent, le système proposé utilise plutôt une projection sphérique (Szeliski et Shum, 1997) afin de limiter ces artéfacts indésirables.

Par la suite, il est nécessaire de choisir la section de la composition qui sera centrée dans la vue finale. Puisque le but de cette opération est de corriger l'image source, cette dernière est utilisée comme référence. Une fois la surface de composition et la vue finale choisies, le système proposé détermine la valeur de chacun des pixels de la vue finale en utilisant une technique de lancer de rayon classique. Après cette étape, les deux images ont été déformées, mais le mélange n'a pas encore été effectué. La figure 4.5 montre un exemple de déformation pour une image source et une image cible.

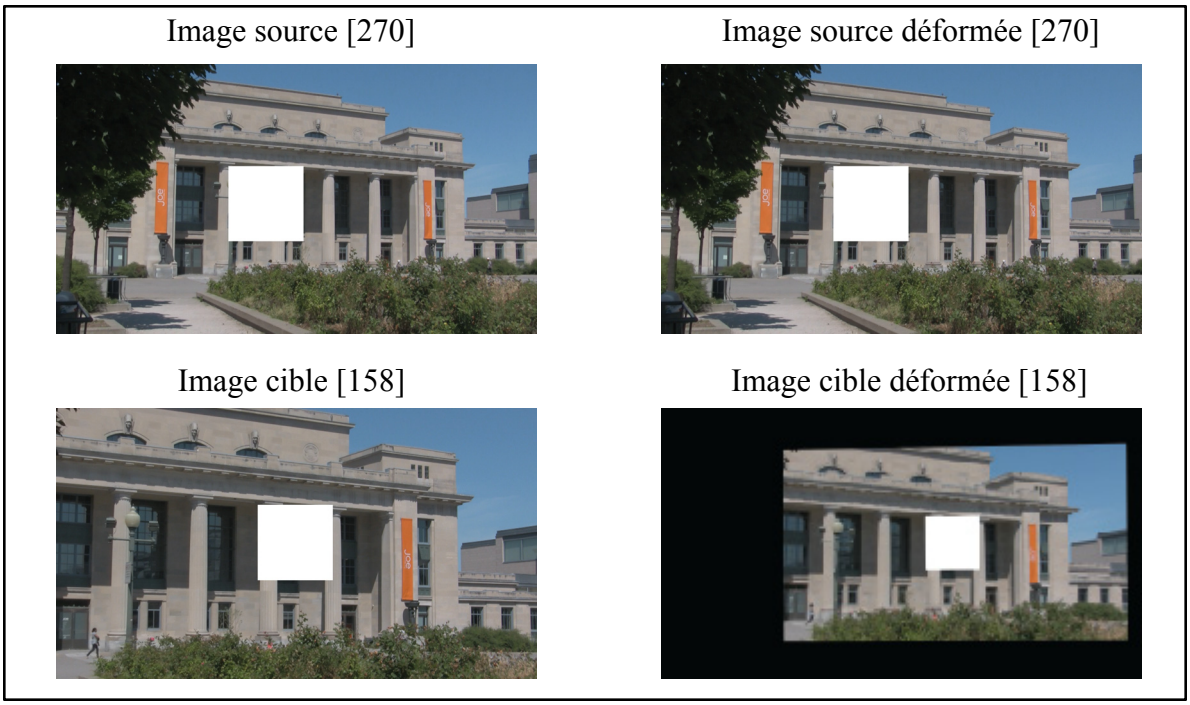


Figure 4.5 Déformation des images source et cible.

4.3.2 Sélection des régions

Lorsque les images source et cible ont été déformées vers la même surface de composition, il est ensuite nécessaire de choisir quelles régions de chacune des images sera conservées dans le mélange final. Dans un monde idéal où toutes les images sont parfaitement alignées et où il n'y a pas de variation d'exposition, cette tâche serait triviale puisque tous les choix de mélange (en omettant les régions manquantes) donneraient le résultat optimal. Malheureusement, la majorité des séquences contiennent des variations d'exposition et plusieurs paires d'images source-cible ne sont pas parfaitement alignées créant ainsi des artéfacts visibles (coutures, fantômes, etc.) dans la séquence vidéo nettoyée. Par conséquent, il est nécessaire de faire une correction des différences d'exposition (section 4.3.3) et de choisir une technique de mélange appropriée (section 4.3.4). Préalablement à ces étapes, le système doit faire la sélection des régions de chacune des images qui contribueront au résultat final.

Évidemment, l'image source est celle qui contribuera le plus à la composition finale. Tous les pixels seront pris en compte, mis à part les pixels manquants identifiés par le masque cible et ceux situés à proximité. Pour y arriver, le système effectue une opération morphologique (dilatation) sur le masque cible afin d'élargir cette région d'environ 40 pixels. Cette étape est importante afin de s'assurer qu'aucun pixel identifié par le masque cible contribuera au mélange final. Inversement, cette région manquante dilatée de l'image source sera celle qui sera conservée de l'image cible déformée. La figure 4.6 montre un exemple de sélection des régions des images source et cible à considérer pour le mélange final. Elle met en évidence que la région non-considérée (carré noir) du masque source déformée est plus grande que la portion manquante de l'image source suite à la dilatation. Un contour rouge a été ajouté au masque cible déformé afin de bien illustrer la position de l'image cible déformée. Celui-ci ne fait cependant pas partie du masque cible déformé à proprement dit.

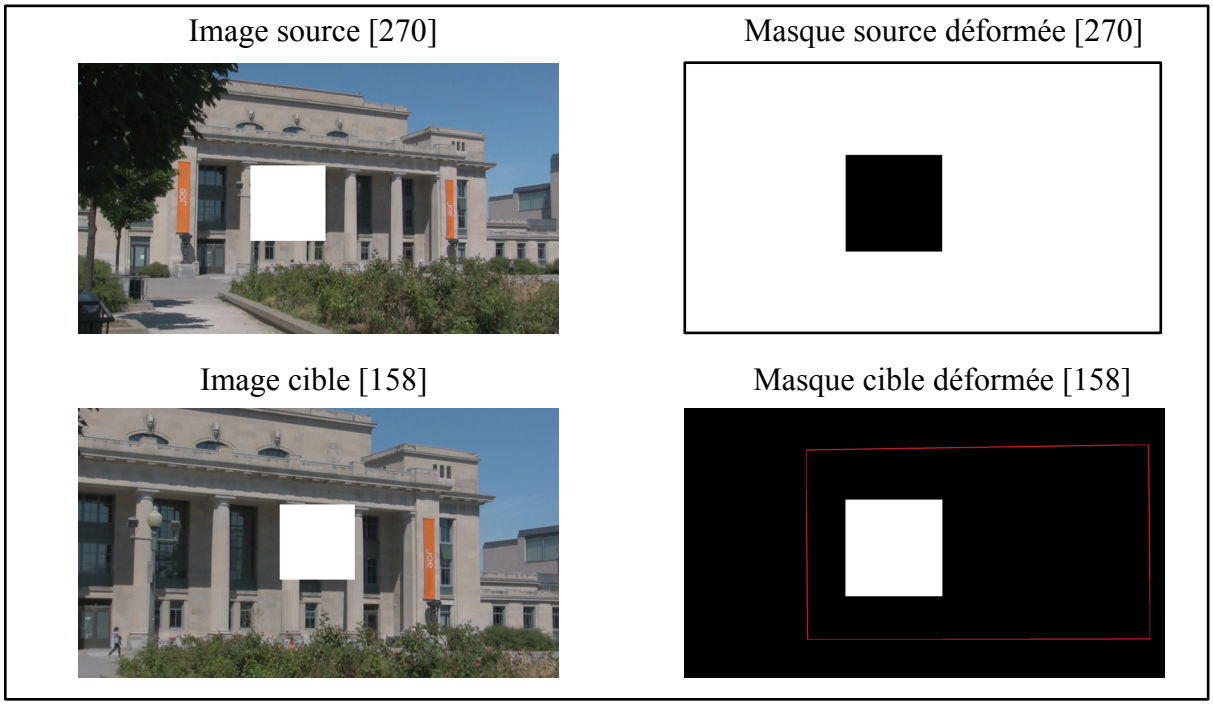


Figure 4.6 Déformation des masques source et cible.

4.3.3 Correction des différences d'exposition

À l'intérieur d'une même séquence vidéo, il arrive fréquemment d'observer une variation d'intensité pour différentes images puisque les caméras changent automatiquement leurs paramètres d'exposition. Par conséquent, même si l'image source et l'image cible sont parfaitement alignées, il arrive qu'une *couture* puisse être visible aux limites des régions à mélanger des deux images. Pour remédier à ce problème, il est nécessaire d'appliquer une technique de correction des différences d'exposition à l'image cible avant d'effectuer le mélange final. Pour y arriver, l'approche proposée ajuste l'exposition de l'image cible en utilisant une technique de spécification d'histogramme (Gonzalez et Woods, 2002). L'idée est de calculer la transformation de couleur à appliquer afin de faire correspondre à l'histogramme des pixels situés en bordure de la zone à remplacer de l'image source celui de la même région de l'image cible déformée. Une fois cette transformation de couleur obtenue, elle est appliquée à toute la région de l'image cible déformée qui est conservée pour la composition finale. La figure 4.7 montre un exemple de correction d'une image source sans la correction préalable des différences d'exposition (gauche) et avec l'utilisation de la méthode de Brown et Lowe (2007) (droite). La différence d'intensité entre les pixels des images source et cible est facilement remarquable dans l'image de gauche.

