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
Olivier CLÉMENT

SYNTHÈSE D'EFFETS DE DÉTÉRIORATION POUR UN RENDU RÉALISTE

MONTRÉAL, LE 11 JANVIER 2011

PRÉSENTATION DU JURY

CETTE THÈSE A ÉTÉ ÉVALUÉE

PAR UN JURY COMPOSÉ DE :

M. Eric Paquette, directeur de thèse
Département de génie logiciel et des TI à l'École de technologie supérieure

M. Mohamed Cheriet, président du jury
Département de génie de la production automatisée à l'École de technologie supérieure

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

M. Richard Egli, examinateur externe
Faculté des sciences à l'Université de Sherbrooke

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

LE 23 NOVEMBRE 2010

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

AVANT-PROPOS

Droits d'auteur

À plusieurs endroits dans ce document, il a souvent été souhaitable de présenter des images provenant de différents travaux dans le domaine de la synthèse d'images réalistes pour permettre au lecteur de mieux en visualiser et en comparer les résultats. Or, les droits d'auteurs associés à ces documents contraignent généralement la reproduction de telles images. Heureusement, certains journaux et conférences scientifiques permettent la reproduction de leur matériel dans un contexte à but non lucratif, tel que la rédaction d'une thèse. Les notes relatives aux droits d'auteurs sont présentées ci-dessous. De plus, la provenance des images utilisées dans ce document est clairement indiquée au bas des différentes figures présentant des résultats découlant d'autres travaux.

Note relative aux droits d'auteur pour les travaux tirés d'une publication d'ACM :

ACM COPYRIGHT NOTICE. Copyright © 2010 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.

Note relative aux droits d'auteur pour les travaux tirés de *Graphics Interface* avant 2001 :

Copyright © 2001 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

J’aimerais d’abord remercier le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), l’École de technologie supérieure (ÉTS) ainsi que les compagnies Ubisoft et Canadel pour le soutien financier qu’ils m’ont fourni tout au long de mon doctorat. Sans le support de ces diverses bourses d’études, la concrétisation de ce projet de recherche aurait été tout simplement impossible.

Je tiens aussi à remercier personnellement mon directeur de recherche, Eric Paquette, pour son implication soutenue dans le projet, pour ses multiples conseils et recommandations ainsi que pour sa capacité à faire renaître ma motivation au travers des étapes plus pénibles de mes études. Sans son assistance et son expertise, la réalisation du projet aurait été beaucoup plus ardue et la qualité du résultat aurait sans doute été grandement diminuée.

Finalement, j’aimerais remercier tous les membres du laboratoire multimédia et toutes les personnes qui ont participé de près ou de loin à ce projet. Je tiens à mentionner particulièrement Jocelyn Benoit, Martin Poirier et François Dagenais pour les contributions considérables qu’ils ont apportées à diverses étapes du projet.

SYNTHÈSE D'EFFETS DE DÉTÉRIORATION POUR UN RENDU RÉALISTE

Olivier CLÉMENT

RÉSUMÉ

De nos jours, le niveau de réalisme attendu des images synthétiques utilisées dans les jeux vidéo, les films d'animation ou les systèmes de réalité virtuelle a augmenté considérablement. Pour produire des répliques conformes à des situations tirées du monde réel, les systèmes de rendu réaliste doivent considérer une quantité impressionnante de détails. Parmi ces détails, les effets produits par le vieillissement des objets comme de la rouille ou des égratignures sont particulièrement difficiles à traiter et requièrent beaucoup de travail manuel. Les approches existantes pour adresser cette problématique, telles que les simulations basées sur la physique et les simulations basées sur des règles empiriques, ne sont pas adaptées aux artistes puisqu'elles demandent la manipulation de paramètres complexes et sont peu polyvalentes puisqu'elles sont très souvent conçues spécifiquement pour la synthèse d'un type d'effet en particulier.

L'objectif de ce projet de recherche est d'offrir une structure polyvalente permettant de générer de nouveaux effets de détérioration à partir d'une image contenant un exemple de l'effet désiré. Pour y parvenir, un algorithme de synthèse de textures par remplissage de trous a été adapté au contexte pour engendrer de nouveaux effets similaires. À partir d'une recette de détérioration basée sur des propriétés locales comme l'accessibilité et la courbure, l'approche proposée permet à un artiste de définir un patron général caractérisant l'usure d'un objet. Ensuite, cette recette est applicable soit à un objet pour en générer une ou plusieurs occurrences similaires sans être identiques, soit à plusieurs objets distincts pour produire des détériorations comparables. Le dernier volet du projet consiste à altérer le processus préalablement défini pour le rendre indépendant de la couleur de l'exemple fourni en entrée par l'utilisateur. L'approche standard exploitant les canaux de couleurs RVB est remplacée par la synthèse d'un niveau d'intensité de détérioration. Le système doit ensuite interpréter cette intensité à l'aide de points de contrôle définis par l'utilisateur et d'un nuanceur exécuté par l'application de manière à produire une couleur à afficher pour chaque pixel formant les effets de détérioration.

Les résultats obtenus à partir des différentes contributions du projet de recherche sont très encourageants. L'approche proposée produit des résultats d'excellente qualité et permet d'améliorer considérablement le réalisme des images synthétiques. Le processus est très polyvalent puisqu'il fonctionne sur une grande variété d'effets et de matériaux. Son utilisation est adaptée aux artistes et aux studios de production puisque ses paramètres sont simples et intuitifs. Finalement, les temps de calculs nécessaires sont minimes et s'intègrent bien dans un processus de création itératif.

Mots-clés : images synthétiques, rendu réaliste, détérioration, usure, apparence des matériaux

AGING EFFECT SYNTHESIS FOR A REALISTIC RENDERING

Olivier CLÉMENT

ABSTRACT

During the last decade, the level of realism expected for synthetic images used in video games, animated movies, or virtual reality systems has increased considerably. To properly replicate real world situations, realistic rendering systems must consider an impressive amount of detail. Among these details, effects produced by aging such as rust or scratches are particularly hard to handle, require a lot of manual work, and adding them is obviously time-consuming. Existing methods addressing this problem, such as physically-based and empirical simulations, are not suitable for artists since they require the manipulation of complex physical parameters, and are not polyvalent since they are often designed for a specific type of aging effect.

The main objective of this research project is to provide artists with a versatile framework to generate new aging effects based on an image containing an example of the desired effect. To achieve this, a constrained texture synthesis algorithm has been adapted to such a context to produce new similar effects. From an aging recipe based on local properties, such as accessibility and curvature, the proposed approach allows an artist to define a general pattern characterizing the weathering of an object. Then, this recipe can be applied either to an object to generate one or more similar instances, yet not identical, or to several different objects to produce comparable aging. The last constituent of the project is to alter the previously defined process to make it independent of the sample color provided by the user. The standard approach exploiting RGB channels is replaced by the synthesis of an aging intensity. The system then interprets this intensity using control points defined by the user and a custom shader executed by the application to produce a color to display for each pixel forming the aging effects.

The results obtained from the different contributions of this research project are very promising. The proposed approach produces high-quality results, and can greatly improve the realism of synthetic images. The process is very versatile since it works on a wide variety of aging effects and materials. Its use is perfectly suitable for artists and production studios since its parameters are simple and intuitive. Finally, required computation times are minimal and thus fit well in an iterative creation process.

Keywords: synthetic images, realistic rendering, aging, weathering, material appearance

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

BRDF	Bidirectional Reflectance Distribution Function
HSV	Triplet « *Hue* – *Saturation* – *Value* » ou Teinte – Saturation – Luminosité
KLT	Karhunen–Loève Transform
OpenCV	Open Computer Vision
PCA	Principal Component Analysis
RVB	Triplet rouge, vert, bleu

INTRODUCTION

De nos jours, l'infographie est un domaine en effervescence dans lequel les innovations ne cessent de faire évoluer la qualité et le réalisme des images synthétiques. Les studios de productions cinématographiques utilisent de plus en plus d'images générées par ordinateurs pour créer des effets spéciaux impressionnants et vont même jusqu'à produire des films composés entièrement d'images synthétiques. Depuis plusieurs années, cette demande accrue stimule la recherche dans le but de produire des images qui se rapprochent encore plus de la réalité.

Figure 1.1 Exemples illustrant l'évolution du réalisme dans les films d'animation.
Adaptée de Tony de Peltrie, de Toy Story (© Pixar Animation Studios), de Shrek (© DreamWorks) et de Ratatouille (© Pixar Animation Studios)

Comme le montre la figure 1.1, le niveau de réalisme des films d'animation s'est grandement amélioré avec les années. Au départ, les films d'animation n'étaient rien de plus qu'une séquence d'images 2D créées manuellement. Par la suite, des outils ont été développés pour modéliser un environnement 3D et produire un rendu 2D. Au fil des années, la simulation de phénomènes plus complexes a été intégrée aux algorithmes de rendu, tels que des surfaces

réfléchissantes, des objets semi-transparents, des fluides en mouvement ou des cheveux d'apparence naturelle. En parallèle, la performance et la capacité des différentes composantes graphiques de l'ordinateur ont été décuplées avec le temps, ce qui a mené à un niveau de réalisme beaucoup plus intéressant dans des films d'animation contemporains tel que « Ratatouille » par *Pixar Animation Studios*.

Figure 1.2 Exemples illustrant l'évolution du réalisme dans les jeux vidéo.
Adaptée de Pole Position (© Namco), de Duke Nukem 3D (© Apogee Software), de Half Life 2 (© Valve Corporation) et de Halo 3 (© Microsoft Game Studios)

De l'autre côté, les créateurs de l'univers grandissant des jeux vidéo tentent par tous les moyens de faire augmenter le niveau de réalisme de leurs environnements virtuels. Coincé dans un contexte d'application exécutée en temps réel, les développeurs de jeux vidéo n'arrivent évidemment pas à atteindre le même niveau de réalisme que les producteurs de

films d'animation. Tout de même, le réalisme représente une préoccupation majeure de cette industrie, ce qui engendre des améliorations considérables. Cette évolution constante (voir figure 1.2) à comme conséquence principale de faire gonfler les attentes des consommateurs, autant dans le domaine cinématographique que dans le monde du jeu vidéo. À chaque nouvelle production, les clients s'attendent à observer une amélioration du niveau de réalisme et à être éblouis par rapport aux produits précédents.

Mais quels sont les aspects qui influencent le niveau de réalisme d'une image synthétique? En fait, l'usure des objets virtuels représente un des facteurs importants du réalisme d'une image de synthèse. Un reproche souvent fait aux images générées par ordinateur est qu'elles sont « impeccables » ou « trop parfaites ». Inévitablement, les objets du monde réel se détériorent avec le temps alors que les objets synthétiques sont épargnés. En effet, une multitude de phénomènes naturels comme l'oxydation, la moisissure ou tout simplement l'accumulation de poussière viennent modifier l'apparence des objets. Comme si ce n'était pas suffisant, l'être humain peut, lui aussi, détériorer son environnement en brisant, égratignant ou tachant ces mêmes objets. Comme le montre la figure 1.3, les objets présents dans le monde réel sont loin d'être impeccables ou d'avoir une apparence parfaite. En plus, les différents effets de détérioration énoncés plus haut peuvent aussi se combiner les uns les autres pour produire une allure bien particulière.

Figure 1.3 Exemples d'effets de détérioration tirés du monde réel.

Produire ce genre d'apparence sur un objet virtuel représente une tâche longue et pénible, et ce, même pour un artiste expérimenté. Il faut d'abord identifier les effets susceptibles d'avoir altérer l'objet en question pour ensuite construire manuellement les textures appropriées. Malheureusement, le problème ne s'arrête pas là. En effet, ce processus doit être répété pour tous les objets de l'environnement, ce qui multiplie la quantité de travail nécessaire. Prenons l'exemple d'un artiste qui doit modéliser une salle de classe dans laquelle se retrouvent plusieurs instances d'un même bureau. Pour représenter correctement la réalité, chacun des bureaux doit posséder des marques d'usure différentes. Certains bureaux auront été remplacés récemment, d'autres non. Quelques étudiants y auront gravé des inscriptions, d'autres y auront simplement crayonné au plomb. Une gomme à mâcher ici, une tache d'encre par là et le niveau de détails nécessaires pour créer un environnement réaliste grimpe à vue d'oeil. Par conséquent, les producteurs vont, plus souvent qu'autrement, négliger cet

aspect du réalisme dans le but de respecter les budgets qui leur sont alloués. Un compromis fréquemment utilisé dans les jeux vidéo est de réutiliser les mêmes effets de détérioration pour chacune des instances d'un même objet. En reprenant l'exemple énoncé précédemment, cela signifierait que les bureaux de la salle de classe seraient tous usés exactement de la même manière. Un spectateur moindrement attentif remarquera rapidement cette façon de faire, ce qui réduira son degré d'immersion dans l'environnement virtuel. Par ailleurs, avec les technologies actuelles, les applications exécutées en temps réel comme les jeux vidéo doivent souvent composer avec des contraintes de mémoire importantes. Par conséquent, l'utilisation de textures différentes pour les diverses instances d'un même objet peut parfois poser problème.

Au fil des années, plusieurs techniques et outils ont été développés dans le milieu de la recherche pour résoudre ces problèmes. Cependant, seulement une portion minime de ces approches est utilisée concrètement en industrie. Dans ce contexte, il est essentiel d'identifier les raisons de cette situation. Tout d'abord, la majorité des techniques et outils qui adressent cette problématique ne sont pas adaptés pour une utilisation par un artiste. En effet, plusieurs approches de simulation demandent de manipuler une série de paramètres physiques, parfois complexes, pour lesquels il est difficile de prévoir l'impact réel sur l'apparence du résultat final. Très souvent, l'artiste n'a qu'un contrôle partiel sur l'allure produite et doit préconiser une approche par essais et erreurs pour atteindre l'apparence désirée. De plus, chacune des approches proposées se limite souvent à un type d'effets de détérioration bien précis. L'artiste doit donc apprendre à utiliser plusieurs outils, ce qui signifie plusieurs ensembles de paramètres, pour pouvoir générer la totalité des effets désirés.

Par ailleurs, les approches proposées ne prennent pas en considération le processus de création pratiqué dans les studios de production. Comme l'explique la figure 1.4, les concepteurs travaillent selon un processus itératif. L'artiste commence par visualiser l'apparence de l'objet et en construit une première version qui sera révisée, soit par des pairs ou par un directeur artistique.

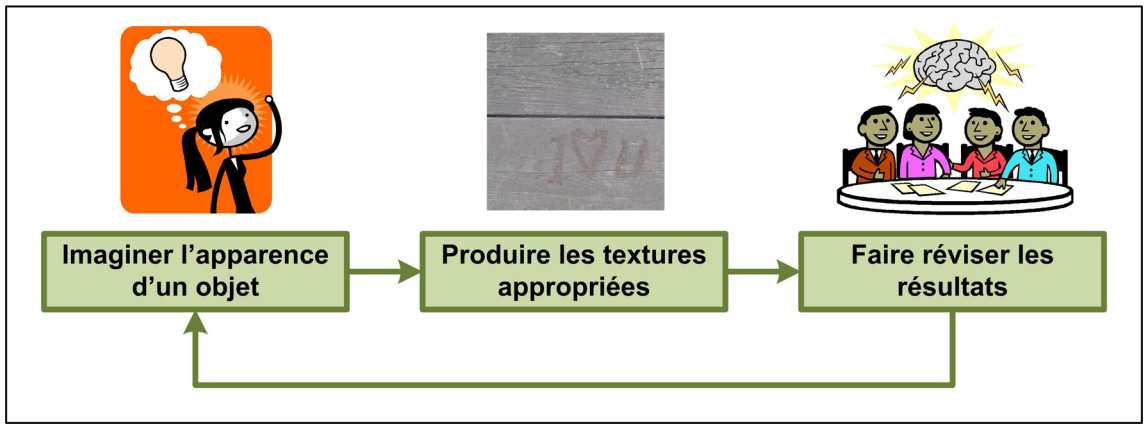

Figure 1.4 Processus de conception itératif en place dans les studios de production.

Pour que ce processus fonctionne correctement, il est important que l'artiste puisse faire hâtivement des modifications sur l'apparence de son objet. Or, certaines approches de simulation proposées demandent des temps de calculs qui s'étalent sur plusieurs jours, ce qui ne répond pas aux attentes des studios de production. Finalement, pour plusieurs techniques, il est impossible de fonctionner dans le contexte d'une application temps réel, limitant ainsi davantage les possibilités d'utilisation concrète en entreprise.

CHAPITRE 1

REVUE DE LA LITTÉRATURE ET OBJECTIFS

Avant de commencer, il est très important d'avoir une vue d'ensemble des travaux antérieurs reliés à la problématique étudiée. Par conséquent, le présent chapitre a pour objectif d'illustrer l'état de l'art dans les domaines du réalisme et des techniques de détériorations, de la synthèse de texture et du rendu interactif. Ces sujets touchent de près les objectifs associés à ce projet de recherche et se doivent donc d'être approfondis. Ce chapitre se termine par un énoncé des objectifs du projet.

1.1 Techniques de détérioration

Le réalisme étant une préoccupation importante dans le monde des images synthétiques, plusieurs chercheurs se sont attaqués aux phénomènes d'usure et de détérioration des objets virtuels. De ce fait, différentes techniques de simulation d'effets de détérioration ont vu le jour. L'ensemble de ces techniques peut être partagé en trois grandes familles : les simulations basées sur la physique, les simulations basées sur des règles empiriques et les simulations basées sur la capture d'images. Il est à noter que le regroupement est donné à titre indicatif seulement, puisque certaines techniques pourraient parfois être classées dans plusieurs de ces familles à la fois. Chacun de ces regroupements sera détaillé et critiqué dans les sections qui suivent. En plus d'être réunies chronologiquement par famille, les techniques seront groupées selon les effets de détérioration qu'elles traitent. Pour une présentation plus détaillée de l'ensemble des méthodes proposées à ce jour, il serait avisé de consulter l'article de Mérillou et Ghazanfarpour (2008) ainsi que l'article de Dorsey et Hanrahan (2000).

1.1.1 Simulations basées sur la physique

À l'origine, les premiers travaux cherchant à simuler des effets de détérioration ont produits des techniques basées sur la physique. L'idée générale derrière cette approche est de reproduire fidèlement les phénomènes naturels qui affectent l'apparence d'un objet à l'aide

d'une simulation physique. Un des premiers exemples de ce type de méthodes a été développé par Dorsey, Pederson et Hanrahan (1996) pour recréer l'effet de l'écoulement d'eau sur l'apparence de la surface d'un objet. Pour y parvenir, ils ont construit un modèle de simulation par particules permettant de représenter l'écoulement, l'absorption et l'accumulation occasionnés par de l'eau se répandant sur un objet. Chacune des particules possède différentes propriétés telles qu'une masse, une position et une certaine vélocité. L'écoulement de ces particules est déterminé à partir de leurs propriétés et de la géométrie de l'objet à user. L'absorption et l'accumulation sont déterminées à partir de l'écoulement et de certaines propriétés reliées à l'environnement comme la quantité de pluie et le niveau d'ensoleillement. La figure 1.1 présente quelques résultats obtenus à l'aide de cette technique.

Figure 1.1 Résultats obtenus en simulant un écoulement d'eau.
Tirée de Dorsey, Pedersen et Hanrahan (1996, p. 419)

D'autres travaux par Dorsey et Hanrahan (1996) s'attardent sur l'oxydation naturelle de certains métaux comme le bronze ou le cuivre. Ce phénomène d'usure résulte principalement d'une série d'étapes de corrosion du métal avec l'atmosphère. L'apparence de la patine métallique dépendra des particules présentes dans l'air. Pour imiter correctement ce processus de transformation, Dorsey et Hanrahan proposent un système par couches évoluant selon trois types de contextes différents : un environnement marin, un environnement rural ou un environnement urbain. Selon le contexte, les étapes d'oxydation ainsi que les particules en présences seront différentes ce qui changera l'apparence de l'objet dans le temps. La figure 1.2 montre le résultat de l'application de cette technique.

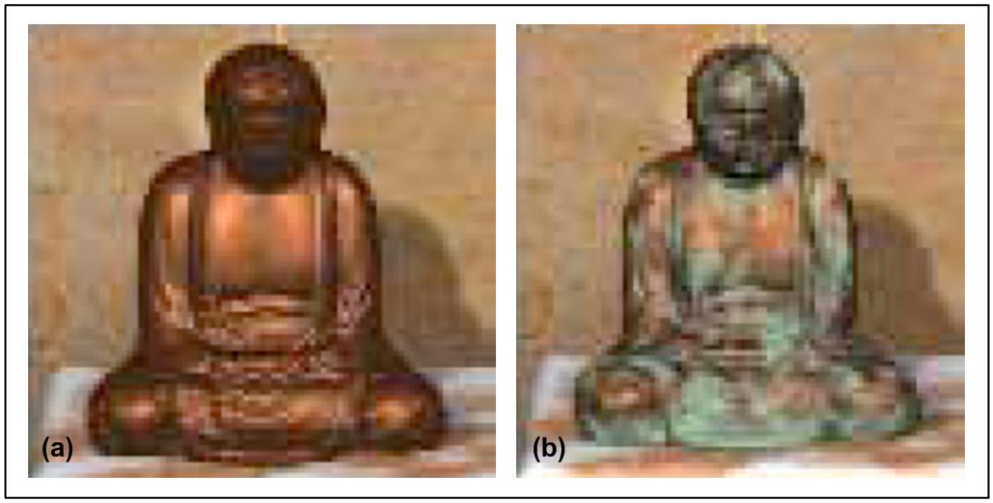

Figure 1.2 Résultats obtenus en simulant une patine métallique.
Adaptée de Dorsey et Hanrahan (1996, p. 395)

Ce phénomène d'oxydation naturelle a aussi été analysé par Chang et Shih (2001). Leur recherche s'adresse spécifiquement à des objets de bronze qui auraient vieillis sous le sol pendant plusieurs années dans le but de représenter correctement d'anciennes statues chinoises. Leur simulation se base sur un modèle qui mesure la tendance à s'oxyder de tous les points d'une surface donnée. Cette tendance est déterminée à partir de la courbure (Turk, 1992) et de l'accessibilité (Miller, 1994) des points de la surface. De plus, certaines caractéristiques du sol comme le niveau d'humidité et le contenu chimique entrent aussi dans le calcul. La distribution de cette tendance est ensuite utilisée pour générer le rendu du

modèle analysé. Avec ce modèle, l'apparence du résultat peut être contrôlée en ajustant des poids associés à chacun de ces paramètres.

Les phénomènes physiques reliés au vieillissement d'objets en pierre ont aussi été étudiés par Dorsey *et al.* (1999). La pierre, étant une substance poreuse, absorbe fortement l'humidité transmise par son environnement. Les minéraux qui forment la pierre vont lentement se dissoudre et se recristalliser en fonction de l'humidité absorbée. Ce processus, répété sur plusieurs années modifiera l'apparence de la pierre en provoquant plusieurs effets comme de l'érosion ou du jaunissement. Pour représenter ce phénomène, la technique proposée introduit une approche par dalles (« *slabs* ») qui modélise la couche externe de l'objet selon une série de quadrilatères. Chacune de ces dalles absorbe une certaine quantité d'humidité déterminée selon un modèle physique et modifie l'apparence de l'objet. Des résultats obtenus avec cette technique sont exposés à la figure 1.3.

Figure 1.3 Résultats obtenus en simulant le vieillissement du marbre.
Tirée de Dorsey *et al.* (1999, p. 233)

La rouille est un autre phénomène d'usure naturelle qui a été étudié par plusieurs chercheurs. Mérillou, Dischler et Ghazanfarpour (2001a) se sont attaqués au problème. Leur technique de simulation propose de représenter l'évolution de l'apparence d'un objet rouillé dans le temps. Pour commencer, une série de points de départ sont identifiés en fonction de certaines propriétés comme des défauts microscopiques ou des points de contacts entre deux objets. Évidemment, ces régions sont plus susceptibles de démarrer le processus d'oxydation. À partir de ces points, une carte de corrosion est calculée pour les points de la surface de l'objet. Cette carte contient plusieurs informations dont la couleur, la porosité, la rugosité et l'élévation de la surface. Selon un processus itératif, les différentes régions rouillées sont étendues suivant un algorithme de progression semi-aléatoire qui prend en considération les propriétés de la surface. Les informations contenues dans la carte de corrosion sont utilisées, à un moment précis de l'évolution, pour faire le rendu de l'objet rouillé. La figure 1.4 montre quelques résultats produits à l'aide de cette technique.

Figure 1.4 Résultats obtenus en simulant la corrosion d'objets métalliques.
Adaptée de Mérillou, Dischler et Ghazanfarpour (2001a, p. 173 et 174)

Chang et Shih (2003) se sont aussi attardés à la simulation d'effet de corrosion en adaptant leur modèle développé pour représenter des patines métalliques (Chang et Shih, 2001). Encore une fois, leur technique de simulation s'adresse à une situation bien spécifique : la corrosion formée sur des objets submergés dans l'eau de mer. Comme dans leur modèle précédent, leur approche se base sur la tendance à rouiller d'un point de surface, calculée à partir de certaines propriétés comme la courbure (Turk, 1992) et l'accessibilité (Miller, 1994). Par contre, les propriétés reliées à l'environnement changent pour représenter fidèlement les caractéristiques de l'eau de mer.

Mérillou, Dischler et Ghazanfarpour (2001b) se sont aussi penchés sur le phénomène d'usure par égratignures. Pour développer leur approche, les auteurs ont d'abord cherché à mesurer des surfaces égratignées provenant du monde réel pour pouvoir en représenter correctement la géométrie. À partir des mesures prises expérimentalement, ils proposent de modéliser une égratignure, d'un point de vue microscopique, selon une forme de « V » définie par des angles $\boldsymbol{\alpha}$ et $\boldsymbol{\beta}$ (angles entrant et sortant) qui varient en fonction du matériau détérioré. Inévitablement, cette géométrie a un effet sur la manière dont la surface réfléchit la lumière. Dans leur technique, Mérillou, Dischler et Ghazanfarpour (2001b) proposent un modèle de réflexion de la lumière spécialisé utilisant les BRDFs (*« Bidirectional Reflectance Distribution Function »*) pour simuler correctement l'apparence des égratignures. Ultérieurement, l'approche proposée a été raffinée par Bosch *et al.* (2004) en incorporant l'utilisant d'un outil pour définir les égratignures et leur géométrie sur un objet.

Plusieurs autres techniques de simulation basées sur des modèles physiques ont été proposées pour adresser des phénomènes d'usure différents. À titre d'exemples, O'Brien, Bargteil et Hodgins (2002) ont construit un modèle pour représenter adéquatement des objets fracturés alors que Aoki *et al.* (2004) ont proposé une approche pour générer des craquelures sur la surface d'un objet, pour ne nommer que ceux-là.

Pour résumer, cette famille de techniques de simulation propose une grande gamme d'approches et couvre une multitude d'effets de détérioration. Dans ce cas, pourquoi les studios de productions éprouvent encore des difficultés à synthétiser des images réalistes qui prennent en considération l'usure naturelle des objets? En fait, bien que ces techniques de simulation produisent des résultats très réalistes, plusieurs défauts qui leurs sont associés font en sorte qu'elles ne sont tout simplement pas utilisées concrètement en entreprise. Comme le montre le tableau 1.1, ces méthodes demandent à l'artiste de manipuler des paramètres physiques complexes pour entreprendre une simulation. Ces paramètres sont loin d'être intuitifs pour un artiste et vont souvent le porter à laisser la technique de côté. De plus, le lien entre les différents paramètres et l'apparence finale de l'objet n'est pas toujours évident. L'artiste doit donc souvent travailler par essais et erreurs pour obtenir l'allure désirée. Puisque les temps de calculs nécessaires à ces simulations sont relativement longs, travailler par essais et erreurs est loin d'être idéal. Par ailleurs, chacune des techniques de simulation adresse un effet de détérioration spécifique. Or, lorsque vient le temps de modéliser un environnement réaliste, tous ces effets de détérioration doivent être combinés. Pour un artiste, apprendre à utiliser l'ensemble de ces techniques représente une quantité de travail invraisemblable dans le contexte d'une entreprise. Pour toutes ces raisons, la grande majorité de ces approches s'intègre avec difficultés dans le processus de création artistique des studios de production, ce qui fait en sorte qu'elles sont plus souvent qu'autrement ignorées.

Tableau 1.1 Bilan des approches de simulation basées sur la physique

Technique	Paramètres	Performance
Dorsey *et al.* (1996) **Écoulement**	Rugosité et absorptivité du matériau Adhésion et solubilité des dépôts Niveaux de pluie et d'ensoleillement	De 20 minutes à 3 heures De 450 à 260000 polygones Processeur de 250 MHz
Dorsey et Hanrahan (1996) **Patine**	Types de particules dans l'air	Données non disponibles
Chang et Shih (2001) **Oxydation**	Propriétés géométriques de l'objet Niveau d'humidité du sol Contenu chimique du sol	Données non disponibles
Dorsey *et al.* (1999) **Vieillissement de pierre**	« Slabs » décomposant l'objet 3D Concentration de sel Perméabilité et porosité du matériau Saturation maximale + 11 autres paramètres	De 3 à 24 heures De 100K à 2,2M de polygones 4 processeurs de 250 MHz
Mérillou *et al.* (2001a) **Corrosion**	Nombre de points de départ Carte d'imperfections structurelles Rugosité, porosité et épaisseur du matériau	Données non disponibles
Chang et Shih (2003) **Corrosion sous-marine**	Propriétés géométriques de l'objet Directions des courants marins Distributions des sels dans l'eau	De 16K à 21K polygones
Mérillou *et al.* (2001b) **Égratignure**	Caractéristiques des égratignures Carte de positionnement	Données non disponibles
O'Brien *et al.* (2002) **Fracture**	Résistance limite du matériau Élasticité du matériau Forces de déformation	Données non disponibles
Aoki *et al.* (2004) **Craquelure de surface**	Taille des cellules d'humidité Forces externes de déformation	De 48 minutes à 25 heures Processeur de 2 GHz

1.1.2 Simulations basées sur des règles empiriques

Bien que les approches basées sur des simulations physiques présentées à la section précédente produisent souvent des résultats de très bonne qualité, elles ne sont pas utilisées concrètement en entreprise. Tel que mentionné préalablement (voir tableau 1.1) ces techniques nécessitent la manipulation de paramètres complexes, ce qui a pour conséquence directe d'effrayer les utilisateurs potentiels, des artistes, et de réduire considérablement leur utilisation. De plus, les temps de calcul des algorithmes de simulations basées sur la physique sont souvent mauvais. Finalement, dans certains cas, le phénomène physique associé aux effets de détérioration est mal compris, ce qui empêche le développement d'une méthode de

simulation. Conséquemment, plusieurs chercheurs ont tenté de régler ces problèmes en proposant des approches basées sur des règles empiriques plus intuitives et plus rapides.

Parmi les premiers travaux de cette famille, l'approche de Wong, Ng et Heng (1997) est particulièrement intéressante. Leur modèle empirique est fondé sur l'utilisation d'une « distribution de tendance » définie selon des sources de détérioration abstraites. Cette distribution représente la probabilité que des marques d'usure apparaissent sur un objet à un point donné. Il s'agit d'un concept plus intuitif pour un artiste et il peut facilement déterminer les régions d'un objet plus susceptibles d'être détériorées. Cette distribution de tendance est ensuite utilisée par un algorithme de synthèse de textures pour produire l'apparence finale de l'objet.

Paquette, Poulin et Drettakis (2001) se sont ensuite attardés au phénomène de détérioration de surface par impacts en proposant eux-aussi un modèle empirique. Leur modèle propose de simuler itérativement la compaction d'un objet suite à une série d'impacts externes. Pour ce faire, leur système offre une gamme d'outils utilisés pour frapper sur un objet statique défini par un maillage standard. Selon l'outil sélectionné et quelques paramètres spécifiés par l'utilisateur (la trajectoire entre autre), la technique calcule d'abord l'intersection de l'outil et de l'objet. Par la suite, le maillage est raffiné à cet endroit pour permettre de représenter correctement les détails à ajouter. Finalement, les différents points formant le maillage sont déplacés selon la forme de l'outil pour simuler la compaction résultante. Évidemment, puisque l'opération modifie la géométrie de l'objet, les normales doivent être recalculées pour produire un affichage cohérent. Comme l'illustre la figure 1.5, la qualité des résultats est excellente.

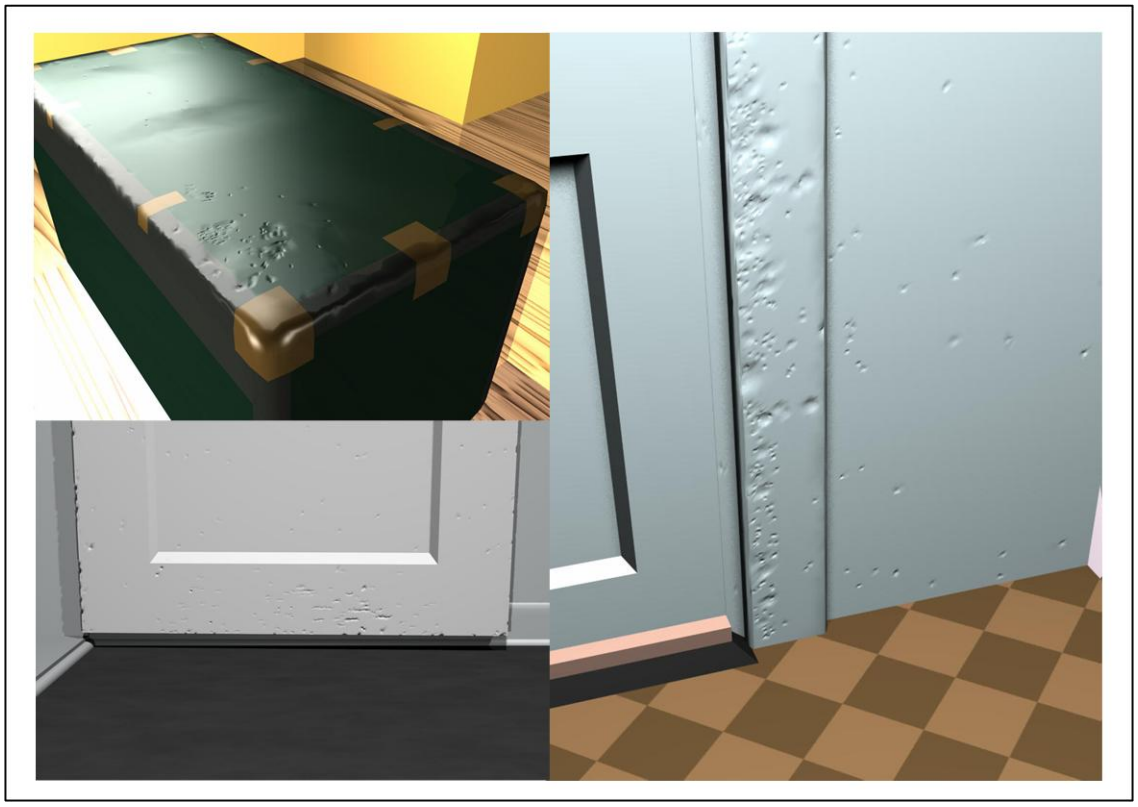

Figure 1.5 Résultats obtenus en simulant la détérioration de surface par impacts.
Adaptée de Paquette, Poulin et Drettakis (2001, p. 180)

Dans d'autres travaux, Paquette, Poulin et Drettakis (2002) ont développé un modèle empirique permettant de traiter les craquelures et l'écaillement de surfaces peinturées. D'autres chercheurs, tels que Gobron et Chiba (2001a; 2001b), s'étaient préalablement attardés à ce problème. L'approche de Paquette, Poulin et Drettakis propose une simplification du phénomène physique sous-jacent fondée sur le stress de tension appliqué sur la couche de peinture. Durant la simulation, des craquelures sont créées puis propagées sur la surface selon une grille de propriétés telles que le stress, la résistance maximale et l'élasticité. Ensuite, la perte d'adhésion de la couche de peinture provoquée par ces craquelures est interprétée lors du rendu pour générer des recourbements simulant le phénomène d'écaillement.

Par la suite, Cutler *et al.* (2002) et Chen *et al.* (2005) proposent des techniques de simulation empirique pour représenter une variété de types d'effets de détérioration. L'approche de Chen *et al.* (2005) est fondée sur l'étude d'un système de particules appelées « *γ-ton* ». Il s'agit d'une technique itérative qui consiste à « émettre » une série de particules à partir d'un ensemble de sources prédéterminées. Ces particules se propagent à l'intérieur de la scène selon certaines propriétés et selon différentes possibilités relatives à leur trajectoire. Après un certains nombre d'itérations, le système interprète la distribution des « *γ-ton* » dans l'environnement pour produire l'apparence finale de la scène. Pour y parvenir, leur méthode utilise, entre autre, un algorithme de synthèse de texture ainsi que des « *displacement maps* ». La figure 1.6 montre quelques résultats obtenus à l'aide de cette technique.

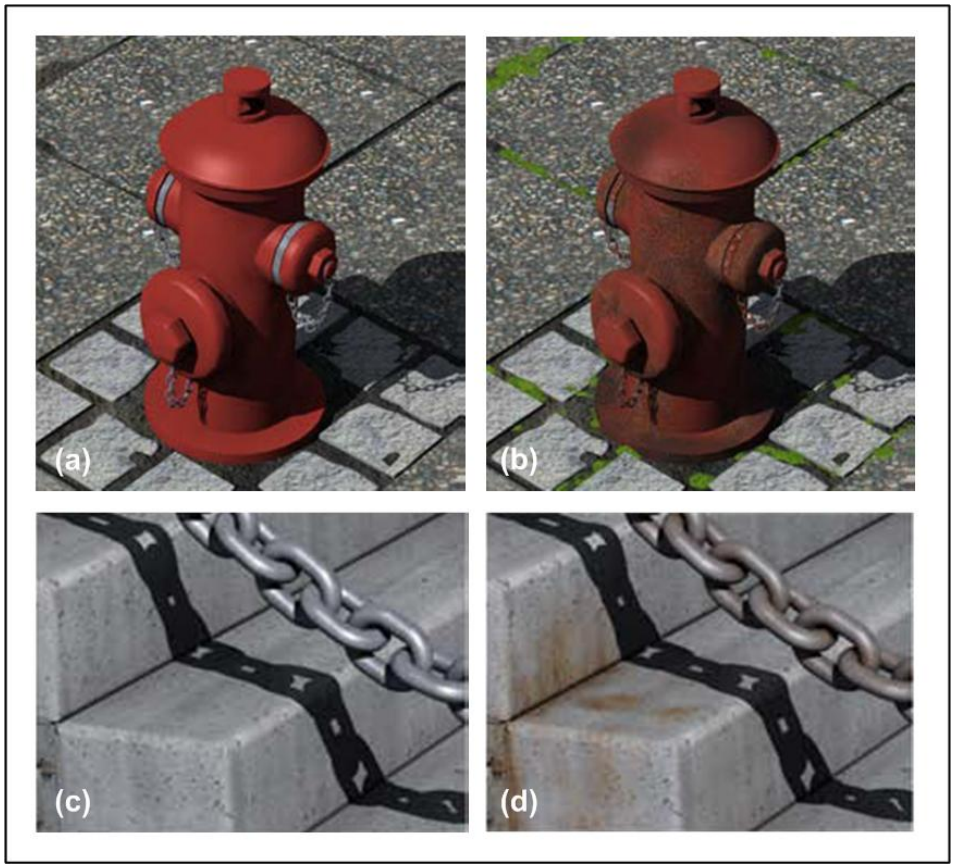

Figure 1.6 Résultats obtenus en simulant la détérioration par γ-ton.
Adaptée de Chen *et al.* (2005, p. 1127 et 1133)

Pour résumer, cette famille de techniques de détérioration apporte certaines solutions aux problèmes associés aux simulations physiques. En effet, elle propose des modèles basés sur des règles empiriques plus simples et plus intuitives que les lois physiques régissant réellement les phénomènes en question. Par conséquent, le potentiel d'utilisation concrète des ces techniques en industrie est considérablement plus élevé, tout simplement parce que les artistes les comprennent mieux et sont plus susceptibles de les exploiter.

Malheureusement, cette famille de technique ne parvient pas à régler tous les problèmes. Encore une fois, la relation entre les différents paramètres et l'apparence finale de l'objet n'est pas toujours claire (voir tableau 1.2). Il est donc difficile pour les artistes de bien contrôler l'apparence de leurs objets sans avoir une excellente compréhension du fonctionnement interne du modèle empirique. Par ailleurs, la majorité des techniques de cette famille s'adresse à un effet de détérioration spécifique. Or, dans un environnement quelconque, plusieurs types d'effets de détérioration se combinent pour engendrer une scène réaliste. De ce fait, les artistes doivent apprendre à utiliser correctement plusieurs de ces techniques, ce qui représente beaucoup de temps et d'argent pour les studios de production.

Tableau 1.2 Bilan des approches de simulation basées sur des règles empiriques

Technique	Paramètres	Performance
Wong *et al.* (1997) **Imperfection de surface**	Distribution de tendance de détérioration Propriétés géométriques de l'objet	Données non disponibles
Paquette *et al.* (2001) **Impact**	Impacts à l'aide d'outils de déformation	De 4 à 9 secondes de calculs De 30 à 120 min. d'interaction De 2 à 6k polygones Processeur de 600 MHz
Paquette *et al.* (2002) **Écaillement**	Contrainte de tension Résistance de tension Force d'adhésion	De 3 à 75 minutes De 130k à 400k polygones Processeur de 400 MHz
Chen *et al.* (2005) **Plusieurs types d'effets**	Sources et propriétés de γ-ton Propriétés des surfaces de l'objet	De 30 à 90 minutes De 50k à 200k polygones Processeur de 3 GHz

1.1.3 Approches basées sur la capture d'images

Plus récemment, une nouvelle famille d'approches a vu le jour. En effet, quelques chercheurs ont développé des approches basées sur la capture d'images. L'idée générale derrière ces techniques est d'utiliser des images comme exemples pour définir l'effet désiré, ce qui est beaucoup plus intuitif pour un artiste. Gu *et al.* (2006) proposent de capter la variation d'apparence d'un objet dans le temps pour pouvoir reproduire ce changement sur un objet synthétique. Pour ce faire, ils ont construit un dôme formé de 16 caméras et de 150 sources de lumières pour faire l'acquisition de la variation dans le temps d'une BRDF. Cette construction est exposée à la figure 1.7.