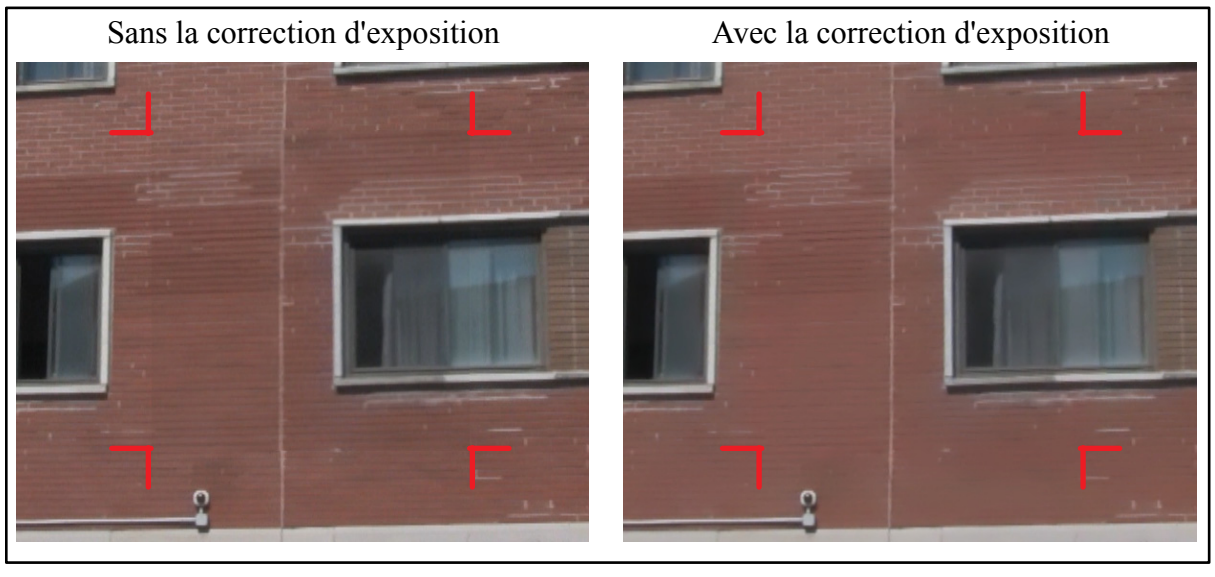


Figure 4.7 Correction des différences d'exposition.

4.3.4 Seconde validation de l'image déformée basée sur PSNR

Tel que mentionné préalablement, il arrive à l'occasion qu'il y ait de petites erreurs lors de l'alignement de l'image cible et de l'image source. Celles-ci entraînent des artéfacts visibles (ex. des « fantômes ») lors de la composition finale des images. Afin de minimiser ces erreurs, l'approche proposée introduit une seconde validation basée sur la métrique PSNR qui permet d'identifier les alignements problématiques et de rejeter l'image cible candidate le cas échéant.

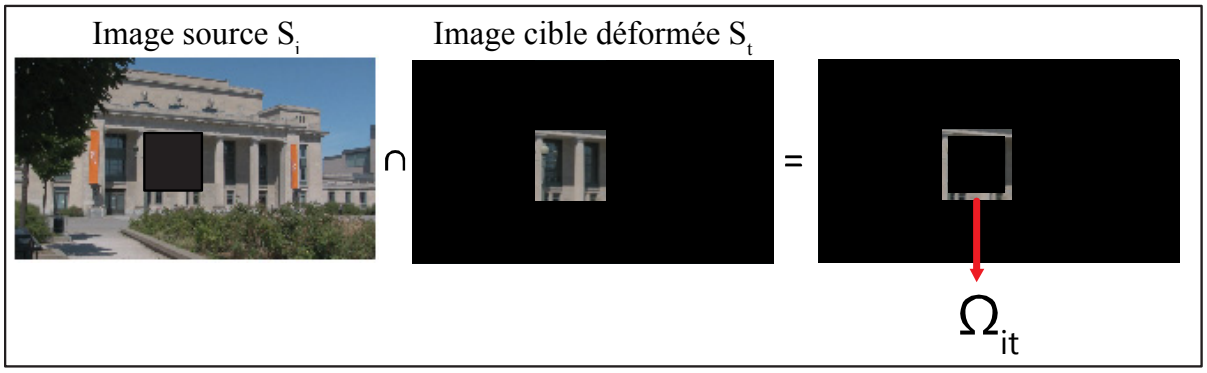


Figure 4.8 Identification de la région Ω_{it}.

Pour chaque paire d'image source et d'image cible déformée, l'approche calcule la valeur $PSNR_{it}$ de la région Ω_{it} (voir la figure 4.8) en fonction de l'équation 4.1

$$PSNR_{it} = 10\log_{10}\cdot(MAX_Y) - 10\log_{10}\cdot(MSE_{it}) \tag{4.1}$$

avec

$$MSE_{it} = \frac{}{|\Omega_{it}|}\sum_{p\in\Omega_{it}}(Y(S_i(p)) - Y(S_t(p)))^2 \tag{4.2}$$

où $|\Omega_{it}|$ est le nombre de pixels dans la région Ω_{it},, Y() retourne la composante Y (luminance) du modèle de couleur YCbCr et MAX_Y représente la luminance maximum (255).

Si la valeur $PSNR_{it}$ (calculé en décibel) est inférieure à un certain seuil, l'image cible déformée est considérée invalide. Si aucune image cible déformée obtient une valeur $PSNR_{it}$ supérieure au seuil, l'image cible déformée ayant obtenu la valeur $PSNR_{it}$ la plus élevée est utilisée pour corriger l'image source. La métrique PSNR est régulièrement utilisée dans le domaine de la compression vidéo et le seuil de 40 décibels est souvent utilisée (Huynh-Thu et Ghanbari, 2008). Des expérimentations empiriques ont également permis de confirmer la validité de cette valeur. Par conséquent, le seuil PSNR a été fixé à 40 décibels pour toutes résultats obtenus.

4.3.5 Mélanges des images avec une approche multi-bandes

Lorsque la correction des différences d'exposition et la seconde validation sont terminées, le système peut alors mélanger les images source et cible afin de générer l'image finale nettoyée. Tel que mentionné précédemment, cette étape serait simple dans un monde idéal puisque tous les pixels qui se superposeraient auraient la même intensité. Or, même suite à la correction des différences d'exposition, il demeure certaines variations d'intensité puisque, par exemple, les images ne sont pas toujours parfaitement alignées. Il est donc primordial de choisir une bonne méthode de mélange.

Le système proposé utilise une technique de mélange multi-bandes basée sur les travaux présentés par Burt et Adelson (1983). Le principe de celle-ci consiste à mélanger les basses fréquences sur de grandes régions spatiales alors que le mélange des hautes fréquences se fait sur de petites régions spatiales. Ceci permet d'avoir un résultat plus lisse tout en gardant les détails plus fins et en éliminant certains *fantômes*. Pour y arriver, une pyramide Laplacienne des images source et cible est générée. Chaque *bande* est par la suite multipliée par une fonction de pondération différente pour chacun des niveaux de la pyramide. Ces fonctions de pondérations sont générées par la construction d'une pyramide de Gaussienne en utilisant le masque source déformé et le masque cible déformé (voir section 4.3.2). Chaque image de la pyramide Laplacienne est donc multipliée par l'image correspondante de la pyramide de Gaussienne. Les résultats obtenus sont par la suite combinés pour créer l'image nettoyée

finale. La figure 4.9 montre un exemple de mélange final des images source et cible sans traitement particulier (gauche) et avec l'utilisation de la méthode de mélange multi-bandes de Burt et Adelson (1983) (droite). Dans l'image de gauche, des artéfacts sont entre autres visibles sur le cadre à droite de la fenêtre, à la bordure de la région à remplacer.

4.4 Résultats

Cette section présente des résultats obtenus avec l'approche de correction de régions manquantes basée sur le suivi de caractéristiques invariantes présentée dans ce chapitre. La figure 4.10 présente une séquence vidéo dans laquelle un cycliste en mouvement a été supprimé. Cette séquence comporte une rotation constante de la caméra du début à la fin.

La figure 4.11 présente quant à elle les résultats de la séquence « CanadianPacific » qui est caractérisée par rotation de la caméra et une grande mise à l'échelle. Puisque l'objet à supprimer dans la séquence « CanadianPacific » a été ajouté de façon synthétique, il est possible de comparer le résultat de la complétion (séquence nettoyée) avec la séquence avant l'ajout de l'objet synthétique (séquence propre). La figure 4.14 présente la comparaison des séquences propre et nettoyée pour « CanadianPacific ». De son côté, la figure 4.12 montre les résultats de la correction de la séquence « Logement » qui est constituée d'une rotation et d'une légère mise à l'échelle. La figure 4.15 présente la comparaison des séquences propre et nettoyée pour « Logement ». La figure 4.13 présente les résultats de la séquence vidéo « Roulement » durant laquelle la caméra subit un roulement marqué vers la droite. Finalement, le tableau 4.1 montre des statistiques des complétions des séquences vidéo présentées dans ce chapitre et celles avec la méthode de Newson *et al.* (2014). La prochaine section discute des avantages et des inconvénients de l'approche proposée.

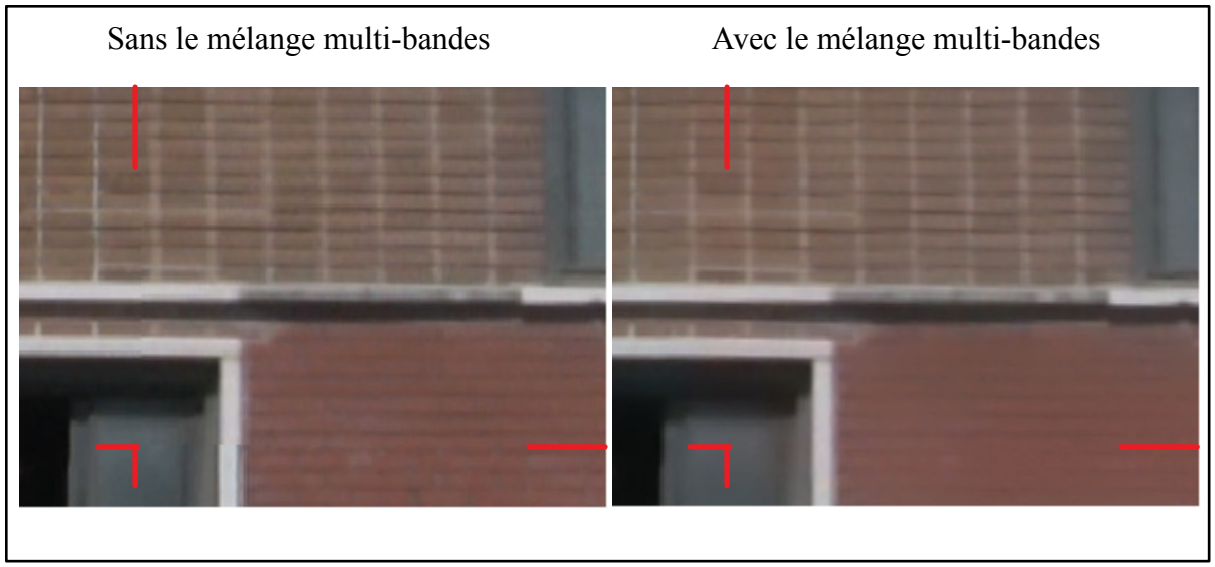
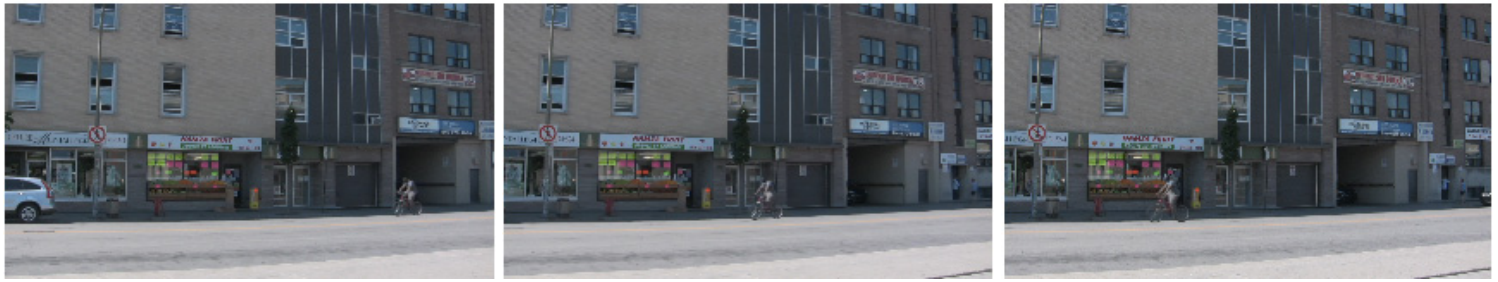
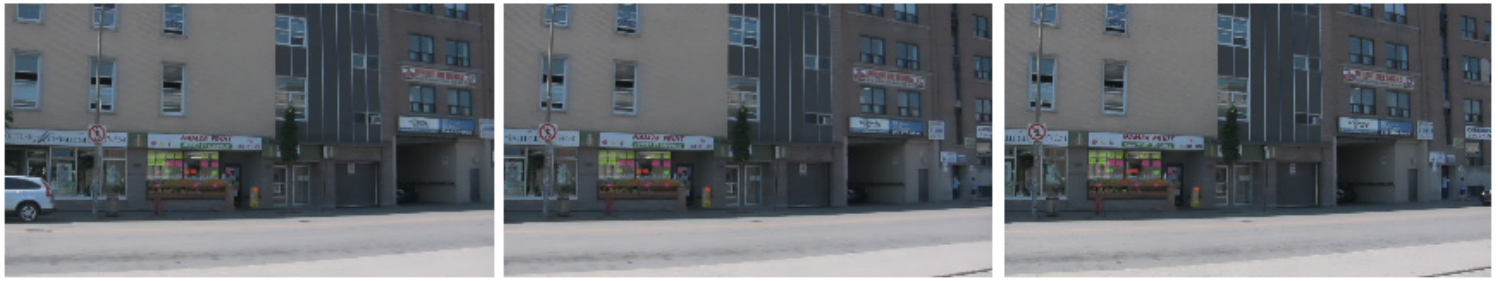


Figure 4.9 Mélange des images avec une approche multi-bandes.



Séquence vidéo originale

Masque source



Séquence vidéo nettoyée

Mouvement de la caméra :	**Résolution :**	**Région manquante :**	**Temps de calcul :**
rotation	1 920 x 1 080	moy. : 83 337 pixels / image	~ 2.5 s. / image

Figure 4.10 Résultats pour la séquence vidéo « Cafe ».

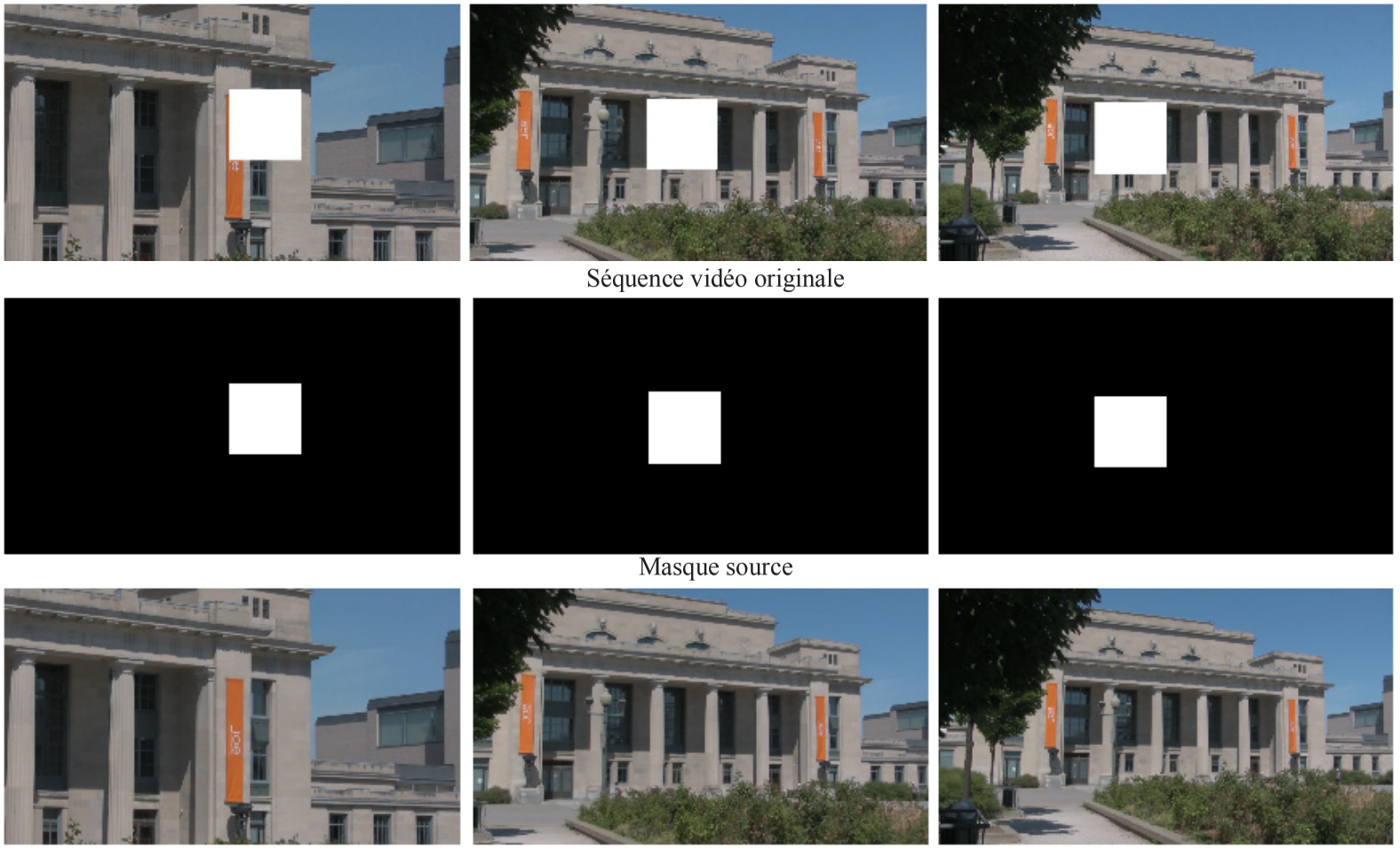
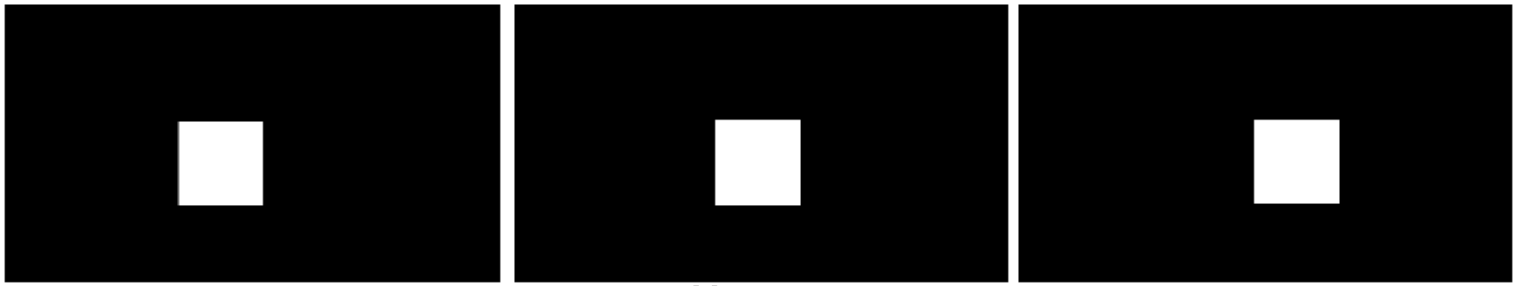


Figure 4.11 Résultats pour la séquence vidéo « CanadianPacific ».