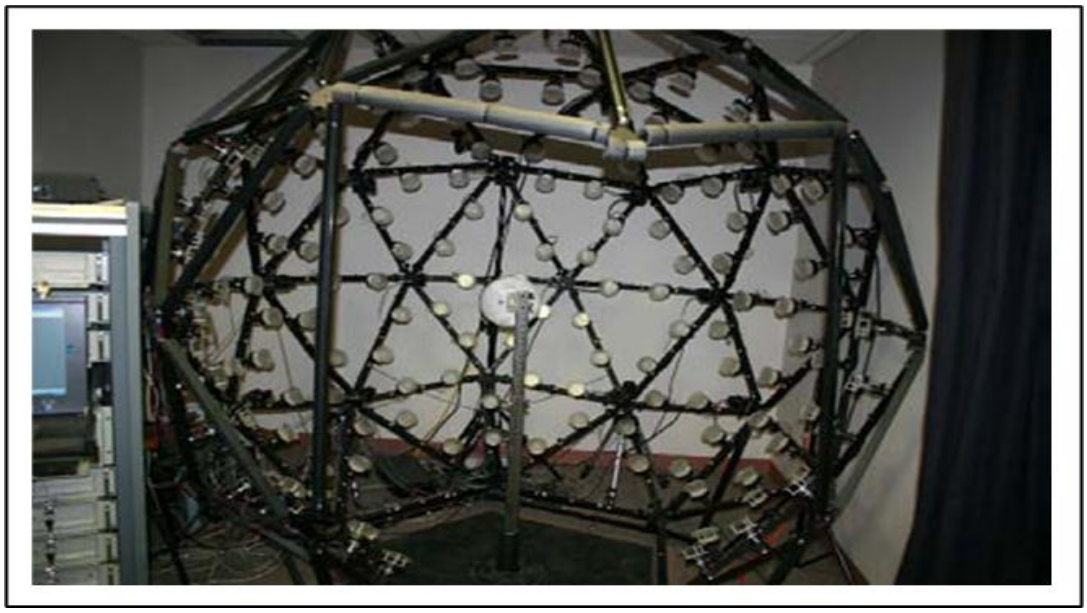

Figure 1.7 Système d'acquisition de la variation d'une BRDF dans le temps.
Tirée de Gu *et al.* (2006, p. 764)

À partir de ce système, les auteurs ont pu construire une base de données de 26 matériaux se détériorant dans le temps selon certains effets. Cette information est ensuite utilisée pour générer des effets semblables à partir d'un algorithme de synthèse de texture et d'un modèle mathématique estimant la courbe de variation de l'usure par rapport au temps. Quelques résultats de cette technique sont présentés à la figure 1.8.

Figure 1.8 Résultats obtenus en simulant l'évolution selon une BRDF.
Adaptée de Gu *et al.* (2006, p. 771)

Wang *et al.* (2006) proposent une technique semblable utilisant un système de capture simplifié. En effet, leur approche demande une seule image contenant plusieurs niveaux de détérioration. À partir de cet échantillon et de quelques interventions de l'utilisateur, le système déduit le degré de détérioration pour tous les pixels de l'image. Cette information est ensuite utilisée par un algorithme de synthèse de texture pour générer l'apparence d'un objet en fonction d'un paramètre de temps. La figure 1.9 montre quelques résultats produits par cette technique.

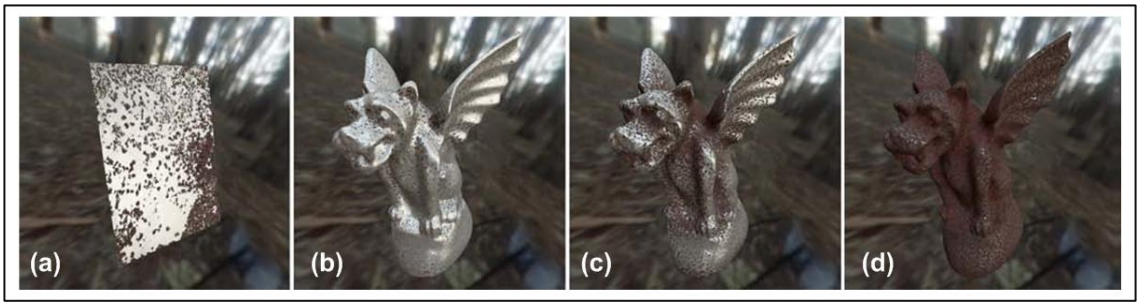

Figure 1.9 Résultats obtenus en simulant l'évolution selon un échantillon unique.
Tirée de Wang *et al.* (2006, p. 754)

Dernièrement, Lu *et al.* (2007) ont aussi développé une approche de simulation basée sur un système de capture d'images. La conception du système de capture utilisé dans le cadre de ce projet, appelé « *ShapeGrabber* » (voir figure 1.10), a été préalablement détaillée par Farouk *et al.* (2003).

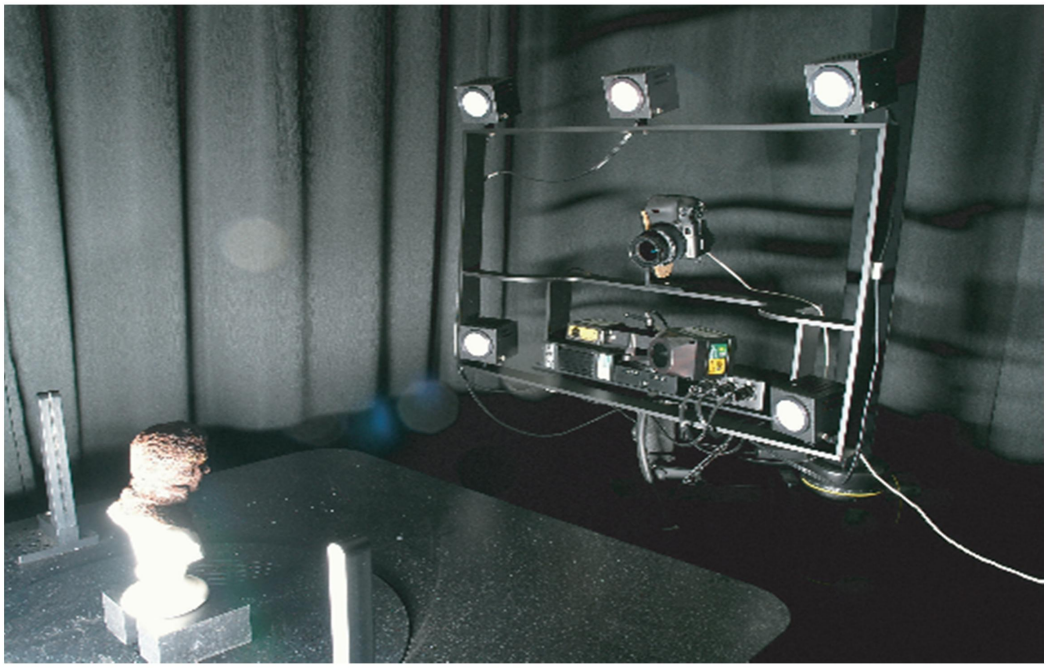

Figure 1.10 Système de capture du changement d'apparence « *ShapeGrabber* ».
Tirée de Lu *et al.* (2007, p. 7)

L'originalité de leur technique est l'utilisation de plusieurs propriétés locales de l'objet pour en contrôler l'apparence finale. En effet, la valeur de différentes propriétés, appelées paramètres de contexte, telles que l'occlusion ambiante (Kontkanen et Laine, 2005) et la courbure moyenne signée (Meyer *et al*., 2003) servent lors de l'étape de synthèse de texture pour générer une apparence plus cohérente avec la géométrie de l'objet. La figure 1.11 montre des résultats produits par cette technique.

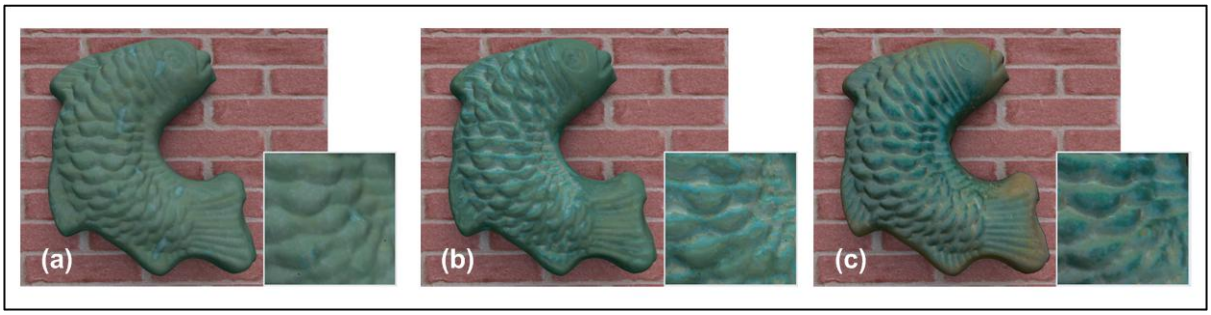

Figure 1.11 Résultats obtenus en simulant l'évolution avec paramètres de contexte.
Images tirées de Lu *et al.* (2007, p. 15)

Cette famille d'approches de simulation est très prometteuse puisqu'elle s'attaque aux problèmes associés aux autres techniques. Tout d'abord, l'utilisation d'images comme paramètres permet de rendre le processus beaucoup plus intuitif pour un artiste : la manipulation de paramètres physiques complexes est complètement exclue de la boucle. De plus, la relation entre les images fournies en entrée et l'apparence du résultat final est considérablement plus claire, ce qui implique un niveau de contrôle plus adéquat. Malheureusement, les techniques présentées dans cette section ne sont pas exemptes de défauts. L'approche de Gu *et al.* (2006) ainsi que celle de Lu *et al.* (2007) demande la mise en place d'un système d'acquisition d'images sophistiqué qui n'est évidemment pas accessible aux studios de production de petite et moyenne envergure. Il serait donc avantageux de concevoir une technique hybride combinant le système d'acquisition très simple de Wang *et al.* (2006) avec les paramètres de contexte proposés par Lu *et al.* (2007) pour obtenir un meilleur contrôle sur l'apparence finale de l'objet. Finalement, il serait intéressant d'étendre ces approches pour leurs permettre de traiter des effets de détérioration qui survienne brusquement, c'est-à-dire dont l'apparence ne varie pas nécessairement dans le temps tel que des graffitis, des gravures ou des traces de souliers.

1.2 Synthèse de textures

Comme mentionné dans la section 1.1, plusieurs techniques de simulation d'effets de détérioration utilisent des algorithmes de synthèse de texture. En effet, quelques approches, particulièrement dans la famille de simulations basées sur des images, se servent de tels algorithmes pour générer l'apparence finale de l'objet à partir d'un exemple. Il est donc important d'avoir une bonne compréhension de ces techniques pour maîtriser les approches de simulations. La présente section fait un résumé des travaux majeurs dans ce domaine en les divisant en deux blocs : les algorithmes de base et leurs applications à la complétion d'images.

1.2.1 Algorithmes de base

Historiquement, plusieurs chercheurs tels que De Bonet (1997) ou Heeger et Bergen (1995) ont proposé des algorithmes permettant de générer de nouvelles textures à partir d'un échantillon original. Cependant, l'approche fondamentale, sur laquelle se base plusieurs méthodes contemporaines, a été développée par Efros et Leung (1999). Leur technique consiste à générer, texel par texel, une nouvelle texture à partir d'un exemple initial. Pour y parvenir, ils assument que la couleur d'un texel *t* donné est dépendante de la couleur des texels voisins et indépendante du reste de l'image. Ainsi, pour déterminer la valeur d'un texel *t* de la nouvelle texture, ils proposent de regarder les couleurs des texels formant le voisinage, de chercher la meilleure correspondance dans l'échantillon original et de transférer la couleur du texel en question. Le seul paramètre de cette méthode, en plus de l'exemple de texture initial, est la taille du voisinage à considérer, aussi appelé la fenêtre. La taille de la fenêtre influencera la composante aléatoire du résultat.

Par la suite, ces travaux ont influencés plusieurs chercheurs qui ont proposé des approches pour améliorer la technique ou pour l'appliquer à un domaine spécifique. Liang *et al.* (2001) présentent une technique « *patch-based* » qui ne génère pas la nouvelle texture texel par texel mais plutôt bloc par bloc. De cette façon, ils arrivent à faire la synthèse de nouvelles textures dans un contexte temps réel. D'autres auteurs comme Lefebvre et Hoppe (2006) ainsi que

Wei et Levoy (2000) ont aussi proposé des améliorations à l'approche classique d'Efros et Leung (1999). Plusieurs chercheurs ont aussi proposé des méthodes permettant de contrôler l'apparence du résultat final. Parmi ceux-ci, Efros et Freeman (2001) ont développé une approche pour transférer une texture vers une autre image et ainsi contrôler l'apparence de l'image synthétisée (voir figure 1.12).

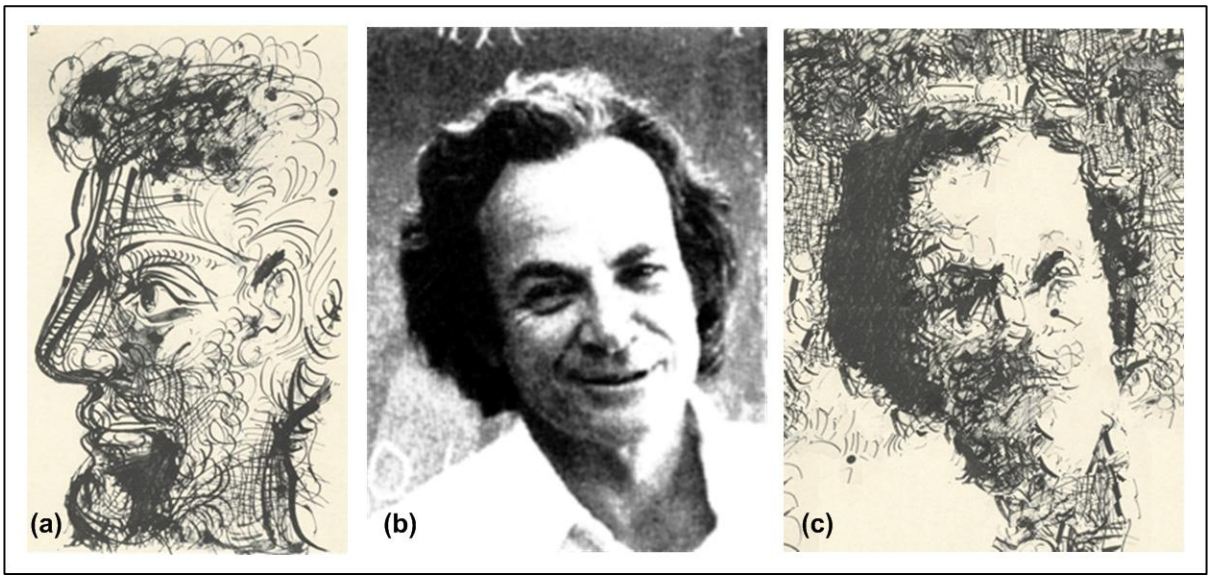

Figure 1.12 Résultats de l'algorithme de transfert de texture d'Efros et Freeman.
Adaptée de Efros et Freeman (2001, p. 345)

Ashikhmin (2001) innove en proposant une méthode permettant à l'utilisateur de contrôler l'apparence du résultat en crayonnant très grossièrement la configuration désirée. Malheureusement, la qualité des résultats obtenus n'est pas suffisante et l'utilisation de la technique est limitée à un type de texture spécifique. D'autres approches améliorant le contrôle ont été développées (Barrett et Cheney, 2002; Brooks et Dodgson, 2002; Kwatra *et al*., 2003) mais ne seront pas détaillées dans cette revue.

D'autres importants travaux de recherche dans ce domaine ont été proposés par Hertzmann *et al.* (2001). Leur méthode, toujours basée sur l'approche d'Efros et Leung (1999), intègre plusieurs améliorations majeures. Tout d'abord, ils présentent un traitement multi-résolutions permettant d'éliminer le paramètre spécifiant la taille de la fenêtre à considérer. En effet, leur

technique capte l'information de haut niveau via des images de basses résolutions et les détails plus fins dans des images de hautes résolutions. Il n'est donc plus nécessaire d'ajuster la fenêtre en fonction du motif contenu à l'intérieur de la texture. En plus, leur méthode introduit l'utilisation d'une technique d'approximation développée par Arya *et al.* (1998) lors de la recherche de la meilleure correspondance dans l'échantillon initial. Cette technique d'approximation construit un arbre regroupant l'ensemble des pixels de l'échantillon pour accélérer la recherche selon un vecteur de caractéristiques. Des résultats obtenus à l'aide de cette approche sont donnés à la figure 1.13.

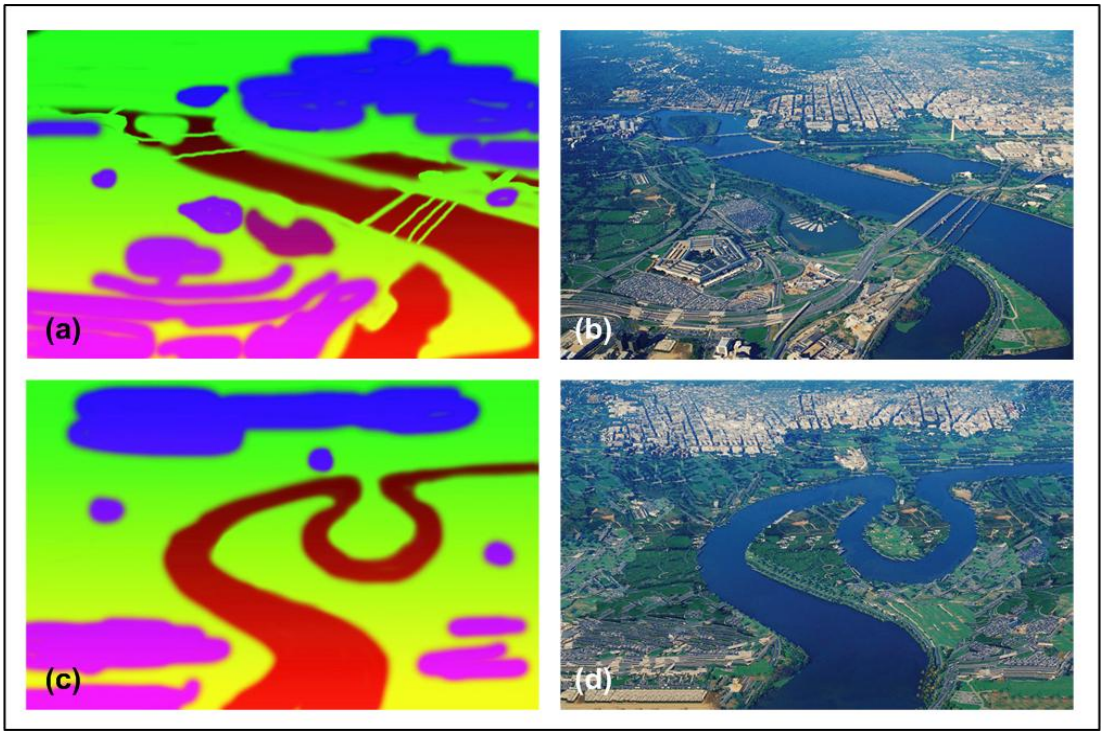

Figure 1.13 Résultats de l'algorithme de synthèse de texture d'Hertzmann *et al.*
Tirée de Hertzmann *et al.* (2001, p. 339)

1.2.2 Complétion d'images

La complétion d'images représente une des applications importantes des algorithmes de synthèse de textures. Elle consiste à remplacer certaines régions indésirables d'une image et peut être utilisée dans plusieurs domaines comme pour la restauration de vieilles toiles ou de vieilles photographies ainsi que pour retirer ou même pour modifier ou retirer des éléments importuns (micros, câbles de soutien, etc.). Bertalmio *et al.* (2000) sont parmi les premiers à proposer une solution à cette problématique basée un algorithme de synthèse de textures. Leur approche consiste à remplir itérativement la région en question en propageant les valeurs des pixels du voisinage et en tentant de préserver la direction des arêtes. Quelques résultats obtenus à l'aide de cette technique sont présentés à la figure 1.14.

Figure 1.14 Résultats produits par l'algorithme de complétion de Bertalmio *et al.*
Tirée de Bertalmio *et al.* (2000, p. 423)

Leur technique produit de bons résultats pour des régions assez étroites mais n'arrive pas à recréer adéquatement les motifs lorsque les zones deviennent plus larges. Drori, Cohen-Or et Yeshurun (2003) ont développé une approche différente pour pallier au problème. Leur méthode multi-résolutions permet de reconstituer ces motifs puisque le processus couvre les patrons de haut niveau ainsi que les détails plus fins. Ils intègrent aussi un concept de « confiance » qui représente un pourcentage de certitude envers le remplacement effectué. Ainsi, il leur est possible de faire évoluer l'image pour obtenir un niveau de confiance maximum. La qualité des résultats est fortement améliorée (voir figure 1.15).

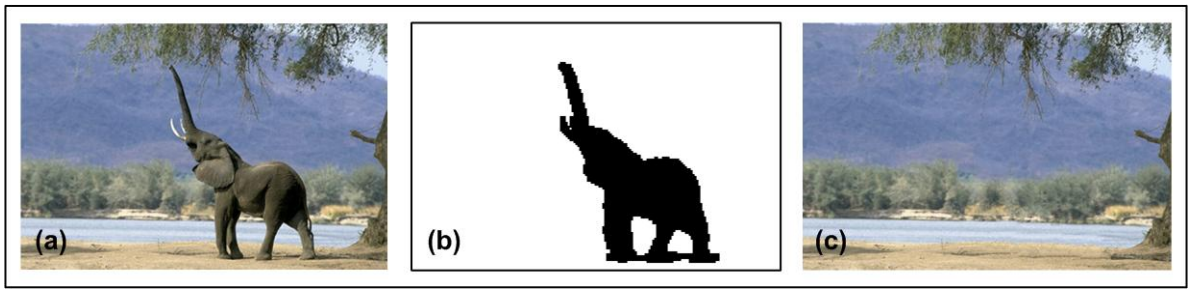

Figure 1.15 Résultats produits par l'algorithme de complétion de Drori *et al.*
Adaptée de Drori, Cohen-Or et Yeshurun (2003, p. 303)

Malheureusement, ces algorithmes éprouvent certains problèmes lorsque des structures sont complètement cachées ou lorsque des ambiguïtés se présentent durant le processus de complétion. Consultez la figure 1.16 pour quelques exemples de ce problème.

Figure 1.16 Quelques exemples d'ambiguïtés de structures.
Adaptée de Sun *et al.* (2005, p. 867 et 868)

Sun *et al.* (2005) proposent d'intégrer l'utilisateur dans le traitement pour lui permettre de fournir des informations additionnelles sur les structures importantes contenues dans l'image. À partir de cette information, la méthode effectue une première phase de remplissage en forçant la continuité des structures identifiées par l'utilisateur. Ensuite, les autres régions de l'image sont complétées selon un algorithme de synthèse de texture standard. Cette méthode produit de très bons résultats pour des problèmes plus complexes (voir figure 1.17) mais nécessite l'intervention de l'utilisateur, ce qui fait en sorte qu'elle n'est plus automatique.

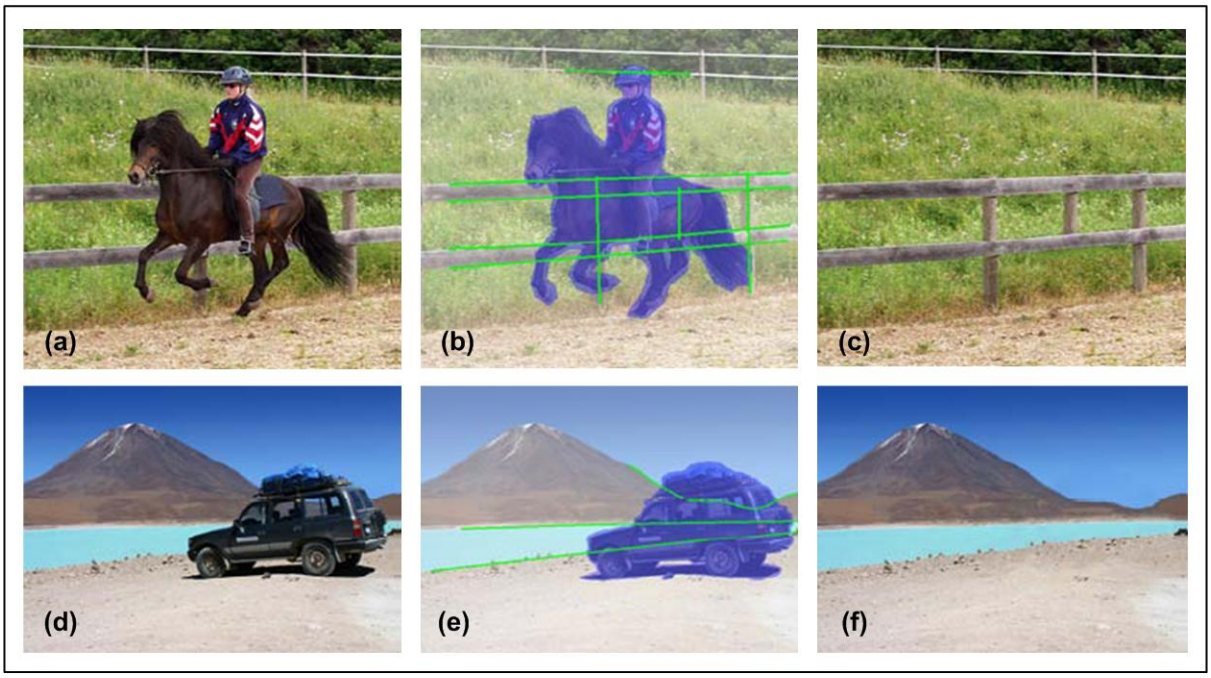

Figure 1.17 Résultats produits par l'algorithme de complétion de Sun *et al.*
Adaptée de Sun *et al.* (2005, p. 867)

En résumé, les différentes techniques de complétion d'images énoncées préalablement ont chacune leurs forces et leurs faiblesses. Certaines produisent de moins bons résultats mais ont l'avantage d'être complètement automatisées alors que d'autres demandent à l'utilisateur d'intervenir pour améliorer la qualité des résultats. Dans le cadre de ce projet de recherche, ces algorithmes seront adaptés pour permettre d'ajouter certains éléments comme des effets de détérioration sur une texture à partir d'un exemple pour en améliorer le réalisme.

1.3 Objectifs et contributions

Pour adresser correctement la problématique énoncée précédemment, les différents objectifs associés à ce projet de recherche ont été partagés selon trois axes distincts : la synthèse d'effets de détérioration à l'aide d'un exemple, une approche de détérioration basée sur les propriétés locales d'un objet et un processus de synthèse indépendant de la couleur. La présente section a pour but de détailler chacun de ces axes et de mettre en évidence comment ils permettront de réduire les difficultés associées aux effets de détérioration pour la synthèse d'images réalistes.

1.3.1 Synthèse d'effets de détérioration à l'aide d'un exemple

Tel qu'exposé précédemment, la synthèse d'effets de détérioration représente un défi important pour les studios de production qui veulent produire des images réalistes. Un premier objectif du projet de recherche est de développer une approche pour ajouter des effets de détérioration sur la surface d'objets virtuels qui soit adaptée aux artistes et à leur environnement. D'une part, cela signifie que les paramètres fournis à la méthode doivent être simples, intuitifs et ne demander aucune connaissance scientifique particulière. Comme l'illustre la figure 1.18, l'utilisation d'un exemple de l'effet désiré, sous forme d'image (créée à la main ou photographiée), représente une excellente façon de fournir l'information nécessaire pour reproduire quelque chose de similaire.

D'autre part, pour être adaptée aux artistes et à leur environnement, l'approché développée ne doit pas être limitée à quelques effets spécifiques mais doit plutôt offrir la possibilité de travailler avec un vaste éventail de types de détérioration touchant la surface d'un objet. Ainsi, l'approche développée doit permettre de transférer et de combiner des effets de détérioration d'une image à l'autre pour obtenir des résultats plus réalistes.

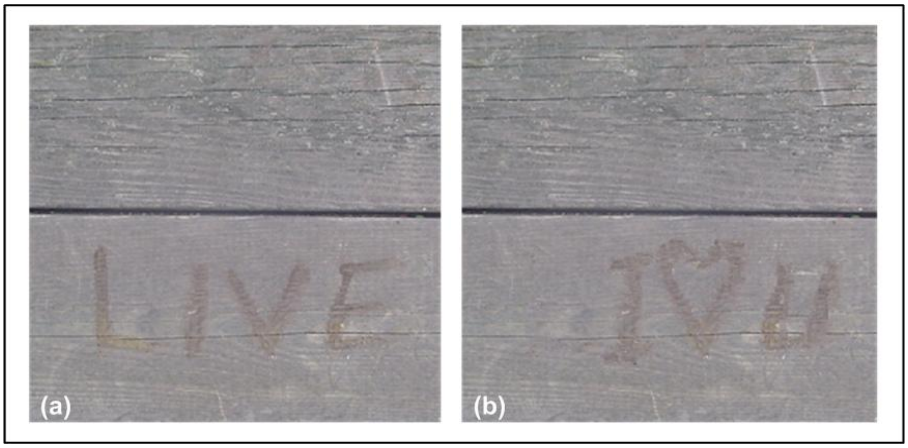

Figure 1.18 Reproduction d'effets de détérioration semblables.
où (a) est l'effet de détérioration original et (b) est une reproduction semblable.

Finalement, l'approche de synthèse d'effets de détérioration proposée doit produire des résultats dans un délai interactif. De cette façon, l'artiste pourra rapidement ajuster l'apparence de son objet sans ralentir le processus itératif de revue par les pairs.

1.3.2 Approche de détérioration basée sur les propriétés locales d'un objet

Un deuxième volet de ce projet de recherche consiste à analyser la corrélation entre les propriétés locales d'un objet et les effets de détérioration associés, dans le but de fournir à l'utilisateur un outil permettant de définir aisément le modèle de détérioration. La disposition de plusieurs effets de détérioration est fortement liée à certaines propriétés de la géométrie d'un objet comme la normale, la courbure ou l'accessibilité. À titre d'exemples, il est juste de présumer que des régions comme les coins ou les arêtes courent un plus grand risque d'être involontairement cognées alors que les surfaces planes sont plus susceptibles d'être vandaliser par graffitis. Dans le même ordre d'idées, de la moisissure a plus de chance de se former dans des régions moins accessibles d'un objet, tout simplement car l'eau qui s'y infiltre sèche beaucoup moins rapidement qu'ailleurs. La figure 1.19 présente quelques exemples illustrant ce propos.

Figure 1.19 Relation entre les effets de détérioration et les propriétés d'un objet.

Il est donc considérablement avantageux d'utiliser cette relation pour simplifier le processus de synthèse. En associant certaines propriétés à son exemple d'effet de détérioration, l'artiste permet au système de déterminer automatiquement les régions plus susceptibles d'être touchées. Pour l'artiste, cette approche demeure relativement simple d'utilisation et offre un niveau de contrôle adéquat sur l'apparence finale de l'objet.

1.3.3 Synthèse indépendante de la couleur de l'exemple

Le troisième et dernier volet de ce projet de recherche consiste à proposer un processus de synthèse indépendant de la couleur de l'exemple fourni. En analysant des cas réels, il est facile de constater qu'il arrive souvent qu'un même objet existe dans une grande variété de couleurs différentes. À titre d'exemple, la majorité des meubles, tels que des tables ou des chaises, peuvent être achetés dans plusieurs teintes de bois (voir figure 1.20).

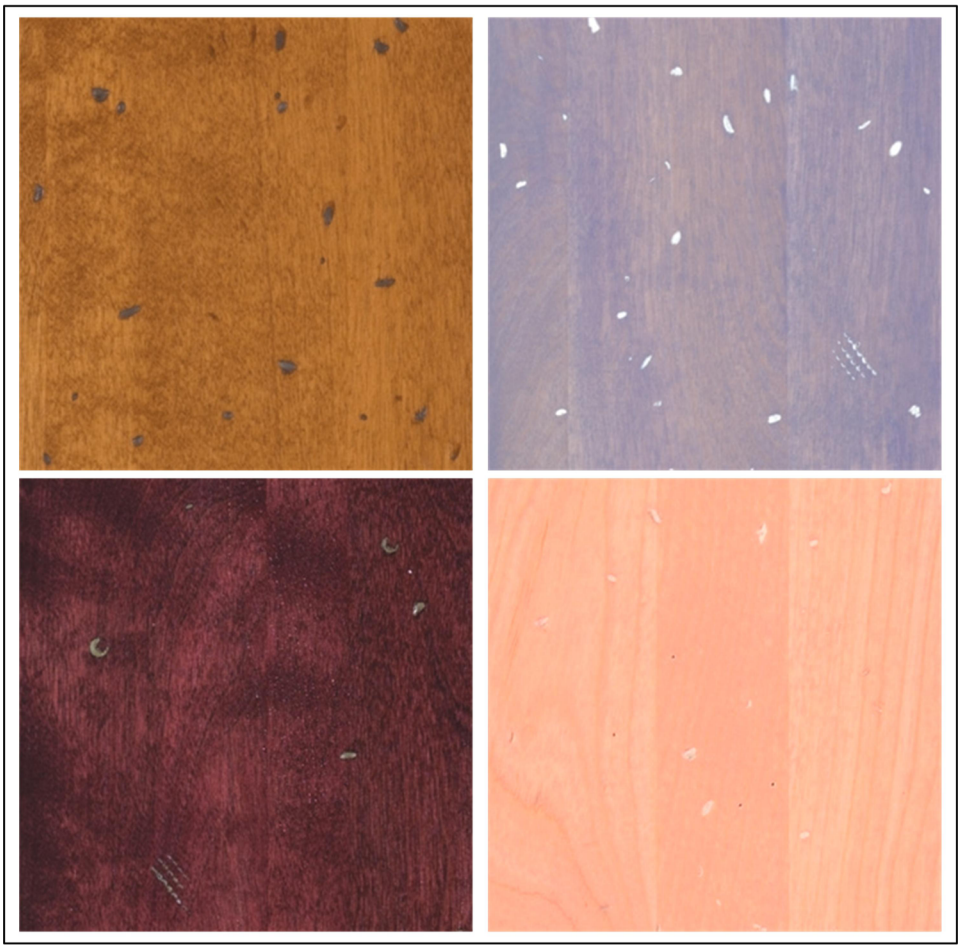

Figure 1.20 Exemple d'effets similaires de couleurs différentes.
Tirée du site Internet de Canadel (2010)

Or, lorsque vient le temps d'appliquer des effets de détérioration sur de tels objets à l'aide d'approches basées sur un exemple, l'utilisateur rencontrera quelques difficultés. En effet, dans le cas où son image d'exemple contient une égratignure bleue, il ne sera pas en mesure de produire une égratignure rouge. De plus, si l'objet sur lequel il veut ajouter son effet n'est pas de la même couleur que l'objet sur l'image d'exemple, le résultat du processus de synthèse aura de la difficulté à produire des transitions adéquates entre les régions détériorées et non-détériorées. Pour produire de bons résultats, l'artiste aurait donc besoin d'un exemple pour toutes les combinaisons possibles de couleur d'effet et de couleur de base de l'objet. Par conséquent, le dernier axe du projet consiste à proposer une solution à ce problème en permettant à l'utilisateur de contrôler la teinte du résultat final à partir d'un seul et même exemple.

CHAPITRE 2

SYNTHÈSE DE DÉTÉRIORATION À L'AIDE D'EXEMPLES[1]

Tel qu'énoncé précédemment, un des objectifs de ce projet de recherche est de développer un processus d'édition d'effets de détérioration adapté pour les artistes et les studios de production. Pour répondre à ces attentes, le processus d'édition proposé doit éviter la manipulation de paramètres complexes, fournir un contrôle intuitif sur l'apparence du résultat final et pouvoir s'utiliser dans des délais interactifs. Dans ces conditions, une approche de simulation basée sur des images, appelées des exemples, semble être appropriée, par opposition à une simulation basée sur un modèle physique. Le présent chapitre présente de façon détaillée les différentes étapes de l'approche proposée, expose les résultats ainsi obtenus et se termine par une discussion sur les avantages et les inconvénients qui lui sont associés.

2.1 Présentation générale du processus

Le processus de synthèse d'effets de détérioration développé dans ce projet de recherche a pour objectif de permettre à un artiste d'éditer l'usure d'un objet virtuel en modifiant la texture qui lui est appliquée dans le but de produire une apparence plus réaliste. L'idée générale du processus est d'utiliser un exemple de l'effet de détérioration désiré, fourni par l'artiste, pour en générer de nouveaux semblables mais non identiques. Comme le montre la figure 2.1, cet exemple, aussi appelé l'image source, provient généralement d'une photographie, bien qu'une image créée par ordinateur ferait aussi parfaitement l'affaire. De plus, une seconde image doit être fournie par l'artiste pour produire l'image éditée : le

---

[1] Le contenu de ce chapitre fait l'objet d'une publication dans une conférence internationale :
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*. p. 151-158. New York (NY): ACM Press.

masque cible. Ce masque est une image binaire spécifiant la région sur laquelle l'artiste désire appliquer le nouvel effet de détérioration.

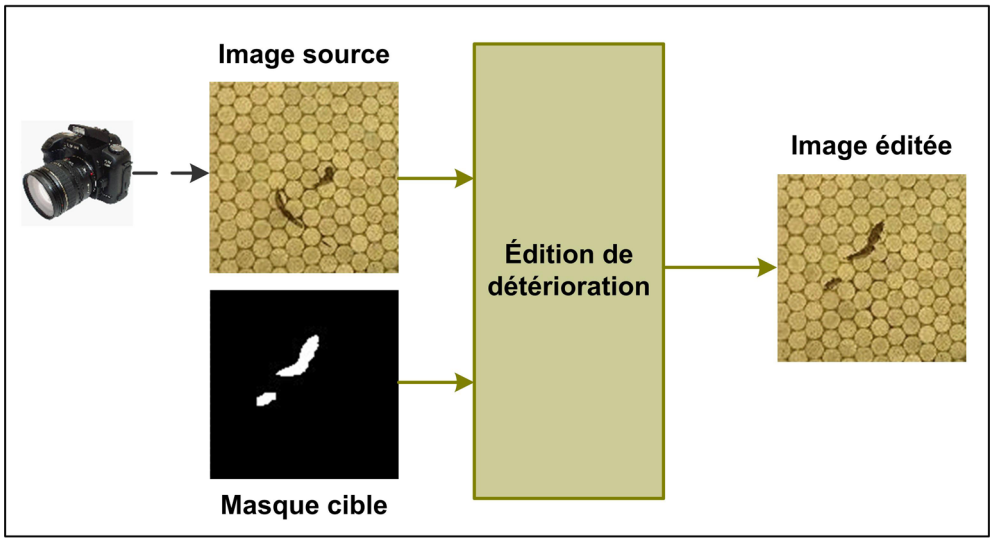

Figure 2.1 Présentation générale du processus de synthèse.

L'utilisation d'un exemple d'effet de détérioration combinée à celle d'un masque cible pour contrôler l'apparence du résultat final a été introduite récemment par quelques auteurs proposant des méthodes d'usure basées sur des images (Gu *et al.*, 2006; Lu *et al.*, 2007; Wang *et al.*, 2006). Par contre, leurs approches ne sont pas adaptées à une utilisation dans un studio de production puisqu'elles impliquent des systèmes de capture complexe (voir section 1.1.3) et ne sont pas efficaces dans un contexte de revue itérative (voir figure 1.4).

Le fonctionnement interne du processus d'édition d'effets de détérioration proposé est présenté à la figure 2.2. Comme le montre cette figure, le processus est divisé en trois étapes principales : la segmentation, l'élimination et la reproduction. L'étape de segmentation consiste à identifier les régions de l'image source contenant les effets de détérioration. Il s'agit d'une étape cruciale du processus puisqu'une identification erronée aura un impact direct sur la qualité des résultats produits. En sortie, l'étape de segmentation produit le masque source qui, tout comme le masque cible, est une image binaire marquant les effets de détérioration. Ensuite, l'étape suivante consiste à utiliser le masque source pour éliminer les effets de détérioration initiaux. Un algorithme de synthèse de texture par remplissage de trous

est utilisé pour y parvenir. En sortie, cette étape produit une image dite nettoyée. Finalement, la dernière étape consiste à reproduire des effets de détérioration semblables à l'exemple original aux endroits spécifiés par le masque cible. Comme pour l'étape d'élimination, un algorithme de synthèse de texture par remplissage de trous est utilisé pour générer ces nouveaux effets de détérioration. En sortie, le système produit une image contenant des effets édités en fonction du masque cible.