Séquence vidéo originale

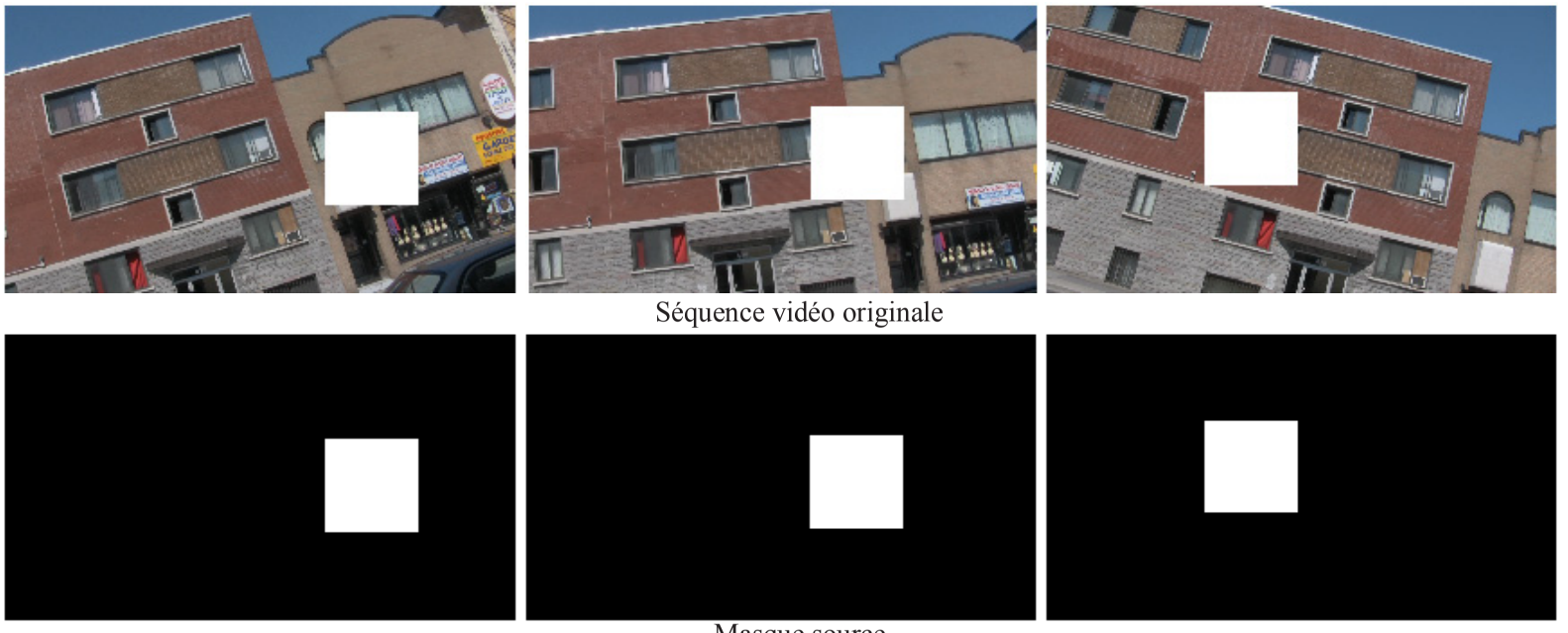


Masque source

Séquence vidéo nettoyée

Mouvement de la caméra :	**Résolution :**	**Région manquante :**	**Temps de calcul :**
mise à l'échelle et rotation	1920 x 1080	moy. : 109 561 pixels / image	~ 2.5 s. / image

Figure 4.12 Résultats pour la séquence vidéo « Logement ».



Séquence vidéo originale

Masque source

Séquence vidéo nettoyée

Mouvement de la caméra :	**Résolution :**	**Région manquante :**	**Temps de calcul :**
Roulement	1920 x 1080	moy. : 121 104 pixels / image	~ 2.5 s. / image

Figure 4.13 Résultats pour la séquence vidéo « Roulement ».

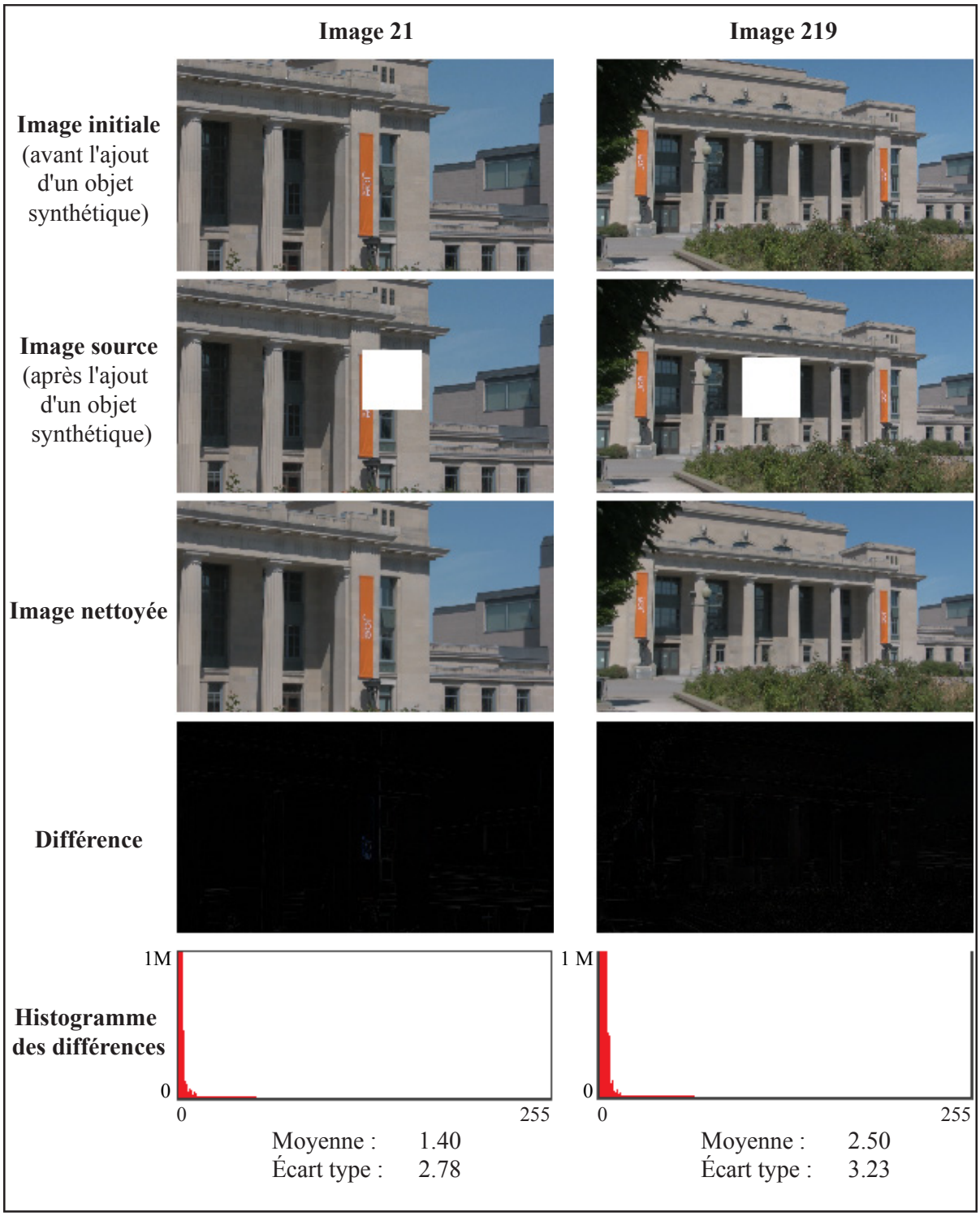


Figure 4.14 Comparaison de la séquence vidéo « CanadianPacific ».

Figure 4.15 Comparaison de la séquence vidéo « Logement ».

Tableau 4.1 Données sur les performances de l'approche de complétion vidéo

Nom	Cadres	Résolution	Durée	Pixels manquants	Temps (m.) Newson (2014)	Temps (m.) Benoit (2015)	Facteur accé.
Logement 01		1920 x 1080	240	26 293 310	470,6	13	36,2
	000-112	1920 x 1080	113	12 379 728	252,9		
	113-239	1920 x 1080	127	13 913 582	217,7		
Cafe 001		1920 x 1080	97	797 639	147,4	2,7	54,6
Cafe 02		1920 x 1080	298	49 362 182	646,1	22,1	29,2
	000-100	1920 x 1080	101	16 729 736	195,4		
	101-201	1920 x 1080	101	16 730 549	249,6		
	202-297	1920 x 1080	96	15 901 897	201,1		
CP 006		1920 x 1080	306	27 813 536	604,7	25,1	24,1
	000-102	1920 x 1080	103	9 362 276	253,6		
	103-206	1920 x 1080	104	9 453 185	216		
	207-305	1920 x 1080	99	8 998 075	135,1		
CP 007		1920 x 1080	121	16 473 265	306,6	10,2	30,1
CP 008		1920 x 1080	241	33 519 268	788,5	21,8	36,2
	000-120	1920 x 1080	121	16 829 012	274,8		
	121-241	1920 x 1080	121	16 690 256	513,7		

4.5 Discussion

Cette section analyse et interprète les résultats obtenus avec l'approche de correction de régions manquantes basée sur le suivi de caractéristiques invariantes présentée dans ce chapitre.

4.5.1 Avantages

Le processus de correction de régions manquantes basé sur le suivi de caractéristiques invariantes présenté dans ce chapitre permet de résoudre un problème majeur des approches de correction de séquences vidéo. Il permet de traiter des séquences vidéo de haute définition et permet, de surcroît, de corriger de très grandes régions manquantes. En effet, puisque le

processus ne requiert aucune structure de recherche, contrairement aux méthodes classiques de correction vidéo telles que Newson *et al.* (2014) ou Ebdelli, Guillemot et Le Meur (2012), l'approche proposée n'est pas limitée par la taille des séquences vidéo. De plus, puisque la méthode proposée n'utilise pas une technique de remplissage *pixel par pixel* ou *patch par patch*, elle est en mesure de compléter de très grandes régions manquantes comparativement à l'état de l'art. La taille des régions n'a pas d'impact sur le temps nécessaire pour compléter l'image ou sur la qualité de la complétion contrairement à Newson *et al.* (2014). Les figures 4.10 à 4.13 montrent d'ailleurs des exemples où de très grandes régions ont été corrigées (jusqu'à 121 000 pixels par image). Le processus proposé est donc adapté à un contexte réel de production où la résolution des séquences vidéo à corriger est de plus en plus grande.

Par ailleurs, puisque le système proposé utilise le suivi de caractéristiques invariantes (SURF), il est très robuste face aux différents types de mouvement de caméra présents dans les séquences vidéo qu'il est en mesure de compléter. En effet, les résultats présentés par les figures 4.10 à 4.13 montrent que le système développé est robuste face aux rotations marquées, aux grandes mises à l'échelle et au roulement. L'utilisation des caractéristiques invariantes permet également au système proposé de se démarquer de l'état de l'art puisqu'il est en mesure de corriger des régions manquantes dont l'information ne se retrouve pas dans le reste de la séquence vidéo source avec exactement la même orientation et le même facteur de mise à l'échelle.

De plus, l'approche proposée permet de compléter des séquences vidéo HD dans un temps plus court que ceux que l'on retrouve dans l'état de l'art. Le tableau 4.1 compare les temps nécessaires pour l'approche de Newson *et al.* (2014) et celle proposée dans ce chapitre. On remarque des facteurs d'accélération qui se situent entre 24.1 et 54.6 comparativement à Newson *et al.* (2014). Il est également important de noter que l'approche de Newson *et al.* (2014) n'est pas en mesure de compléter les séquences vidéo HD qui dépassent 120 cadres puisqu'elle nécessite trop de mémoire (voir section 3.5.3 pour l'explication).

Finalement, le système proposé est bien adapté au pipeline de production et au contexte d'utilisation de l'artiste. D'une part, l'artiste n'a pas à manipuler ou à spécifier de paramètres complexes; il doit simplement fournir la séquence vidéo source et identifier la zone à remplacer via le masque source. D'autre part, le temps de calcul plus que raisonnable (quelques secondes par image) nécessaire à l'édition d'une séquence vidéo rend le système proposé utilisable dans un contexte réel de production. Pour un artiste, la correction de régions manquantes contenues dans une séquence vidéo est une tâche répétitive et machinale qui requiert peu de talent artistique. Avec l'utilisation de l'approche proposée, l'artiste peut s'afférer à des tâches plus créatives. Le tableau 4.2 synthétise la comparaison de l'approche proposée avec les travaux récents.

Tableau 4.2 Comparaison de l'approche proposée avec l'état de l'art

Auteurs	**HD?**	**Résolution maximum**	***Nb. cadres maximum***	**Caméras complexes?**	**Grandes régions manquantes?**
Benoit et Paquette (2015)	Oui	1920 x 1080	250	Limité	Supérieures
Benoit et Paquette (2016)	Oui	1920 x 1080	306	Oui	Oui
Newson *et al.* (2014)	Limité	1120 x 754	200	Limité	Supérieures
Ebdelli *et al.* (2015)	Limité	1440 x 1056	180	Limité	Oui
Newson *et al.* (2013)	Non	1120 x 754	200	Non	Supérieures
Daisy *et al.* (2015)	Non	960 x 544	106	Non	Non
Xu *et al.* (2015)	Non	960 x 540	93	Limité	Supérieures
Herling et Broll (2014)	Non	640 x 320	?	Limité	Non
Zarif, Faye et Rohaya (2013)	Non	640 x 480	250	Non	Non
Mosleh *et al.* (2012)	Non	320 x 240	ND	Non	Non
Vijay Venkatesh *et al.* (2009)	Non	320 x 240	140	Non	Non
Koochari et Soryani (2010)	Non	320 x 240	105	Non	Non
Xiao *et al.* (2011)	Non	320 x 130	150	Non	Non

4.5.2 Limitations

Bien que l'approche proposée de correction de séquence vidéo basée sur le suivi de caractéristiques invariantes remplisse les objectifs fixés au départ, elle possède néanmoins certaines limitations. Premièrement, puisque le système cherche et déforme une région complète afin de corriger une région manquante, les résultats obtenus pour les zones stochastiques (par exemple des vagues ou des branches d'arbre en mouvement) sont peu concluants. En effet, les petits mouvements dans la zone remplacée ne sont pas exactement les mêmes d'une image à l'autre. Ces variations sont facilement perceptibles par l'œil et l'audience remarque rapidement la zone qui a été altérée. De plus, l'approche proposée ne fonctionne pas bien pour les séquences vidéo contenant des textures sans contours nets puisqu'il est plus difficile de trouver des points d'intérêt valides à suivre d'une image à l'autre. Le travail de Ebdelli, Guillemot et Le Meur (2012) et spécialement celui de Newson *et al.* (2014), qui utilisent des algorithmes itératifs de complétion, obtiennent de meilleurs résultats dans ces conditions. Afin de régler ces problèmes, il serait intéressant de combiner l'approche de complétion vidéo présentée au chapitre 3 et celle du présent chapitre.

Deuxièmement, l'approche proposée n'est pas en mesure de corriger convenablement les séquences vidéo lorsqu'un objet en mouvement se trouve à l'intérieur de la zone à remplacer. Prenons l'exemple d'une séquence vidéo où un camion en mouvement cache momentanément un piéton marchant sur le trottoir opposé. Si l'on désire retirer le camion dans la séquence vidéo corrigée, l'approche proposée ne sera pas en mesure de recréer correctement le mouvement du piéton puisqu'elle se base uniquement sur les caractéristiques qui sont à l'extérieur de la zone à remplacer (le camion dans ce cas-ci) et ne tient pas compte de l'information des images précédentes et suivantes de la séquence vidéo. Ceci dit, la majorité des techniques d'édition de séquences vidéo ne sont pas en mesure de recréer les objets en mouvement, sauf dans le cas spécifique de mouvements cycliques tel qu'expliqué au chapitre 1.

Troisièmement, puisque l'approche présentée se base sur une homographie pour déterminer les paramètres de la caméra et déformer l'image cible, le système est sensible à la validité de la matrice d'homographie. Or, la matrice d'homographie n'a pas été en mesure de déformer correctement l'image cible pour 0,0025 % des images sources (deux images sur une total de 794 images pour les quatre séquences présentées à la section 4.4) ce qui a entraîné des erreurs lors de la validation de l'image cible. La figure 4.16 présente un exemple de cette situation. Pour l'instant, le système proposé ne détecte pas automatiquement ces cas et l'artiste doit manuellement corriger ces images lorsque le traitement est terminé. Dans des recherches futures, il serait intéressant de pousser plus loin la validation de l'image cible; favoriser les images cibles avec les plus petites déformations par rapport à l'image source, choisir les images cibles ayant le moins de différences d'exposition ou sélectionner celles ayant le plus de cohérence avec l'image précédente.

Figure 4.16 Erreur de validation de l'image cible.

Finalement, puisque le système n'utilise qu'une seule image cible pour compléter l'image source, il est impératif que la région à remplacer soit entièrement présente dans une même image cible afin que le résultat de la complétion soit valide. Or, il y a des situations où

l’ensemble de l’information manquante d’une région à remplacer ne se trouve pas dans une seule image cible, mais se trouve cependant en combinant plusieurs images cibles différentes. Dans des recherches futures, il serait intéressant d’étudier la possibilité de pouvoir compléter une même image source en se basant sur plusieurs images cibles et ainsi pouvoir compléter un plus grand ensemble de séquences vidéo.

CONCLUSION

À notre époque, la majorité des séquences vidéo filmées pour le cinéma ou la télévision doivent être altérées durant l'étape de postproduction afin d'y ajouter des objets manquants ou bien remplacer des régions indésirables. En effet, le coût et la disponibilité des acteurs, des lieux de tournage, du matériel et de l'équipe technique font en sorte que le temps alloué pour les tournages est de plus en plus limité, augmentant ainsi les chances qu'un oubli ou qu'une erreur soit présent dans la séquence vidéo filmée. De plus, la nécessité d'avoir recours à des trucages et le désir d'incorporer des éléments irréels ou imaginaires sont d'autres facteurs expliquant le besoin de modifier les séquences vidéo à posteriori. Afin que l'audience puisse croire en l'illusion que celles-ci soient réelles, la retouche ou l'édition de séquences vidéo doit être imperceptible. Lors de l'ajout d'un objet, il est donc important que celui-ci ait une apparence réaliste et qu'il présente de l'usure ou des effets de détérioration. Or, la création et l'ajout de ces effets ne sont pas des tâches triviales. Tel qu'expliqué dans cette thèse, les méthodes de l'état de l'art traitant de ce problème comme les simulations basées sur la physique ou les simulations basées sur des paramètres empiriques ne sont pas adaptées puisqu'elles utilisent des paramètres peu intuitifs pour l'artiste. De plus, la majorité des techniques sont généralement conçues pour un seul effet de détérioration ce qui implique qu'un artiste doit utiliser et apprendre plusieurs systèmes différents. Lors du retrait d'une région indésirable, il est tout aussi important que le résultat soit imperceptible. Malheureusement, les travaux antérieurs ne sont pas adaptés au contexte actuel de production puisqu'ils ne sont pas en mesure de traiter les séquences vidéo de haute résolution, maintenant devenues monnaie courante. De plus, la majorité des techniques sont limitées par le type de mouvement de caméra et la taille des régions manquantes qu'elles sont en mesure de traiter. Par conséquent, autant pour l'ajout d'un objet que pour la suppression d'une région manquante, le manque d'une technique efficace fait en sorte que l'artiste réalise généralement ces tâches manuellement. En tenant compte de ce fait, les objectifs de cette thèse étaient de concevoir un système de simulations d'effets de détérioration et un système de remplissage automatique et efficace de régions manquantes dans une séquence vidéo haute définition qui étaient intuitifs pour les artistes et adaptés aux pipelines de production.