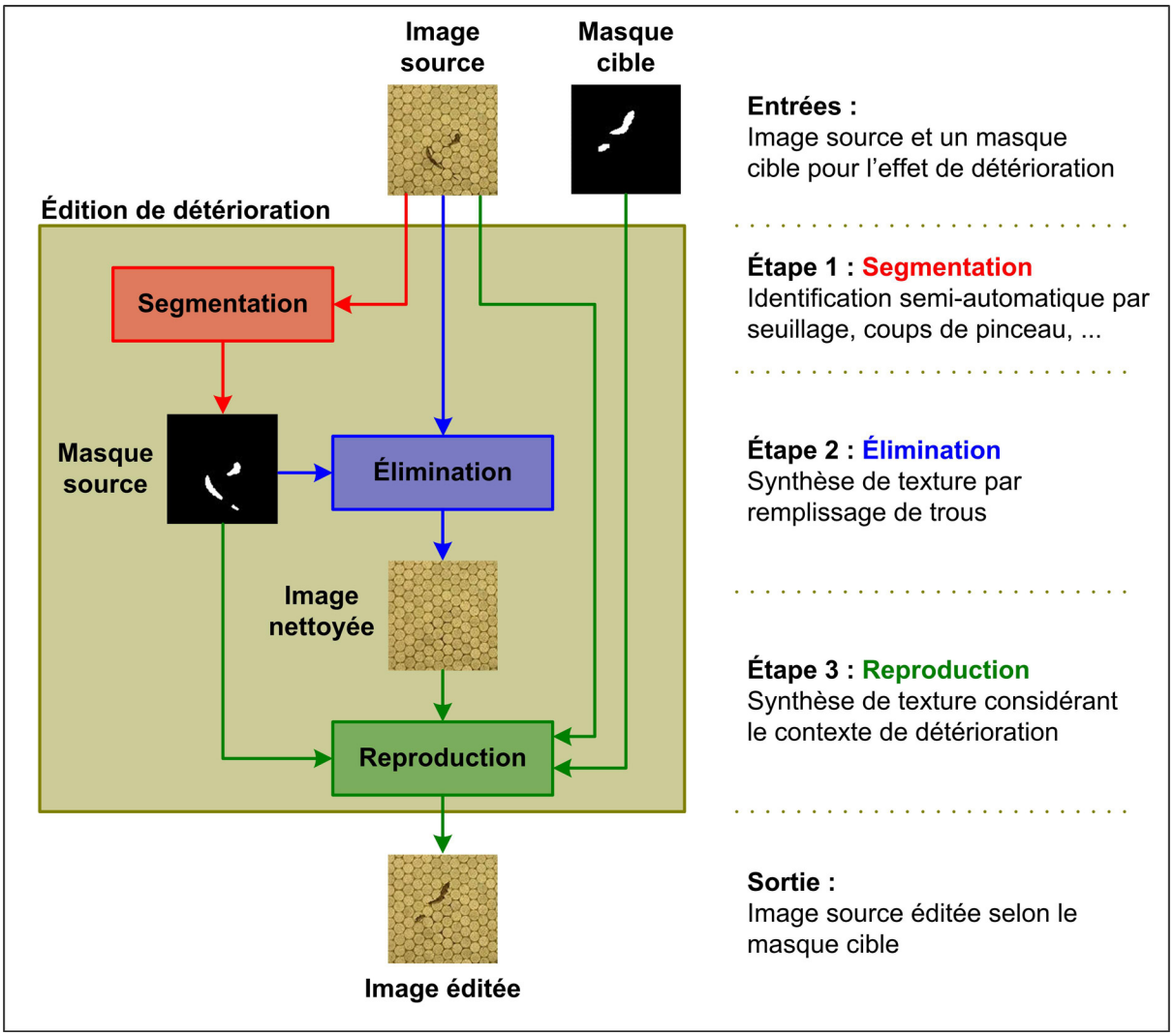

Figure 2.2 Présentation du fonctionnement interne du processus.

Le fonctionnement de chacune des étapes formant le processus d'édition d'effets de détérioration proposé sera détaillé dans les sections 2.2, 2.3 et 2.4 respectivement.

2.2 Étape de segmentation interactive

L'étape de segmentation interactive se retrouve au tout début du processus d'édition d'effets de détérioration. Tel que mentionné précédemment, cette étape consiste à identifier les régions de l'image source contenant les effets de détérioration. Elle joue un rôle critique dans le processus puisqu'une identification erronée ou imprécise entraînera inévitablement des problèmes durant les étapes ultérieures, ce qui aura pour effet de réduire la qualité des résultats obtenus. C'est dans cette optique qu'il a été décidé d'intégrer l'artiste dans ce processus d'identification. En effet, il est pratiquement impossible de développer un algorithme de segmentation complètement automatique qui produira des résultats fiables, peu importe l'image fournie en entrée. L'idée est donc de fournir différents outils à l'artiste pour faciliter l'identification des régions détériorées tout en lui permettant de valider et de corriger le résultat aux besoins. Évidemment, pour produire le masque cible, un artiste pourrait utiliser des fonctionnalités standards fournies par des logiciels d'édition d'images commerciaux. Néanmoins, le prototype développé regroupe différents outils particulièrement efficaces dans le contexte du projet. Par ailleurs, bien que les méthodes de segmentation présentées dans cette section soient déjà connues et utilisées dans d'autres domaines du traitement d'images, elles permettent de se différencier de travaux antérieurs (Gu *et al.*, 2006; Lu *et al.*, 2007; Wang *et al.*, 2006) en permettant de traiter efficacement des effets locaux (des bosses, des égratignures, des taches, etc.) et non seulement des effets évolutifs (rouille, patine, moisissure, etc.). Une séquence vidéo montrant l'interaction entre l'utilisateur et le prototype est disponible à l'adresse suivante :
http://profs.etsmtl.ca/epaquette/Research/Papers/Clement.2007/

2.2.1 Segmentation par seuillage

Une première grande famille de méthode consiste à produire la segmentation à l'aide d'opérations de seuillage. Il s'agit d'une famille de méthodes très simples qui peuvent produire d'excellents résultats lorsque le contexte s'y prête bien. Considérons d'abord une image représentée selon l'aide de 255 niveaux de gris. Une opération de seuillage consiste

tout simplement à comparer la valeur de chacun des pixels à un seuil minimum S_{min} et un seuil maximum S_{max} pour ensuite appliquer la règle suivante :

$$\begin{aligned} \mathbf{1}: &\quad si\ S_{min} < I < S_{max} \\ \mathbf{0}: &\quad sinon \end{aligned} \tag{2.1}$$

Bien que cette approche soit plutôt simpliste, elle produit d'excellents résultats lorsque la région à identifier présente une valeur d'intensité suffisamment distincte du reste de l'image. La figure 2.3 montre une image source contenant une tâche de peinture blanche pour laquelle cette méthode s'applique bien.

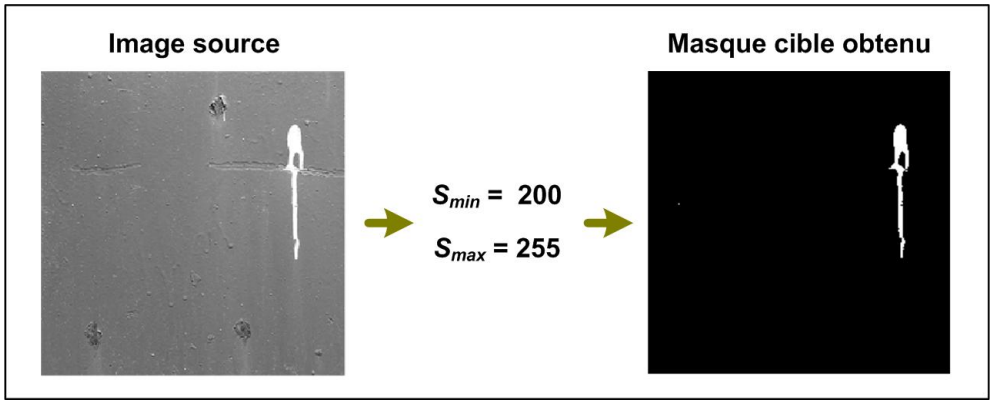

Figure 2.3 Segmentation par seuillage sur le niveau de gris.

Évidemment, il est possible d'étendre ce concept pour une image RVB (Triplet rouge, vert, bleu). L'opération reste la même mais est effectuée sur chaque canal de façon indépendante.

$$\begin{aligned} \mathbf{1}: &\quad si\ SR_{min} < R < SR_{max}\ et \\ &\quad si\ SV_{min} < V < SV_{max}\ et \\ &\quad si\ SB_{min} < B < SB_{max}\ et \\ \mathbf{0}: &\quad sinon \end{aligned} \tag{2.2}$$

Le seuillage sur les composantes RVB permet d'identifier efficacement des régions détériorées qui présente une couleur distincte du reste de l'image. Encore une fois, il s'agit d'une approche assez simpliste qui fonctionne adéquatement que dans certains cas spécifiques. Un exemple d'un tel cas est présenté à la figure 2.4.

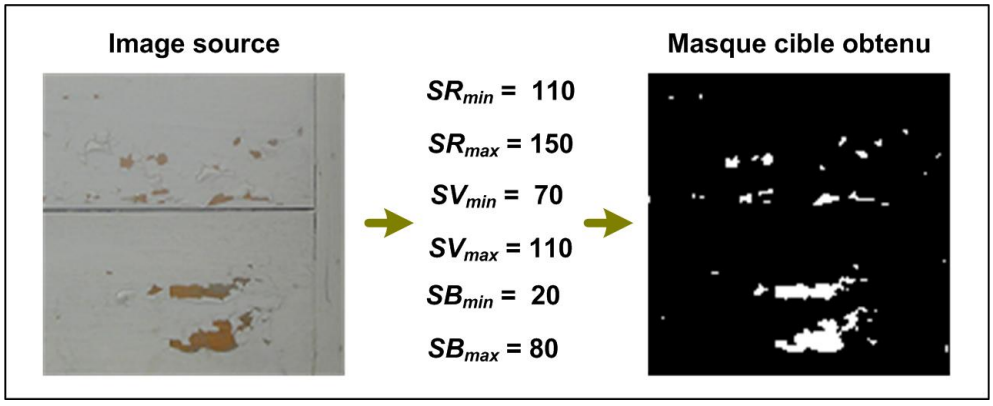

Figure 2.4 Segmentation par seuillage sur les canaux RVB.

Cependant, lorsqu'il est possible d'identifier les régions par seuillage sur la couleur, il est souvent plus simple d'utiliser un modèle de couleur différent de RVB, tel que le modèle HSV. En effet, le modèle HSV permet de définir une couleur à l'aide de trois composantes : la teinte, la saturation (la pureté de la couleur) et l'intensité. Il s'agit d'un modèle de représentation plus intuitif pour l'être humain ce qui, dans le contexte du projet, facilite la sélection des seuils à utiliser. Par conséquent, le prototype permet à l'artiste de choisir le modèle de couleurs qu'il préfère.

En somme, il faut retenir que la segmentation par seuillage est une méthode simple, rapide et intuitive pour un artiste mais qui s'applique seulement à des contextes très spécifiques. Cependant, lorsque les conditions le permettent, ces méthodes vont produire une identification très précise en demandant peu d'efforts de l'artiste. La sélection des valeurs des différents seuils nécessaire peut parfois être un peu ardue, mais l'utilisation d'un histogramme permet de bien visualiser les différentes régions contenues dans l'image source. Avant de passer à des méthodes de segmentation plus complexes, il est souvent avantageux de faire quelques essais de seuillage pour se sauver du travail, dans l'optique où il serait ainsi possible d'identifier les régions détériorées.

2.2.2 Segmentation par coups de pinceau

Puisque l'utilisation des techniques de segmentation par seuillage est assez limitée, une approche de segmentation par coups de pinceau adaptée des travaux de Lischinski *et al.* (2006) a été intégrée dans le prototype. Il s'agit d'une méthode interactive qui demande à l'artiste de marquer grossièrement les régions détériorées à l'aide de coups de pinceau et qui assigne ensuite automatiquement l'étiquette « détérioré » ou l'étiquette « non détérioré » à chacun des pixels de l'image source. Pour effectuer cette assignation, le système cherche à minimiser une fonction d'énergie définissant les diverses régions de l'image. La figure 2.5 illustre les différentes étapes associées à cette approche.

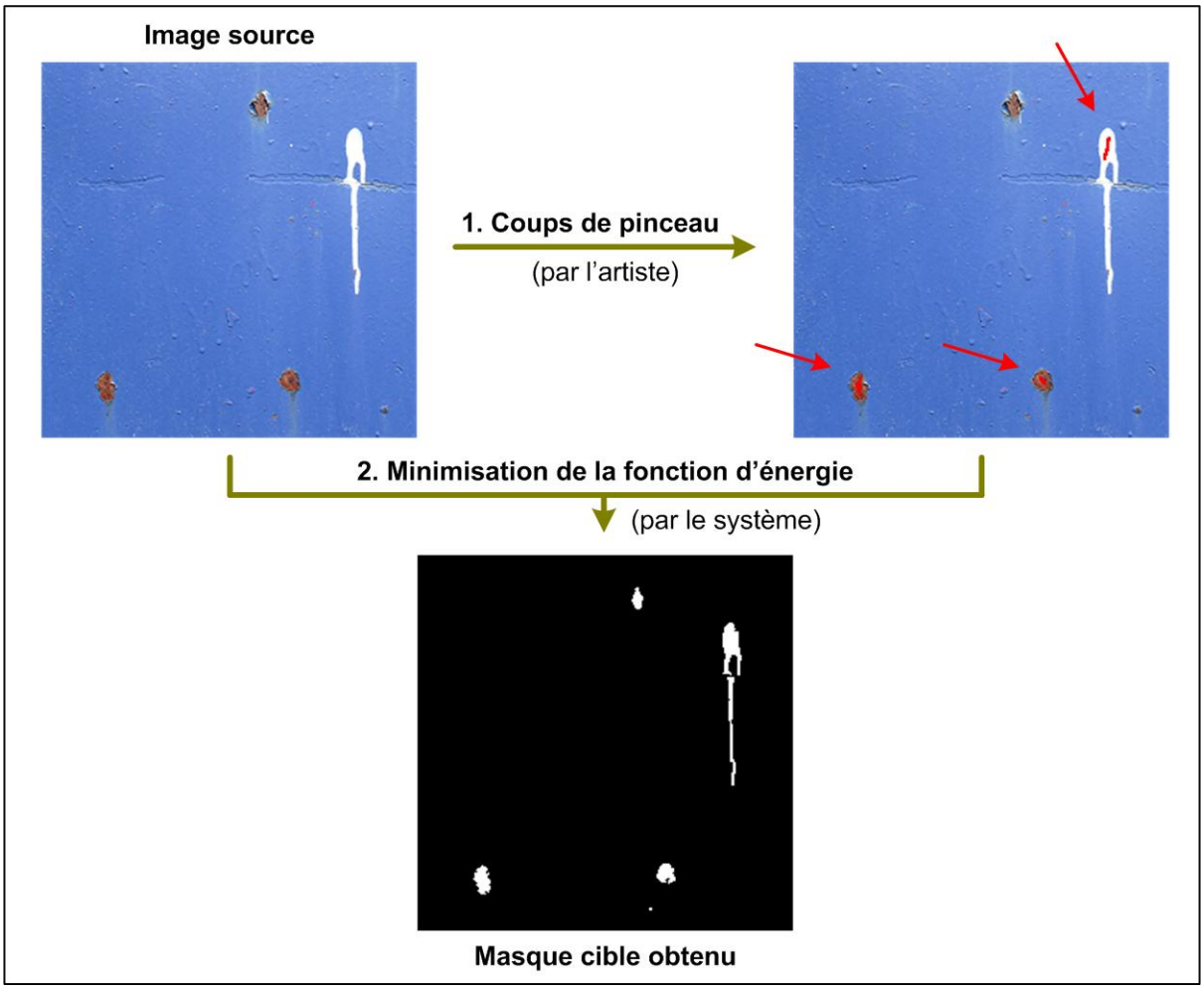

Figure 2.5 Segmentation par coups de pinceau.

Bien évidemment, le cœur de cette méthode réside dans l'étape de minimisation d'énergie qui permet de transformer quelques coups de pinceau grossiers de l'artiste en une identification précise des régions détériorées. La première étape consiste à définir une fonction d'énergie qui permet de bien caractériser les différentes régions contenues dans l'image. À l'aide de multiples expérimentations, une fonction d'énergie formée de deux termes a été développée pour adresser ce problème.

$$\boldsymbol{E}\ (et) = \boldsymbol{E}_{couleurs}\ (et) + \boldsymbol{E}_{contours}\ (et) \tag{2.3}$$

Cette fonction définie l'énergie associée à un pixel de l'image en fonction de l'étiquette « *et* » qui lui est assignée. Tel que mentionné précédemment, l'étiquette peut prendre les valeurs « détérioré » ou « non détérioré ». Le premier terme de la fonction consiste à accorder un coût énergétique supplémentaire à un pixel si sa couleur est semblable aux pixels identifiés par les coups de pinceau et que l'étiquette « non détérioré » lui est assigné. Ainsi, lors de l'étape de minimisation, les pixels dont la couleur correspond aux régions marquées par l'artiste auront tendance à porter l'étiquette « détérioré ». Le deuxième terme de la fonction d'énergie a pour objectif d'assurer l'intégrité des régions à l'aide d'une détection de contours. Tout d'abord, l'algorithme de Canny (1986) est utilisé pour faire la détection des contours sur l'image source. Ensuite, un coût énergétique supplémentaire est accordé à un pixel si l'étiquette qui lui est associée est différente de celle de ses voisins et qu'aucune arête n'a été identifiée à cet endroit. Inversement, pour les régions où une arête a été identifiée par l'algorithme de détection de contours, aucun coût énergétique ne sera accordé pour un changement d'étiquette. Bref, ce deuxième terme favorise la discontinuité des étiquettes en fonction des contours de l'image.

Maintenant que la fonction d'énergie est définie, l'étape suivante consiste à assigner une étiquette à chacun des pixels de façon à minimiser l'énergie engendrée pour l'ensemble l'image. Pour ce faire, le prototype utilise une librairie développée par Boykov, Veksler et Zabih (2001) qui implémente une approximation rapide basée sur des coupes de graphes (« *graph cuts* »). Pour plus de détails concernant le fonctionnement de cette librairie, veuillez consulter leur article.

Dans le cadre de multiples expérimentations, cette approche de segmentation a été la plus utilisée de toutes. En effet, cette technique fonctionne adéquatement dans la majorité des cas, en plus d'être très simple d'utilisation. À titre de paramètres, elle ne nécessite que quelques coups de pinceau grossiers lorsqu'utilisée telle quelle. Pour des segmentations plus complexes, la fonction d'énergie peut être ajustée de façon spécifique, ce qui la rend très polyvalente. Finalement, cette approche fonctionne dans des délais interactifs, permettant à l'artiste d'obtenir rapidement un résultat et d'ajuster les paramètres au besoin.

2.2.3 Combinaisons de différentes techniques

Durant l'étape de segmentation, il arrive parfois qu'une seule technique n'arrive pas à identifier correctement toutes les régions détériorées d'une image, particulièrement lorsque que cette dernière contient plusieurs effets différents. Conséquemment, il serait utile de pouvoir utiliser diverses techniques de segmentation sur une même image dans le but de combiner leurs résultats. La figure 2.6 présente un cas de segmentation où la combinaison de différentes techniques permet d'obtenir une meilleure segmentation. Dans cet exemple, la segmentation par coups de pinceau permet de bien identifier les régions rouillées de l'image source, mais l'identification de tache de peinture blanche contient quelques discontinuités. Or, la segmentation par seuillage sur niveau de gris permet d'obtenir un résultat impeccable pour la peinture blanche. Pour obtenir un meilleur résultat, il est donc possible d'additionner le résultat de plusieurs techniques de manière à les combiner. Le système permet de faire l'addition, la soustraction et l'intersection de plusieurs techniques.

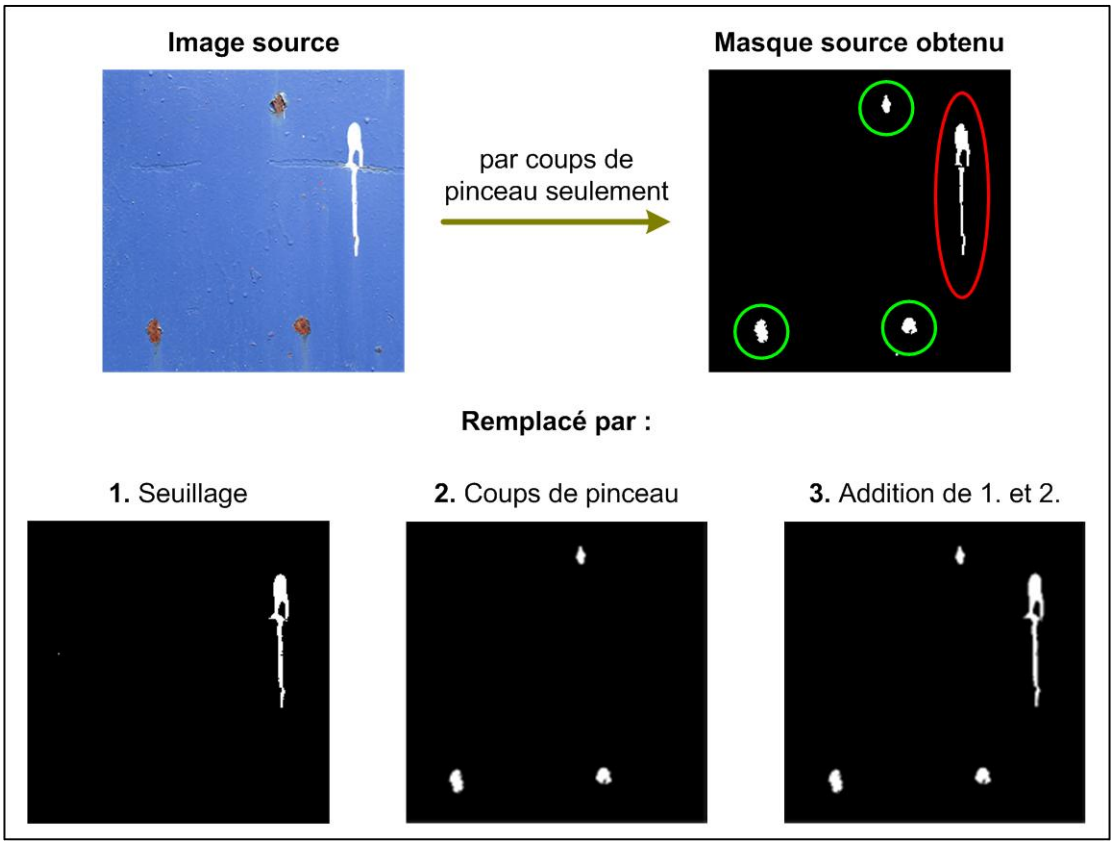

Figure 2.6 Segmentation par combinaisons de techniques.

2.2.4 Ajustements manuels

Puisque l'étape de segmentation initiale est cruciale au bon fonctionnement du reste du processus, l'artiste a donc la responsabilité d'en inspecter le résultat et d'apporter des correctifs au besoin. En effet, dans le cas d'effets plus complexes, il peut arriver qu'aucune des techniques présentées précédemment n'arrive à produire automatiquement une identification acceptable. Dans ces situations, l'artiste a la possibilité de faire des ajustements manuels en utilisant des outils standards de dessin : le pinceau, la gomme à effacer et des opérateurs morphologiques. La figure 2.7 montre un exemple d'une telle situation.

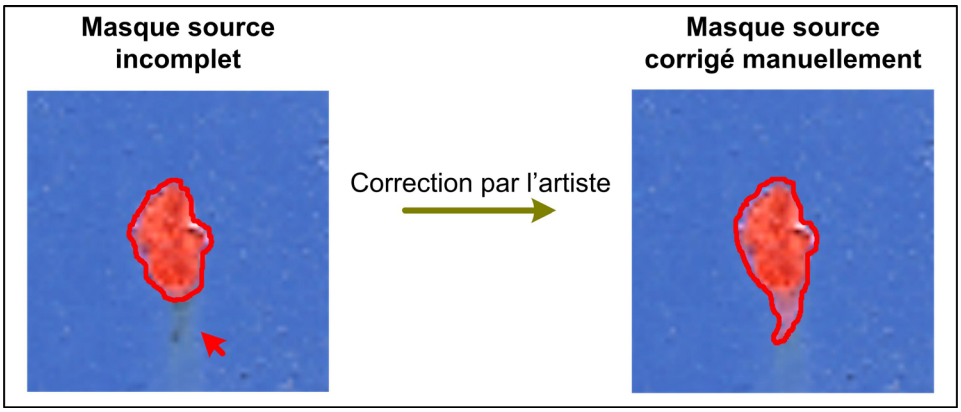

Figure 2.7 Segmentation par ajustements manuels.

Pour lui simplifier la tâche, le système affiche la segmentation à l'artiste sous forme de couche rouge semi-transparente apposée sur l'image source. De cette façon, la correspondance entre les pixels des deux images est mise en évidence et l'artiste peut détecter rapidement les régions problématiques. De plus, l'artiste est en mesure d'agrandir ces régions de manière à effectuer les corrections avec un niveau de précision approprié. Les opérateurs morphologiques disponibles (érosion et dilatation) permettent à l'artiste d'ajuster rapidement la taille des régions identifiées. En effet, il arrive souvent que les méthodes de segmentation automatique rencontrent des problèmes dans les transitions entre les régions détériorées et non-détériorées de l'image. En appliquant quelques érosions ou quelques dilatations, l'artiste peut facilement ajuster la précision de la segmentation produite par le système. Bref, ces outils permettent, en quelques minutes, de valider et de corriger le résultat de l'étape de segmentation pour assurer le bon fonctionnement des étapes suivantes.

2.3 Étape d'élimination

À partir de l'image source et du masque source produit par l'étape de segmentation, il est maintenant essentiel d'éliminer les effets de détérioration initiaux pour préparer la synthèse de nouveaux effets. Tel que mentionné précédemment, l'étape d'élimination consiste à produire une image dite nettoyée, qui n'est rien d'autre que l'image source débarrassée de toutes traces de détérioration. Il s'agit d'une étape complètement automatique qui utilise un algorithme de synthèse de texture par remplissage de trous. Tous les détails relatifs au fonctionnement de cet algorithme sont fournis dans cette section.

2.3.1 Survol de l'algorithme

L'algorithme de synthèse de texture utilisé durant l'étape d'élimination est inspiré et adapté des travaux de Efros et Leung (1999) ainsi que de ceux de Hertzmann *et al.* (2001). Comme discuté à la section 1.2.1, il s'agit de deux approches fondamentales dans le domaine de la synthèse de texture qui reposent sur la théorie des « *Markov Random Fields (MRF)* » (Kindermann et Snell, 1980). Dans le contexte du projet, ces approches ont donc été ajustées pour la synthèse d'images réalistes en incorporant des caractéristiques spécifiques aux effets de détérioration et en améliorant les possibilités de synthèse par remplissage de trous. Tel qu'illustré à la figure 2.8, l'algorithme consiste à parcourir l'image source dans le but de remplacer les pixels identifiés par le masque source. Les pixels de remplacement sont sélectionnés dans le reste de l'image source, c'est-à-dire parmi ses pixels non-détériorés. Pour choisir le bon pixel de remplacement, l'algorithme utilise le voisinage du pixel à remplacer (aussi appelé la « fenêtre ») de façon à trouver la meilleure correspondance en le comparant aux candidats potentiels.

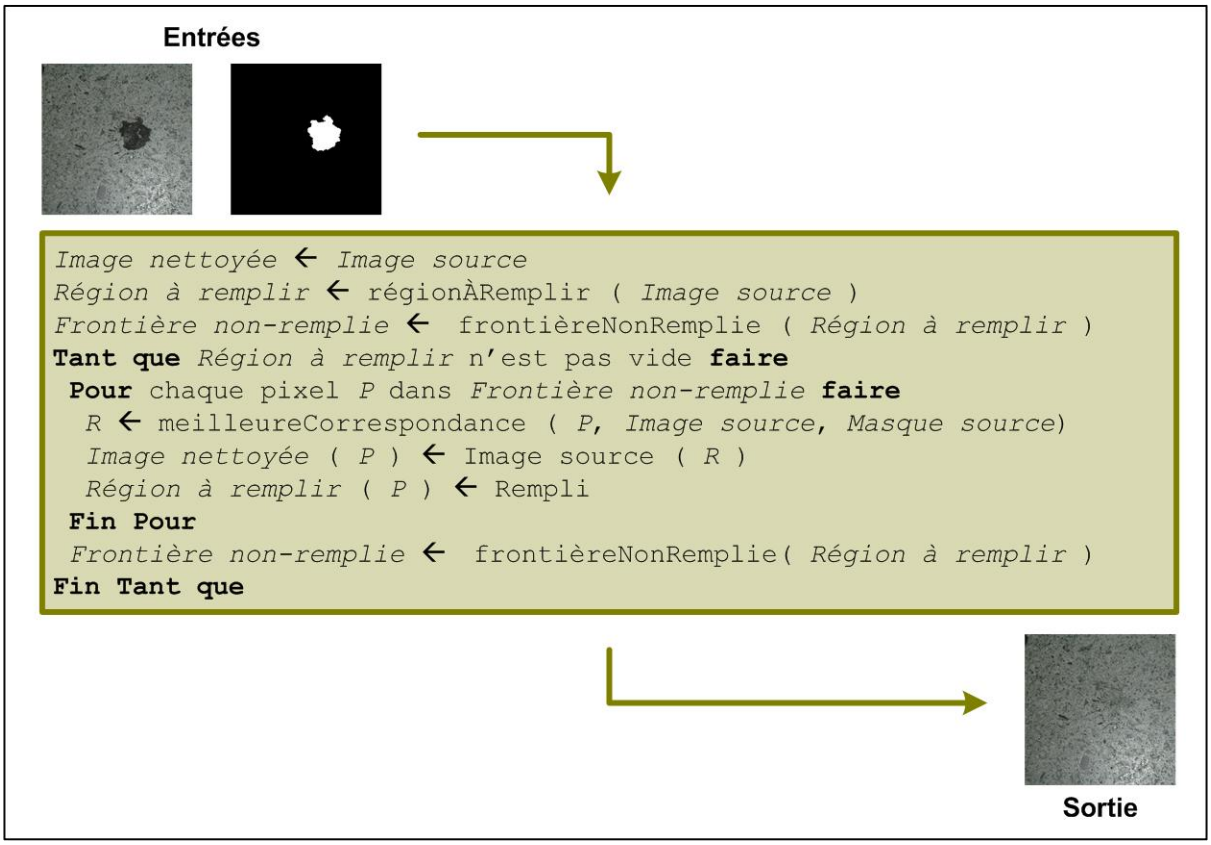

Figure 2.8 Pseudo-code de l'algorithme d'élimination.

Bien que les concepts derrière cet algorithme soient relativement simples, deux aspects fondamentaux se doivent d'être étudiés plus attentivement. Premièrement, il est essentiel d'approfondir la relation entre l'ordre de remplissage utilisé et la qualité des résultats obtenus. En effet, il est évident que la séquence employée lors du remplissage aura un impact sur le résultat final puisque les pixels de remplacement sont sélectionnés à partir du voisinage d'un pixel. Ainsi, le premier remplacement influencera, s'il est dans son voisinage, le calcul du second et ainsi de suite. Les enjeux de l'ordre de remplissage sont détaillés dans la section 2.3.2. Ensuite, la manière de trouver la meilleure correspondance, qui représente le coeur de l'algorithme, soulève plusieurs questionnements. Quels pixels dans le voisinage doivent être considérés? Est-ce qu'une recherche exhaustive est envisageable? Conséquemment, cet aspect de l'algorithme est détaillé à la section 2.3.3.

2.3.2 Synthèse par remplissage de trous

Avant d'étudier plus en détails la façon de déterminer les meilleurs pixels de remplacement, il est important d'examiner l'ordre de remplissage à privilégier selon le contexte. Comme illustré à la figure 2.8, l'algorithme parcourt la région à remplir tant que tous ses pixels n'ont pas été traités. Mais quelle est la séquence de traitements idéale pour obtenir les meilleurs résultats possibles? L'approche séquentielle traditionnelle, aussi appelée *« scan-line »*, consiste à débuter la séquence en haut à gauche et de se déplacer colonne par colonne puis ligne par ligne. En privilégiant cette séquence de traitements, il est possible d'utiliser une fenêtre constante en forme de « *L* » sur le côté (voir figure 2.9), puisqu'il est juste de prendre pour acquis que ces pixels voisins auront été préalablement traités.

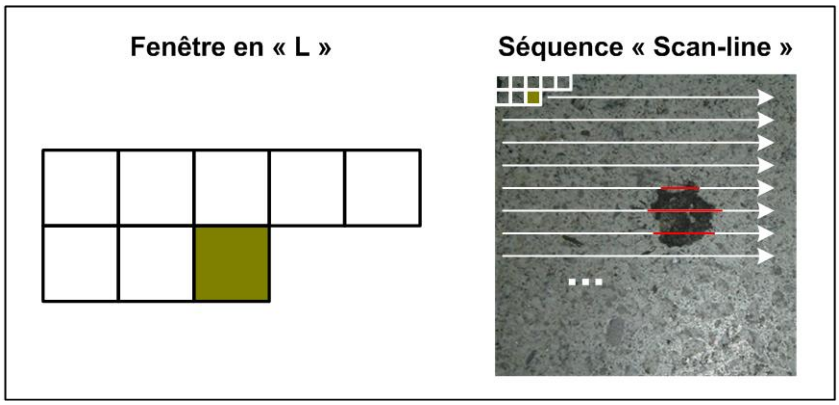

Figure 2.9 Fenêtre en « *L* » et remplissage « *scan-line* ».

Cette façon de faire fonctionne correctement lorsque l'objectif est de générer complètement une nouvelle texture. Cependant, durant l'étape d'élimination, l'objectif est de faire la synthèse d'une portion spécifique de l'image. Dans cette situation, l'approche traditionnelle n'est plus adéquate puisqu'elle ne considère que le voisinage supérieur gauche, et ce, pour tous les pixels à remplacer. Par conséquent, certaines discontinuités peuvent apparaître en bas à droite de la région à traiter comme le montre la figure 2.10.

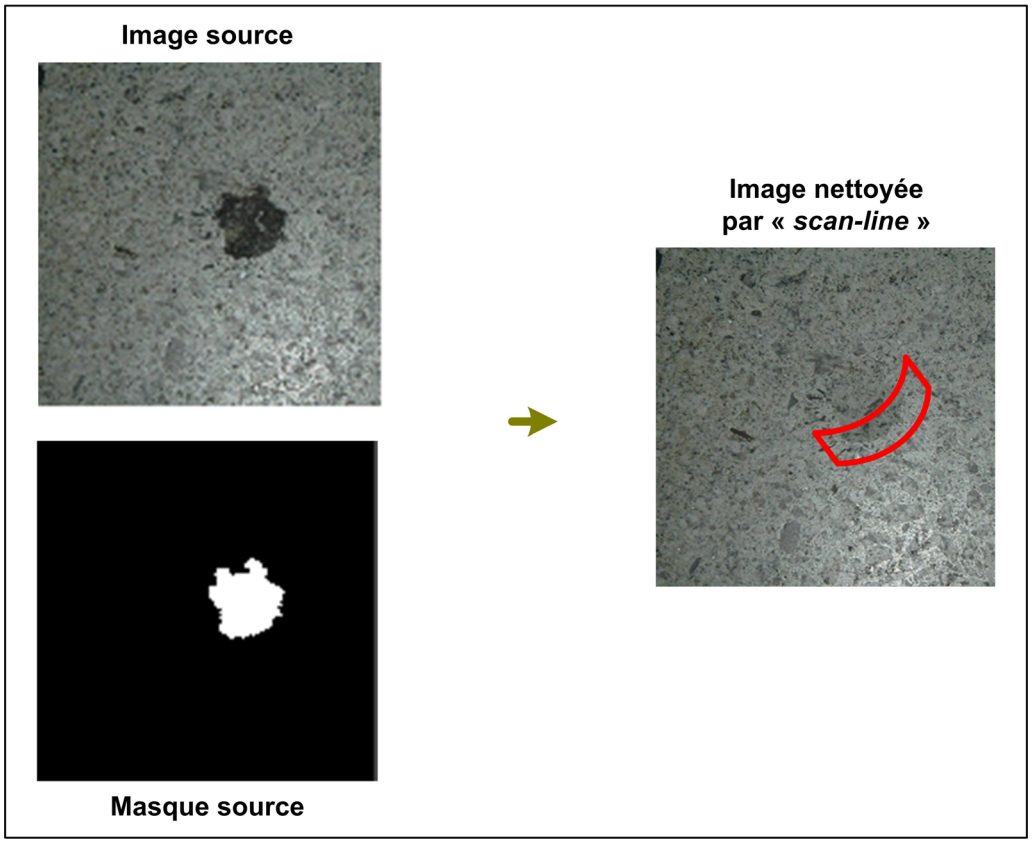

Figure 2.10 Cas de discontinuité causée par le remplissage « *scan-line* ».

Pour régler ce problème, il serait préférable de modifier la séquence traditionnelle de traitements de manière à remplacer d'abord les pixels formant la bordure de la région détériorée. En effet, parmi tous les pixels à remplacer, ceux-ci possèderont initialement le maximum de pixels non-détériorés dans leur voisinage, ce qui permettra de trouver une meilleure correspondance dans le reste de l'image source. Ainsi, en remplissant itérativement, couche par couche, la bordure de la région détériorée, un peu comme si on pelait un oignon, il sera possible de maximiser le contexte non-détérioré dans le voisinage de chaque pixel à nettoyer. La figure 2.11 illustre le fonctionnement de cette approche par remplissage de trous pour une itération donnée.

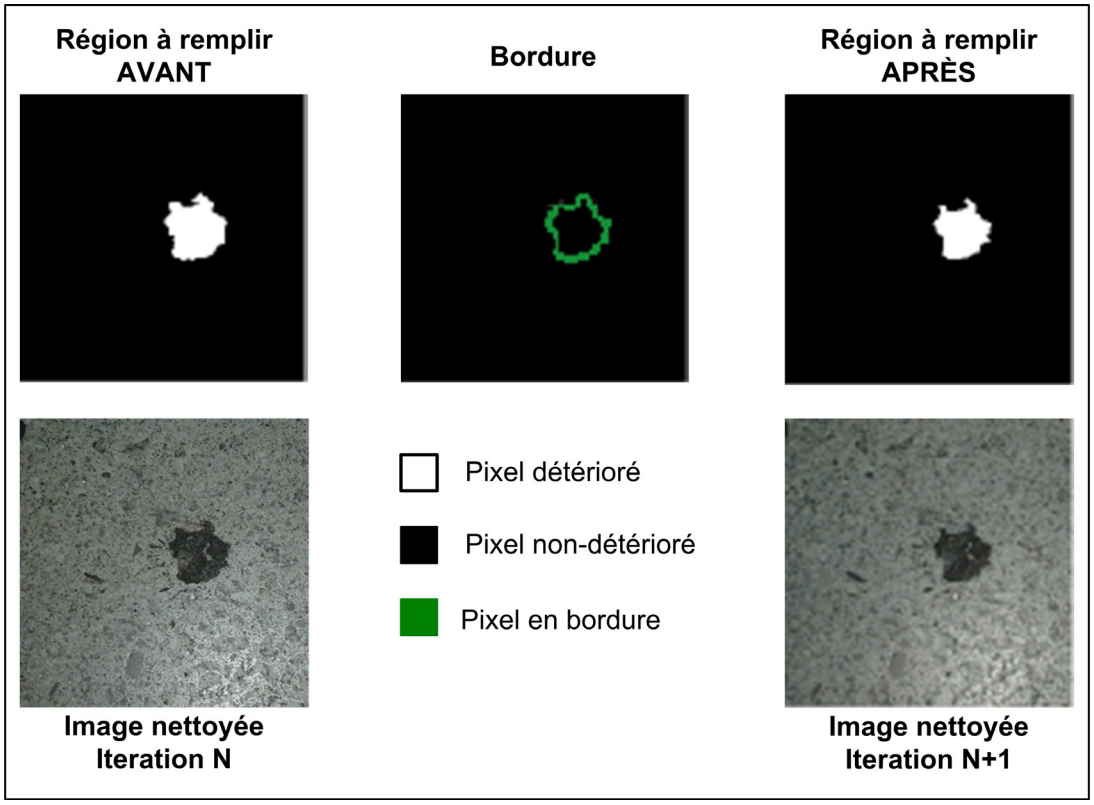

Figure 2.11 Une itération de l'approche de synthèse par remplissage de trous.

Le problème avec cette approche, c'est que le voisinage à considérer pour déterminer la meilleure correspondance (la fenêtre) n'est pas constant pour tous les pixels. Par exemple, pour un pixel situé en bas à droite dans la région à remplacer, il faudrait utiliser les pixels non-détériorés positionnés en bas à droite. Similairement, pour un pixel situé en haut à droite, il faudrait considérer les pixels non-détériorés positionnés en haut à droite. Ainsi, la fenêtre en traditionnelle « L » présentée à la figure 2.9 ne s'applique pas avec cette approche par remplissage de trous. Pour remédier au problème, l'approche proposée consiste à considérer quatre fenêtres potentielles pour la recherche et d'en sélectionner une pour chaque pixel, en fonction des pixels non-détériorés dans son voisinage lors de son traitement. La figure 2.12 montre les quatre fenêtres potentielles utilisées dans le système ainsi que la distribution de leur utilisation par pixel de la région à remplacer. Selon la forme du masque à traiter, il est possible qu'aucune de ces fenêtres ne renferme que des pixels non-détériorés. Dans cette situation, le pixel en question sera ignoré et traité à l'itération suivante.

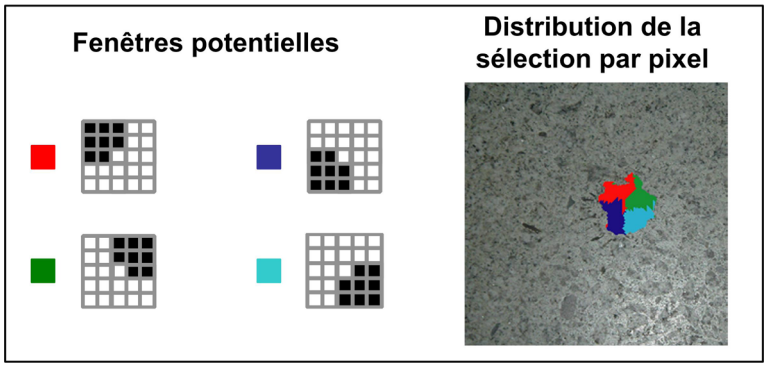

Figure 2.12 Les fenêtres potentielles et leur utilisation.

En somme, cette séquence de remplissage permet de maximiser le contexte non-détérioré dans le voisinage du pixel en traitement de façon à trouver la meilleure correspondance possible. De plus, puisque les quatre fenêtres utilisées se superposent partiellement, cette approche permet d'éliminer les effets de discontinuités produits par la séquence de remplissage « *scan-line* » et la fenêtre en « *L* » traditionnelle (voir figure 2.13).