La première section de cette thèse reposait sur le développement d'un système de création d'effet de détérioration pour des objets de synthèse basé sur une image échantillon contenant un exemple de l'effet voulu. L'acquisition de cette image échantillon peut être réalisée avec une simple caméra et ne requiert pas un mécanisme complexe de capture d'images. Le système de création automatique présenté propose un processus novateur de synthèse de textures qui tient compte du contexte détérioré et non-détérioré de la texture à remplir afin d'augmenter le réalisme des résultats. De plus, un nouvel ordre de remplissage par remplissage de trou permet de diminuer les discontinuités en bordures des zones à remplir. Le système ne comporte aucun paramètre à manipuler par l'artiste le rendant ainsi très facile d'utilisation. L'allure finale de l'effet de détérioration est contrôlée par le masque cible, une simple image que l'artiste peut facilement créer. Si l'artiste n'est pas satisfait du résultat initial, il peut simplement modifier le masque cible afin d'obtenir une nouvelle version, rendant ainsi le système proposé adapté au processus itératif de création de l'artiste. Les résultats obtenus avec le système montrent qu'il permet la création d'une vaste gamme d'effets de détérioration. Les temps de calcul nécessaires pour obtenir ces résultats demeurent minimes. Bref, le système de création d'effet de détérioration proposé répond aux besoins de l'artiste et s'intègre facilement dans le pipeline de création préconisé par les studios de production.

La deuxième section de cette thèse consistait à la conception d'une approche de remplissage de régions manquantes d'une séquence vidéo de haute définition. Le processus d'édition de séquence vidéo proposé se fonde sur une approche de synthèse de textures basée sur les champs aléatoires de Markov qui ne comporte aucun paramètre à manipuler de la part de l'artiste. Ce dernier doit simplement identifier la région indésirable par la création du masque cible. La majorité des techniques de l'état de l'art nécessite la création d'une structure de recherche coûteuse en espace mémoire et en temps de calcul limitant leur utilisation sur des séquences de plus haute résolution. L'approche proposée réduit plutôt l'espace de recherche en introduisant un processus de remplissage novateur qui se base sur le principe de la cohérence des recherches effectuées à basse résolution et sur une recherche locale pour compléter les séquences HD. Les résultats obtenus montrent que l'approche proposée est en

mesure de traiter des séquences vidéo de haute définition et de corriger de plus grandes régions manquantes. Le système proposé est extensible puisqu'il est indépendant de la mesure de similarité choisie. Aussi, les résultats des recherches sont robustes puisque les calculs ne sont pas effectués sur des données compressées qui pourraient entraîner des erreurs d'approximation.

La dernière section de cette thèse consistait à la conception d'une approche d'édition de régions manquantes d'une séquence vidéo de haute définition comportant des mouvements de caméras non-triviaux et de très grandes régions à remplacer. Le processus d'édition de séquence vidéo proposé se base sur le suivi de caractéristiques invariantes afin de compléter les régions manquantes. Par conséquent, le processus ne requiert aucune structure de recherche et n'est donc pas limité par la taille des séquences vidéo. De plus, les résultats obtenus avec la technique proposée montrent qu'elle est en mesure de compléter de très grandes régions manquantes. L'utilisation du suivi des caractéristiques invariantes implique également que la méthode proposée est robuste vis-à-vis des mouvements de caméras non-triviaux comme la rotation, la mise à l'échelle et le roulement. L'approche proposée introduit également une étape innovante d'atténuation des différences d'exposition basée sur la spécification d'histogramme permettant de rendre moins perceptibles les corrections d'exposition apportées aux séquences vidéo. Aussi, la validation de l'image cible à l'aide d'une métrique basée sur PSNR et l'utilisation du mélange multi-bandes augmentent également la qualité des résultats. Finalement, le système proposé est bien adapté au pipeline de création des studios de production puisque l'artiste n'a pas à manipuler des paramètres complexes et que les temps de calcul sont raisonnables.

Les systèmes qui ont été développés dans cette thèse fournissent de nouveaux outils pour l'édition et la retouche de séquences vidéo de haute définition. Ils ont été conçus en respectant les critères de réalisme, de facilité d'utilisation, de polyvalence et d'efficacité. Ils répondent également aux besoins de l'artiste puisqu'ils s'intègrent dans le pipeline de création des studios de production. Voici un résumé des contributions associées à chacune des trois sections :

Système de création d'effet de détérioration :

- Pipeline novateur basé sur une image échantillon
- Nouvelle fonction d'énergie spécifique au contexte de détérioration
- Remplissage par trou optimisé
- Définitions de nouvelles fenêtres de recherche

Remplissage vidéo à l'aide d'une recherche locale :

- Approche de complétion hybride *inpainting*-échantillonage
- Processus de raffinement basé sur la cohérence des recherches
- Approche de complétion basée sur une recherche locale

Remplissage vidéo à l'aide du suivi des caractéristiques invariantes :

- Approche de complétion basée sur le suivi de caractéristiques invariantes
- Technique de correction des différences d'exposition
- Validation de l'image cible avec une métrique basée sur PSNR
- Composition finale avec une approche multi-bandes.

Les solutions proposées dans cette thèse résolvent seulement quelques problèmes dans l'ensemble de ceux touchant la retouche et l'édition de séquences vidéo. Évidemment, plusieurs autres pistes de recherche sont toujours à explorer. Tout d'abord, il serait intéressant de modifier le système de création d'effets de détérioration afin que l'artiste puisse spécifier le degré de dégradation plutôt que de simplement indiquer l'état détérioré ou non-détérioré de chacun des pixels dans le masque cible. De cette façon, l'artiste aurait un meilleur contrôle sur l'aspect final des résultats obtenus. Pour y parvenir, le masque cible définit comme une image binaire pourrait être modifié de façon à avoir des tons de gris représentant le degré de dégradation plutôt que de représenter uniquement le contexte détérioré ou non-détérioré de l'effet. Dans le même ordre d'idées, il serait intéressant de pouvoir générer automatiquement des phénomènes d'usure qui modifient la géométrie 3D de l'objet en plus de la texture 2D.

De plus, il serait intéressant de modifier l'approche de correction de régions manquantes d'une séquence vidéo basée sur le suivi de caractéristiques invariantes de façon à lui permettre de reconstituer de grandes régions manquantes qui ne sont pas nécessairement statiques (exemple : mouvement de vague, déplacement des branches d'un arbres, etc.). Pour y arriver, les approches présentées aux chapitres 3 et 4 pourraient être combinées. Une fois l'image cible trouvée et déformée selon les paramètres de la caméra, la région manquante de l'image source pourrait être corrigée avec le processus itératif de complétion à l'aide d'une recherche locale en se basant uniquement sur l'image cible déformée. Ceci permettrait de compléter des séquences vidéo qui contiennent des régions stochastiques.

Finalement, il a été soulevé que l'évaluation quantitative et objective de la qualité visuelle des résultats obtenus est difficile dans le domaine de la complétion vidéo puisqu'aucune métrique objective fiable et qu'aucun ensemble de données (séquences vidéo) standardisé existe. Il serait donc intéressant de régler ces problèmes, ce qui serait des contributions scientifiques plus qu'intéressante pour le domaine. Une approche qui pourrait être envisagée pour valider cette métrique consisterait à sonder qualitativement l'opinion d'un ensemble de spectateurs quant à la qualité de différentes complétions vidéo et à vérifier s'il existe une corrélation entre ces opinions et les valeurs obtenues à l'aide de la métrique proposée.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Arya, Sunil, David M. Mount, Nathan S. Netanyahu, Ruth Silverman et Angela Y. Wu. 1998. « An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions ». *Journal of the ACM,* vol. 45, nº 6, p. 891-923.

Ashikhmin, Michael 2001. « Synthesizing Natural Textures ». In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*. p. 217-226 New York, NY, USA: ACM.

Barnes, Connelly, E. Shechtman, Dan B Goldman et Adam Finkelstein. 2010. « The generalized patchmatch correspondence algorithm ». In *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III*. p. 29-43. Heraklion, Crete, Greece: Springer-Verlag.

Barnes, Connelly, Eli Shechtman, Adam Finkelstein et Dan B Goldman. 2009. « PatchMatch: a randomized correspondence algorithm for structural image editing ». *ACM Transactions on Graphics,* vol. 28, nº 3, p. 1-11.

Bay, Herbert, Andreas Ess, Tinne Tuytelaars et Luc Van Gool. 2008. « Speeded-Up Robust Features (SURF) ». *Computer Vision and Image Understanding,* vol. 110, nº 3, p. 346-359.

Bellini, Rachele, Yanir Kleiman et Daniel Cohen-Or. 2016. « Time-varying weathering in texture space ». *ACM Transactions on Graphics,* vol. 35, nº 4 (July 2016).

Benoit, Jocelyn, et Eric Paquette. 2015. « Localized search for high definition video completion ». *Journal of WSCG,* vol. 23, nº 1, p. 45-54.

Benoit, Jocelyn, et Eric Paquette. 2016. « Fast High-Definition Video Background Completion using Features Tracking ». In *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)* (Oct. 24-27, 2016). Phuket, Thailand: IEEE Computer Society.

Bertalmio, Marcelo, Andrea L. Bertozzi et Guillermo Sapiro. 2001. « Navier-Stokes, Fluid Dynamics, and Image and Video inpainting ». In *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1, p. 355-362. New York, NY, USA IEEE Computer Society.

Bertalmio, Marcelo, Guillermo Sapiro, Vincent Caselles et Coloma Ballester. 2000. « Image Inpainting ». In *Proceedings of the 27th annual Conference on Computer Graphics and Interactive Techniques*. p. 417–424. New York (NY): ACM Press / Addison-Wesley Publishing Co.

Bézin, Richard, Benoît Crespin, Xavier Skapin, Olivier Terraz et Philippe Meseure. 2014. « Generalized Maps for Erosion and Sedimentation Simulation ». *Computer and Graphics,* vol. 45, nº C, p. 1-16.

Bosch, Carles, Pierre-Yves Laffont, Holly Rushmeier, Julie Dorsey et George Drettakis. 2011. « Image-Guided Weathering: A new Approach Applied to Flow Phenomena ». *ACM Transactions on Graphics,* vol. 30, nº 3.

Boykov, Yury, Olga Veksler et Ramin Zabih. 2001. « Fast Approximate Energy Minimization Via Graph Cuts ». *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, nº 11, p. 1222-1239.

Brown, Matthew, et David G. Lowe. 2007. « Automatic Panoramic Image Stitching using Invariant Features ». *International Journal of Computer Vision,* vol. 74, nº 1, p. 59-73.

Burt, Peter J., et Edward H. Adelson. 1983. « The Laplacian pyramid as a compact image code ». In *Readings in computer vision: issues, problems, principles, and paradigms*, sous la dir. de Martin, A. Fischler, et Firschein Oscar. p. 671-679. Morgan Kaufmann Publishers Inc.

Chang, Yao-Xun, et Zen-Chung Shih. 2001. « Physically-Based Patination for Underground Objects ». *Computer Graphics Forum,* vol. 19, nº 3, p. 109-117.

Chang, Yao-Xun, et Zen-Chung Shih. 2003. « The Synthesis of Rust in Seawater ». *The Visual Computer,* vol. 19, nº 1, p. 50-66.

Chen, Yanyun, Lin Xia, Tien-Tsin Wong, Xin Tong, Hujun Bao, Baining Guo et Heung-Yeung Shum. 2005. « Visual Simulation of Weathering by Gamma-ton Tracing ». *ACM Transactions on Graphics,* vol. 24, nº 3, p. 1127-1133.

Cheung, Vinvent, Brendan J. Frey et Nebojsa Jojic. 2005. « Video Epitomes ». In *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1, p. 42-49 vol. 1. Los Alamitos, CA: IEEE Computer Society Press.

Chunxiao, Liu, Peng Qunsheng et Wang Xun. 2011. « Recent development in image completion techniques ». In *Int. Conf. on Computer Science and Automation Engineering* (10-12 June 2011). Vol. 4, p. 756-760. Shanghai: IEEE.

Clément, Olivier, Jocelyn Benoit et Eric Paquette. 2007. « Efficient Editing of Aged Object Textures ». In *Proceedings of the 5th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*. Grahamstown, South Africa: ACM.

Criminisi, Antonio, Patrick Perez et Kentaro Toyama. 2003. « Object Removal by Exemplar-Based Inpainting ». In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2, p. 721-728. IEEE Computer Society.

Criminisi, Antonio, Patrick Perez et Kentaro Toyama. 2004. « Region Filling and Object Removal by Exemplar-Based Image Inpainting ». *IEEE Transactions on Image Processing,* vol. 13, nº 9, p. 1200-1212.

Daisy, Maxime , Pierre Buyssens, David Tschumperlé et Olivier Lézoray. 2015. « Exemplar-based Video Completion with Geometry-guided Space-time Patch Blending ». In *SIGGRAPH Asia 2015 Technical Briefs*. Kobe, Japan: ACM.

Dorsey, Julie, Alan Edelman, Henrik Wann Jensen, Justin Legakis et Hans Kohling Pedersen. 1999. « Modeling and Rendering of Weathered Stone ». In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. p. 225-234. New York (NY): ACM Press / Addison-Wesley Publishing Co.

Dorsey, Julie, et Pat Hanrahan. 1996. « Modeling and Rendering of Metallic Patinas ». In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. p. 387-396. New York (NY): ACM Press.

Dorsey, Julie, Hans Kohling Pedersen et Pat Hanrahan. 1996. « Flow and Changes in Appearance ». In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. New York (NY): ACM Press.

Drori, Iddo, Daniel Cohen-Or et Hezy Yeshurun. 2003. « Fragment-Based Image Completion ». *ACM Transactions on Graphics,* vol. 22, nº 3, p. 303-312.

Ebdelli, M. , O. Le Meur et C. Guillemot. 2015. « Video Inpainting With Short-Term Windows: Application to Object Removal and Error Concealment ». *IEEE Transactions on Image Processing,* vol. 24, nº 10 (Oct. 2015), p. 3034-3047.

Ebdelli, M., C. Guillemot et O. Le Meur. 2012. « Examplar-based video inpainting with motion-compensated neighbor embedding ». In *Proceedings of the 19th IEEE International Conference on Image Processing* (Sept. 30 2012-Oct. 3 2012). p. 1737-1740. Orlando, FL IEEE Computer Society.

Efros, Alexei A. , et William T. Freeman. 2001. « Image Quilting for Texture Synthesis and Transfer ». In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. p. 341-346. New York, NY, USA: ACM.

Efros, Alexei A. , et Thomas K. Leung. 1999. « Texture Synthesis by Non-Parametric Sampling ». In *Proceedings of the International Conference on Computer Vision*. Vol. 2, p. 1033-1038 Kerkyra IEEE Computer Society.

Endo, Yuki, Yoshihiro Kanamori, Jun Mitani et Yukio Fukui. 2010. « An Interactive Design System for Water Flow Stains on Outdoor Images ». In *Smart Graphics: 10th International Symposium on Smart Graphics* (June 24-26, 2010). p. 160-171. Banff, Canada: Springer Berlin Heidelberg.

Fischler, Martin A., et Robert C. Bolles. 1981. « Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography ». *Communications of the ACM* vol. 24, nº 6, p. 381-395.

Glondu, Loeiz, Maud Marchal et Georges Dumont. 2013. « Real-Time Simulation of Brittle Fracture Using Modal Analysis ». *IEEE Transactions on Visualization and Computer Graphics,* vol. 19, nº 2, p. 201-209.

Gonzalez, Rafael C., et Richard E. Woods. 2002. *Digital Image Processing*, 2nd ed. Upper Saddle River (NJ): Prentice Hall, 822 p.

Gossow, David, Peter Decker et Dietrich Paulus. 2011. « An Evaluation of Open Source SURF Implementations ». In *RoboCup 2010*, sous la dir. de Ruiz-del-Solar, Javier, Eric Chown et Paul G. Plöger. Gossow2011. p. 169-179. Springer-Verlag.

Granados, M., J. Tompkin, K. Kim, O. Grau, J. Kautz et C. Theobalt. 2012. « How Not to Be Seen - Object Removal from Videos of Crowded Scenes ». *Computer Graphics Forum,* vol. 31, nº 2, p. 219-228.

Gu, Jinwei, Chien-I Tu, Ravi Ramamoorthi, Peter Belhumeur, Wojciech Matusik et Shree Nayar. 2006. « Time-Varying Surface Appearance: Acquisition, Modeling and Rendering ». *ACM Transactions on Graphics,* vol. 25, nº 3, p. 762-771.

Hartley, Richard, et Andrew Zisserman. 2003. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 700 p.

Hays, James, et Alexei A. Efros. 2008. « Scene Completion using Millions of Photographs ». *Communications of the ACM* vol. 51, nº 10, p. 87-94.

Heckbert, Paul Seagrave. 1986. « Survey of texture mapping ». *IEEE Computer Graphics and Applications* vol. 6, nº 11, p. 56-67.

Herling, J., et W. Broll. 2014. « High-Quality Real-Time Video Inpaintingwith PixMix ». *IEEE Transactions on Visualization and Computer Graphics,* vol. 20, nº 6, p. 866-879.

Hertzmann, Aaron , Charles E. Jacobs, Nuria Oliver, Brian Curless et David H. Salesin. 2001. « Image Analogies ». In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. p. 327-340. ACM.

Hsu, Siu-chi, et Tien-Tsin Wong. 1995. « Simulating Dust Accumulation ». *IEEE Computer Graphics and Applications,* vol. 15, nº 1, p. 18-22.

Huynh-Thu, Q. , et M. Ghanbari. 2008. « Scope of validity of PSNR in image/video quality assessment ». *Electronics Letters,* vol. 44, nº 13 (June 19 200), p. 800-801.

Iben, Hayley, et James O'Brien. 2009. « Generating Surface Crack Patterns ». *Graphical Models* vol. 71, nº 6 (November, 2009), p. 198-208.

Jensen, H. W. 1996. « Global illumination using photon maps ». In., p. 21-30. Coll. « Rendering Techniques '96. Proceedings of the Eurographics Workshop. Eurographics ». Wein, Austria: Springer-Verlag.

Jia, Jiaya, Yu-Wing Tai, Tai-Pang Wu et Chi-Keung Tang. 2006. « Video Repairing under Variable Illumination using Cyclic Motions ». *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, nº 5, p. 832-839.

Jia, Jiaya, Tai-Pang Wu, Yu-Wing Tai et Chi-Keung Tang. 2004. « Video Repairing: Inference of Foreground and Background under Severe Occlusion ». In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1, p. 364-371. IEEE Computer Society.

Kider, Josseph, Samantha Raja et Norman Badler. 2011. « Fruit Senescence and Decay Simulation ». *Computer Graphics Forum,* vol. 30, nº 2 (April 2011), p. 257-266.

Kokaram, A. C., B. Collis et S. Robinson. 2005. « Automated rig removal with Bayesian motion interpolation ». In *Proceeding of the IEEE Conference on Vision, Image and Signal Processing*. Vol. 152, p. 407-414. IET

Kokaram, Anil. 2004. « Practical, Unified, Motion and Missing Data Treatment in Degraded Video ». *Journal of Mathematical Imaging and Vision* vol. 20, nº 1, p. 163-177.

Koochari, Abbas, et Mohsen Soryani. 2010. « Exemplar-based Video Inpainting with Large Patches ». *Journal of Zhejiang University - Science C,* vol. 11, nº 4, p. 270-277.

Kumar, Neeraj, Li Zhang et Shree Nayar. 2008. « What Is a Good Nearest Neighbors Algorithm for Finding Similar Patches in Images? ». In *Proceedings of the 10th European Conference on Computer Vision: Part II*. p. 364-378. Marseille, France: Springer-Verlag.

Kwatra, Vivek , Arno Schödl, Irfan Essa, Greg Turk et Aaron Bobick. 2003. « Graphcut Textures: Image and Video Synthesis using Graph Cuts ». *ACM Transactions on Graphics,* vol. 22, nº 3, p. 277-286.