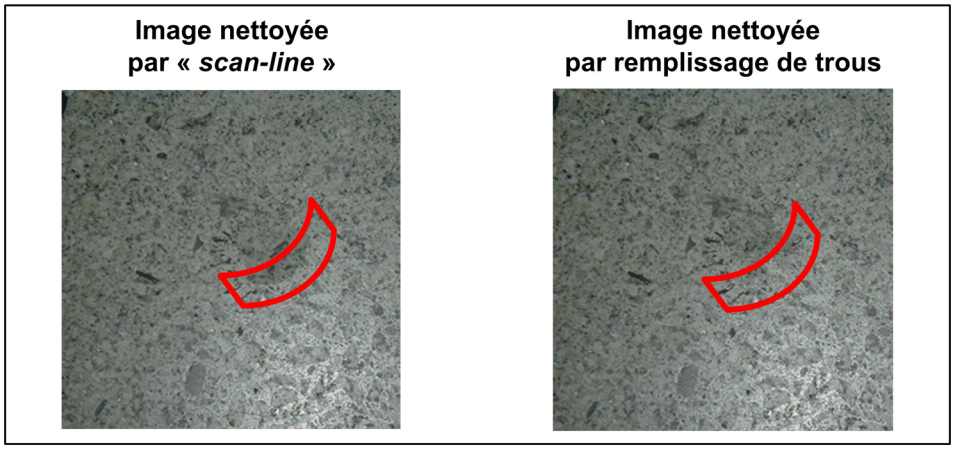

Figure 2.13 Comparaison selon la séquence de remplissage utilisée.

2.3.3 Recherche des meilleures correspondances

Après avoir déterminé la séquence de remplissage préférable, il est important de bien définir la façon de sélectionner les pixels de remplacement pour « nettoyer » la région détériorée. Tel que mentionné précédemment, l'algorithme de synthèse recherchera, pour chaque pixel de la région à corriger, un pixel dans le reste de l'image source pour lequel le contexte (le voisinage) correspondra le mieux possible à celui du pixel à remplacer. Conséquemment, il est nécessaire de spécifier quelle sera la mesure permettant de quantifier cette correspondance. Or, la norme L_2, aussi appelée norme euclidienne, est couramment utilisée dans plusieurs algorithmes de synthèse de textures (Efros et Leung, 1999; Hertzmann *et al.*, 2001) et permet généralement d'obtenir d'excellents résultats. Lorsqu'appliquée à notre contexte, l'équation ci-dessous est dérivée, où *f* représente la fenêtre sélectionnée, *s* la valeur RVB du pixel dans le voisinage du pixel à remplacer et *r* la valeur RVB du pixel dans le voisinage du pixel de remplacement.

$$\sum_{(i,j)\in f} (s(i,j) - r(i,j))^2 \tag{2.4}$$

Évidemment, pour trouver la meilleure correspondance possible, l'algorithme aura comme objectif de minimiser cette mesure de distance parmi tous les pixels candidats contenus dans les régions non-détériorées de l'image source. Maintenant que la mesure à considérer est définie, il faut déterminer comment l'algorithme trouvera le pixel la minimisant. À première vue, puisque l'étape de segmentation présentée à la section 2.2 permet de bien identifier les pixels des régions non-détériorées, une recherche séquentielle exhaustive serait très simple à effectuer et déterminerait de façon absolue la meilleure correspondance. Malheureusement, en utilisant des textures 2D de taille habituelle avec un ratio de pixels détériorés typique, les temps de calculs nécessaire pour effectuer une telle recherche sont inacceptables. En effet, avec de tels paramètres, le processus de synthèse s'étale sur plusieurs heures, ce qui implique que l'artiste doit souvent attendre après les résultats. Dans le contexte d'un studio de production où l'utilisateur doit faire réviser ses résultats par ses pairs ou un directeur artistique, il doit être en mesure d'ajuster rapidement l'apparence de ses objets selon les

commentaires ainsi recueillis. Par conséquent, le système doit faire la synthèse dans un délai interactif.

Pour y parvenir, l'approche proposée exploite une méthode développée par Arya *et al.* (1998) permettant de faire une recherche approximative. En effet, *Library for Approximate Nearest Neighbor Searching* (ANN) propose un algorithme de recherche basé sur une structure en arbre. Cette librairie a déjà fait ses preuves dans le domaine de la synthèse de textures et est utilisée par plusieurs méthodes de complétion d'images. Pour déterminer la meilleure correspondance entre un élément donné et un ensemble quelconque, leur algorithme calcule une norme L_2 sur un vecteur de caractéristiques de taille fixe. Pour faire correspondre ce calcul à la métrique présentée précédemment, ce vecteur de caractéristiques est formé des valeurs RVB des pixels non-détériorés correspondant à la fenêtre sélectionnée pour faire le remplacement. Plusieurs paramètres peuvent venir ajuster le fonctionnement de cet algorithme tels que la marge d'erreur acceptable pour l'approximation ainsi que le type d'arbre utilisé. Pour plus de détails sur le fonctionnement interne de la librairie, veuillez vous référer à l'article d'Arya *et al.* (1998).

Concrètement dans l'application, l'étape initiale consiste à construire les quatre structures de recherche utilisées par ANN correspondant aux fenêtres potentielles présentées à la figure 2.12. Ce traitement ne doit être fait qu'une seule fois au tout début du processus de synthèse. Ensuite, au moment de traiter un pixel donné, il est d'abord nécessaire de sélectionner la fenêtre qui sera considérée pour déterminer la correspondance. À ce moment, le vecteur de caractéristiques est rempli en fonction de la fenêtre sélectionnée. Finalement, le vecteur est fourni à ANN qui trouve une approximation de la meilleure correspondance dans la structure de recherche appropriée. Une analyse sur les performances découlant de l'utilisation d'un tel processus de recherche sera présentée à la section 2.6.

2.4 Étape de reproduction

À partir de l'image source, du masque source produit par l'étape de segmentation, de l'image nettoyée engendrée à l'étape d'élimination et du masque cible fourni par l'utilisateur, le système est maintenant prêt à passer à l'étape de reproduction pour générer de nouveaux effets de détérioration similaires. Tel que mentionné précédemment, l'étape de reproduction consiste à produire une image dite éditée, qui n'est rien d'autre que l'image nettoyée sur laquelle de nouveaux effets correspondant au masque cible ont été ajoutés. Il s'agit d'une étape complètement automatique qui utilise encore une fois un algorithme de synthèse de texture par remplissage de trous. Tous les détails relatifs au fonctionnement de cet algorithme sont présentés dans cette section.

2.4.1 Survol de l'algorithme

L'algorithme de synthèse de texture utilisé durant l'étape de reproduction est en fait une variation de l'algorithme développé pour l'étape d'élimination. En effet, puisque que l'objectif est toujours de remplacer une portion des pixels de l'image, il est possible d'utiliser une approche similaire. Évidemment, certains rôles doivent être inversés puisqu'on veut maintenant détériorer la texture plutôt que de la réparer. Tel qu'illustré à la figure 2.14, l'algorithme consiste à parcourir l'image nettoyée dans le but de remplacer les pixels identifiés par le masque cible. Cette fois, les pixels de remplacement sont sélectionnés parmi les pixels détériorés de l'image source. Pour choisir le bon pixel de remplacement, l'algorithme utilise encore le voisinage du pixel à remplacer de façon à trouver la meilleure correspondance dans les candidats potentiels.

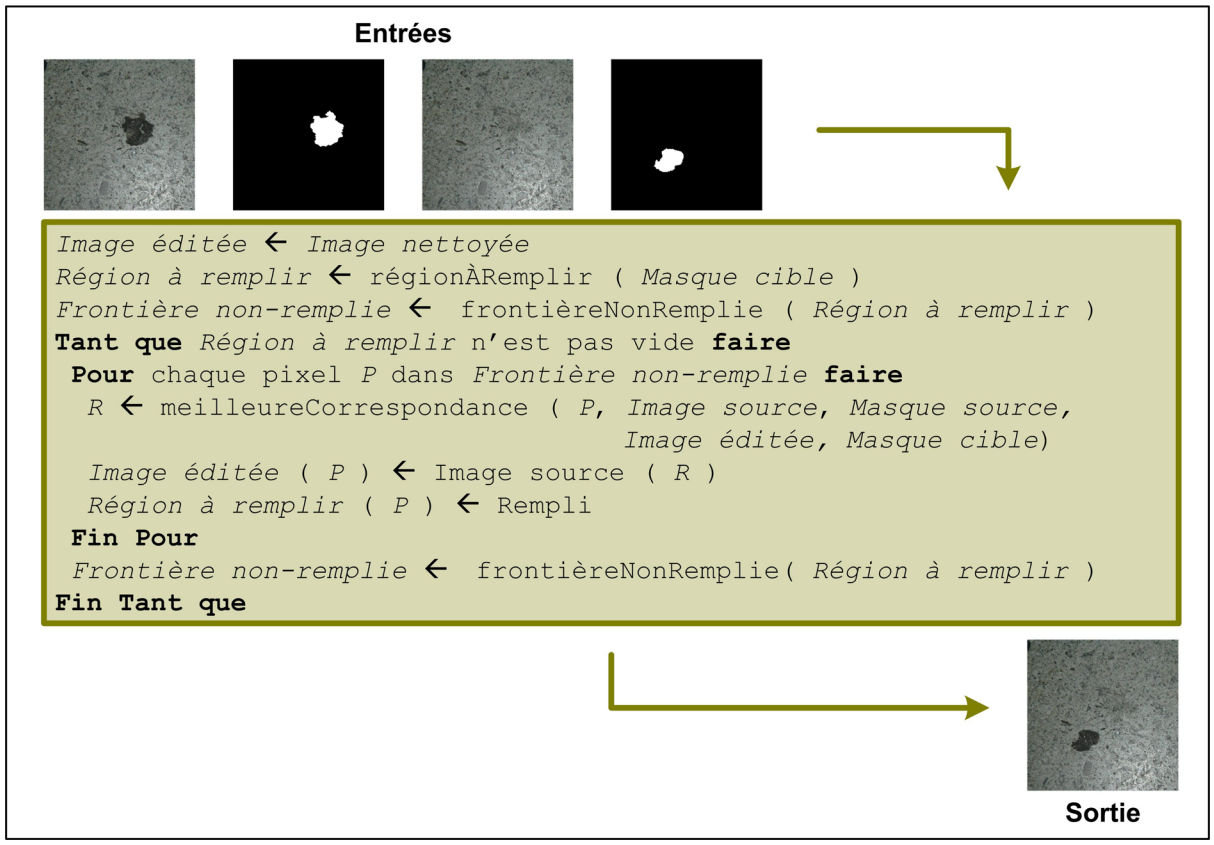

Figure 2.14 Pseudo-code de l'algorithme de reproduction.

Puisqu'il faut encore remplacer une portion spécifique de l'image, l'approche par remplissage de trous détaillée à la section 2.3.2 est encore exploitée pour définir la séquence de remplacement des pixels. Cependant, le processus de recherche des meilleures correspondances doit être légèrement altéré pour s'adapter au contexte spécifique de l'étape de reproduction.

2.4.2 Recherche des meilleures correspondances

À cette étape, des changements sont de mise pour faire fonctionner correctement l'algorithme de synthèse de textures original. Tout d'abord, il est nécessaire de changer la banque de candidats potentiels pour le remplacement. En effet, puisque l'algorithme cherche maintenant à ajouter des effets de détérioration, il doit nécessairement faire sa recherche de correspondances parmi les pixels détériorés de l'image source, et non l'inverse comme c'était

le cas à l'étape d'élimination. De plus, la métrique utilisée pour mesurer la correspondance entre deux contextes est légèrement modifiée. En plus de considérer la couleur des pixels avoisinant, l'algorithme prend aussi en compte le contexte binaire des masques, comme l'illustre la figure 2.15.

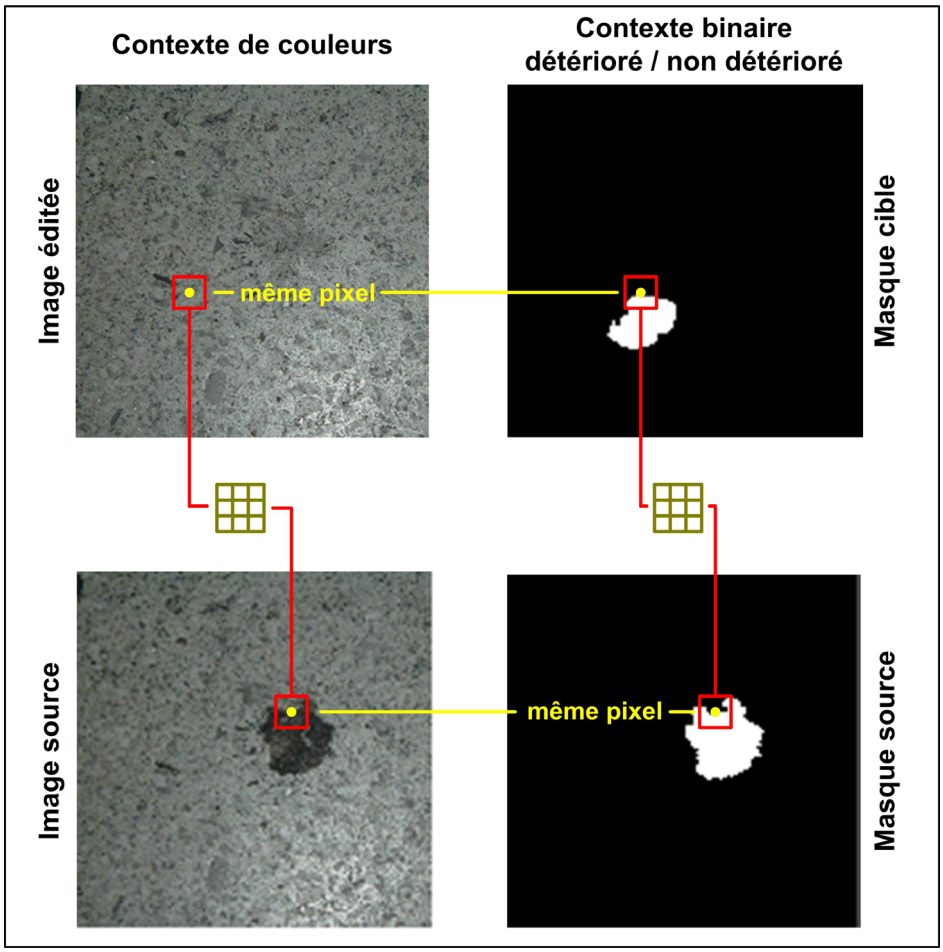

Figure 2.15 Le nouveau terme dans la mesure de correspondance.

En ajoutant ce terme à la mesure de correspondance, l'algorithme ne se base plus uniquement sur la couleur des pixels voisins mais aussi sur le contexte binaire « détérioré » ou « non-détérioré ». Cela permet de recréer plus fidèlement les motifs internes des effets de détérioration. En effet, pour générer la bordure d'un nouvel effet, l'algorithme favorisera un pixel de remplacement en bordure de l'effet contenu dans l'image source. Similairement, l'algorithme préférera utiliser un pixel interne, ou de milieu de région, lorsqu'il génère un

pixel équivalent pour le nouvel effet. La métrique présentée à la section 2.3.3 est donc modifiée en conséquence.

$$\sum_{(i,j)\in f}(1-\alpha)(s(i,j)-r(i,j))^2+\alpha(s_m(i,j)-r_m(i,j))^2 \tag{2.5}$$

Dans cette équation, *f* représente la fenêtre sélectionnée, *s* la valeur RVB du pixel dans le voisinage du pixel à remplacer, *r* la valeur RVB du pixel dans le voisinage du pixel de remplacement, s_m la valeur binaire du pixel dans le masque source et r_m la valeur binaire du pixel dans le masque cible. De plus, pour contrôler la balance entre les deux termes de l'équation, un paramètre α, variant entre 0 et 1, a été ajouté. Lorsqu'il prend une valeur se rapprochant de zéro, la métrique favorisera les correspondances de couleurs RVB. À l'opposé, une valeur près de un favorisera les correspondances sur le contexte binaire des masques. Évidemment, l'objectif de l'algorithme est encore de minimiser cette mesure pour déterminer la meilleure correspondance possible parmi les pixels candidats.

2.5 Transferts et combinaisons d'effets

En utilisant le processus de synthèse tel que présenté jusqu'à maintenant, il est possible pour un artiste d'éditer le niveau de détérioration d'une texture via les étapes de segmentation, d'élimination et de reproduction. Cependant, en pratique, il arrive souvent que l'utilisateur veuille transférer des effets de détérioration d'une photographie vers une autre texture et même de combiner plusieurs effets sur une même image. Pour permettre une telle utilisation, chaque étape du processus peut être employée indépendamment. De cette façon, l'artiste peut passer par l'étape de segmentation pour chacun des effets qu'il veut ajouter. Ensuite, il peut utiliser itérativement l'étape de reproduction pour construire la texture désirée, comme le montre la figure 2.16.

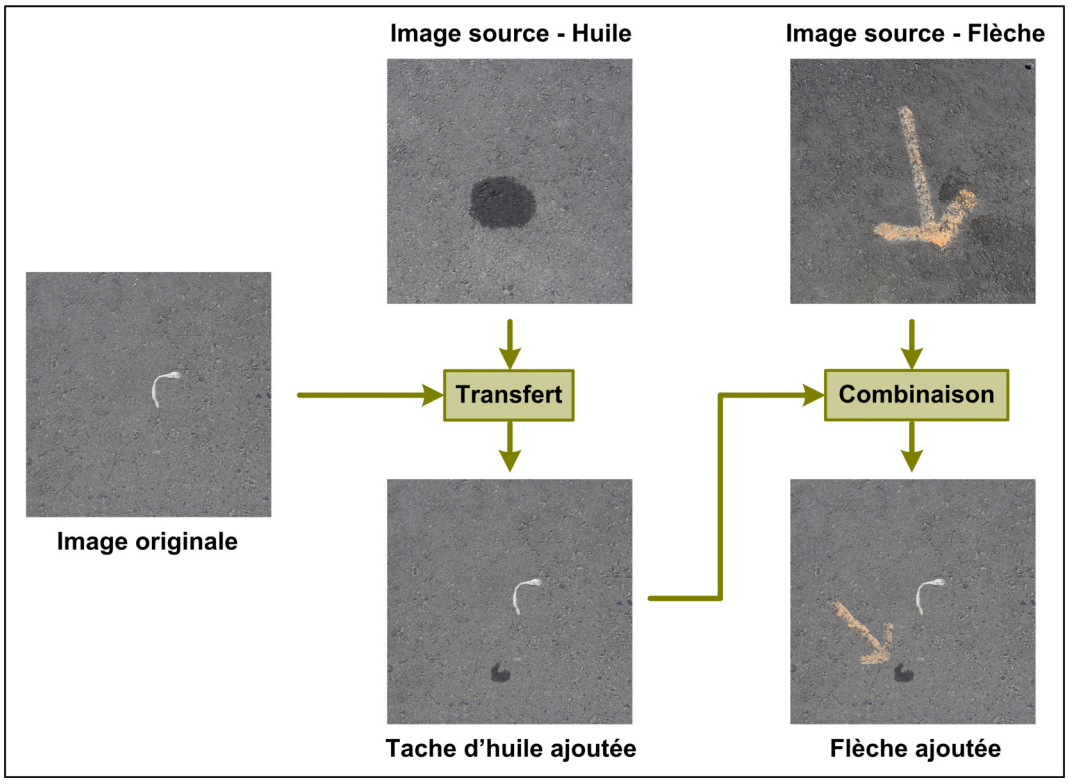

Figure 2.16 Exemple de transferts et de combinaisons d'effets.

Cette approche itérative fonctionne correctement puisque l'algorithme de synthèse ne génère de nouveaux pixels que pour les régions identifiées dans le masque cible, contrairement à certains travaux antérieurs (Gu *et al.*, 2006; Lu *et al.*, 2007; Wang *et al.*, 2006). Il s'agit donc d'une approche simple et efficace pour étendre l'utilisation du prototype à des situations concrètes.

2.6 Présentation des résultats

La présente section a comme objectif de présenter les résultats obtenus à l'aide du processus de synthèse d'effets de détérioration décrit dans ce chapitre. La figure 2.17 montre deux exemples de résultats, en grand format, alors que la figure 2.18 présente un éventail plus complet, en petit format. Finalement, pour souligner la polyvalence de la technique, la figure 2.19 expose un cas où la forme de la région détériorée est non triviale (du texte) et où le masque source et le masque cible se superposent.

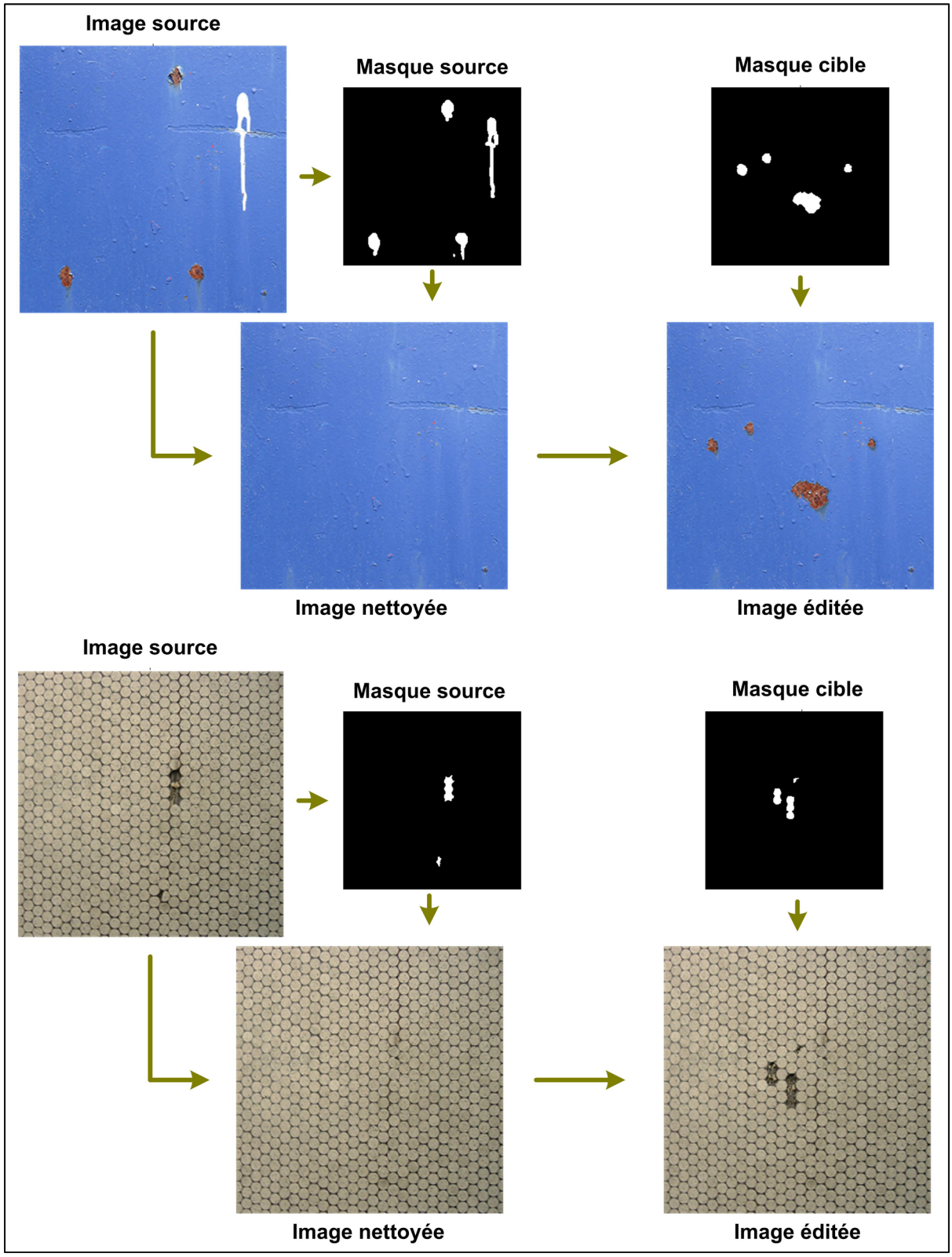

Figure 2.17 Résultats grand format du processus de synthèse d'effets de détérioration.

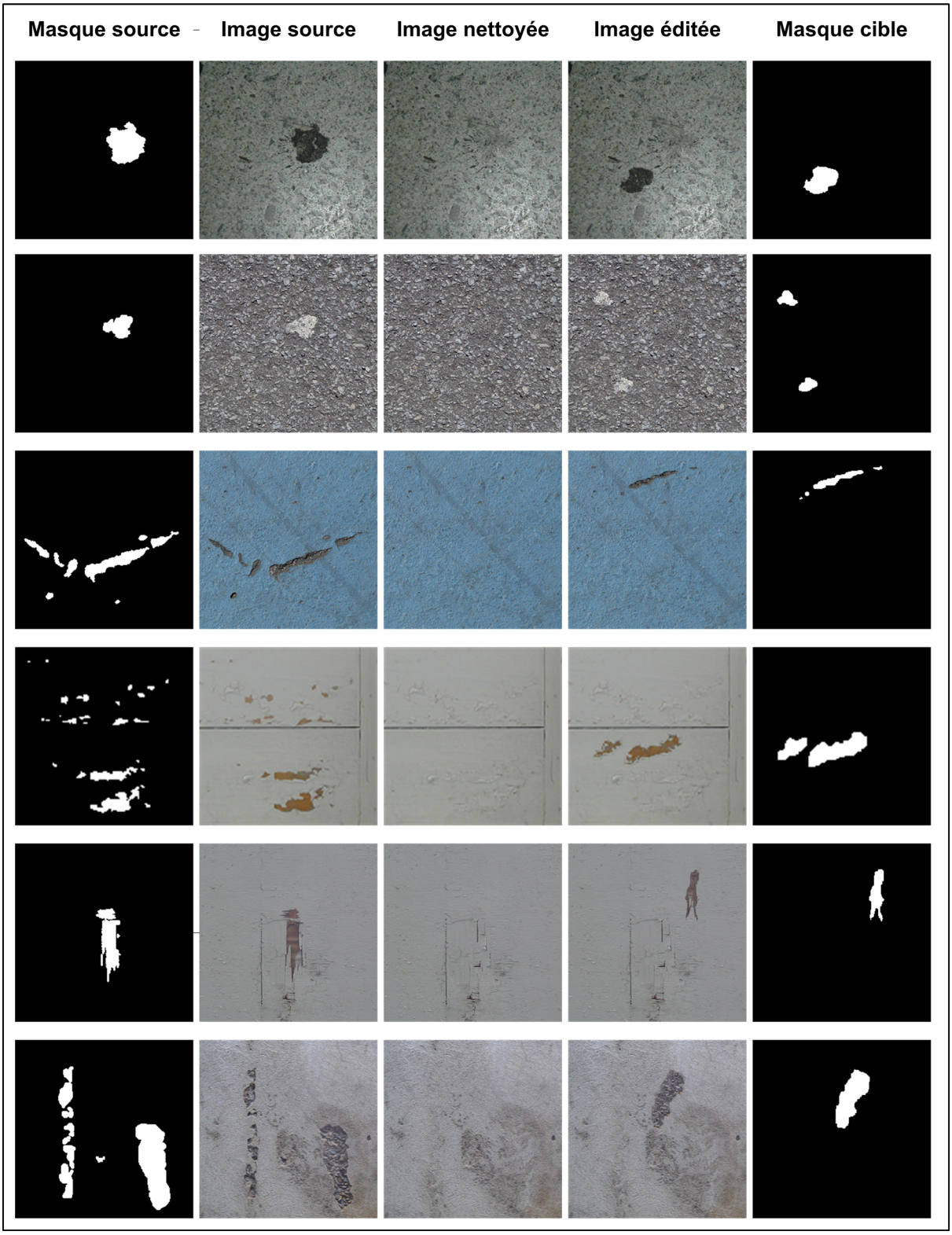

Figure 2.18 Résultats petit format du processus de synthèse d'effets de détérioration.

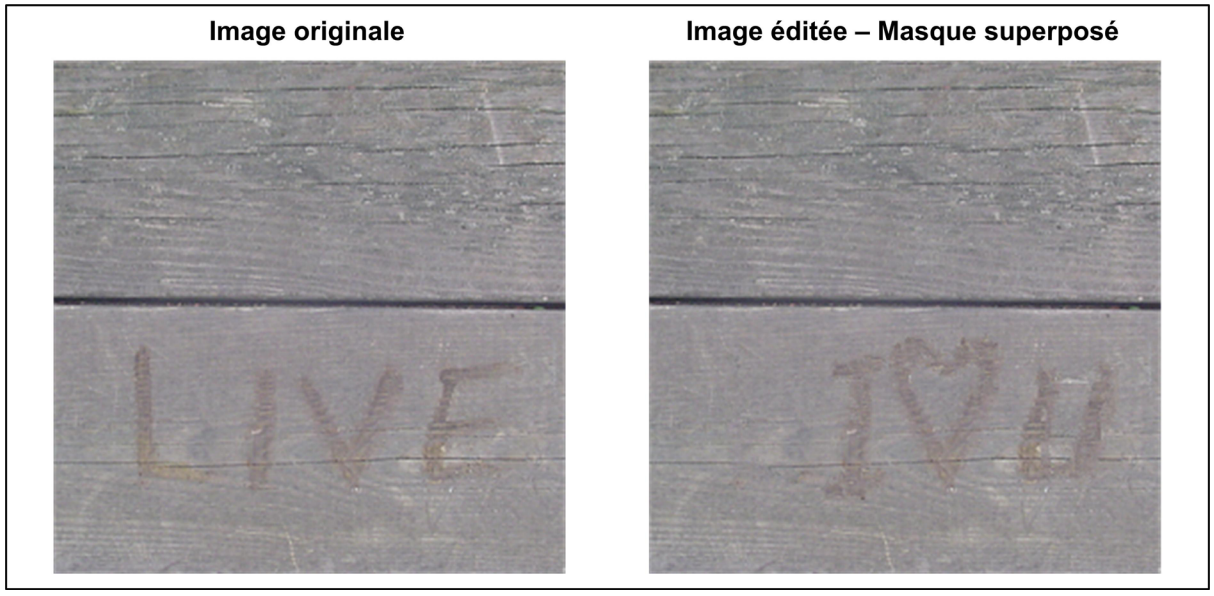

Figure 2.19 Résultats du processus de synthèse pour des masques superposés.

Comme on peut l'observer sur les figures présentées dans cette section, le processus de synthèse d'effets de détérioration fonctionne sur une grande variété de matériaux, tels que le bois, le métal, la céramique, le marbre, le gravier et le ciment. De plus, le processus fonctionne correctement peu importe la nature des effets de détérioration (rouille, graffiti, impact, écaillement, etc.). Tant et aussi longtemps que l'utilisateur fournit un exemple adéquat de l'effet désiré, le système sera en mesure de reproduire quelque chose de semblable, sans pour autant être identique. Finalement, la figure 2.20 et le tableau 2.1 regroupent les données pertinentes pour analyser la performance des différentes étapes de l'approche proposée. Pour une critique détaillée de l'ensemble du processus, veuillez consulter la section 2.7.

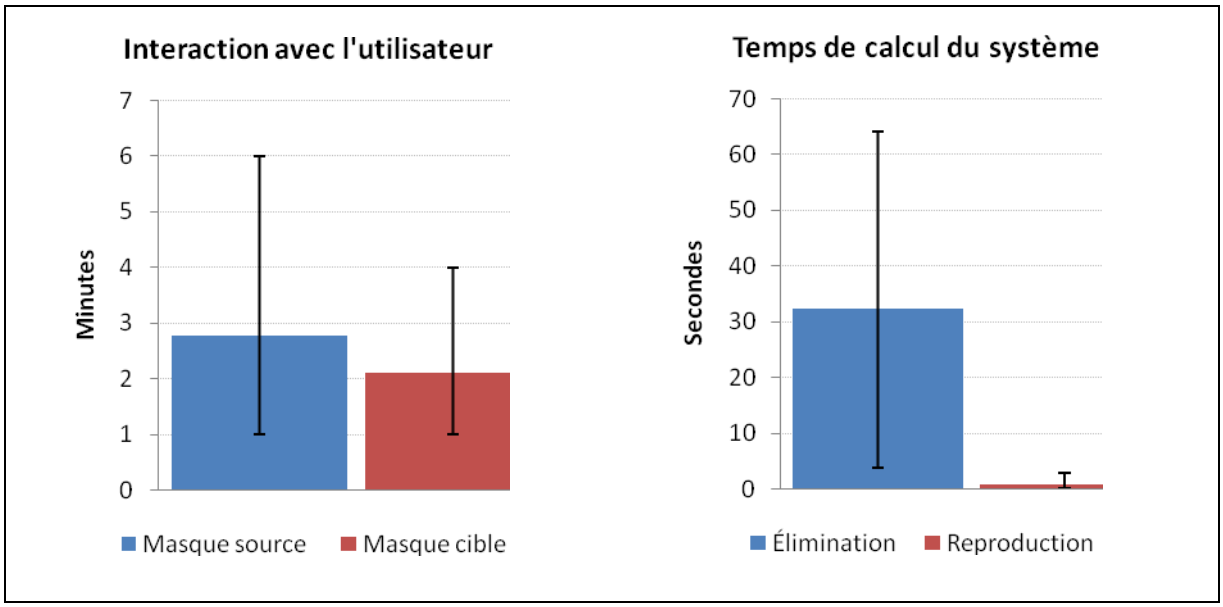

Figure 2.20 Le temps moyen et la dispersion pour les différentes étapes du processus.

Tableau 2.1 Données sur les performances du processus de synthèse

		Interaction avec l'utilisateur (minutes)		Temps de calcul du système (secondes)	
Résultats	**Dimension**	**Masque source**	**Masque cible**	**Étape d'élimination**	**Étape de reproduction**
Figure 2.17 (métal)	256x256	4	4	8.64	0.68
Figure 2.17 (céramique)	256x256	2	2	26.41	0.12
Figure 2.18 (marbre)	256x256	2	2	27.55	0.78
Figure 2.18 (gravier)	256x256	3	2	36.77	0.55
Figure 2.18 (ciment bleu)	256x256	1	2	56.79	1.12
Figure 2.18 (bois blanc)	128x128	2	2	3.89	0.26
Figure 2.18 (bois gris)	256x256	2	2	12.75	0.27
Figure 2.18 (ciment gris)	256x256	6	1	64.08	3.00
Figure 2.19 (graffiti)	256x256	3	2	54.05	1.79

2.7 Discussion

Cette section contient un bilan du processus de synthèse d'effets de détérioration à l'aide d'exemples décrit dans ce chapitre. Une discussion divisée en deux volets permet de mettre en évidence les avantages de la méthode proposée, ainsi que ses limitations.

2.7.1 Les avantages

Lorsque comparée avec une grande partie des travaux antérieurs dans le domaine, l'approche proposée se démarque par le fait qu'elle est parfaitement adaptée pour une utilisation par un artiste. En effet, dans un studio de productions, l'utilisateur typique sera inévitablement un artiste, sans connaissances scientifiques particulières. Or, plusieurs méthodes de détérioration présentées à la section 1.1 nécessitent la manipulation de paramètres physiques complexes pour contrôler l'apparence finale du résultat. Bien que ces méthodes parviennent à produire d'excellents résultats, elles sont plus souvent qu'autrement laissée de côté par les artistes qui préfèrent accomplir la tâche manuellement, malgré les délais importants associés. Notre approche basée sur l'utilisation d'exemples est intuitive et conviviale pour un artiste puisque les seuls paramètres à fournir sont des images. C'est un langage avec lequel ils sont familiers, ce qui les rend beaucoup plus enclin à utiliser la technique. Il est aussi important de mentionner que notre méthode est très polyvalente et que, contrairement à plusieurs autres techniques (Aoki *et al.*, 2004; Bosch *et al.*, 2004; Paquette, Poulin et Drettakis, 2001; 2002), elle ne se limite pas à un seul type de détérioration.

De plus, l'approche proposée s'intègre adéquatement dans le processus de création itérative préconisé dans les studios de production (voir figure 1.4). Comme le montre le tableau 2.1 et la figure 2.20, les temps de calcul associés au processus sont très rapides et impliquent des délais que l'on peut qualifier d'interactifs. L'utilisateur n'a jamais à attendre plusieurs heures avant de pouvoir réviser l'apparence de son résultat et peut donc facilement appliquer les correctifs proposés par ses pairs ou par son directeur artistique. L'intégration de notre approche est donc transparente puisqu'elle ne demande pas de modifications au processus de

création déjà en place. Par ailleurs, le processus de capture des exemples est simplifié au maximum (une photographie ordinaire), ce qui n'implique aucun coût additionnel pour le studio, par opposition à d'autres techniques basées sur la capture d'exemples (Gu *et al.*, 2006; Lu *et al.*, 2007).

Bien que l'approche proposée soit intuitive pour les artistes et qu'elle s'intègre bien à l'intérieur d'un studio de productions, il est aussi très important de commenter la qualité des résultats qu'elle produit. Malheureusement, il n'existe pas de métrique reconnue pour mesurer le niveau de réalisme d'une image de synthèse. Il est donc assez difficile de quantifier la qualité de nos résultats. Par contre, puisque notre approche cherche à reproduire un exemple, une excellente base de comparaison est disponible pour chaque résultat obtenu. Une inspection visuelle des figures 2.17, 2.18 et 2.19 permet rapidement de constater que la ressemblance entre l'exemple contenu dans l'image source et les nouveaux effets de détérioration générés sur l'image éditée est toujours très forte. Une telle inspection visuelle ne permet pas de quantifier le niveau de réalisme des images produites mais plutôt de vérifier que le résultat produit correspond à l'exemple fourni. Puisque cet exemple provient généralement d'un cas réel, il est juste de déduire que si les nouveaux effets sont semblables, ils seront réalistes. Évidement, d'autres facteurs comme la forme des effets et leur positionnement sur l'objet à user (paramètres contrôler par l'utilisateur via le masque cible) influenceront grandement l'impression de réalisme.

2.7.2 Les limitations

Malgré ses nombreux avantages, l'utilisation de l'approche proposée présente aussi quelques aspects négatifs. Tout d'abord, notre méthode est soumise à quelques limitations. En effet, puisqu'elle utilise des textures 2D pour représenter les effets de détérioration à la surface d'un objet, elle ne permet pas de travailler avec des phénomènes qui modifieraient la géométrie 3D de cet objet. Par exemple, des déformations majeures ou même des fractures associées à un impact important ne peuvent pas être reproduit à l'aide de notre méthode. Elle se limite donc aux effets de surface qui ne changent pas significativement la forme ou les dimensions d'un objet.

Ensuite, il est important d'analyser un peu plus en détails les paramètres nécessaires à notre méthode, soit l'image source et le masque cible. Il a été mentionné précédemment que ces paramètres sont intuitifs pour un artiste et qu'il s'agissait d'un point fort de la technique. Par contre, selon le cas, la création du masque cible peut devenir longue et pénible. Si l'artiste veut ajouter une simple tache isolée sur une table, le masque cible correspondant sera très simple à produire. À l'opposé, s'il désire détériorer sévèrement un objet en lui ajoutant des centaines de petites égratignures, le masque cible sera beaucoup moins facile à créer. En plus, si l'artiste doit produire un environnement contenant plusieurs occurrences différentes de cet objet, il devra répéter le travail plusieurs fois. Dans plusieurs cas, la création du masque cible représente donc un goulot d'étranglement du processus et en réduit l'efficacité générale. Une solution à ce problème est proposée au chapitre 3.

Par ailleurs, il est important de mentionner que, selon le cas, l'étape de segmentation semi-automatique présentée à la section 2.2 peut elle aussi devenir un goulot d'étranglement important. En effet, dans certaines situations (plusieurs effets différents sur l'image source, contraste faible entre les régions détériorées et non-détériorées, etc.) la création du masque source par segmentation automatique devient assez difficile. L'artiste doit donc corriger manuellement cette segmentation ce qui accroît le temps requis par le processus.

Finalement, un dernier aspect négatif de l'approche proposée concerne la quantité d'exemples (images sources) nécessaires pour posséder une base de données d'effets représentative. Pour générer un effet sur un objet, l'artiste doit absolument avoir un exemple correspondant. Cependant, un problème survient lorsque l'artiste possède un exemple de l'effet qu'il veut ajouter qui n'est pas de la couleur désirée. À titre d'exemple, un artiste pourrait vouloir ajouter une tache bleue sur un objet alors qu'il ne possède qu'un exemple pour une tache noire. Conséquemment, en plus d'avoir accès à un exemple pour chaque type d'effets (rouille, tache, poussière, etc.), l'artiste doit aussi en considérer la teinte, ce qui multiplie la quantité d'images sources nécessaires. Ce genre de situation peut devenir problématique, particulièrement si ce dernier ne parvient pas à trouver un cas réel de la teinte qui lui convient. Une solution à ce problème est proposée au chapitre 4.

CHAPITRE 3

GÉNÉRATION DE MASQUES À L'AIDE DE PROPRIÉTÉS LOCALES[2]

Comme mentionné antérieurement, un deuxième objectif de ce projet de recherche est d'automatiser davantage le processus d'édition d'effets de détérioration décrit au chapitre 2 en intégrant une approche de détérioration basée sur les propriétés locales d'un objet. Pour remplir cet objectif, l'approche proposée doit tirer profit de la relation entre le positionnement des effets de détérioration et de certaines propriétés géométriques de l'objet à user. Par ailleurs, la méthode développée doit permettre de générer rapidement plusieurs occurrences d'un même objet en plus de rester intuitive et simple d'utilisation pour un utilisateur typique. Le présent chapitre détaille les différentes étapes de l'approche proposée, expose les résultats obtenus et se conclut par une discussion sur les avantages et les limitations qui lui sont associés.

3.1 Présentation générale du processus

L'utilisation du processus de synthèse de détérioration présenté au chapitre 2 requiert la création manuelle, par un artiste, d'un masque cible pour contrôler le positionnement des effets sur l'objet à user (voir figure 2.1). Il s'agit d'une approche simple et efficace pour avoir un contrôle complet sur l'apparence finale de cet objet. Néanmoins, dans certaines situations, la création manuelle de ce masque peut s'avérer longue et pénible pour un artiste. En effet, si ce dernier désire détériorer un objet sévèrement en appliquant une grande variété d'effets différents, la création du masque cible n'aura rien de triviale. Même dans les cas plus simples, cette étape du processus constitue un goulot d'étranglement important (voir figure 2.20). De plus, pour produire plusieurs occurrences différentes d'un même objet, comme les

---

[2] Le contenu de ce chapitre fait l'objet d'une publication dans une revue scientifique internationale :
Clément, Olivier, et Eric Paquette. 2010. « Adaptable Aging Factory for Multiple Objects and Colorations ». *Computers & Graphics,* vol. 34, n° 4, p. 460-467.

chaises d'un ensemble de salle à dîner, l'artiste doit créer un masque cible distinct pour chaque instance, ce qui a pour effet de multiplier la quantité de travail nécessaire. De plus, il arrive souvent qu'un artiste désire user de façon similaire des objets différents, comme c'est le cas pour la table et les chaises d'un même ensemble de salle à dîner. Pour répondre à ces situations, l'objectif est donc d'éliminer l'étape manuelle de création des masques en tirant profit de la relation entre le positionnement des effets de détérioration et de certaines propriétés des objets à user. La figure 3.2 présente un survol des possibilités d'utilisation du système.

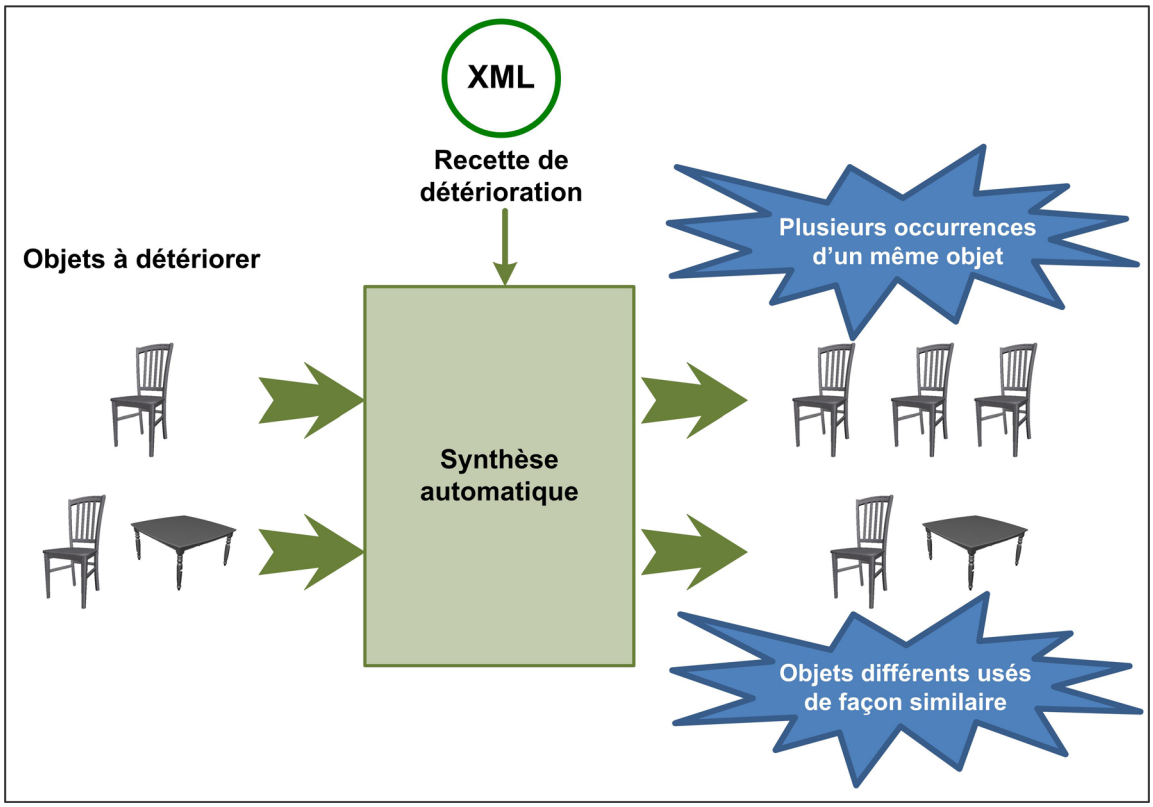

Figure 3.1 Présentation générale du processus de génération automatique.

Pour y parvenir, le processus proposé pour la génération automatique de masques cibles se décompose en trois étapes principales : le calcul des propriétés locales (voir section 3.2), la création de la recette de détérioration (voir section 3.3) puis la génération du masque cible (voir sections 3.4 et 3.5). La figure 3.2 présente le fonctionnement global du processus.

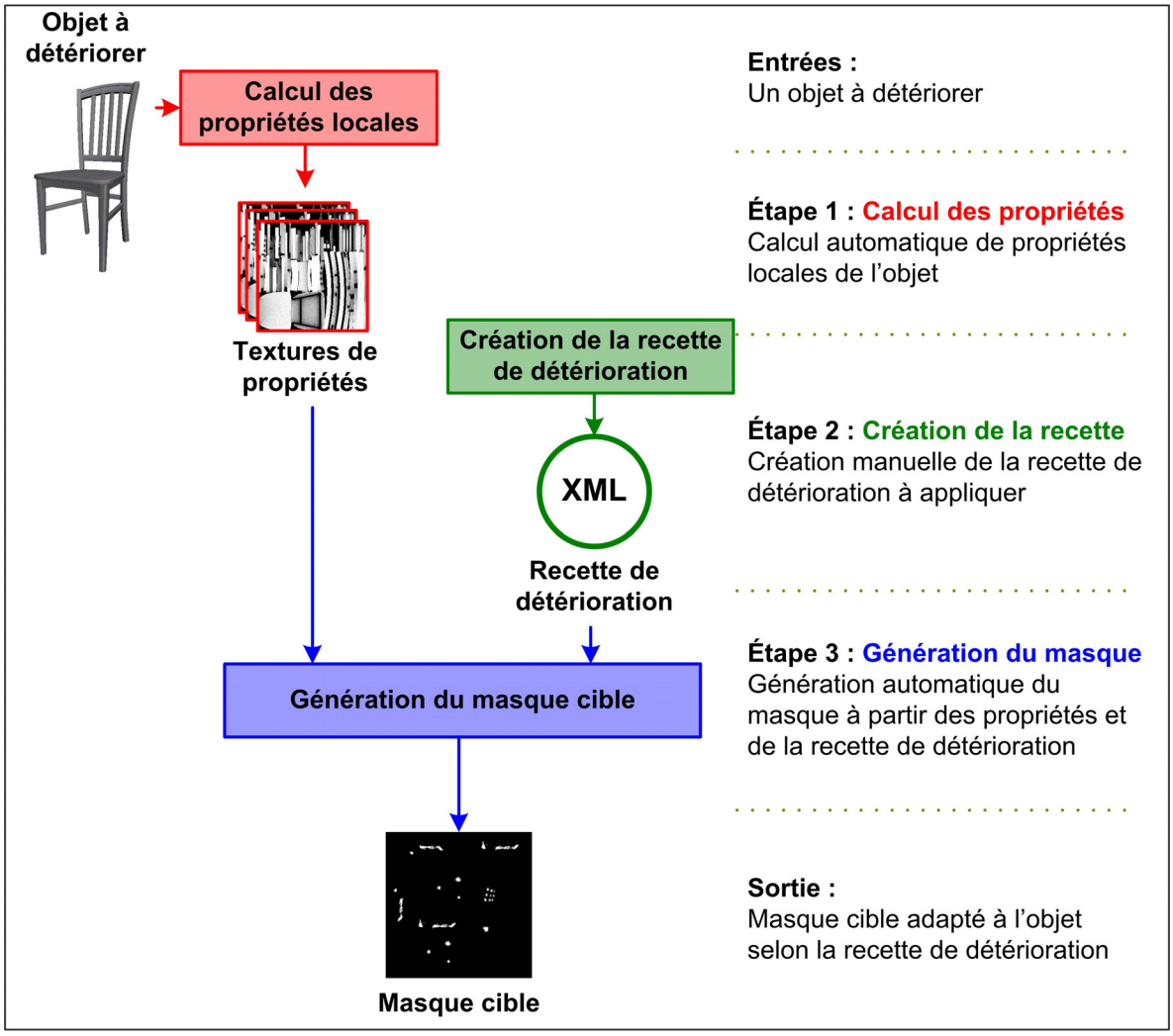

Figure 3.2 Présentation détaillée du processus de génération de masques cibles.

Finalement, le masque cible produit en sortie de ce processus doit être utilisé à l'entrée du système décrit préalablement au chapitre 2 pour engendrer la texture RVB de l'objet détérioré. L'utilisation de propriétés pour déterminer le positionnement des effets de détérioration a récemment été introduite par Lu *et al.* (2007). Néanmoins, comme il sera détaillé à la section 3.2, la solution proposée dans ce projet de recherche innove dans sa manière d'exploiter ces propriétés pour permettre de générer rapidement et efficacement plusieurs occurrences d'un même objet, détériorées de façon semblable mais non identique. De plus, la solution proposée innove en introduisant le concept de recette de détérioration qui permet à un artiste de définir, à haut niveau, un patron de détérioration général applicable peu importe l'objet à user.

3.2 Calcul des propriétés locales

L'étape du calcul des propriétés locales se trouve dès le début du processus de génération de masque cible. Avant d'approfondir davantage le fonctionnement et l'utilité de cette étape, il est important de définir ce qu'est une propriété locale. Sur le plan théorique, il s'agit en fait d'une propriété mathématique qui peut être vérifiée localement pour tout point d'un espace topologique quelconque. Dans notre contexte, il est possible d'interpréter cette définition théorique pour affirmer qu'il s'agit d'une propriété mathématique qui peut être vérifiée localement pour tout point à la surface de l'objet. Or, il existe une forte relation entre la disposition des effets de détérioration sur un objet et certaines propriétés locales de celui-ci, telles que l'accessibilité, la courbure et l'orientation de la surface. En effet, les régions fortement accessibles d'un objet (les coins) sont susceptibles de subir une détérioration par impacts alors que les régions moins accessibles sont plus sujettes à retenir la moisissure. Cette étape consiste donc à calculer les valeurs de ces propriétés mathématiques pour un objet donné. Il s'agit d'une étape automatique qui doit être exécutée une seule fois pour chaque objet à détériorer. Cette opération permettra par la suite de caractériser l'objet en termes d'attributs plutôt qu'en termes de texels de surface. De cette manière, l'artiste pourra utiliser ces propriétés pour définir aisément les régions à détériorer. Cette caractérisation sera détaillée à la section 3.3 qui introduit le concept de recette de détérioration.

3.2.1 Accessibilité

La toute première propriété locale considérée pour caractériser les objets à détériorer est l'accessibilité. Il s'agit d'une mesure introduite par Miller (1994) permettant de définir la facilité d'accès pour un point de surface. Selon Miller (1994), l'accessibilité se définit comme étant le rayon de la plus grande sphère possible touchant un point de surface sans pour autant faire intersection avec la surface. La figure 3.3 illustre clairement cette définition à l'aide d'un schéma en deux dimensions.

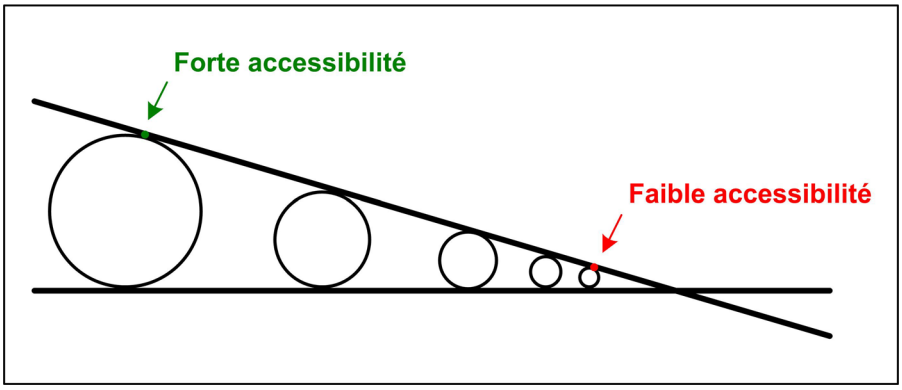

Figure 3.3 Définition de la propriété d'accessibilité.

Ce schéma montre la mesure d'accessibilité pour deux plans faisant intersection. Les cercles montrent le rayon maximum possible pour toucher à un point de surface sans la couper. Plus le rayon est grand, plus l'accessibilité est forte (à gauche dans le schéma). À l'opposé, plus le rayon est petit, plus l'accessibilité est faible (à droite dans le schéma). Bien que Miller (1994) détaille aussi comment calculer cette propriété, plusieurs auteurs tels que Wong *et al.* (1997) ou Kontkanen et Laine (2005) ont repris la définition de l'accessibilité pour proposer de nouvelles façons de la calculer efficacement pour des maillages contemporains plus complexes. Dans le cadre de ce projet, l'implémentation de calcul d'accessibilité du logiciel de modélisation 3D *Blender* a été utilisée. Elle est basée sur un tracé de rayons similaires aux travaux de Hsu et Wong (1995). La figure 3.4 montre visuellement la distribution de cette propriété locale sur différents objets communs. Plus la couleur est claire en un point de surface, plus l'accessibilité est forte. À l'inverse, plus la couleur est foncée, plus l'accessibilité est faible.

Figure 3.4 Exemples visuels de distributions pour l'accessibilité.

Sur ces quelques exemples, il est évident que cette propriété permet de distinguer différents types de régions d'un objet. Par exemple, on remarque que le dessus de la table de salle à manger et l'extérieur de ses pattes sont fortement accessible, ce qui les rend plus susceptibles d'être frappés et de montrer des marques d'usure par impacts. À l'opposé, les barreaux formant l'intérieur du dossier de la chaise montrent une accessibilité plus faible. Logiquement, ils sont moins susceptibles d'être cognés que barreaux formant l'extérieur du dossier. Sur la statue de Buddha, la métrique d'accessibilité permet d'identifier les zones difficiles à atteindre qui retiendraient probablement l'humidité et risqueraient de développer

de la moisissure ou de la rouille. Bref, l'utilisation de cette métrique permet de faire un premier pas vers une caractérisation générale des objets par propriétés locales.

3.2.2 Courbure moyenne

Une autre propriété importante pour caractériser les différentes régions d'un objet à détériorer est la courbure. Par définition, il s'agit d'une métrique permettant de mesurer le degré de changement du vecteur normal d'un point de surface par rapport à son voisinage. Par conséquent, une région plane (aucun changement dans les vecteurs normaux) aura une courbure moyenne faible alors qu'une arête (changements importants dans les vecteurs normaux) montrera une courbure moyenne élevée. Puisque les effets susceptibles de toucher les arêtes et les surfaces planes sont souvent très différents, cette propriété locale a beaucoup d'importance dans la caractérisation des régions de l'objet à user. La définition mathématique théorique de cette métrique est très intuitive et largement répandue. Cependant, la façon de la calculer se complexifie lorsque vient le temps d'en estimer la valeur pour un maillage formé de polygones triangulaires. Plusieurs auteurs proposent des approches d'approximation pour estimer la courbure moyenne sur de tels maillages (Hamann, 1993; Meyer *et al.*, 2003; Taubin, 1995). Dans le cadre de ce projet de recherche, une implémentation de la technique proposée par Theisel *et al.* (2004) a servi pour estimer la courbure moyenne sur les objets à détériorer. La figure 3.5 montre quelques exemples visuels de la distribution de cette propriété locale sur différents objets. Pour plus de détails sur le fonctionnement de la méthode, veuillez vous référer à l'article de Theisel *et al.* (2004).

Figure 3.5 Exemples visuels de distributions pour la courbure moyenne.

Sur la figure 3.5, les texels clairs représentent des régions avec une courbure convexe forte tandis que les texels foncés représentent des régions avec une courbure concave forte. Les texels ayant une valeur médiane représentent des régions avec une faible courbure. Ces exemples montrent clairement que la courbure moyenne permet rapidement d'identifier les arêtes et les coins d'un objet. Puisque ces régions sont hautement susceptibles d'être détériorées, cette propriété locale est essentielle pour la caractérisation des objets.

3.2.3 Orientation de la surface

La dernière propriété utilisée pour caractériser les régions des objets à détériorer est l'orientation de la surface. Il s'agit d'une propriété très simple qui définit la direction d'un point de surface à l'aide de son vecteur normal (utilisé dans le calcul d'éclairage). Comme le montre la figure 3.6, le vecteur normal d'un point est, par définition, perpendiculaire à la surface.

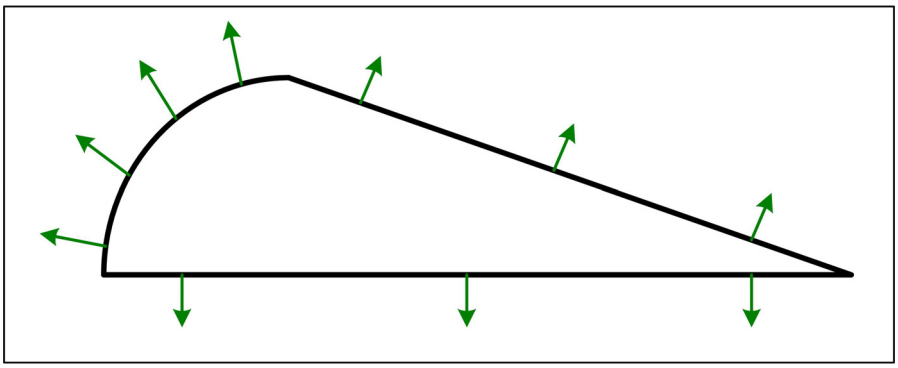

Figure 3.6 Définition de la propriété d'orientation de surface.

Bien qu'elle soit très simple à déterminer, cette propriété est très utile pour différencier certaines régions d'un objet qui seraient impossibles à distinguer à l'aide de l'accessibilité ou de la courbure. À titre d'exemple, le dessus et le dessous d'une table de salle à manger ont sensiblement les mêmes valeurs pour les autres propriétés (forte accessibilité et faible courbure). Or, il est fort probable que le dessus de la table présente des marques de détérioration par impacts légers et très peu probable que le dessous de la table exhibe de tels effets. De plus, cette propriété permet d'établir une relation entre un effet de détérioration potentielle et sa source. Par exemple, la moisissure se formant sur une statue de pierre risque d'être positionnée en fonction de la source d'humidité présente dans l'environnement. Conséquemment, si un point d'eau est situé à gauche de cette statue, la moisissure devrait commencer à se s'installer sur les surfaces orientées vers cette direction. La figure 3.7 montre visuellement la distribution de cette propriété locale sur différents objets communs. Pour des fins d'affichage, les coordonnées (x, y, z) du vecteur normal ont été associées dans l'ordre aux canaux RVB de la texture pour chaque point de surface. La plage de -1 à +1 est décalée pour présenter des valeurs variant de 0 à 255.

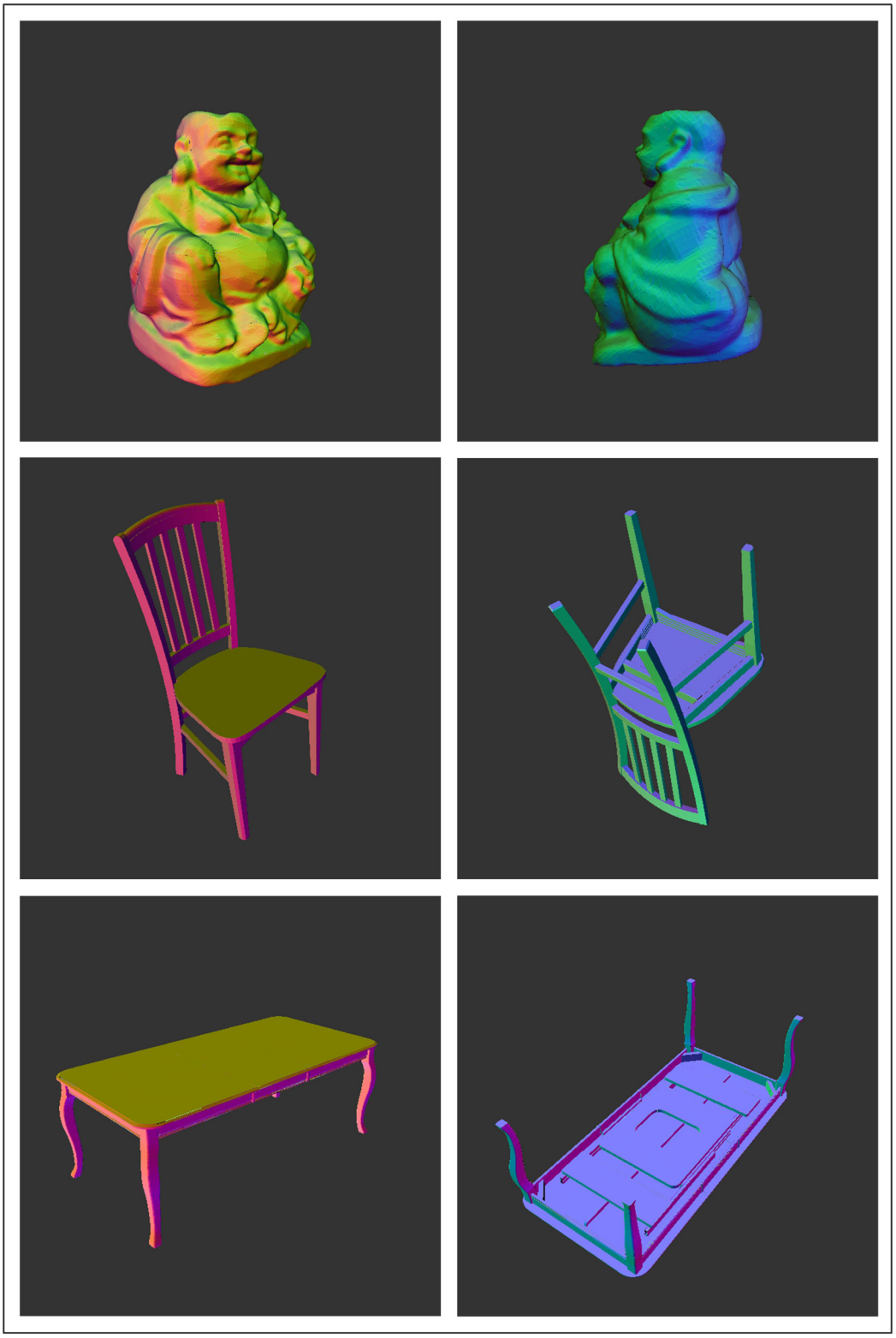

Figure 3.7 Exemples visuels de distributions pour l'orientation de surface.

3.2.4 Autres propriétés

Dans le cadre de ce projet de recherche, seules les propriétés d'accessibilité, de courbure et d'orientation ont été utilisée pour caractériser les différentes régions des objets à détériorer. Cependant, il est important de comprendre que cet ensemble de propriétés locales peut facilement être étendu selon les besoins de l'utilisateur. Par exemple, dans leurs travaux, Lu *et al.* (2007) proposent d'utiliser l'épaisseur du matériau (dans leur cas, l'épaisseur de la couche de peinture) comme propriété locale pour contrôler l'apparition de craquelures à la surface. Par ailleurs, l'utilisateur peut se servir du mécanisme de propriétés locales pour disposer d'un contrôle manuel sur le positionnement de ses effets sur l'objet. Il arrive parfois qu'un artiste veuille positionner un effet à un endroit particulier sur un objet et que cette région ne soit pas distinguable d'une autre à partir des propriétés prédéfinies. À titre d'exemple, il serait possible qu'un artiste veuille ajouter un effet à un endroit précis sur le dessus d'une table de salle à manger. Il peut s'agir d'une décision basée sur des considérations esthétiques, sémantiques ou même complètement arbitraires. Dans ces situations, l'utilisateur peut mettre en place une nouvelle propriété locale, qu'il produit manuellement, et qu'il utilisera dans sa recette de détérioration pour définir l'emplacement de certains effets. La figure 3.8 montre quelques exemples d'une telle situation.

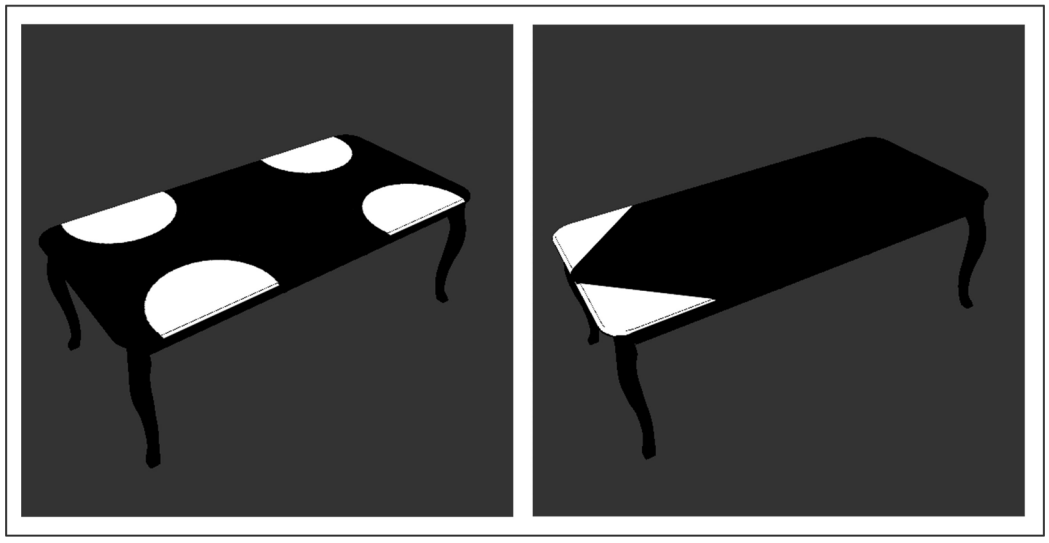

Figure 3.8 Exemples visuels de génération d'une propriété manuelle.

3.3 Création de la recette de détérioration

La prochaine étape du processus consiste à produire une recette de détérioration. Il s'agit d'une étape manuelle effectuée par l'artiste ayant pour but d'engendrer un « *patron de détérioration* ». Avant d'aller plus loin, il est important de bien définir le concept de patron de détérioration. Dans le cadre de ce projet de recherche, c'est tout simplement un modèle d'usure général qui n'est pas spécifique à un objet en particulier. Il est défini à partir des propriétés locales présentées à la section 3.2 de manière à créer une association entre ces caractéristiques et les effets de détérioration à appliquer. Exprimé en mots, un tel modèle d'usure général aurait l'allure suivante :

- Appliquer une détérioration par impacts sur les coins et les arêtes.
- Ajouter de la moisissure dans les recoins difficiles d'accès.
- Appliquer un graffiti sur une surface plane et facile d'accès.

De cette façon, une recette de détérioration est exprimée en termes de caractéristiques générales. Comme mentionné précédemment, plusieurs approches de synthèse d'effets de détérioration basées sur des images demandent à l'utilisateur de créer manuellement leurs masques cibles (Clément, Benoit et Paquette, 2007; Gu *et al.*, 2006; Wang *et al.*, 2006). Dans ce contexte, cela signifie que l'artiste doit définir les effets à ajouter en termes de texels, ce qui constitue une approche spécifique à l'objet traité. La méthode par recette de détérioration générale proposée dans ce projet de recherche innove en permettant à l'artiste d'appliquer un même patron à différents objets de manière à produire des détériorations similaires. De plus, la même recette de détérioration peut être appliquée plusieurs fois à un même objet pour générer des instances semblables mais non identique. La figure 3.9 montre un exemple de recette de détérioration en format XML et la figure 3.10 présente le prototype d'interface utilisé pour la définir.

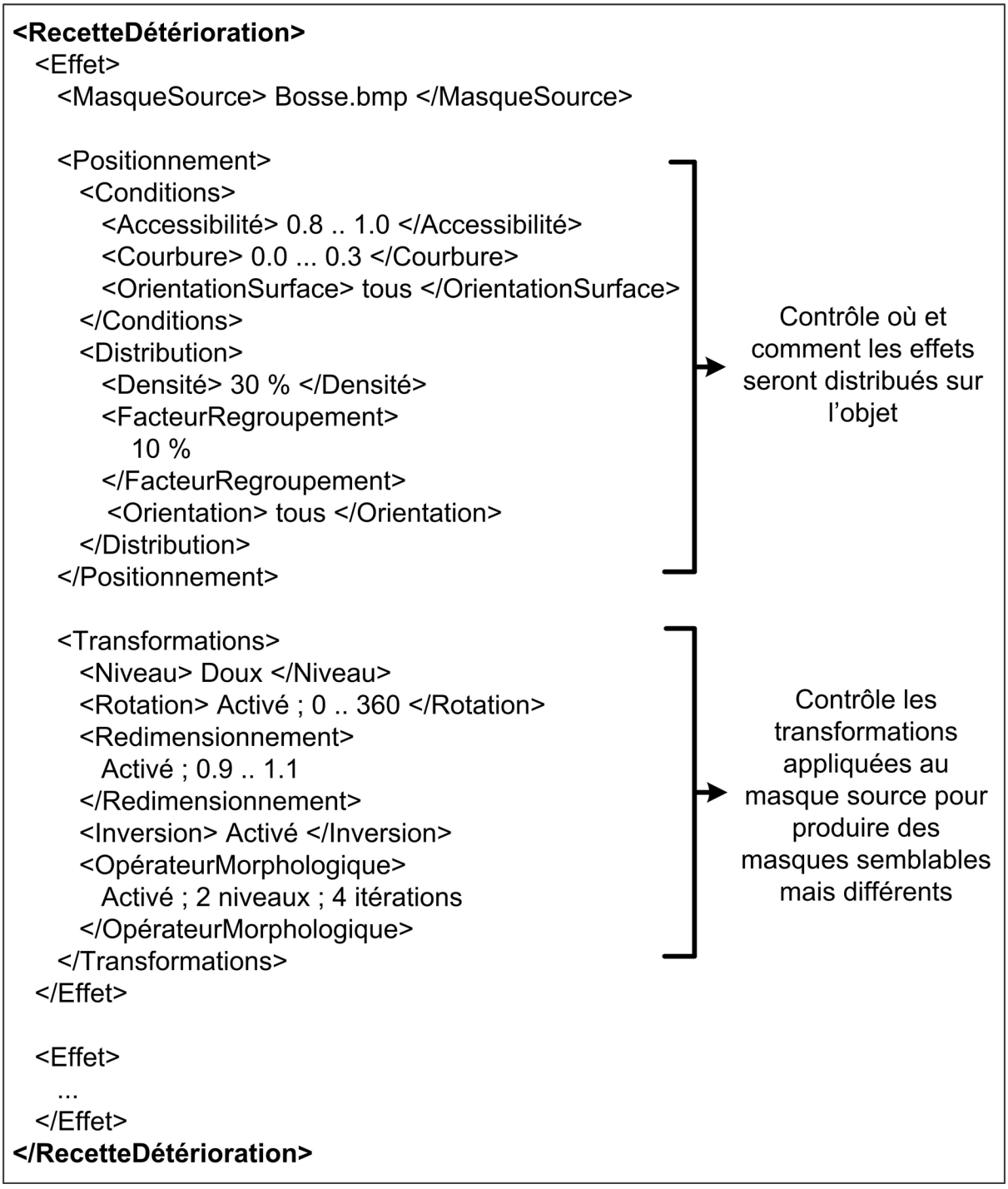

Figure 3.9 Exemple de recette de détérioration en format XML.

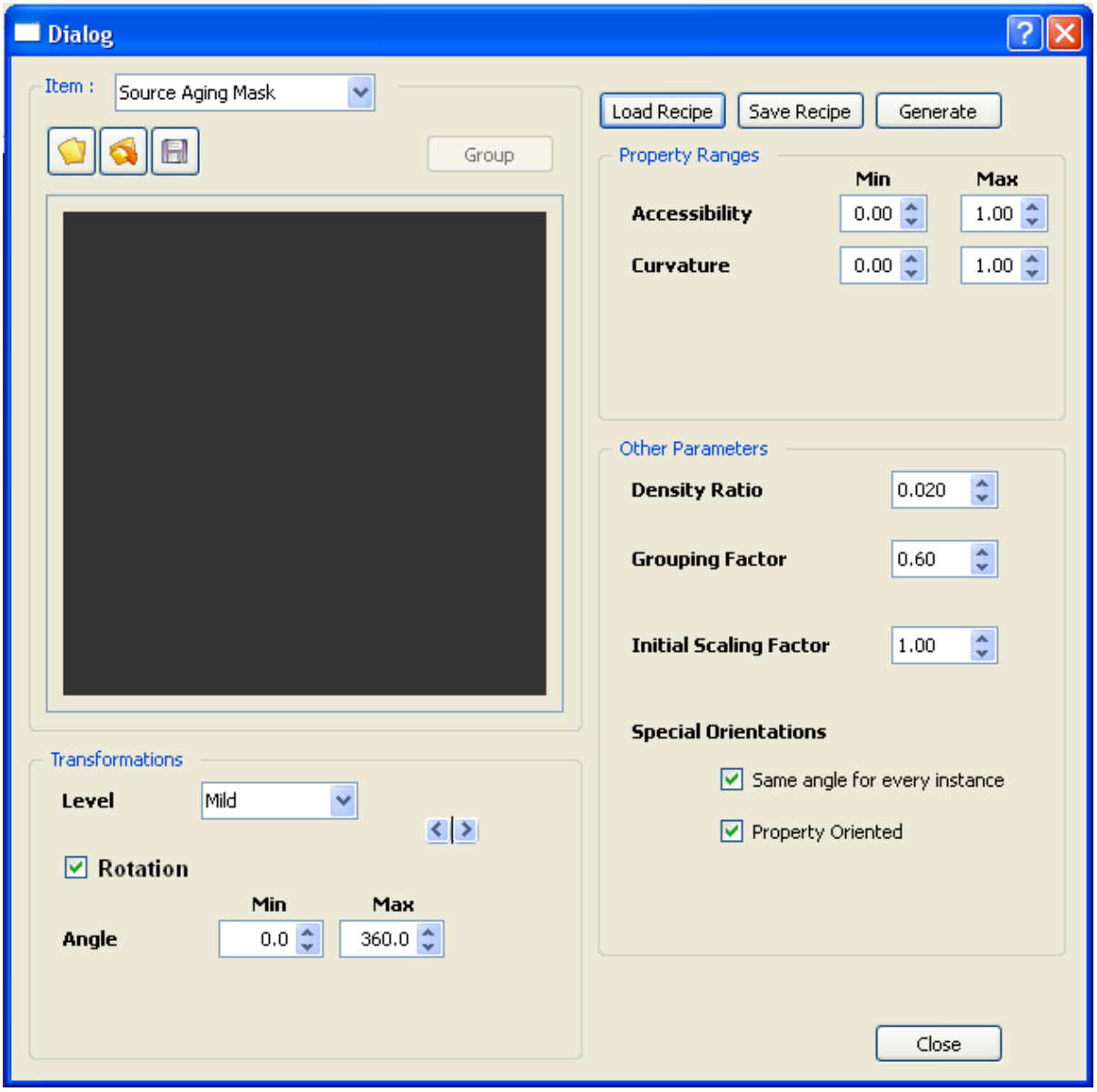

Figure 3.10 Prototype d'interface utilisé pour la recette de détérioration.

La recette de détérioration est tout simplement constituée d'une séquence d'effets à appliquer. Pour chacun de ceux-ci, la recette contient une référence vers un masque source contenant un exemple de l'effet désiré. Par la suite, une série de paramètres, regroupés en deux segments principaux, doivent être spécifiés.

Tout d'abord, le premier segment, englobé dans la balise « *Positionnement* », permet de contrôler où et comment l'effet sera distribué sur l'objet. Les conditions de positionnement limitent les régions disponibles pour placer un effet à partir d'intervalles sur les propriétés locales. Contrairement aux travaux de Lu *et al.* (2007), l'approche proposée exploite les propriétés locales seulement pour définir les régions valides mais n'influence pas la forme des effets de détérioration, ce qui facilite grandement la génération de plusieurs occurrences non identiques d'un même objet. D'autres paramètres contenus dans la balise « *Positionnement* » comme la densité, le facteur de regroupement et l'orientation contrôlent la distribution des effets sur l'objet. Cette portion de la recette de détérioration sera étudiée plus en détails à la section 3.4.

Par la suite, le deuxième segment, englobé par la balise « *Transformations* », permet de contrôler les modifications apportées au masque source de manière à produire des effets semblables mais non identiques. Pour y parvenir, l'utilisateur doit activer ou désactiver une série de transformations (rotation, redimensionnement, inversion et opérateur morphologique) et fournir les paramètres nécessaires aux opérateurs activés. Cette portion de la recette de détérioration sera détaillée à la section 3.5.

À ce stade, le système peut maintenant générer un masque cible en interprétant la recette de détérioration pour un objet donné et ses propriétés locales. La figure 3.11 présente l'algorithme développé pour traduire une recette de détérioration générale en un masque cible spécifique à un objet et ses propriétés locales.

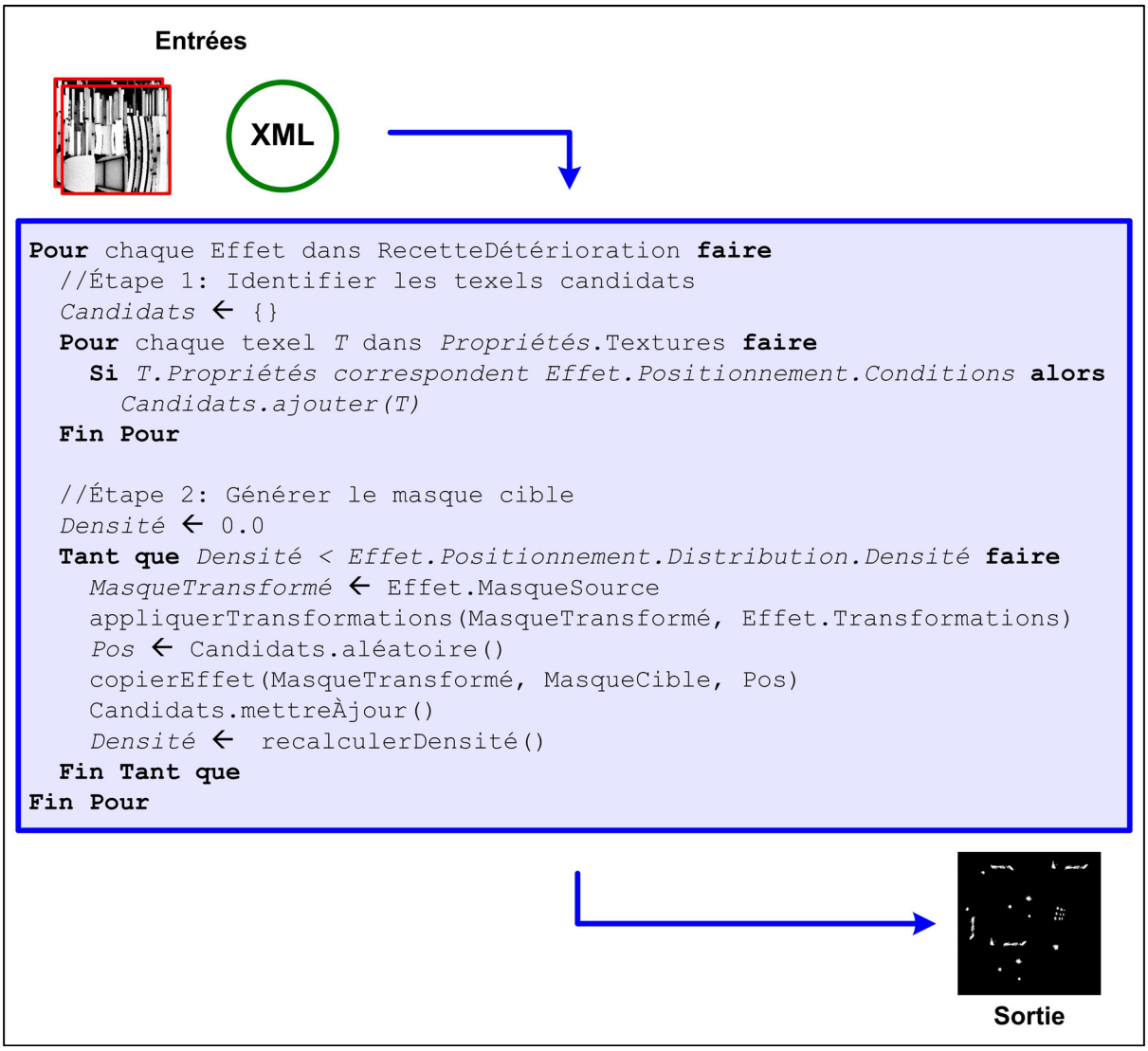

Figure 3.11 Pseudo-code de l'algorithme de génération de masques.

Cet algorithme pour la génération automatique fonctionne en deux étapes, répétées pour chaque type d'effets à ajouter. La première étape consiste à identifier les positions candidates à partir des propriétés locales de l'objet et des conditions de positionnement spécifiées dans la recette. Cette portion de l'algorithme sera étudiée à la section 3.4. La deuxième étape consiste à générer le masque cible en appliquant des transformations sur le masque source et en positionnant le nouvel effet à un endroit sélectionné aléatoirement parmi les candidats. La section 3.5 détaille ce segment du processus.

3.4 Positionnement automatique des effets

À cette étape, il est important de bien comprendre comment l'algorithme présenté à la figure 3.11 pourra utiliser la recette de détérioration et les propriétés locales de l'objet pour positionner automatiquement les effets sur le masque cible. Tout d'abord, l'algorithme parcourt les textures de propriétés en comparant les valeurs de chaque texel aux intervalles acceptables spécifiés dans la balise « *Conditions* » de la recette de détérioration. Cette opération permet d'identifier les candidats initiaux, c'est-à-dire les texels qui répondent aux contraintes définies par l'artiste. Lorsque vient le temps de copier un nouvel effet sur le masque cible à générer, l'algorithme sélectionne aléatoirement une position parmi les candidats potentiels. Cet aspect aléatoire est très important puisqu'il permet d'utiliser la recette de détérioration plusieurs fois sur un même objet pour obtenir différentes versions, semblables, sans être identiques. Par la suite, la densité est recalculée pour s'assurer de ne pas dépasser la valeur du paramètre spécifié dans la recette. La métrique utilisée pour mesurer la densité est tout simplement le ratio entre le nombre de texels détériorés sur le nombre de texels formant la liste de candidats. Tant que la densité désirée n'est pas atteinte, l'algorithme continuera d'ajouter de nouveaux effets sur le masque cible.