Lefebvre, Sylvain, et Hugues Hoppe. 2006. « Appearance-Space Texture Synthesis ». *ACM Transactions on Graphics,* vol. 25, nº 3, p. 541-548.

Liang, Lin, Ce Liu, Ying-Qing Xu, Baining Guo et Heung-Yeung Shum. 2001. « Real-Time Texture Synthesis by Patch-Based Sampling ». *ACM Transactions on Graphics,* vol. 20, nº 3, p. 127-150.

Lischinski, Dani, Zeev Farbman, Matt Uyttendaele et Richard Szeliski. 2006. « Interactive Local Adjustment of Tonal Values ». *ACM Transactions on Graphics,* vol. 25, nº 3, p. 646-653.

Lowe, David G. 1999. « Object Recognition from Local Scale-Invariant Features ». In *Proceedings of the International Conference on Computer Vision*. Vol. 2, p. 1150-1157. Kerkyra: IEEE Computer Society.

Lu, Jianye, Athinodoros S. Georghiades, Andreas Glaser, Hongzhi Wu, Li-Yi Wei, Baining Guo, Julie Dorsey et Holly Rushmeier. 2007. « Context-Aware Textures ». *ACM Transactions on Graphics,* vol. 26, nº 1, p. 3.

Mérillou, N., S. Mérillou, D. Ghazanfarpour, J. M. Dischler et M. Galin. 2010. « Simulating Atmospheric Pollution Weathering on Buildings ». In *18th International Conference in Central Europe on Computer Graphics (WSCG)* (February 1-4, 2010). p. 65-72. Pilsen, Czech Republic.

Mérillou, Stéphane, Jean-Michel Dischler et Djamchid Ghazanfarpour. 2001a. « Corrosion: Simulating and Rendering ». In *Proc. of Graphics Interface*. p. 167-174. Toronto, Canada: Canadian Information Processing Society.

Mérillou, Stéphane, Jean-Michel Dischler et Djamchid Ghazanfarpour. 2001b. « Surfaces Scratches: Measuring, Modeling and Rendering ». *The Visual Computer,* vol. 17, nº 1, p. 30-45.

Mérillou, Stéphane, et Djamchid Ghazanfarpour. 2008. « Technical Section : A Survey of Aging and Weathering Phenomena in Computer Graphics ». *Computers & Graphics,* vol. 32, nº 2, p. 159-174.

Mosleh, A., N. Bouguila et A. Ben Hamza. 2012. « Video Completion Using Bandlet Transform ». *IEEE Transactions on Multimedia,* vol. 14, nº 6, p. 1591-1601.

Muja, Marius, et David G. Lowe. 2009. « Fast approximate nearest neighbors with automatic algorithm configuration ». In *Proceeding of International Conference on Computer Vision Theory Applications VISAPP*. p. 331-340. INSTICC Press.

Newson, Alasdair, Andrés Almansa, Matthieu Fradet, Yann Gousseau et Patrick Pérez. 2013. « Towards fast, generic video inpainting ». In *Proceedings of the 10th European Conference on Visual Media Production*. p. 1-8. London, United Kingdom: ACM.

Newson, Alasdair, Andrés Almansa, Matthieu Fradet, Yann Gousseau et Patrick Pérez. 2014. « Video Inpainting of Complex Scenes ». *SIAM Journal of Imaging Science,* vol. 7, nº 4, p. 1993-2019.

Paquette, Eric, Pierre Poulin et George Drettakis. 2001. « Surface aging by impacts ». In *Proceedings of Graphics Interface 2001*. p. 175-182. Ottawa, Ontario, Canada: Canadian Information Processing Society.

Paquette, Eric, Pierre Poulin et George Drettakis. 2002. « The Simulation of Paint Cracking and Peeling ». In *Proceedings of Graphics Interface 2002 Conference* p. 59-68.

Patwardhan, Kedar A., Guillermo Sapiro et Marcelo Bertalmio. 2005. « Video Inpainting of Occluding and Occluded Objects ». In *Proceedings of the IEEE International Conference on Image Processing*. Vol. 2, p. 69-72. IEEE Computer Society.

Patwardhan, Kedar A., Guillermo Sapiro et Marcelo Bertalmio. 2007. « Video Inpainting Under Constrained Camera Motion ». In *IEEE Transactions on Image Processing* Vol. 16, p. 545-553. 2. IEEE Computer Society.

Pritch, Y., E. Kav-Venaki et S. Peleg. 2009. « Shift-map image editing ». In *Proceedings of the IEEE 12th International Conference on Computer Vision* (Sept. 29 2009-Oct. 2 2009). p. 151-158. Kyoto: IEEE Computer Society.

Shen, Yuping, Fei Lu, Xiaochun Cao et Hassan Foroosh. 2006. « Video Completion for Perspective Camera Under Constrained Motion ». In *Proceedings of the 18th*

International Conference on Pattern Recognition. Vol. 3, p. 63-66. Hong Kong IEEE Computer Society.

Shih, Timothy K. , Nick C. Tang, Wei-Sung Yeh, Ta-Jen Chen et Wonjun Lee. 2006. « Video Inpainting and Implant via Diversified Temporal Continuations ». In *Proceedings of the 14th annual ACM international conference on Multimedia*. Santa Barbara, CA, USA: ACM.

Shiratori, Takaaki, Yasuyuki. Matsushita, Tang Xiaoou et Kang Sing Bing. 2006. « Video Completion by Motion Field Transfer ». In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1, p. 411-418. IEEE Computer Society.

Sun, Jian, Lu Yuan, Jiaya Jia et Heung-Yeung Shum. 2005. « Image Completion With Structure Propagation ». *ACM Transactions on Graphics,* vol. 24, nº 3, p. 861-868.

Szeliski, Richard, et Heung-Yeung Shum. 1997. « Creating full view panoramic image mosaics and environment maps ». In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. p. 251-258. ACM Press/Addison-Wesley Publishing Co.

Ting, Huang, Shifeng Chen, Jianzhuan Liu et Xiaoou Tang. 2007. « Image Inpainting by Global Structure and Texture Propagation ». In *Proceedings of the 15th International Conference on Multimedia*. Augsburg, Germany: ACM.

Tong, Xin, Jingdan Zhang, Ligang Liu, Xi Wang, Baining Guo et Heung-Yeung Shum. 2002. « Synthesis of bidirectional texture functions on arbitrary surfaces ». *ACM Transactions on Graphics,* vol. 21, nº 3, p. 665-672.

Venkatesh, M. Vijay, Sen-ching Samson Cheung et Jian Zhao. 2009. « Efficient Object-based Video Inpainting ». *Pattern Recognition Letters,* vol. 30, nº 2, p. 168-179.

Wang, Chao, Xin Tong, Stephen Lin, Minghao Pan, Chao Wang, Hujun Bao, Baining Guo et Heung-Yeung Shum. 2006. « Appearance Manifolds for Modeling Time-Variant Appearance of Materials ». *ACM Transactions on Graphics,* vol. 25, nº 3, p. 754-761.

Wang, Z., E. P. Simoncelli et A. C. Bovik. 2003. « Multiscale Structural Similarity for Image Quality Assessment ». In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers*. Vol. 2, p. 1398-1402.

Wang, Zhou , A. C. Bovik, H. R. Sheikh et E. P. Simoncelli. 2004. « Image Quality Assessment: from Error Visibility to Structural Similarity ». *IEEE Transactions on Image Processing,* vol. 13, nº 4 (April 2004), p. 600-612.

Wei, Li-Yi , et Marc Levoy. 2000. « Fast texture Synthesis using Tree-Structured Vector Quantization ». In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Co.

Wei, Li-Yi, Sylvain Lefebvre, Vivek Kwatra et Greg Turk. 2009. « State of the Art in Example-based Texture Synthesis ». In *Proceedings of Eurographics* (2009-03-30). p. 93-117. Munich, Germany: Eurographics Association.

Wexler, Yonatan, Eli Shechtman et Michal Irani. 2004. « Space-Time Video Completion ». In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1, p. 120-127. Washington D.C.: IEEE Computer Society.

Wexler, Yonatan, Eli Shechtman et Michal Irani. 2007. « Space-Time Completion of Video ». *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, nº 3, p. 463-476.

Wong, Tien-Tsin, Wai-Yin Ng et Pheng-Ann Heng. 1997. « A Geometry Dependent Texture Generation Framework for Simulating Surface Imperfections ». In *Proceedings of the Eurographics Workshop on Rendering Techniques*. p. 139-150. London (UK): Springer-Verlag.

Xiao, C. X., M. Liu, Y. W. Nie et Z. Dong. 2011. « Fast Exact Nearest Patch Matching for Patch-Based Image Editing and Processing ». *Ieee Transactions on Visualization and Computer Graphics,* vol. 17, nº 8 (Aug), p. 1122-1134.

Xiao, Chunxia , Shu Liu, Hongbo Fu, Chengchun Lin, Chengfang Song, Zhiyong Huang, Fazhi He et Qunsheng Peng. 2008. « Video Completion and Synthesis ». In *Computer Animation and Virtual Worlds* Vol. 19, p. 341-353. 3-4. Chichester, UK: John Wiley and Sons Ltd.

Xu, Z., Q. Zhang, Z. Cao et C. Xiao. 2015. « Video Background Completion Using Motion-guided Pixels Assignment Optimization ». *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. PP, nº 99, p. 1-1.

Xue, Su, Julie Dorsey et Holly Rushmeier. 2011. « Stone Weathering in a Photograph ». In *Proceedings of the Twenty-second Eurographics conference on Rendering* (June 27 - 29, 2011). p. 1189-1196 Prague, Czech Republic: Eurographics Association.

Zarif, S., I. Faye et D. Rohaya. 2013. « Static object removal from video scene using local similarity ». In *2013 IEEE 9th International Colloquium on Signal Processing and its Applications (CSPA),* (8-10 March 2013). p. 54-57.

Zhang, Yunjun , Jiangjian Xiao et Mubarak Shah. 2005. « Motion Layer Based Object Removal in Videos ». In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*. Vol. 1, p. 516-521.