Bien évidemment, suite à l'ajout d'un nouvel effet sur le masque cible, la liste de candidats doit être mise à jour. À ce stade, il y a deux situations possibles à considérer : le nouvel effet peut être positionné tout près d'un des effets précédents de manière à former un groupe ou, à l'inverse, être placé à une certaine distance des effets placés antérieurement. Dans des cas réels, il arrive souvent que plusieurs effets de détérioration soient regroupés à un endroit particulier sur la surface d'un objet, formant ainsi une agglomération. Pour permettre à l'artiste de contrôler cette possibilité, la recette de détérioration inclut un paramètre appelé « *facteur de regroupement* » qui représente la chance qu'a un nouvel effet de s'agglomérer avec un autre et dont la valeur peut varier entre 0 et 100 %. En spécifiant une valeur près de 0 %, les effets seront distribués relativement uniformément sur les candidats potentiels alors qu'avec une valeur se rapprochant de 100 %, tous les effets seront regroupés les uns les autres. Ainsi, lorsque vient le moment de positionner un nouvel effet, l'algorithme choisit

aléatoirement parmi l'une de ces deux options en tenant compte de la valeur du facteur de regroupement puis met la liste de candidats à jour. La figure 3.12 illustre graphiquement les nouveaux candidats à considérer selon la situation sélectionnée.

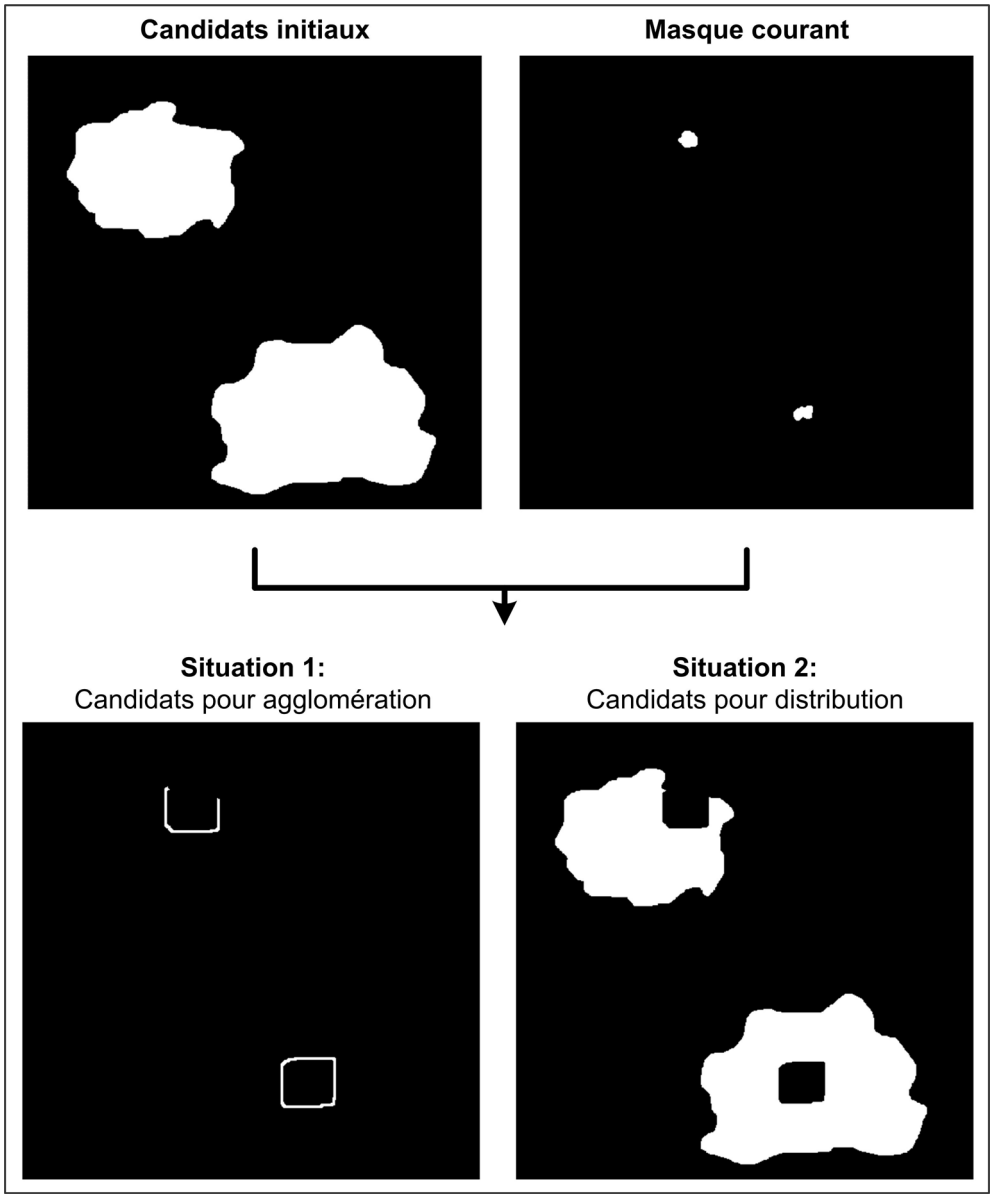

Figure 3.12 Exemple de la mise à jour des candidats.

Pour déterminer la liste des candidats dans la situation 1 (agglomération), l'algorithme applique deux séries de dilations au masque courant. La première série est fixe et permet d'assurer une distance minimale entre les effets dans l'agglomération. Le nombre de dilations dans la deuxième série varie en fonction de la taille du nouvel effet à ajouter. Plus l'effet est gros, plus l'algorithme appliquera de dilations. À l'inverse, si l'effet est petit, peu de dilations seront effectuées. Pour déterminer le nombre de dilations « ***d*** » à appliquer, l'algorithme calcule la longueur de la demi-diagonale du plus petit rectangle englobant l'effet à partir de l'équation suivante :

$$\boldsymbol{d} = arrondi\left(\frac{\sqrt{L\arg eur^2 + Hauteur^2}}{2}\right) \tag{3.1}$$

Ensuite, pour obtenir la liste de candidats mise à jour, l'algorithme calcule la différence entre l'image résultant des deux séries de dilations et l'image résultant seulement de la première série. Comme on le voit sur la figure 3.12, le noyau utilisé dans les étapes de dilation tend à rendre la région carrée. Évidemment, il est nécessaire de s'assurer que tous les candidats de cette liste répondent aux critères de sélection basés sur les propriétés locales. Pour s'en assurer, l'algorithme calcule l'intersection entre la liste ainsi mise à jour et les candidats initiaux. Dans le cas où la liste de candidats est vide, le système déduit qu'il n'y a aucune position valide pour produire une telle agglomération.

Dans le deuxième cas (distribution uniforme), l'algorithme commence par appliquer une série de dilations au masque courant. Le nombre de dilations à effectuer « ***d*** » est déterminé exactement de la même façon que dans le cas de l'agglomération. Par la suite, l'algorithme calcule la différence entre les candidats initiaux et l'image résultant de la série de dilations. Puisque le processus débute avec la liste de candidats initiaux, les texels ainsi sélectionnés répondent aux critères de positionnement basés sur les propriétés locales. De plus, à cause de la série de dilations appliquée au masque courant, la nouvelle liste de candidats ne contient que des texels éloignés des effets placés préalablement. Des résultats illustrant l'impact des paramètres contrôlant le positionnement automatique des effets sont présentés à la section 3.6.

3.5 Transformations appliquées sur les effets

Avant de positionner un nouvel effet sur le masque cible, le processus doit appliquer une série de transformations au masque source de manière à produire quelque chose de semblable, sans pour autant être identique. Pour contrôler cette portion importante du processus de génération de masques, la recette de détérioration prévoit un ensemble de paramètres, englobés dans la balise « *Transformations* ». Tout d'abord, l'artiste doit sélectionner le niveau de transformation désiré parmi trois possibilités : doux (1 à 3 transformations), moyen (4 à 6 transformations) et fort (7 à 9 transformations). Dans le cas du niveau « doux », peu de transformations sont appliquées, ce qui implique un effet très proche du masque source. À l'opposé, dans le cas du niveau « fort », un grands nombre de transformations sont appliquées, ce qui résulte en un effet considérablement différent du masque source. Ensuite, l'artiste doit activer ou désactiver un assortiment d'opérateurs de transformation, tels que rotation, redimensionnement, inversion et opérateur morphologique. De plus, des intervalles doivent être définis pour les différents paramètres des opérateurs activés comme un angle de rotation en degrés ou un facteur de redimensionnement. Au moment de modifier le nouvel effet, l'algorithme sélectionne des opérateurs parmi les transformations activées et génère aléatoirement des paramètres compris dans les intervalles définis. L'ensemble des opérateurs est détaillé dans les sections 3.5.1 et 3.5.2. Des considérations spéciales concernant l'orientation des effets et les effets groupés sont discutées à la section 3.5.3. Finalement, des résultats illustrant l'impact des paramètres contrôlant les transformations appliquées sur les effets sont présentés à la section 3.6.

3.5.1 Opérateurs de transformation de base

Tout d'abord, l'ensemble des transformations utilisé par le processus de génération automatique de masques cibles contient quelques opérateurs de base : rotation, redimensionnement (mise à l'échelle) et inversion (réflexion miroir). Bien que ces opérations soient relativement simples, elles permettent rapidement d'engendrer des différences mineures entre les effets générés et le masque source, diminuant grandement la sensation de

percevoir des effets identiques d'un objet à l'autre. Dans le cadre de ce projet de recherche, l'implémentation de ces opérations faite dans la librairie OpenCV (*« Open Computer Vision »*) est utilisée. Pour plus de détails sur cette librairie, veuillez consulter le livre de Bradski et Kaehler (2008).

L'opérateur de rotation permet de changer l'orientation d'un effet et nécessite un seul paramètre, c'est-à-dire l'angle de rotation. Le processus de transformation génère aléatoirement l'angle de rotation à partir de l'intervalle possible défini par l'artiste dans la recette de détérioration puis applique la rotation résultante. De son côté, l'opérateur d'inversion ne demande aucun paramètre à l'artiste. Lorsque cette transformation est sélectionnée, le processus choisit aléatoirement parmi trois types d'inversions, soit un par rapport à l'axe des x, à l'axe des y ou aux deux axes à la fois. Pour ce qui est de l'opérateur de redimensionnement, son implantation est légèrement plus complexe et demande de prendre en considération quelques particularités des effets de détérioration. Dans certains cas, il est possible d'utiliser un facteur de redimensionnement uniforme et de l'appliquer directement sur le masque source représentant l'effet à ajouter. Cependant, il arrive souvent qu'un redimensionnement homogène soit inadéquat à cause de la forme de l'effet. Par exemple, une égratignure a généralement une forme mince et allongé pour laquelle la longueur totale peut varier considérablement (70 % à 130 %), alors que la largeur sera plutôt constante (95 % à 105 %). La figure 3.13 présente un tel cas d'exemple dans lequel un redimensionnement uniforme altère de façon inadéquate la forme de l'effet contenu dans le masque source. Un redimensionnement selon l'axe principal de l'effet de détérioration permet d'obtenir un résultat correspondant beaucoup mieux à l'exemple réel.

Pour parvenir à appliquer un tel redimensionnement, le processus de transformation doit identifier l'axe principal de l'effet de détérioration. Dans ce contexte, la KLT (Karhunen–Loève Transform), parfois appelée transformée d'Hotelling, est utilisée pour déterminer cet axe. Il s'agit d'une méthode statistique très répandue basée sur l'analyse en composantes principales, mieux connue sous l'acronyme PCA (Principal Component Analysis). Pour plus

de détails sur cette transformée, veuillez consulter le livre de Loève (1978) ou celui de Gonzalez et Woods (2002).

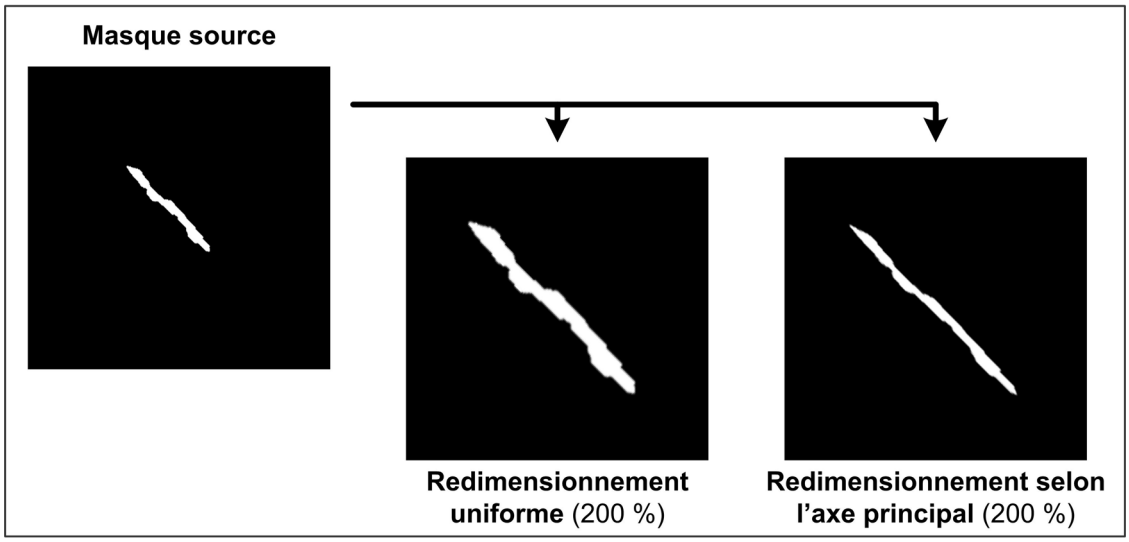

Figure 3.13 Exemple du problème de redimensionnement uniforme.

La KLT permet de déterminer un vecteur unitaire orienté par rapport à l'axe principal de l'effet de détérioration. À partir de ce vecteur, il est possible de déterminer l'angle de rotation nécessaire pour aligner la composante principale de l'effet sur l'axe des *x*. Dans cette situation, la recette de détérioration doit contenir deux facteurs de redimensionnement : un en *x* pour l'axe principal puis un en *y* pour l'axe secondaire.

3.5.2 Opérateur morphologique

Bien que les opérateurs de transformation de base permettent de modifier légèrement l'apparence des effets à ajouter, ils n'altèrent pas réellement la forme du masque source. Pour compléter la gamme d'outils du processus de transformation, un opérateur morphologique a été développé spécifiquement pour le contexte des effets de détérioration. La figure 3.14 présente l'algorithme général de cet opérateur.

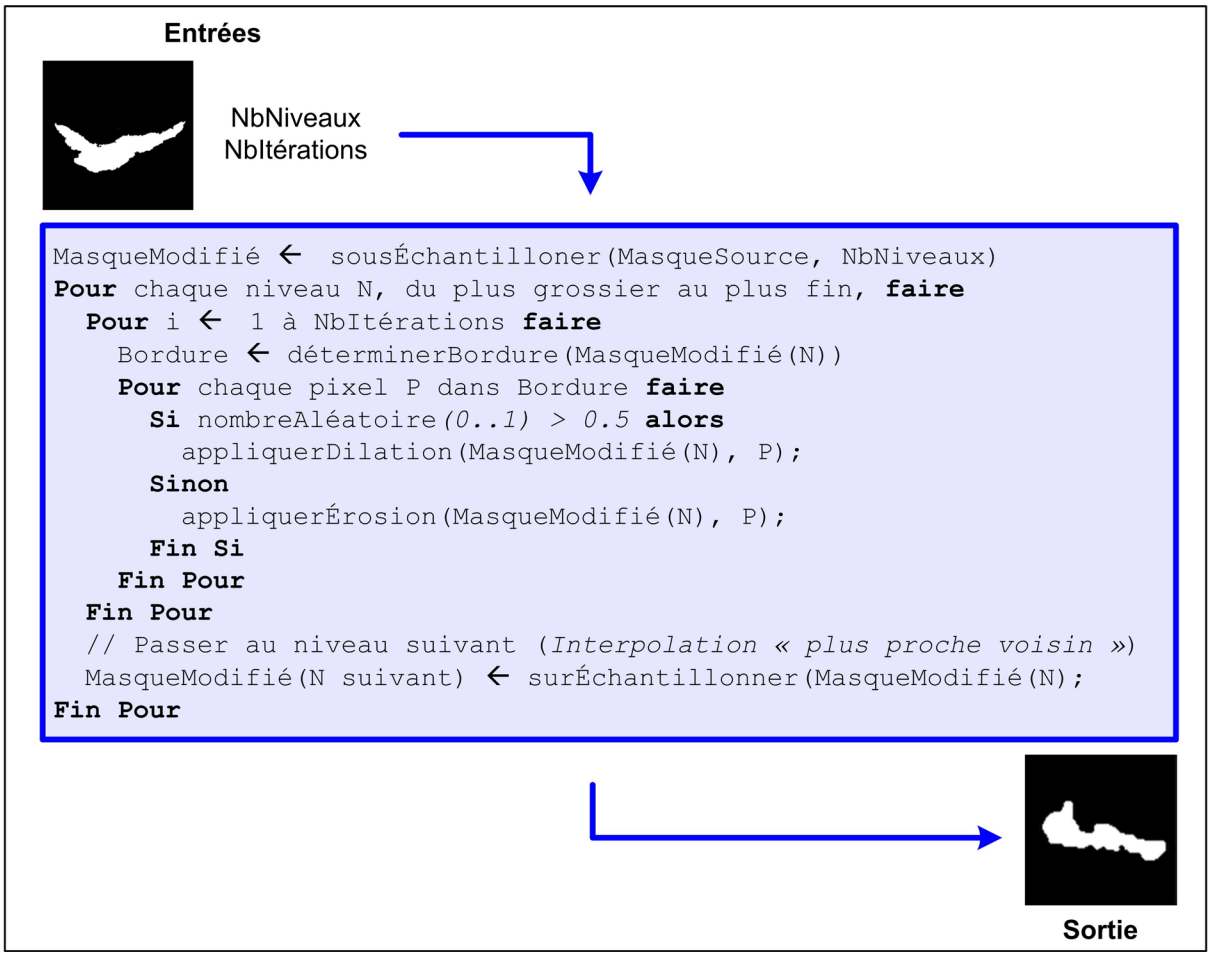

Figure 3.14 Pseudo-code de l'algorithme de l'opérateur morphologique.

L'idée générale de cet opérateur morphologique est de modifier la bordure du masque de l'effet de détérioration en appliquant aléatoirement une dilation ou une érosion. Ce traitement est répété pour un certain nombre d'itérations. Bien évidemment, plus il y a d'itérations appliquées par l'algorithme, plus le masque cible produit en sortie divergera du masque source. De plus, le processus est appliqué à différents niveaux de résolution. Une image sous-échantillonnée est initialement calculée en utilisant une interpolation linéaire. Ensuite, le traitement morphologique est appliqué sur la bordure pour chacun des niveaux entre la résolution de l'image sous-échantillonnée et la résolution maximale. Aux niveaux grossiers, chaque érosion ou dilatation change considérablement la forme du masque. À l'opposé, aux niveaux plus fins, les modifications sont plus subtiles. Le nombre d'itérations à effectuer et le nombre de niveaux à considérer sont fournis en paramètres dans la recette de détérioration.

Lorsque sélectionné, cet opérateur morphologique modifie la forme du masque source permettant de générer des nouveaux effets semblables sans pour autant être identiques.

3.5.3 Considérations spéciales sur les transformations des effets

Durant l'étape de transformation du processus de génération de nouveaux effets de détérioration, l'algorithme doit tenir compte de certains cas spéciaux concernant leur orientation. Tout d'abord, il arrive souvent que certains effets soient alignés en fonction d'une ou de plusieurs propriétés locales. À titre d'exemple, comme le montre la figure 3.15, une marque de détérioration par impact risque logiquement d'être orientée par rapport à une arête d'un objet.

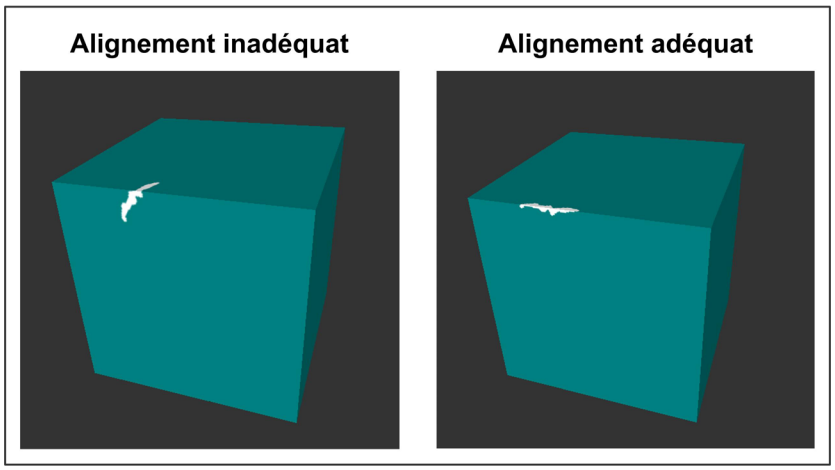

Figure 3.15 Exemple d'alignement basé sur les propriétés.

Tel que décrite précédemment, l'approche proposée trouve un texel répondant aux conditions de positionnement définies dans la recette de détérioration et y centre l'effet, sans considérer son orientation. Or, dans des situations comme celle-ci, l'algorithme devrait identifier le comportement spécial de la propriété locale (la courbure dans cet exemple) et chercher à aligner l'effet en conséquence. Pour y parvenir, la recette de détérioration contient un paramètre « *Orientation* » dans la balise « *Positionnement* » spécifiant si l'effet à ajouter doit être aligné en fonction d'une propriété. Dans ces cas, l'algorithme cherche à maximiser le nombre de texels formant le masque de l'effet positionnés à un endroit répondant aux

conditions de positionnement. Une métrique mesurant la qualité de l'alignement d'un masque par rapport à une propriété est définie en fonction de l'intervalle de valeurs accepté dans la recette de détérioration et d'une distribution normale.

$$\sum_{(p)\in Masque} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(p-\mu)^2}{2\sigma^2}} \tag{3.2}$$

Dans cette équation, p représente la valeur de la propriété pour un texel (x, y) donné du masque, σ représente l'écart-type de la distribution et est égale à la moitié de la taille de l'intervalle de valeurs acceptables pour la propriété considérée puis μ représente la moyenne de la distribution et est égale au point milieu du même intervalle. Maintenant, pour trouver l'orientation optimisant cette métrique, l'algorithme commence par la calculer pour quatre orientations primaires, soit 0º, 45º, 90º et 135º. À partir de ces résultats initiaux, l'algorithme peut déterminer une orientation approximative maximisant la métrique selon les deux meilleures orientations. Par exemple, si les tests effectués pour 0º et 45º produisent les meilleurs résultats, il est généralement juste de conclure que l'orientation optimale se situera entre 0º et 45º. À ce stade, l'algorithme effectue trois tests supplémentaires pour des orientations sélectionnées aléatoirement dans l'intervalle d'angles choisi. L'orientation pour laquelle la métrique montre la valeur maximale sera conservée, permettant ainsi d'appliquer la rotation adéquate pour aligner le masque en fonction d'une propriété locale lors de son positionnement. Cette approximation trouve un maximum local à plus ou moins 6º de précision. Pour une plus grande précision, l'utilisateur peut contrôler le nombre de tests supplémentaires effectués. Un algorithme de recherche plus sophistiqué pourrait être utilisé.

Une deuxième situation spéciale à considérer lors de l'étape de transformation concerne l'orientation des effets les uns avec les autres. Dans plusieurs cas, l'orientation d'un effet donné est complètement indépendante des effets positionnés préalablement. Par contre, il arrive parfois qu'il existe une relation entre l'orientation de deux effets donnés. À titre d'exemple, il arrive souvent qu'un ensemble d'égratignures couvrant la surface d'un objet soient toutes alignées dans une même direction à cause de phénomènes de détérioration répétés sans cesse de la même façon. La figure 3.16 montre un tel exemple d'alignement.

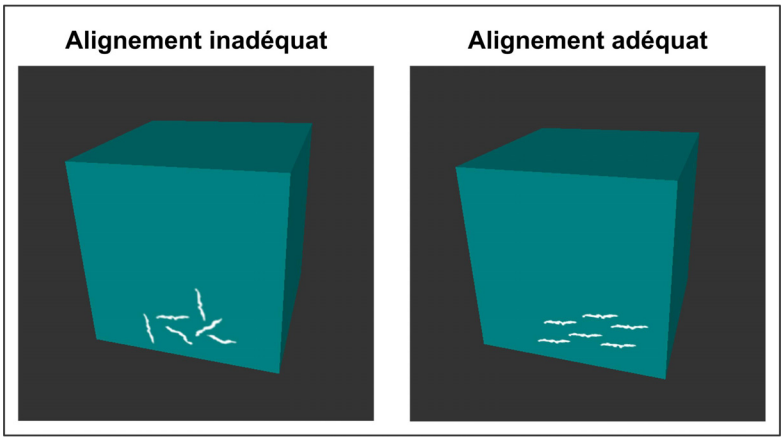

Figure 3.16 Exemple d'alignement entre effets.

Pour parvenir à effectuer un tel alignement, la recette de détérioration prévoit un paramètre « *Orientation* » dans la balise « *Positionnement* » contrôlant ces cas. Lorsqu'il est activé, ce paramètre fait en sorte que l'algorithme applique exactement la même rotation pour chaque nouvelle instance du même groupe d'effets de détérioration traité. Il s'agit d'une façon simple mais efficace de s'assurer que tous les effets aient la même orientation.

Finalement, la dernière situation à considérer concerne les effets formés d'une association de régions distinctes. Pour l'instant, les effets étudiés ont toujours été composés d'une zone unique à l'intérieur du masque. Or, comme le montre la figure 3.17, il arrive parfois qu'un effet soit formé d'un regroupement de plusieurs zones. Dans ces situations, l'utilisateur doit spécifier manuellement les zones à regrouper pour s'assurer que les transformations s'appliquent sur le groupe dans son ensemble et non les régions individuellement.

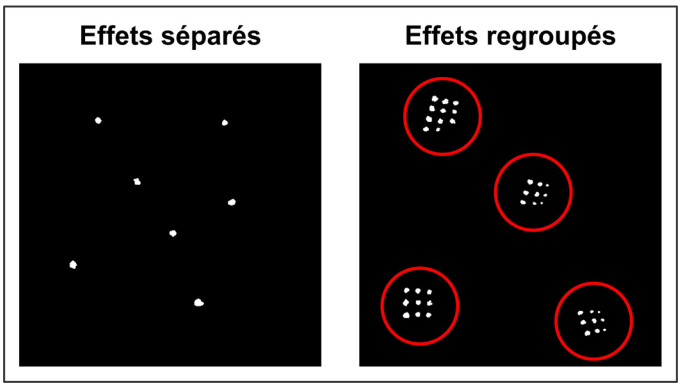

Figure 3.17 Exemple d'effets regroupés.

3.6 Présentation des résultats

Dans cette section, un ensemble de résultats obtenus avec le processus de génération automatique de masque à l'aide de propriétés locales décrit dans ce chapitre sont présentés. Tout d'abord, la figure 3.18 montre plusieurs masques cibles résultant de l'outil de génération automatique. Ensuite, la figure 3.19 illustre le processus complet, allant de la génération des masques cibles à la synthèse des textures RVB. Les figures 3.20 et 3.21 présentent des résultats finaux pour la production de plusieurs occurrences, respectivement sur une chaise en bois et un bloc de ciment (masque pour la moisissure générée automatiquement et masque pour la gravure fait manuellement). La figure 3.22 montre une comparaison entre une distribution uniforme et une agglomération d'effets de détérioration et la figure 3.23 présente deux cas spéciaux concernant l'orientation des effets de détérioration. Finalement, la figure 3.24 montre l'application de la même recette de détérioration sur différents objets pour produire une usure comparable.

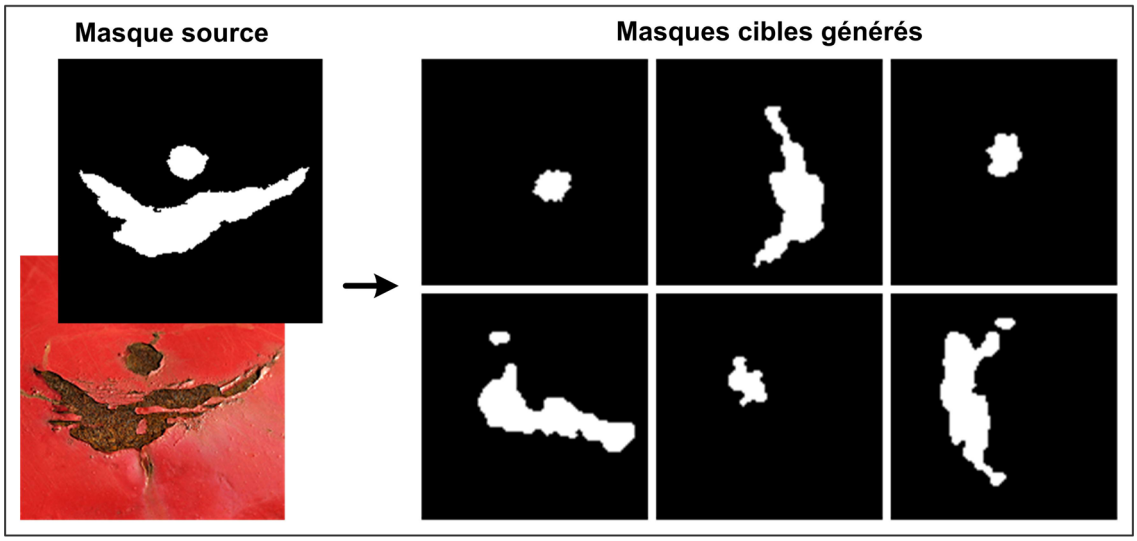

Figure 3.18 Masques cibles résultant de l'outil de génération automatique.

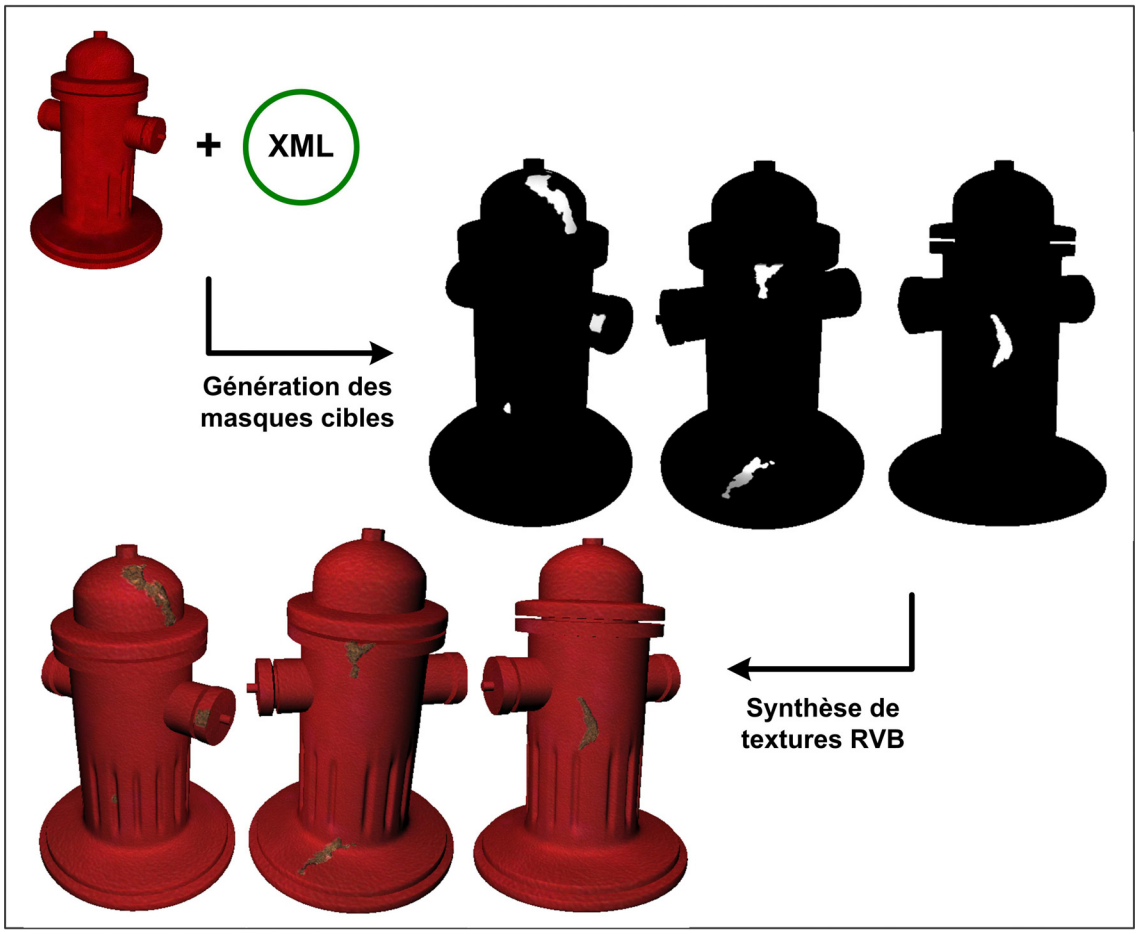

Figure 3.19 Masques cibles appliqués à trois occurrences d'une borne-fontaine.

Figure 3.20 Résultats illustrant plusieurs occurrences d'une chaise en bois.

Figure 3.21 Résultats illustrant plusieurs occurrences d'un bloc de ciment.

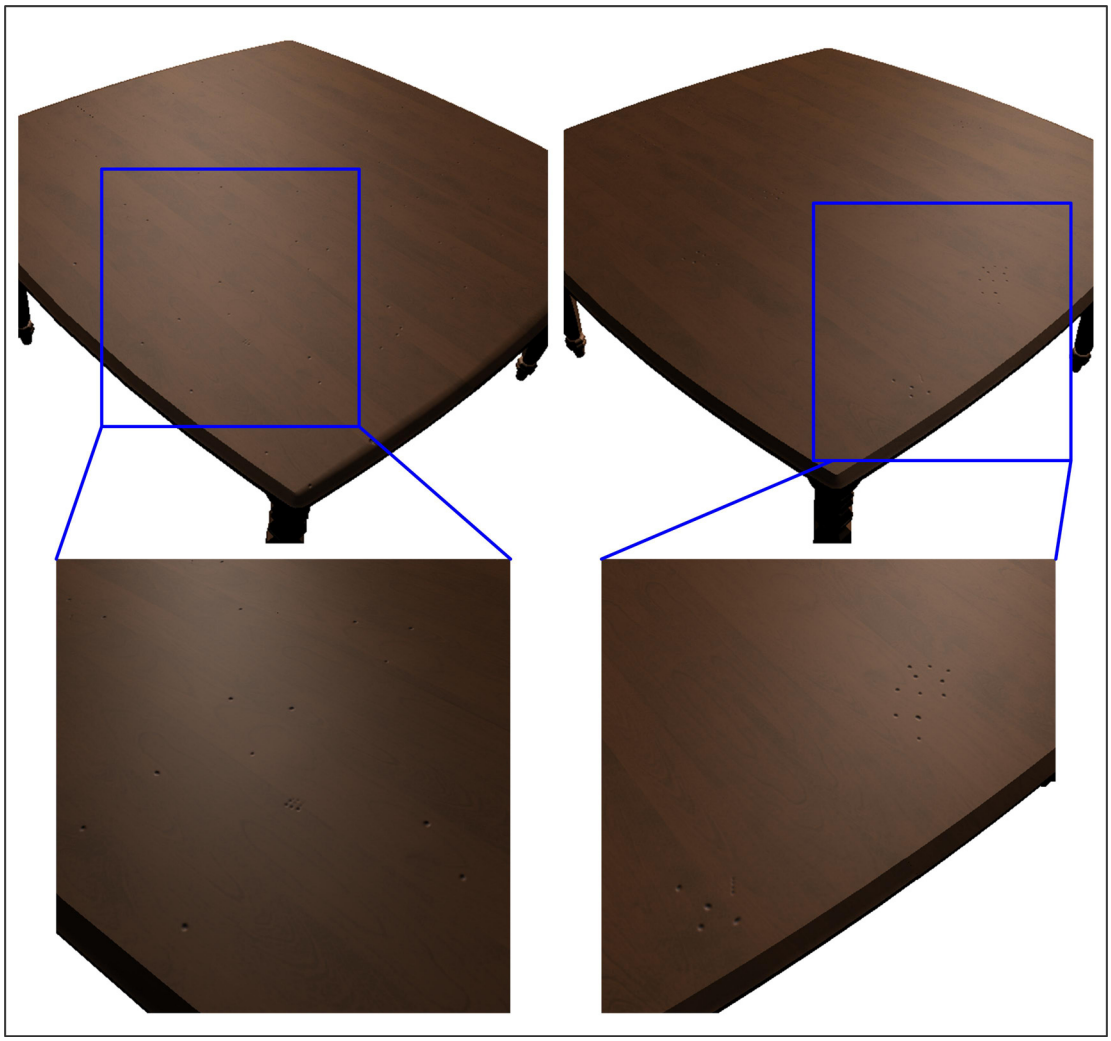

Figure 3.22 Résultats illustrant l'utilisation du paramètre d'agglomération.

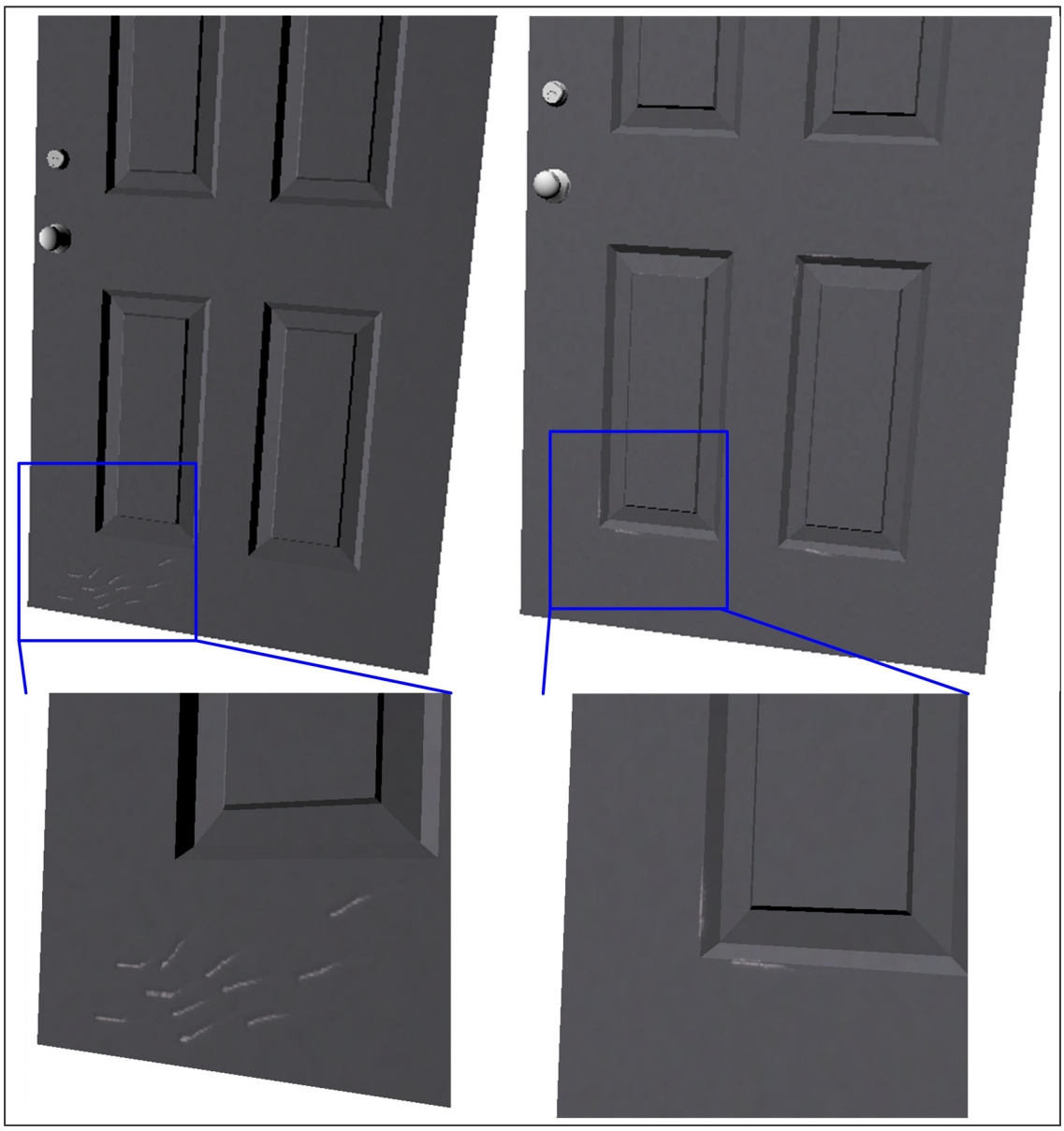

Figure 3.23 Résultats illustrant les cas spéciaux concernant l'alignement.

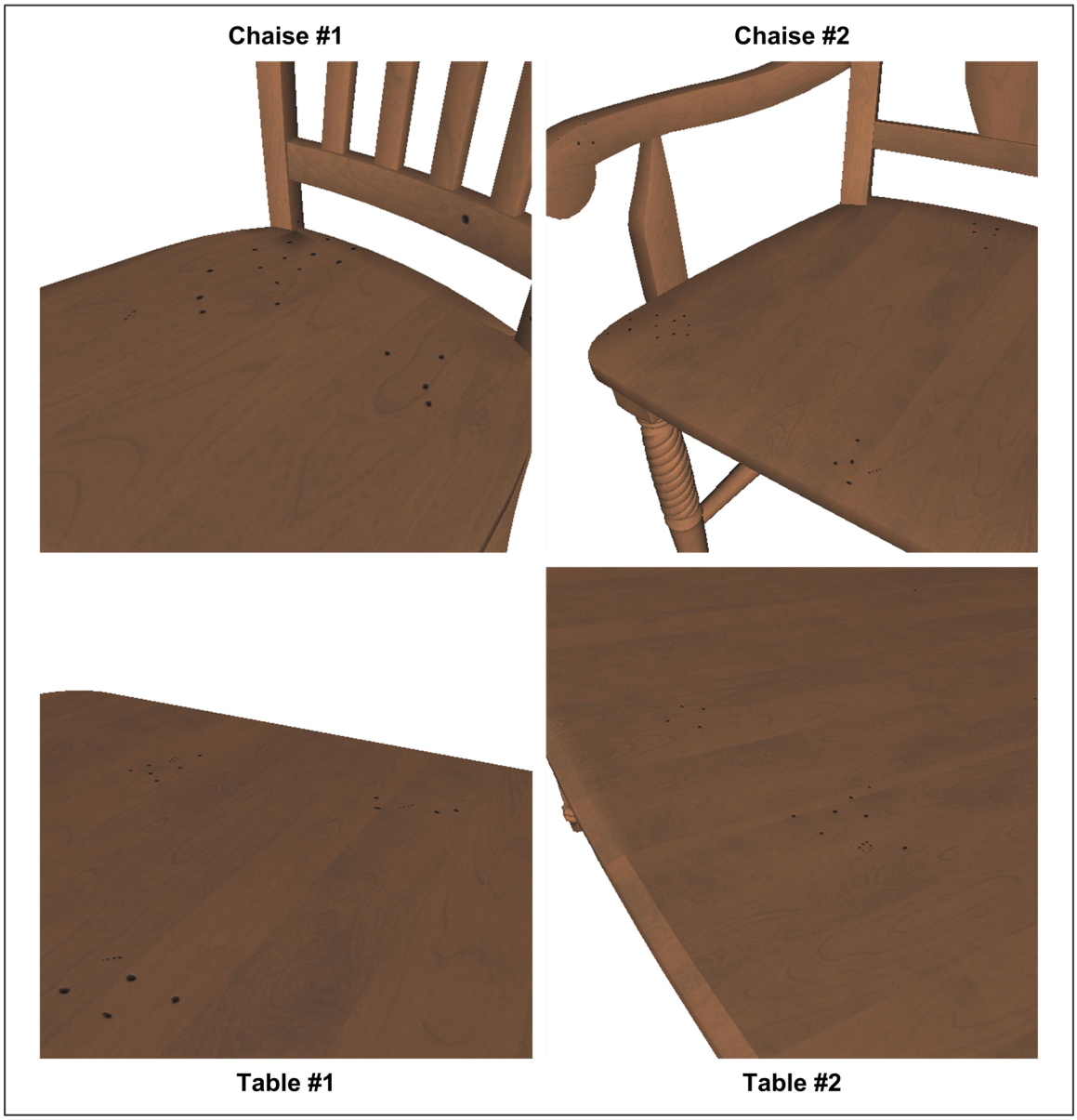

Figure 3.24 Résultats illustrant l'utilisation de la même recette sur plusieurs objets.

Tout comme le processus présenté au chapitre 2, l'approche proposée dans ce chapitre pour la génération automatique de masques cibles fonctionne sur une variété de matériaux, tels que le bois, le métal et le ciment. Tant et aussi longtemps que l'utilisateur fournit un exemple adéquat de l'effet désiré, le système sera en mesure de reproduire quelque chose de semblable. Finalement, pour clore cette section, le tableau 3.1 regroupe les données relatives à la performance de l'approche proposée pour tous les résultats obtenus. Pour une critique détaillée de l'ensemble du processus, veuillez consulter la section 3.7.

Tableau 3.1 Données sur les performances du processus de génération de masques

Résultats	Dimension	Nombre d'effets	Temps de calcul du système (secondes)
Figure 3.18 (1)	1024x1024	1	3
Figure 3.18 (5)	1024x1024	1	3
Figure 3.18 (3)	1024x1024	1	4
Figure 3.18 (4)	1024x1024	1	10
Figure 3.18 (2)	1024x1024	1	11
Figure 3.18 (6)	1024x1024	1	12
Figure 3.19 (1)	1024x1024	3	13
Figure 3.19 (3)	1024x1024	3	13
Figure 3.19 (2)	1024x1024	4	15
Figure 3.21 (1)	1024x1024	16	18
Figure 3.21 (2)	1024x1024	18	18
Figure 3.23 (2)	1024x1024	5	20
Figure 3.23 (1)	1024x1024	11	24
Figure 3.20 (1)	1024x1024	42	24
Figure 3.20 (3)	1024x1024	43	24
Figure 3.20 (2)	1024x1024	50	26
Figure 3.22 (1)	2048x2048	68	48
Figure 3.22 (2)	2048x2048	72	49

3.7 Discussion

Cette section contient un bilan du processus de génération automatique de masques cibles décrit dans ce chapitre. Une discussion divisée en deux volets permet de mettre en évidence les avantages de la méthode proposée, ainsi que ses limitations.

3.7.1 Les avantages

Le processus de génération automatique de masques cibles présenté de ce chapitre permet de résoudre un des problèmes majeurs des approches de détérioration basées sur la capture d'images telle que celle décrite au chapitre 2. Elle permet d'éliminer l'étape de création manuelle qui représentait un goulot d'étranglement important pour l'utilisateur. En effet, en introduisant le concept de recette de détérioration, l'approche proposée permet à l'artiste de définir la façon de détériorer un objet en termes d'attributs (les propriétés locales) plutôt qu'en termes de texels. Il s'agit d'un procédé simple et intuitif qui élimine le travail manuel tout en offrant un contrôle complet sur l'apparence du résultat final. De plus, puisque que le processus de génération automatique doit être utilisé de concert avec l'approche de synthèse du chapitre précédent, l'utilisateur en conserve tous les avantages. Comme le montrent les résultats de la section 3.6, la polyvalence des approches de détérioration basées sur la capture d'images en ce qui a trait à la variété d'effets et de matériaux est toujours présente.

Par ailleurs, le processus développé propose une façon novatrice d'utiliser les propriétés locales comme attributs pour définir l'usure à appliquer. Conséquemment, l'utilisateur peut facilement et rapidement générer des détériorations similaires sur plusieurs objets différents. De plus, la même recette de détérioration peut être réutilisée pour produire plusieurs occurrences différentes d'un même objet, ce qui était difficile, voire impossible, à obtenir avec d'autres techniques (Gu *et al.*, 2006; Lu *et al.*, 2007; Wang *et al.*, 2006). Dans un contexte réel d'utilisation, ces possibilités réduisent grandement le temps nécessaires à la production de la totalité d'un environnement.

Finalement, les données sur les performances recueillies dans le tableau 3.1 montrent que le processus de génération de masques s'exécute dans des délais interactifs. En effet, ces données illustrent clairement que pour des textures de dimensions typiques et un nombre d'effets raisonnable, le temps nécessaire à la génération automatique se mesure en secondes (de 3 à 49 secondes pour les résultats présentés dans ce chapitre). Il s'agit d'un avantage important de la technique puisque cela permet d'intégrer aisément le processus dans un cycle

de développement itératif où les résultats sont révisés et demandent des ajustements rapides. Ces données ont été obtenues en utilisant un PC standard avec un processeur de 2,4 GHz et 3 Go de mémoire vive. Il est important de mentionner que puisque le processus demande en paramètre les textures 2D des propriétés locales, le modèle 3D utilisé n'a aucun impact sur le temps nécessaire à la génération du masque. Évidement, le calcul des propriétés doit être fait préalablement pour chaque modèle à détériorer. D'autres variables influencent les performances du processus telles que la taille des effets ou le nombre de transformations à appliquer. Cependant, puisque l'impact de ces facteurs est négligeable (souvent moins d'une seconde de différence), ils n'ont pas été considérés dans le tableau 3.1 par soucis de simplicité.

3.7.2 Les limitations

Bien évidemment, l'approche définie dans le chapitre possède aussi certains problèmes. Tout d'abord, puisque le processus de génération de masque nécessite l'utilisation de l'approche de synthèse présentée au chapitre précédent, elle souffre des mêmes limitations. En effet, puisque le processus se base sur l'utilisation de textures 2D pour représenter les effets de détérioration à la surface d'un objet, elle ne permet pas de travailler avec des phénomènes qui modifieraient la géométrie 3D de cet objet. Son utilisation est donc limitée aux effets de surface qui ne changent pas significativement la forme ou les dimensions d'un objet. De plus, tel que discuté précédemment, la synthèse des canaux RVB nécessite une image exemple contenant l'effet spécifique de la teinte désirée par l'artiste. Ce problème est étudié plus en détails au chapitre 4.

En ce qui concerne le processus de génération lui-même, quelques difficultés mineures font surface. Tout d'abord, la recette de détérioration contient un nombre relativement important de paramètres devant être spécifiés par l'utilisateur. Heureusement, la grande majorité de ces paramètres sont simples et peuvent être ajustés intuitivement par un artiste n'ayant aucune connaissance scientifique particulière. Par ailleurs, il est important de mentionner qu'au fur et à mesure qu'un artiste apprend à utiliser le système, il pourra se construire un ensemble de

préréglages spécifiques à différents types d'effets de détérioration, qu'il pourra réutiliser pour limiter le nombre de paramètres à ajuster. Un autre problème associé au processus de génération concerne l'ensemble de propriétés locales considérées dans la recette de détérioration. En effet, dans certains cas, il est possible que cet ensemble ne permette pas de bien distinguer chaque région de l'objet à détériorer rendant plus difficile le positionnement automatique des effets. Pour régler ce problème, l'ensemble de propriétés locales considérées peut être étendu facilement. Aussi, pour un contrôle complet sur le positionnement des effets, l'artistes a toujours l'option d'utiliser une propriété manuelle telle que décrite à la section 3.2.4.

Finalement, l'étape de transformation comprise dans le processus de génération ne fonctionne pas correctement pour certains types d'effets de détérioration. L'exemple le plus commun concerne les effets formés de texte, comme les graffitis ou les gravures (voir figure 3.21). Dans ces cas particuliers, les opérateurs de transformations utilisés ne permettent évidemment pas de générer du texte différent et l'artiste doit donc retravailler manuellement les masques pour produire différentes occurrences de l'objet.

CHAPITRE 4

PROCESSUS DE SYNTHÈSE INDÉPENDANT DE LA COULEUR[3]

Le dernier volet de ce projet de recherche consiste à adapter l'approche de synthèse d'effets de détérioration décrite aux chapitres 2 et 3 de manière à proposer un processus indépendant de la couleur de l'exemple fourni par l'utilisateur. Pour y parvenir, l'utilisation d'un nuanceur, souvent appelé un « *shader* », permettra d'interpréter un carte d'élévation à l'exécution de l'application de façon à contrôler les couleurs à afficher pour les régions détériorées de l'objet. Le présent chapitre explique les différentes étapes du processus proposé, présente les résultats ainsi obtenus puis se termine avec une discussion sur ses avantages et ses limitations.

4.1 Présentation générale du processus

Tel que mentionné précédemment, l'approche de synthèse de détérioration présentée au chapitre 2 requiert l'utilisation d'une image source contenant un exemple de l'effet désiré. Or, puisque l'algorithme employé pour générer de nouveaux effets se base sur les canaux RVB de l'image source, l'exemple fourni doit inévitablement être de la teinte souhaitée. Cependant, comme le montre la figure 4.1, il arrive souvent qu'un même effet de détérioration puisse se manifester sous plusieurs colorations différentes. Pour bien traiter ces cas, l'utilisateur doit être en mesure de fournir au système une image source pour chacune des colorations différente. Conséquemment, la taille de la base de données d'images sources nécessaires augmente et devient rapidement impraticable. Cette situation peut devenir problématique, particulièrement si l'artiste ne parvient pas à trouver un exemple du monde réel pour la teinte désirée.

---

[3] Le contenu de ce chapitre fait l'objet d'une publication dans une revue scientifique internationale :
Clément, Olivier, et Eric Paquette. 2010. « Adaptable Aging Factory for Multiple Objects and Colorations ». Computers & Graphics, vol. 34, n° 4, p. 460-467.

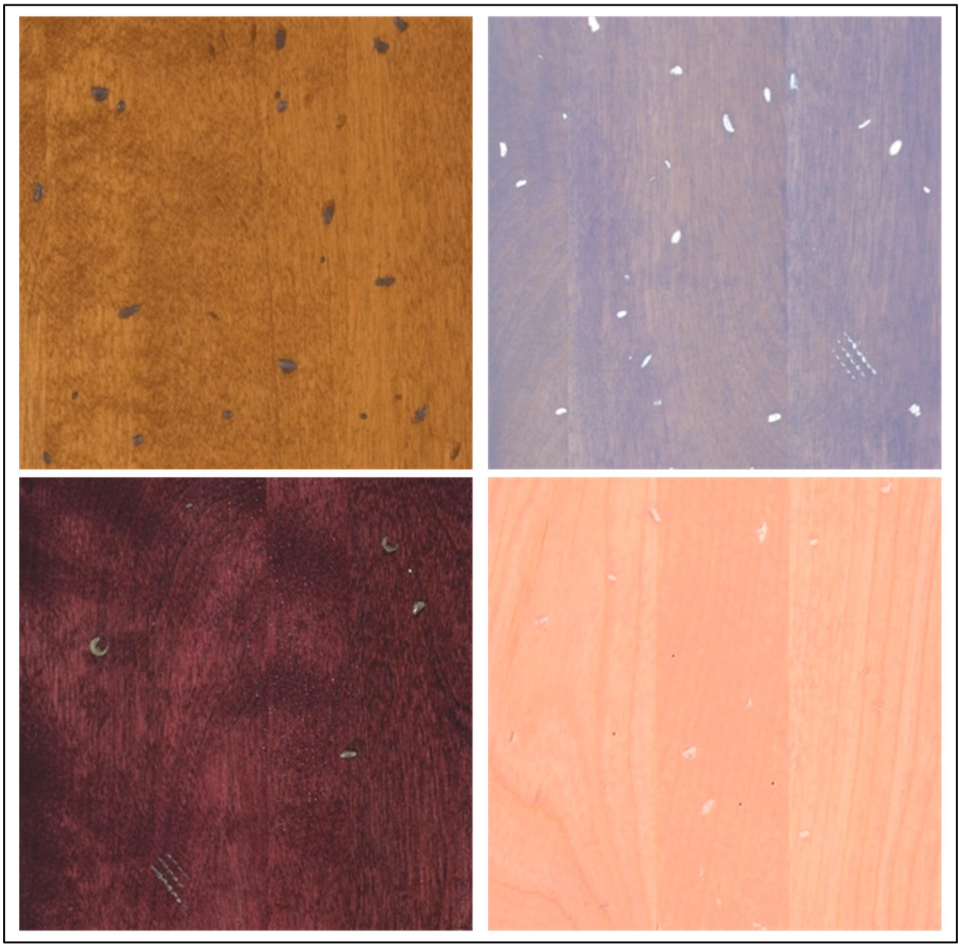

Figure 4.1 Exemple d'effets similaires de couleurs différentes.
Tirée du site Internet de Canadel (2010)

Dans ce contexte, il serait intéressant de pouvoir générer des effets de détérioration indépendamment de la couleur désirée. Évidemment, pour adresser ce problème, il est nécessaire de modifier l'approche de synthèse de texture pour en éliminer l'information des canaux RVB. Par conséquent, les paramètres fournis au processus doivent être modifiés pour remplacer l'image source RVB par autre chose. Ensuite, ce nouveau paramètre doit permettre à l'utilisateur de contrôler l'apparence du résultat final en ajustant la teinte des effets de détérioration générés. La figure 4.2 présente un survol du processus de synthèse indépendant de la couleur introduit dans ce chapitre.

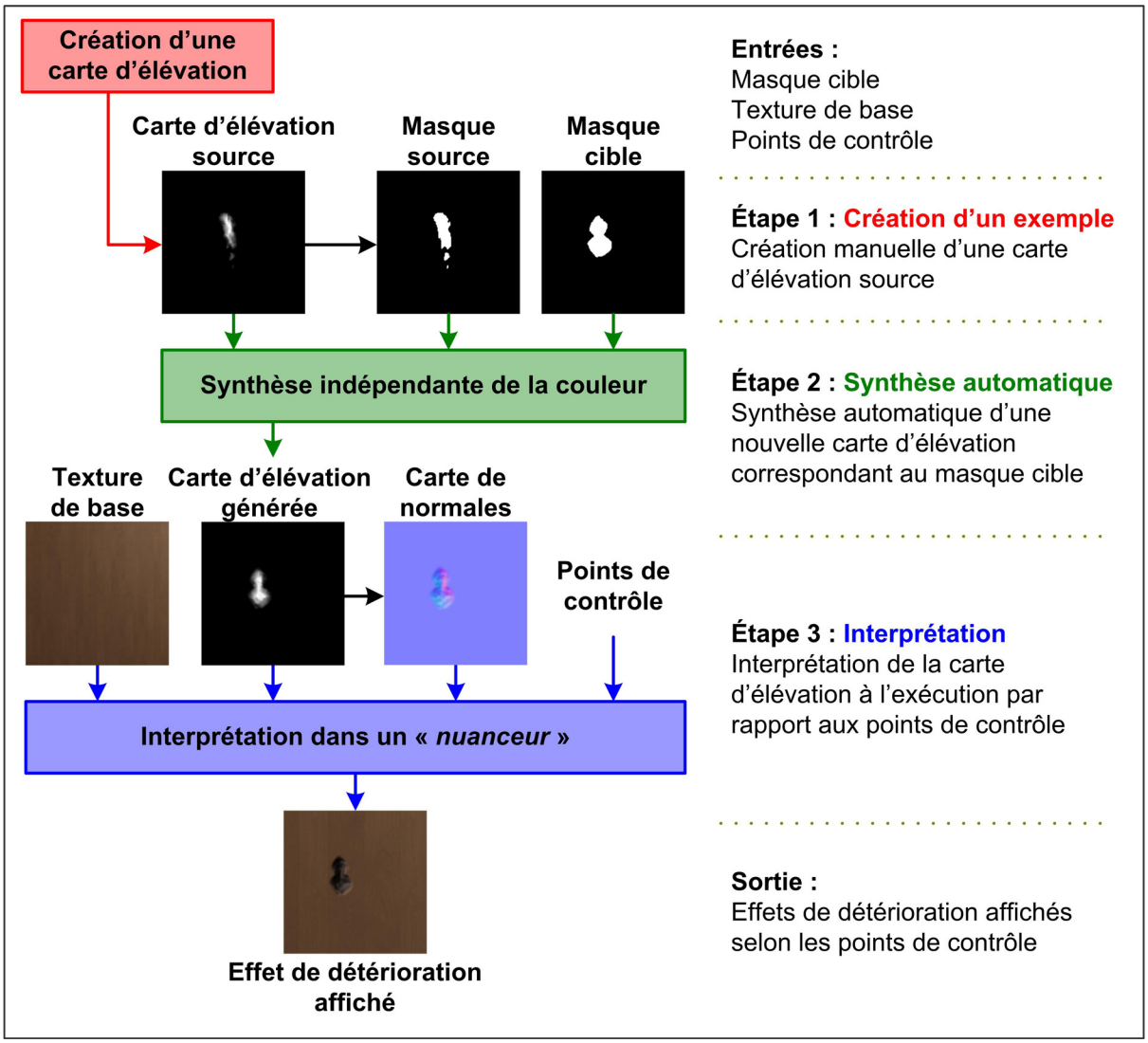

Figure 4.2 Présentation générale du processus indépendant de la couleur.

Le processus de synthèse proposé se décompose en trois étapes : la création d'une carte d'élévation source, la synthèse indépendante de la couleur puis l'interprétation de la carte d'élévation dans un nuanceur. La carte d'élévation remplace l'image RVB source et représente l'intensité de la détérioration à un texel donné. Puisque dans la majorité des cas, l'intensité de la détérioration varie en fonction de la profondeur de l'effet, cette image est appelée une carte d'élévation. Ensuite, un algorithme de synthèse de texture semblable à celui décrit à la section 2.4 permet de générer une nouvelle carte d'élévation correspondant au masque cible fourni en paramètre au processus. Finalement, cette carte d'élévation est interprétée à l'exécution de l'application par un nuanceur permettant de produire une couleur

à afficher pour chaque texel en fonction de points de contrôle fournis par l'utilisateur. Chacune de ces trois étapes est détaillée dans ce chapitre, dans les sections 4.2, 4.3 et 4.4 respectivement.

4.2 Création d'une carte d'élévation source

La première étape du processus de synthèse indépendant de la couleur consiste à créer une carte d'élévation source. Il s'agit d'une image contenant un exemple d'effet, en niveaux de gris, représentant l'intensité de la détérioration pour chaque texel. Tel que mentionné précédemment, puisque l'intensité de la détérioration varie souvent en fonction de la profondeur de l'effet, cette image est appelée une carte d'élévation. Plusieurs méthodes peuvent être utilisées pour produire cette carte d'élévation. Tout d'abord, elle peut être créée manuellement à partir d'outils de sculpture présents dans la majorité des logiciels typiques de modélisation 3D. Il s'agit d'une approche intuitive pour un artiste puisqu'il est déjà familier avec ce genre d'outils. Comme le montre la figure 4.3, cette approche permet de visualiser simplement l'allure 3D de l'effet de détérioration correspondant à la carte d'élévation produite. Dans le cadre de ce projet, le logiciel de modélisation « *Blender* » a été utilisée pour construire les cartes d'élévation sources.

Une autre approche pour créer la carte d'élévation source est d'utiliser des outils comme le pinceau ou l'aérographe présents dans plusieurs logiciels de dessin comme « *Adobe Photoshop* » ou « *Gimp* ». En ajustant correctement les paramètres de ces outils, il est souvent possible de simuler efficacement certains effets de détérioration comme des égratignures fines ou des bosses à la surface d'un objet. Cette deuxième méthode nécessite très peu de temps et ne demande aucune connaissance spécifique préalable.

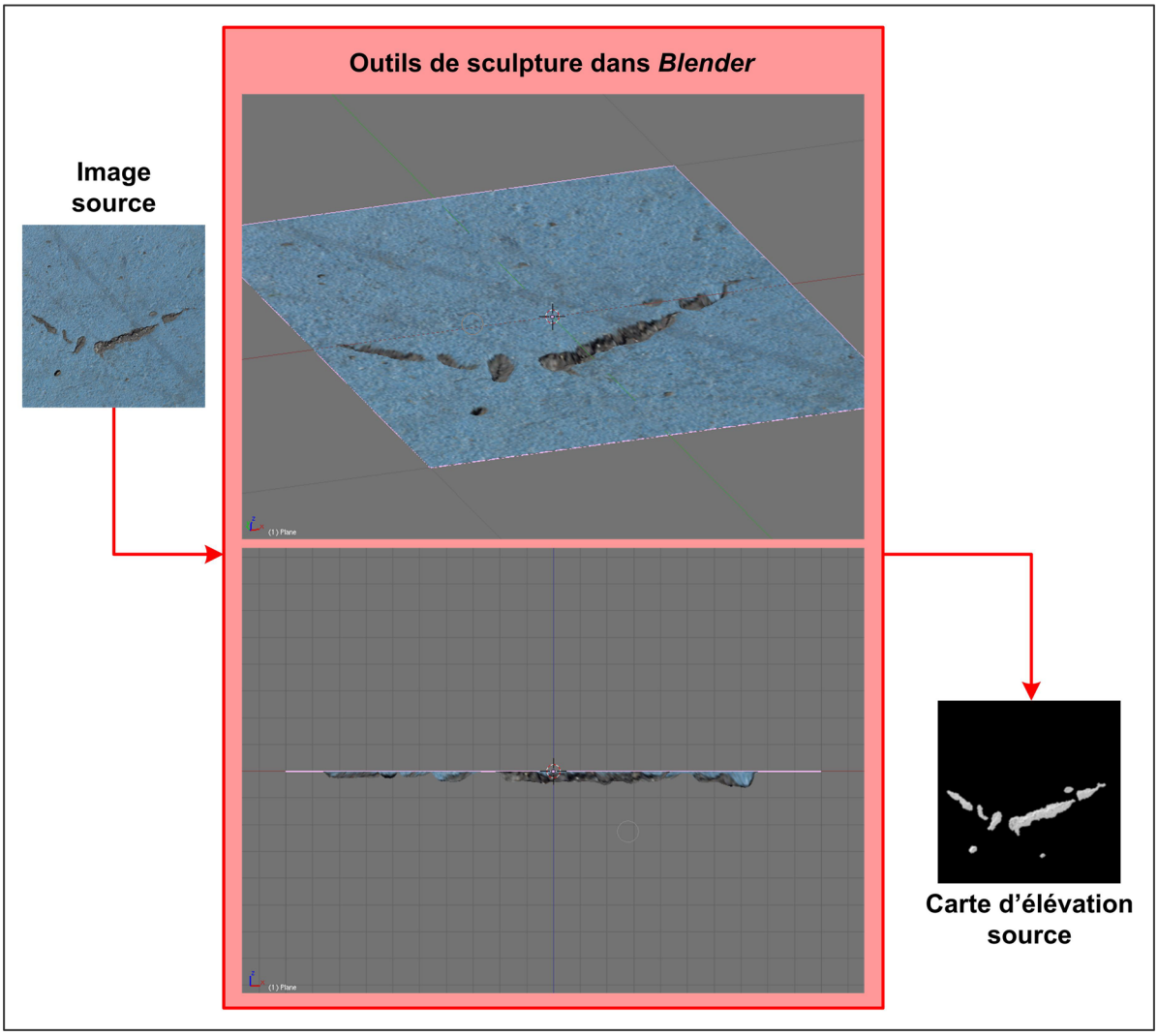

Figure 4.3 Utilisation d'outils de sculpture pour créer une carte d'élévation.

Finalement, une dernière approche pour créer la carte d'élévation source consiste à utiliser un système de numérisation automatique. En effet, il existe plusieurs systèmes à base de laser permettant de générer automatique une carte de profondeur, tel que celui proposé par Beraldin *et al.* (1997). Si, en plus d'avoir accès à un tel système, l'utilisateur a en sa possession un exemple physique de l'effet de détérioration à reproduire, il lui est possible de créer automatiquement la carte d'élévation source nécessaire au processus. L'avantage majeur de cette façon de faire est bien entendu son aspect automatique. Cependant, outre le coût associé, quelques inconvénients liés au système de capture peuvent venir déranger les

utilisateurs tels que les formats admissibles pour les exemples physiques ou les types de surfaces acceptables pour éviter des problèmes de réflexion du laser.

Bref, peu importe l'approche préconisée par l'utilisateur, la création d'une carte d'élévation source ne prend que quelques minutes. De plus, lorsque l'on considère l'utilisation du processus sur une longue période de temps, il est fort possible que l'artiste parvienne à se créer une base de données de cartes d'élévation dans laquelle il pourra puiser pour générer de nouveaux objets détériorés. Dans ce contexte, cette étape demandant un travail manuel de la part de l'artiste devient négligeable avec le temps.

4.3 Processus de synthèse des nouvelles cartes d'élévation

À cette étape, le système doit générer de nouvelles cartes d'élévation ressemblant à l'exemple fourni par l'utilisateur et correspondant aux masques cibles générés en utilisant l'approche décrite au chapitre 3. Pour y parvenir, l'algorithme de reproduction introduit à la section 2.4 est légèrement modifié pour lui permettre de traiter des cartes d'élévation sous forme d'images en niveaux de gris. Comme le montre la figure 4.4, le processus de synthèse prend en paramètres une carte d'élévation source, le masque source correspondant et le masque cible décrivant la nouvelle distribution à générer. En sortie, une nouvelle carte d'élévation est produite en fonction du masque cible.

Pour produire cette nouvelle carte d'élévation, un algorithme de synthèse de textures est utilisé. Pour maximiser le contexte non-détérioré lors de la recherche pour la meilleure correspondance, il est, une fois de plus, nécessaire de traiter d'abord les pixels en bordure des régions à remplacer. Il s'agit donc d'un algorithme par remplissage de trous. Pour un rappel du fonctionnement d'un tel algorithme, veuillez consulter la section 2.3.2. Comme pour la synthèse RVB, la recherche de la meilleure correspondance pour effectuer le remplacement se fait à partir du voisinage du pixel à traiter. Cependant, le contexte RVB est remplacé par le niveau de gris tiré de la carte d'élévation. Le contexte binaire « détérioré » ou « non-détérioré » extrait du masque source et du masque cible est encore considérer dans la mesure

de correspondance. Pour un rappel sur le processus de recherche de la meilleure correspondance et sur la métrique utilisée, veuillez vous rapporter à la section 2.4.2.

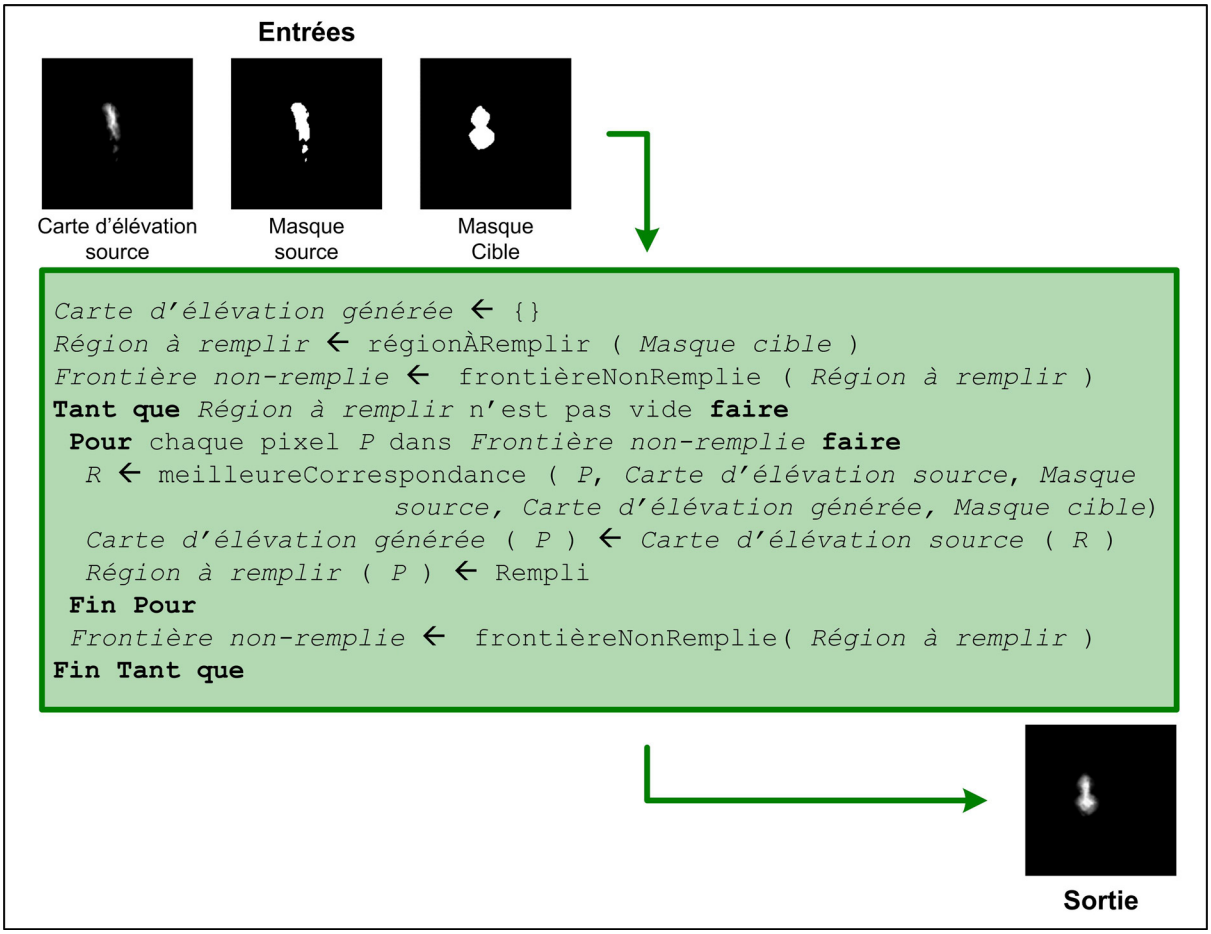

Figure 4.4 Pseudo-code de l'algorithme de synthèse de cartes d'élévation.

Ce processus de synthèse permet donc de générer une carte d'élévation correspondant à l'exemple et au masque cible fournis. À ce stade, aucune information sur la couleur n'est générée. La carte d'élévation ne mesure que la profondeur ou l'intensité de l'effet de détérioration pour un texel donné, ce qui rend le processus indépendant de la couleur. Tel que mentionné précédemment, la carte d'élévation doit maintenant être interprétée à l'exécution de l'application pour afficher les effets de détérioration en couleurs.

4.4 Interprétation dans un nuanceur

Pour produire des couleurs pour chaque texel formant les effets de détérioration, le système doit interpréter la carte d'élévation générée à l'étape précédente. Pour y parvenir, un nuanceur, exécuté en temps réel, permet de transformer l'information contenue dans la carte d'élévation en couleurs affichées à l'écran. Comme le montre la figure 4.2, les paramètres du nuanceur incluent, outre la carte d'élévation, la texture de base de l'objet, une carte de normales et une série de points de contrôle. La définition des points de contrôle est étudiée en détails à la section 4.4.1 puis l'étape de création de la carte de normales est précisée à la section 4.4.2. Finalement, l'algorithme complet du nuanceur est présenté à la section 4.4.3.

4.4.1 Définition des points de contrôle

Pour ajuster l'apparence et la couleur des effets de détérioration à partir de la mesure de profondeur (ou d'intensité) contenue dans la carte d'élévation, l'approche proposée consiste à définir une série de points de contrôle. Comme le montre la figure 4.5, un point de contrôle est tout simplement une couleur associée à une certaine profondeur. Pour les valeurs de profondeur intermédiaires, le système effectue une interpolation linéaire entre les points de contrôle correspondant pour générer la couleur à afficher. De cette manière, en fournissant des points de contrôle différents pour une même carte d'élévation, l'artiste peut facilement changer l'apparence d'un effet de détérioration à partir du même exemple.

Par ailleurs, il est important de remarquer que les points de contrôle sont définis dans l'espace de couleurs HSV (« *Hue – Saturation – Value* ») plutôt que dans l'espace RVB standard. Ce modèle de représentation de la couleur permet de séparer les canaux RVB en valeurs plus intuitives pour un utilisateur, soit la teinte, la saturation et l'intensité. La teinte prend une valeur entre 0º et 360º sur le cercle des couleurs et définit la coloration dominante (0º : du rouge, 60º du jaune, 120º du vert, etc.). La mesure d'intensité varie entre 0% et 100% et définit la brillance de la couleur. Plus la valeur est faible, plus la couleur résultante sera sombre. À l'opposé, plus la valeur est élevée, plus la couleur résultante sera pâle. La

saturation varie elle aussi entre 0% et 100% et mais définit plutôt la pureté de la couleur. Plus la saturation est faible, plus la couleur résultante sera fade, donnant l'impression qu'elle contient un ton de gris (de blanc à noir selon l'intensité).

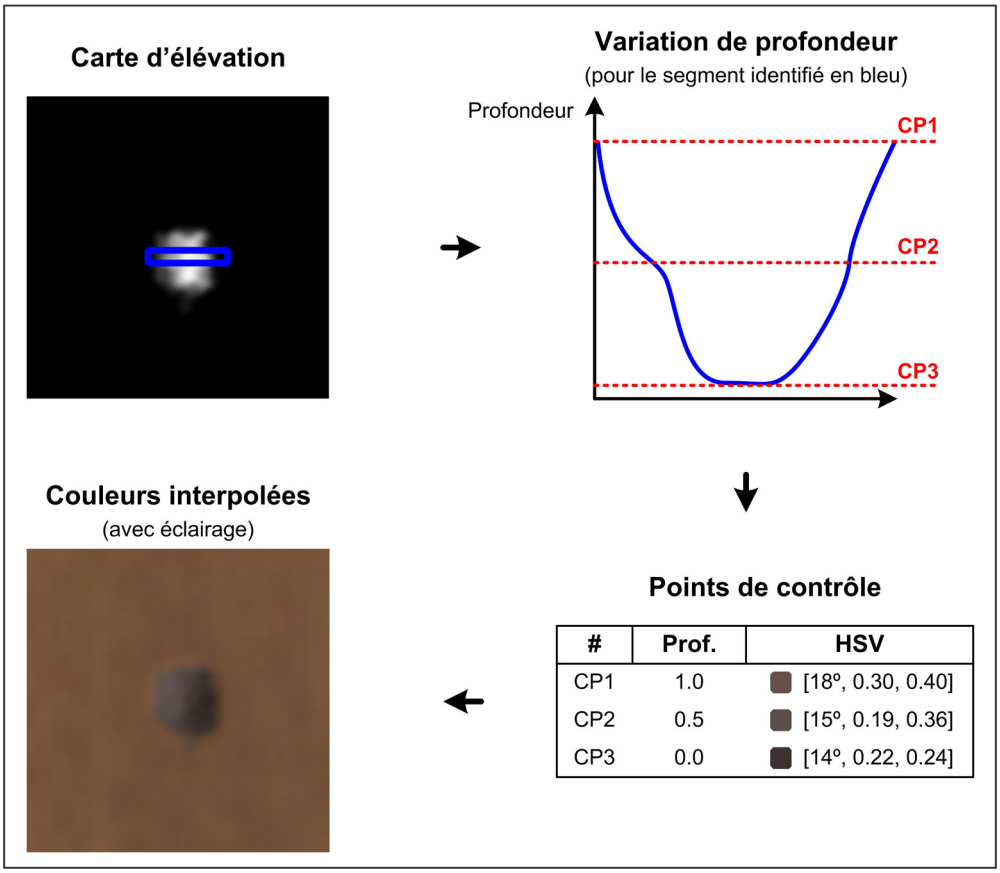

Figure 4.5 Définition des points de contrôle pour l'interpolation.

L'utilisation de cet espace de couleurs permet à l'artiste de définir les points de contrôle nécessaires à l'interpolation beaucoup plus intuitivement. En effet, dans le contexte des effets de détérioration, la teinte ne varie pas vraiment en fonction de la profondeur. C'est plutôt la saturation et l'intensité qui changent selon la profondeur. Dans plusieurs cas, l'effet de détérioration montre la même couleur dominante mais s'assombrit avec la profondeur de l'effet. En utilisant des points de contrôle en RVB, l'artiste aurait eu de la difficulté à simuler ce type de relation alors que c'est très simple en utilisant l'espace HSV.

Bien que l'approche décrite par la figure 4.5 soit préconisée pour une majorité de résultats présentés à la section 4.5, quelques alternatives ont été explorées. Une option intéressante a retenue l'attention. Dans certains cas où l'effet de détérioration est relativement mineur, il arrive parfois que la couleur à l'intérieur de la région usée soit fortement corrélée à la couleur de la texture de base de l'objet à cet endroit. Or, l'approche décrite précédemment ne tient pas compte de la texture de base pour générer les couleurs des effets de détérioration. L'apparence des effets est entièrement ajuster par les points de contrôle fournis par l'artiste. Pour régler ce problème, il est possible pour l'utilisateur de spécifier, toujours à l'aide des points de contrôle, une variation plutôt qu'une couleur absolue. Les points de contrôle prennent donc une forme semblable à celle-ci :

Profondeur de 0.0 → +10º pour la teinte, -10% pour la saturation et -10% pour l'intensité
Profondeur de 0.5 → +15º pour la teinte, -15% pour la saturation et -20% pour l'intensité
Profondeur de 1.0 → +20º pour la teinte, -20% pour la saturation et -30% pour l'intensité

Dans cette situation, le système utilise la couleur contenue dans la texture de base de l'objet et lui applique une variation interpolée en fonction des points de contrôle et de la profondeur de l'effet. À titre d'exemple, si la couleur de base en HSV est [245º, 60%, 80%] pour une profondeur d'effet de 0.5, la couleur affichée sera de [260º, 45%, 60%]. Évidemment, la variation est interpolée linéairement pour les valeurs de profondeur intermédiaires. Bref, cette alternative pour la définition des points de contrôle permet au système de prendre en considération les couleurs contenues dans la texture de base de l'objet pour générer l'apparence finale des effets de détérioration.

4.4.2 Création d'une carte de normales

À partir de la carte d'élévation et des points de contrôle définis par l'utilisateur, le système est en mesure de déterminer la couleur à afficher pour chaque pixel des effets de détérioration. Bien que cette approche permette de générer des valeurs RVB à afficher, son niveau de réalisme n'est pas toujours suffisant. En effet, dans cette situation, il arrive souvent que les effets ajoutés à la surface d'un objet paraissent aplatis et sans relief. Or, les effets de détérioration comme des bosses ou des égratignures altèrent légèrement la géométrie à la surface de l'objet usé. Malheureusement, ces changements de géométrie ont un impact important sur l'apparence finale des effets. Pour considérer ce nouvel aspect, le processus de synthèse proposée intègre l'utilisation d'une carte de normales qui est dérivée de la carte d'élévation. Cette carte de normales contient un vecteur trois dimensions représentant la variation de l'orientation de la surface pour chaque texel. Comme le montre la figure 4.6, ces vecteurs normaux sont relativement simples à calculer à partir de la carte d'élévation en considérant le changement de profondeur d'un texel par rapport à son voisinage.

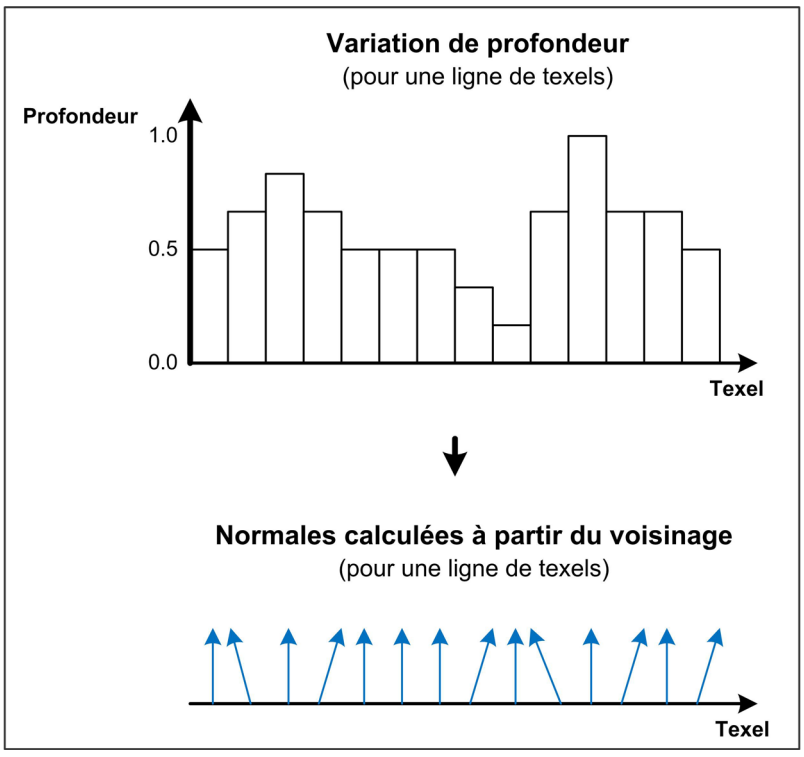

Figure 4.6 Explication du calcul de la carte de normales.

Dans le cadre du projet, la carte de normales est calculée à partir de la carte d'élévation en utilisant l'outil « *NormalMapFilter* » développé par NVIDIA. Plusieurs autres outils disponibles dans les logiciels d'édition d'images standards permettraient d'obtenir un résultat similaire. Par la suite, comme l'explique la figure 4.2, la carte de normales est fournie au nuanceur décrit à la section 4.4.3. Ce dernier prend en compte les variations de normales dans le calcul d'éclairage de manière à donner une apparence de relief aux effets de détérioration contenus dans la carte d'élévation. La figure 4.7 montre le processus complet avec et sans l'utilisation d'une carte de normales.

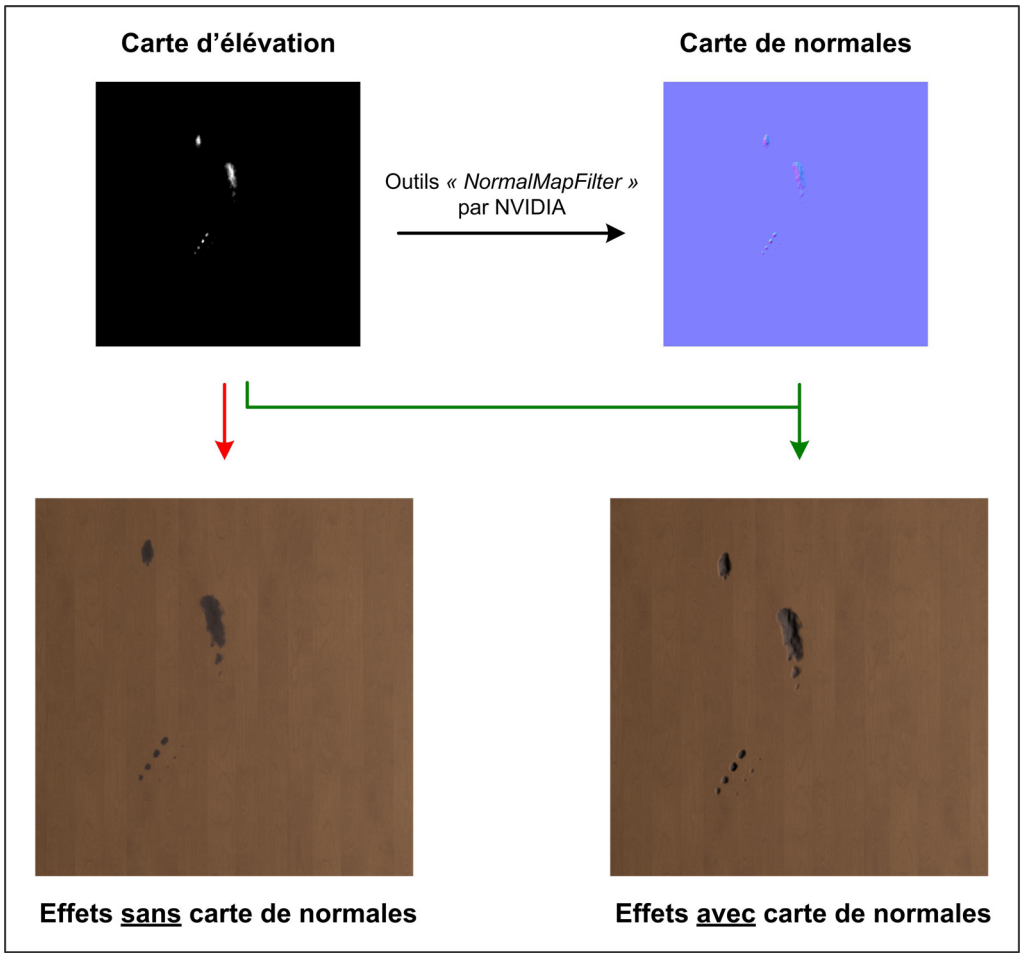

Figure 4.7 Comparaison des effets avec et sans une carte de normales.

4.4.3 Algorithme en temps réel

À partir de la carte d'élévation générée par le processus de synthèse de textures, de la carte de normales dérivée et des points de contrôle spécifiés par l'utilisateur, le système est maintenant en mesure d'interpréter l'information de manière à produire une couleur pour chaque pixel à afficher. La figure 4.8 présente l'algorithme du nuanceur exécuté par l'application pour interpoler une couleur à afficher à partir des points de contrôle.

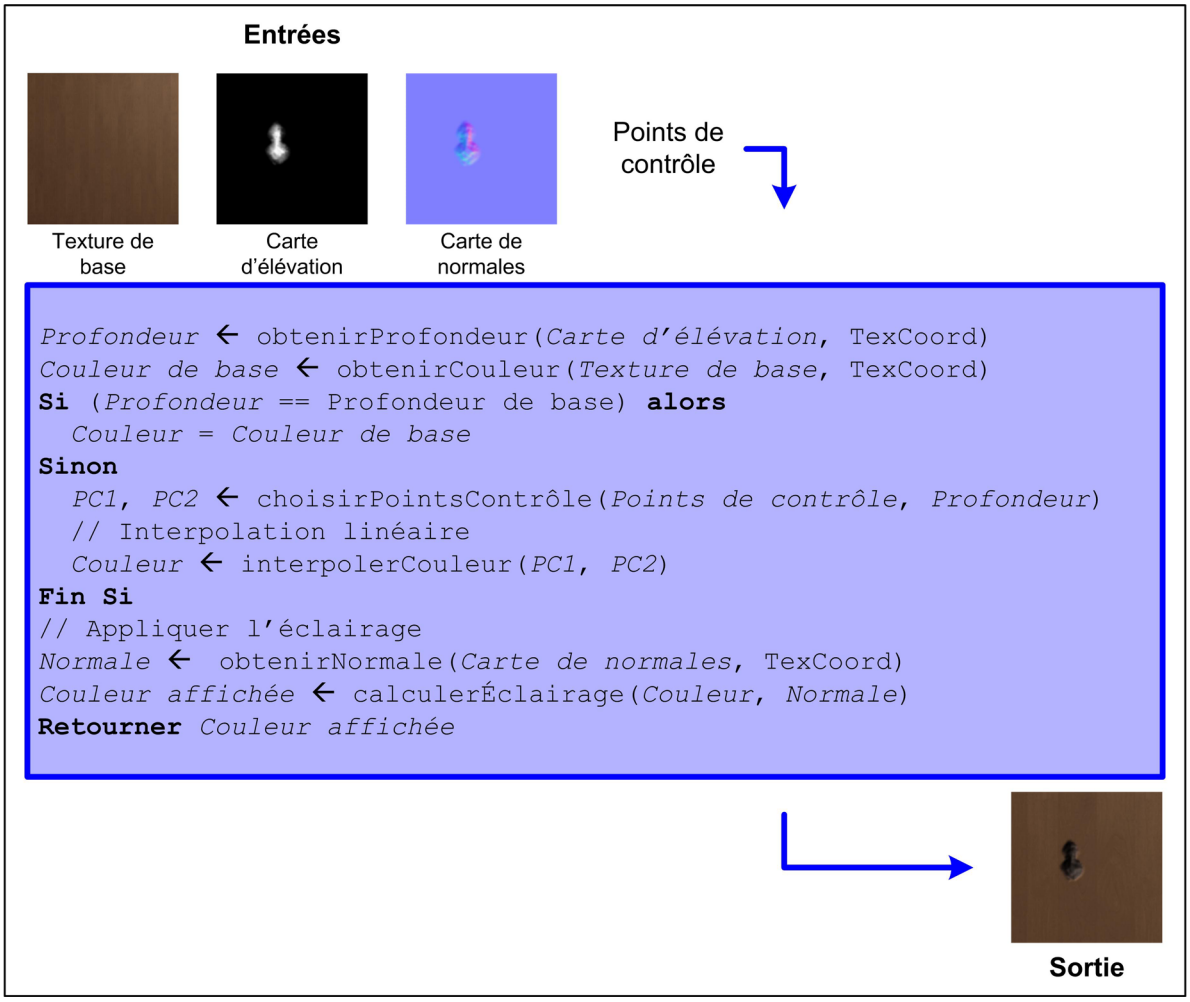

Figure 4.8 Pseudo-code du nuanceur pour l'interpolation d'une couleur.

Pour un pixel donné, l'algorithme consiste à déterminer la profondeur de l'effet à partir de la carte d'élévation et des coordonnées de texture associées à l'objet 3D. Si la profondeur est égale à la profondeur de base, cela signifie qu'il n'y a aucun effet de détérioration à cet endroit et donc que le système affiche la couleur de base tirée de la texture RVB. À l'inverse, si la profondeur n'est pas égale à la profondeur de base, cela signifie qu'il y a un effet de détérioration positionné à cet endroit sur l'objet. L'algorithme cherche donc dans la liste de points de contrôle pour déterminer lesquels il doit utiliser. À titre d'exemple, étudions la situation où quatre points de contrôle sont définis comme suit :

$$\begin{aligned} \text{PC1} = \text{Profondeur de } 0.0 &\rightarrow [15^\circ, 20\%, 40\%]_{\text{HSV}} \\ \text{PC2} = \text{Profondeur de } 0.2 &\rightarrow [20^\circ, 30\%, 30\%]_{\text{HSV}} \\ \text{PC3} = \text{Profondeur de } 0.5 &\rightarrow [25^\circ, 40\%, 20\%]_{\text{HSV}} \\ \text{PC4} = \text{Profondeur de } 1.0 &\rightarrow [45^\circ, 50\%, 10\%]_{\text{HSV}} \end{aligned}$$

Si la profondeur du pixel courant est de *0.3*, l'algorithme sélectionnera les points de contrôle *PC2* et *PC3* pour effectuer l'interpolation puisque l'intervalle formé par ces derniers contient la valeur étudiée. Ensuite, pour déterminer la couleur à afficher le système effectue une interpolation linéaire entre les points de contrôle sélectionné. Dans la situation décrite précédemment, la couleur affichée serait donc calculée de cette façon :

$$\begin{aligned} \textit{ratio} &= (0.3 - 0.2) / (0.5 - 0.2) = 33\% \\ \text{H} &= (25^\circ - 20^\circ) * \textit{ratio} + 20^\circ = 21.67^\circ \\ \text{S} &= (40\% - 30\%) * \textit{ratio} + 30\% = 33.33\% \\ \text{V} &= (20\% - 30\%) * \textit{ratio} + 30\% = 26.67\% \end{aligned}$$

La dernière étape de l'algorithme consiste à appliquer l'éclairage en fonction des sources d'illumination contenues dans la scène. Dans le prototype développé, le processus de calcul d'éclairage est fortement basé sur l'approche standard décrite dans le livre de Rost (2004) qui prend en considération la variation dans l'orientation des surfaces contenue dans la carte de normales. Pour résumer, selon la profondeur associée au pixel courant, l'algorithme

sélectionnera des points de contrôle correspondant et la couleur affichée variera en fonction de l'interpolation effectuée. Le calcul d'éclairage standard est ensuite appliqué en tenant compte de normales modifiées. Évidemment, cette opération doit être exécutée pour chaque pixel à afficher.

Dans le cas où les points de contrôle servent à définir une variation sur la couleur de base (voir section 4.4.1), le nuanceur doit être légèrement modifié pour fonctionner adéquatement. La figure 4.9 présente le pseudo-code pour cette nouvelle version de l'algorithme. Les changements par rapport à l'algorithme original y sont indiqués en rouge.

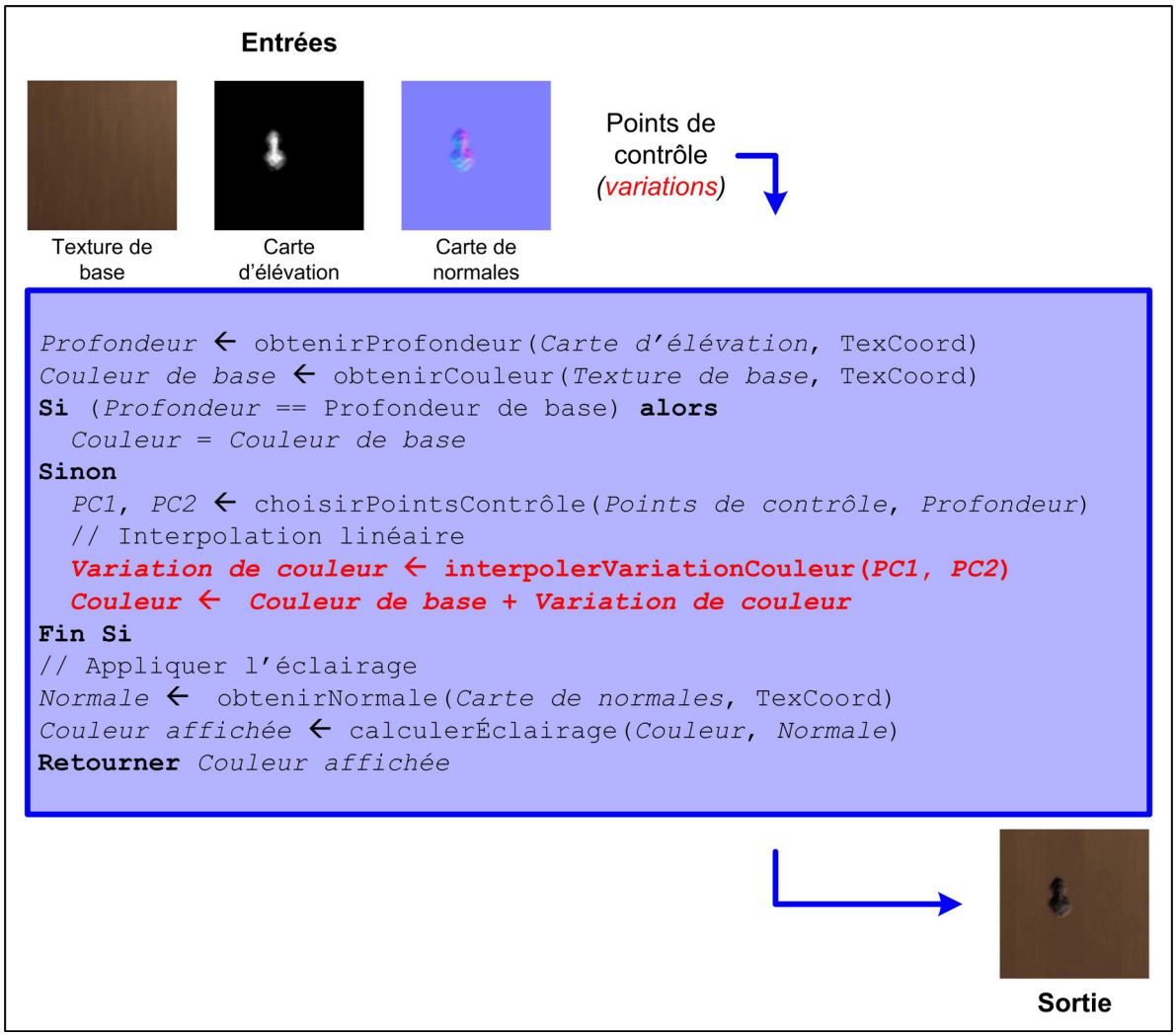

Figure 4.9 Pseudo-code du nuanceur pour l'interpolation d'une variation de couleur.

4.5 Présentation des résultats

Cette section présente un ensemble de résultats obtenus à l'aide de l'approche de synthèse indépendante de la couleur décrite dans ce chapitre. Tout d'abord, la figure 4.10 imite les cas réels de la figure 4.1 en présentant plusieurs teintes d'effets similaires sur du bois.

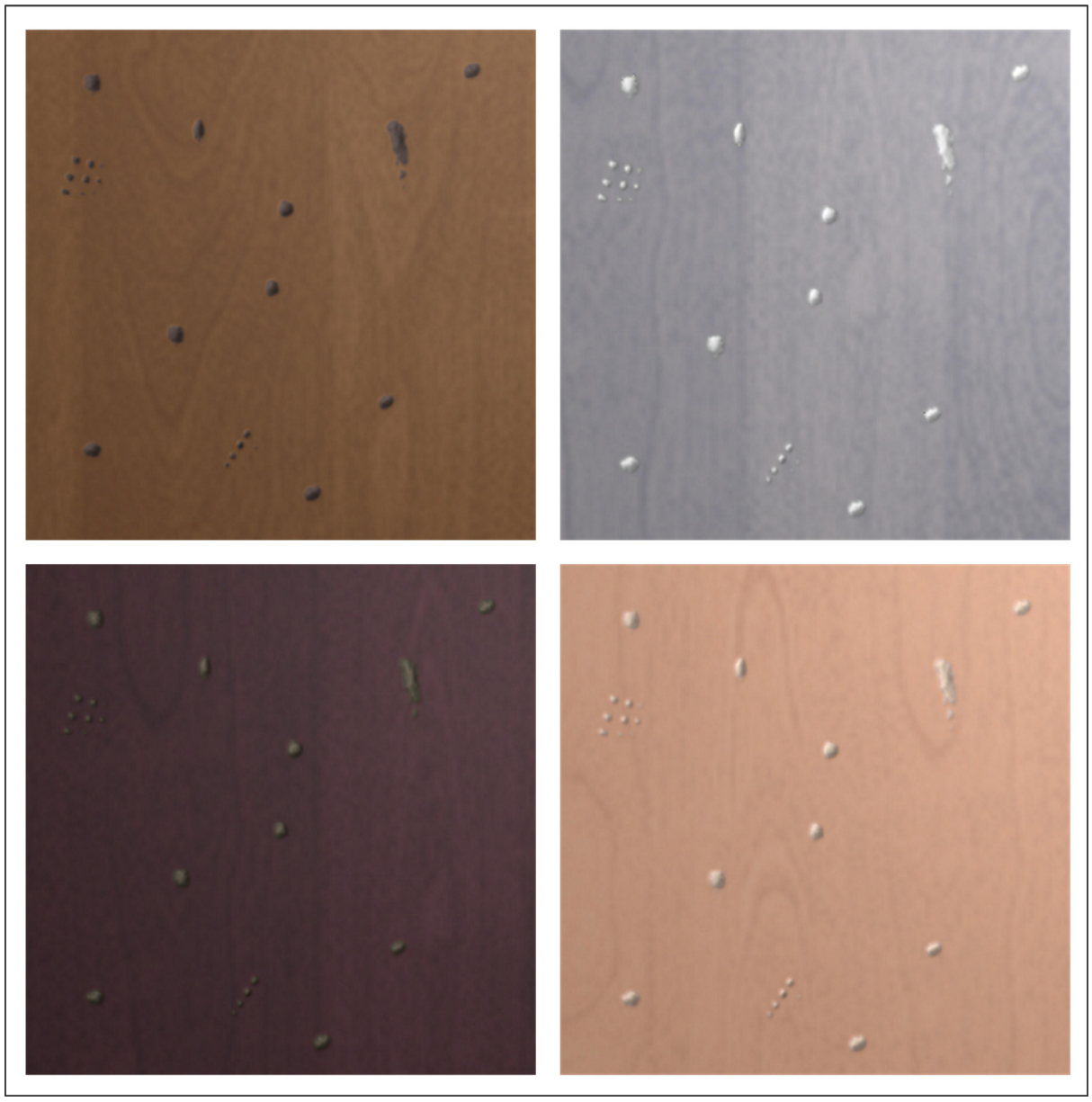

Figure 4.10 Résultats illustrant plusieurs teintes d'effets similaires sur du bois.

La figure 4.11 montre la propagation de patine et de moisissure, respectivement sur une sculpture de bronze et de pierre. De son côté, la figure 4.12 contient plusieurs occurrences détériorées pour différentes teintes d'une même chaise.

Figure 4.11 Résultats illustrant la propagation de patine et de moisissure.

Figure 4.12 Résultats illustrant plusieurs occurrences de teintes différentes.

Tout comme le processus présenté au chapitre 2, l'approche proposée dans ce chapitre pour la synthèse indépendante de la couleur fonctionne sur une variété de matériaux, tels que le bois, le bronze et la pierre. L'implémentation du nuanceur décrit à la section 4.4.3 contient 52 instructions spécifiques à l'interpolation dont seulement une boucle. En utilisant des textures de 1024 par 1024 et une carte vidéo commerciale commune, un objet présentant une détérioration typique est affiché sans ralentissement, soit à 60 images par seconde. Même dans le cas extrême où tous les pixels seraient usés, l'exécution du nuanceur n'a aucun impact sur la vitesse de défilement de l'affichage. Finalement, pour clore cette section, le tableau 4.1 regroupe les données relatives à la performance de l'algorithme de synthèse de l'approche proposée pour tous les résultats présentés de cette section. Pour une critique détaillée de l'ensemble du processus, veuillez consulter la section 4.6.

Tableau 4.1 Données sur les performances du processus indépendant de la couleur

Résultats	Dimension	Nombre de pixels	Temps de calcul du système (secondes)
Figure 4.10	1024x1024	1492	2
Figure 4.12 (2)	1024x1024	8835	8
Figure 4.12 (1)	1024x1024	8900	8
Figure 4.12 (3)	1024x1024	9122	8
Figure 4.11	1024x1024	140708	108

4.6 Discussion

Cette section contient un bilan du processus de synthèse indépendant de la couleur décrit dans ce chapitre. Une discussion divisée en deux volets permet de mettre en évidence les avantages de la méthode proposée, ainsi que ses limitations.

4.6.1 Les avantages

Tout d'abord, l'approche de synthèse indépendante de la couleur proposée dans ce chapitre permet de solutionner un des problèmes majeurs des méthodes de détérioration basées sur la capture d'images en réduisant la quantité d'exemples nécessaires. En effet, le processus permet à l'utilisateur d'ajuster la teinte de l'effet à l'aide de points de contrôle sans nécessiter un exemple spécifique. Conséquemment, l'utilisateur peut démarrer le processus avec un exemple d'une teinte quelconque et produire en sortie l'effet dans la couleur désirée. De plus, comme expliqué dans les sections 4.4.1 et 4.4.3, la définition et l'interpolation des points de contrôle peuvent être modifiés pour générer une couleur complètement indépendante de la texture de base de l'objet ou pour produire une simple variation. Dans les deux cas, il s'agit d'une approche simple et intuitive pour l'artiste qui lui permet une plus grande latitude d'utilisation que les approches standards.

Par ailleurs, l'introduction du concept de carte d'élévation dans le processus de synthèse permet la génération d'une carte de normales décrivant les changements dans l'orientation de la surface des régions détériorées de l'objet. Or, le processus de synthèse standard utilisant les canaux RVB ne permettait pas de dériver facilement une telle carte de normales. Tel qu'illustré à la section 4.4.2, l'exploitation des variations dans les vecteurs normaux permet de créer une illusion de relief qui augmente considérablement le réalisme des effets générés.

Du côté de la performance, le tableau 4.1 montre clairement la rapidité de l'algorithme de synthèse des cartes d'élévation utilisées dans le processus. Dans plusieurs cas, quelques secondes suffisent au système pour générer une carte d'élévation correspondant aux paramètres fournis par l'utilisateur. Même dans des cas plus complexes comme la figure 4.11 où un nombre important de texels sont détériorés, le prototype ne prend que quelques minutes à générer son résultat. Ces délais interactifs peuvent s'expliquer en partie à cause de l'utilisation d'un seul canal dans le processus de synthèse (une image en niveau de gris représentant la profondeur de l'effet) par opposition aux canaux RVB standards. Cette simplification réduit le temps de calcul nécessaire par l'algorithme, particulièrement pour la recherche des meilleures correspondances. Cette interactivité représente un atout important de la technique puisqu'elle favorise l'utilisation d'une approche itérative permettant à un artiste de retravailler rapidement ses résultats suite aux commentaires et critiques qu'il reçoit. Finalement, le nuanceur développé pour l'interprétation des cartes d'élévation et des points de contrôle est relativement simple et n'affecte pas significativement les performances d'une application exécutée en temps réel. Dans ce contexte, son exploitation n'est pas limitée aux applications pouvant se permettre de générer un rendu en différé.

4.6.2 Les limitations

Tout comme le processus présenté au chapitre 2, l'approche de synthèse indépendante de la couleur ne permet pas de traiter les effets modifiant considérablement la géométrie 3D d'un objet, comme un des fractures ou des déformations majeures. L'utilisation des cartes de

normales augmente les possibilités par rapport à la production de relief pour les effets générés mais cette approche montre des limites et ne fonctionne adéquatement que pour des déformations mineures. De plus, l'approche indépendante de la couleur ajoute quelques limitations en ce qui concerne les types d'effets supportés. L'interpolation de points de contrôle pour générer les couleurs à afficher fonctionne parfaitement dans les cas où les effets montrent des variations simples par rapport à la couleur des régions détériorées (voir les figures 4.10, 4.11 et 4.12). Par contre, dans des cas plus difficiles où les effets présentent une texture interne plus complexe (voir les figures 2.18 et 3.18), l'utilisation de points de contrôle ne permet pas de recréer efficacement ces patrons. Pour régler ce problème, il serait possible d'utiliser à la fois la synthèse RVB standard et la synthèse de carte d'élévation. La synthèse RVB permettrait de reproduire correctement la texture interne des régions détériorées alors que la synthèse de carte d'élévation permettrait l'exploitation de points de contrôle pour générer des variations sur la teinte. De cette façon, le processus pourrait rester indépendant de la couleur de l'exemple fourni, tout en permettant de recréer des patrons internes plus complexes.

Finalement, un dernier problème associé au processus de synthèse indépendant de la couleur concerne la calibration des points de contrôle. En effet, dans son état actuel, le système demande à l'utilisateur de fournir des valeurs HSV pour les points de contrôle avant l'application du calcul d'éclairage. Conséquemment, si l'artiste souhaite avoir des effets de détérioration bleus, il fournira la couleur $[240º, 100\%, 100\%]_{HSV}$. Cependant, selon les conditions d'éclairage de la scène, la couleur affichée divergera légèrement de la couleur spécifiée, ce qui peut devenir irritant puisque l'artiste doit souvent réajuster les points de contrôle suite à un premier affichage. Il serait plus intuitif que les couleurs spécifiées correspondant exactement aux couleurs affichées. Pour ce faire, le système devrait prendre en compte les conditions d'éclairage de la scène et faire le calcul inverse de l'éclairage pour ajuster adéquatement les points de contrôle fournis par l'utilisateur.

CONCLUSION

De nos jours, le niveau de réalisme attendu des images synthétiques utilisées dans les jeux vidéo, les films d'animation ou les systèmes de réalité virtuelle a augmenté considérablement. Pour produire des répliques conformes à des situations réelles, les systèmes de rendu réaliste doivent considérer une quantité impressionnante de détails. Parmi ces détails, les effets produits par le vieillissement des objets sont particulièrement difficiles à traiter et requièrent beaucoup de travail manuel. Tel qu'exposé dans cette thèse, les approches existantes adressant cette problématique comme les simulations basées sur la physique ou les simulations basées sur des règles empiriques ne sont pas adaptées aux artistes puisqu'elles offrent souvent un contrôle peu intuitif sur l'apparence des résultats finaux. Par ailleurs, la grande majorité des ces approches sont conçues spécifiquement pour traiter un type d'effet de détérioration particulier, ce qui a comme conséquence de multiplier le temps d'apprentissage nécessaire pour être en mesure d'ajouter un ensemble complet d'effets de détérioration à un environnement virtuel. Faute d'outils appropriés pour ajouter ces effets de détérioration, cette tâche fondamentale pour l'obtention d'images synthétiques réalistes est plus souvent qu'autrement fait complètement manuellement. Dans ce contexte, l'objectif principal de ce projet de recherche était d'offrir une structure polyvalente permettant aux artistes et aux studios de production de contrôler la détérioration de leurs objets synthétiques de façon simple et intuitive.

Le premier volet du projet consistait à développer une approche de génération d'effets de détérioration basée sur la capture d'une image contenant un exemple de l'effet désiré. Le processus de capture de cet exemple, souvent une simple photographie, représente un paramètre avec lequel un artiste est familier. Le contrôle sur l'apparence finale du résultat s'effectue à l'aide d'un masque cible spécifiant les régions à détériorer. Encore une fois, ce paramètre de contrôle est intuitif pour l'artiste puisque son lien avec l'apparence finale est direct, contrairement aux paramètres physiques traditionnels. L'approche de génération automatique proposée exploite un algorithme de synthèse de textures par remplissage de

trous adapté pour la détérioration. Les résultats obtenus à partir du prototype montrent clairement la polyvalence de la technique puisqu'elle fonctionne avec une grande variété d'effets de détérioration et de matériaux. Les temps de calculs nécessaires sont minimes et l'interaction avec l'utilisateur est limitée à quelques minutes, selon la complexité de la détérioration à appliquer. Bref, l'approche proposée possède toutes les qualités requises pour s'intégrer efficacement dans le processus de création itératif préconisé dans les studios de production.

Le deuxième volet du projet de recherche consistait à proposer une approche de génération automatique des masques cibles utilisés pour contrôler le processus de synthèse. Dans la majorité des cas, la création manuelle de ces masques représente un goulot d'étranglement important de l'approche initialement proposée. En intégrant le concept de recette de détérioration basée sur des propriétés locales comme l'accessibilité et la courbure, l'artiste est maintenant en mesure de définir un patron général caractérisant l'usure d'un objet. Cette recette peut être appliquée soit à un objet pour en générer une ou plusieurs occurrences, soit à plusieurs objets distincts pour produire des détériorations comparables. Les masques cibles correspondant sont automatiquement générés en ne demandant aucun travail manuel par l'artiste. Les résultats obtenus montrent la même polyvalence en termes de variété d'effets de détérioration et de matériaux. Encore une fois, les temps de calculs nécessaires sont minimes et favorisent une utilisation interactive sans délais importants.

Le dernier volet de ce projet de recherche consistait à altérer le processus de synthèse préalablement défini de manière à le rendre indépendant de la couleur de l'exemple fourni en entrée par l'utilisateur. Dans ce contexte, l'approche standard exploitant les canaux de couleurs RVB est remplacée par la synthèse d'un niveau d'intensité de détérioration. Ensuite, le système interprète cette intensité à l'aide de points de contrôle définis par l'utilisateur et d'un nuanceur exécuté par l'application de manière à produire une couleur à afficher pour chaque pixel faisant partie des effets de détérioration. Bien que la variété d'effets de détérioration représentables à l'aide de ce processus indépendant de la couleur soit un peu limitée, il offre une polyvalence intéressante dans les situations appropriées. Les temps de

calculs nécessaires à la synthèse sont négligeables et le nuanceur peut être exécuté sans problème dans le contexte d'une application temps réel.

Bref, le regroupement des trois volets de ce projet de recherche représente une structure complète pour faire la synthèse d'effets de détérioration sur l'ensemble d'un environnement virtuel. Contrairement à plusieurs autres approches existantes, elle a été développée en considérant les exigences fondamentales de ses utilisateurs, des artistes, en ce qui a trait à la complexité des paramètres, à la simplicité du contrôle sur l'apparence finale, aux temps de calculs nécessaires puis à la polyvalence de la technique relativement à la variété d'effets supportés. Pour toutes ces raisons, l'approche proposée constitue une contribution scientifique importante dans le domaine de la synthèse d'images réalistes.

Évidemment, plusieurs améliorations pourraient être apportées à l'approche proposée pour la rendre encore plus polyvalente. Tout d'abord, il serait intéressant d'intégrer au processus un module permettant d'appliquer des modifications à la géométrie 3D d'un objet correspondant aux effets de détérioration générés. Tel que mentionné précédemment, l'approche proposée se limite aux effets de surface n'altérant pas substantiellement la structure d'un objet. Or, une telle limitation est considérable puisque les effets touchant de façon significative la géométrie d'un objet sont courants dans les situations réelles. Pour augmenter la polyvalence de la technique, il serait possible d'intégrer des paramètres de déformations 3D à la recette de détérioration pour traiter ces situations. Dans le même ordre d'idées, il arrive régulièrement que des effets de détériorations prennent la forme de lettres ou de symboles distinctifs (graffiti, gravure, etc.). Malheureusement, les transformations appliquées pendant le processus de génération des masques cibles ne traitent pas adéquatement ces situations. Pour réduire davantage la quantité de travail manuel fait par un artiste, il serait intéressant de développer de nouveaux opérateurs de transformation permettant de produire des masques cibles cohérents dans ces situations spécifiques. Finalement, il pourrait aussi être intéressant de proposer une solution à la consommation considérable de mémoire causée par l'utilisation de textures. En effet, le processus proposé demande l'utilisation d'une texture distincte, contenant des effets de détérioration différents, pour chaque instance d'un objet à user.

Particulièrement, dans le contexte d'applications exécutées en temps réel, la consommation de mémoire associée à un tel processus peut devenir problématique. Dans ces situations, il serait avantageux de développer une structure permettant de contenir l'information sur les détériorations de manière plus efficace. À titre d'exemple, il serait possible de regrouper l'ensemble des effets de détérioration dans une seule et unique texture, semblable à un atlas, et de conserver leur positionnement respectif sur les objets dans une structure de données correspondante. À l'exécution de l'application, cette structure de données serait parcourue pour positionner correctement les effets extraits de l'atlas. Ainsi, la consommation de mémoire pourrait être diminuée considérablement.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Aoki, Kimiya, Ngo Hai Dong, Toyohisa Kaneko et Shigeru Kuriyama. 2004. « Physically Based Simulation of Cracks on Drying 3D Solids ». In *Computer Graphics International 2004 Proceedings*. p. 357-364. Washington (DC): IEEE Computer Society.

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, n° 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): ACM Press.

Barrett, William A., et Alan S. Cheney. 2002. « Object-Based Image Editing ». *ACM Transactions on Graphics,* vol. 21, n° 3, p. 777-784.

Beraldin, J.-A., L. Cournoyer, M. Rioux, F. Blais, S. F. El-Hakim et G. Godin. 1997. « Object Model Creation From Multiple Range Images: Acquisition, Calibration, Model Building and Verification ». In *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*. p. 326-333. Washington (DC): 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.

Bosch, Carles, Xavier Pueyo, Stéphane Mérillou et Djamchid Ghazanfarpour. 2004. « A Physically-Based Model for Rendering Realistic Scratches ». *Computer Graphics Forum,* vol. 23, n° 3, p. 361-370.

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, n° 11, p. 1222-1239.

Bradski, Gary, et Adrian Kaehler. 2008. *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol (CA): O'Reilly Media, 576 p.

Brooks, Stephen, et Neil Dodgson. 2002. « Self-Similarity Based Texture Editing ». *ACM Transactions on Graphics,* vol. 21, n° 3, p. 653-656.

Canny, John F. 1986. « A Computational Approach to Edge Detection ». *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, n° 6, p. 679-698.

Chang, Yao-Xun, et Zen-Chung Shih. 2001. « Physically-Based Patination for Underground Objects ». *Computer Graphics Forum,* vol. 19, n° 3, p. 109-117.

Chang, Yao-Xun, et Zen-Chung Shih. 2003. « The Synthesis of Rust in Seawater ». *The Visual Computer,* vol. 19, n° 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, n° 3, p. 1127-1133.

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*. p. 151-158. New York (NY): ACM Press.

Clément, Olivier, et Eric Paquette. 2010. « Adaptable Aging Factory for Multiple Objects and Colorations ». *Computers & Graphics,* vol. 34, n° 4, p. 460-467.

Cutler, Barbara, Julie Dorsey, Leonard McMillan, Matthias Müller et Robert Jagnow. 2002. « A Procedural Approach to Authoring Solid Models ». *ACM Transactions on Graphics,* vol. 21, n° 3, p. 302-311.

De Bonet, Jeremy S. 1997. « Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images ». In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. p. 361-368. New York (NY): ACM Press / Addison-Wesley Publishing Co.

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, et Pat Hanrahan. 2000. « Digital Materials and Virtual Weathering ». *Scientific American,* vol. 282, n° 2, p. 46-53.

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, n° 3, p. 303-312.

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. 341-349. New York (NY): ACM Press.

Efros, Alexei A., et Thomas K. Leung. 1999. « Texture Synthesis by Non-Parametric Sampling ». In *Proceedings of the International Conference on Computer Vision - Volume 2*. p. 1033-1038. Washington (DC): IEEE Computer Society Press.

Farouk, Mohamed, Ibrahim El-Rifai, Shady El-Tayar, Hisham El-Shishiny, Mohamed Hosny, Mohamed El-Rayes, Jose Gomes, Frank Giordano, Holly Rushmeier, Fausto Bernardini et Karen Magerlein. 2003. « Scanning and Processing 3D Objects for Web Display ». In *Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling, 2003*. p. 310-317. Washington (DC): IEEE Computer Society Press.

Gobron, Stéphane, et Norishige Chiba. 2001a. « Crack Pattern Simulation Based on 3D Surface Cellular Automata ». *The Visual Computer,* vol. 17, n° 5, p. 287-309.

Gobron, Stéphane, et Norishige Chiba. 2001b. « Simulation of Peeling Using 3D-Surface Cellular Automata ». In *Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*. p. 338-347. Washington (DC): IEEE Computer Society Press.

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

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, n° 3, p. 762-771.

Hamann, Bernd. 1993. « Curvature Approximation for Triangulated Surfaces ». In *Geometric Modelling*. p. 139-153. London (UK): Springer-Verlag.

Heeger, David J., et James R. Bergen. 1995. « Pyramid-Based Texture Analysis / Synthesis ». In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. New York (NY): ACM Press.

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. New York (NY): ACM Press.

Hsu, Siu-chi, et Tien-Tsin Wong. 1995. « Simulating Dust Accumulation ». *IEEE Computer Graphics and Applications,* vol. 15, n° 1, p. 18-22.

Kindermann, Ross, et J. Laurie Snell. 1980. *Markov Random Fields and Their Applications*. Providence (RI): American Mathematical Society, 142 p.

Kontkanen, Janne, et Samuli Laine. 2005. « Ambient Occlusion Fields ». In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*. p. 41-48. Washington (DC): ACM Press.

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, n° 3, p. 277-286.

Lefebvre, Sylvain, et Hugues Hoppe. 2006. « Appearance-Space Texture Synthesis ». *ACM Transactions on Graphics,* vol. 25, n° 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, n° 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, n° 3, p. 646-653.

Loève, Michel. 1978. *Probality Theory II*, 4th ed. Coll. « Graduate Texts in Mathematics », vol. 46. New York (NY): Springer-Verlag, 436 p.

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, n° 1, p. 3.

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

Mérillou, Stéphane, Jean-Michel Dischler et Djamchid Ghazanfarpour. 2001b. « Surfaces Scratches: Measuring, Modeling and Rendering ». *The Visual Computer,* vol. 17, n° 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, n° 2, p. 159-174.

Meyer, Mark, Mathieu Desbrun, Peter Schröder et Alan H. Barr. 2003. « Discrete Differential-Geometry Operators for Triangulated 2-Manifolds ». In *Visualization and Mathematics III*. p. 35-57. Heidelberg: Springer Verlag.

Miller, Gavin. 1994. « Efficient Algorithms for Local and Global Accessibility Shading ». In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*. New York (NY): ACM Press.

O'Brien, James F., Adam W. Bargteil et Jessica K. Hodgins. 2002. « Graphical Modeling and Animation of Ductile Fracture ». *ACM Transactions on Graphics,* vol. 21, n° 3, p. 291-294.

Paquette, Eric, Pierre Poulin et George Drettakis. 2001. « Surface Aging by Impacts ». In *Graphics Interface 2001 Conference Proceedings*. p. 175-182. Toronto (Ont.): Canadian Information Processing Society.

Paquette, Eric, Pierre Poulin et George Drettakis. 2002. « The Simulation of Paint Cracking and Peeling ». In *Graphics Interface 2002 Conference Proceedings*. p. 59-68. Toronto (Ont.): Canadian Information Processing Society.

Rost, Randi J. 2004. *OpenGL(R) Shading Language*. Boston (MA): Addison-Wesley, 565 p.

Sun, Jian, Lu Yuan, Jiaya Jia et Heung-Yeung Shum. 2005. « Image Completion With Structure Propagation ». *ACM Transactions on Graphics,* vol. 24, n° 3, p. 861-868.

Taubin, Gabriel. 1995. « Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation ». In *Proceedings of the 5th International Conference on Computer Vision*. p. 902-908. Washington (DC): IEEE Computer Society.

Theisel, Holger, Christian Rössl, Rhaleb Zayer et Hans-Peter Seidel. 2004. « Normal Based Estimation of the Curvature Tensor for Triangular Meshes ». In *Proceedings of the 12th Pacific Conference on Computer Graphics and Applications* (6-8 Oct. 2004). p. 288-297. Washington (DC): IEEE Computer Society Press.

Turk, Greg. 1992. « Re-tiling Polygonal Surfaces ». *ACM SIGGRAPH Computer Graphics,* vol. 26, n° 2, p. 55-64.

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, n° 3, p. 754-761.

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*. p. 479-488. New York (NY): ACM Press / Addison-Wesley Publishing Co.

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 '97*. p. 139-150. London (UK): Springer-Verlag.