



HAL
open science

Distributions cohérentes de primitives pour le rendu expressif de scènes animées et le rendu en demi-tons

David Vanderhaeghe

► **To cite this version:**

David Vanderhaeghe. Distributions cohérentes de primitives pour le rendu expressif de scènes animées et le rendu en demi-tons. Interface homme-machine [cs.HC]. Université Joseph-Fourier - Grenoble I, 2008. Français. NNT: . tel-00345353v1

HAL Id: tel-00345353

<https://theses.hal.science/tel-00345353v1>

Submitted on 11 Dec 2008 (v1), last revised 6 Jun 2011 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER GRENOBLE 1

Spécialité : Mathématiques et Informatique

préparée au

Laboratoire Jean Kuntzmann

dans le cadre de

l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique

Distributions cohérentes de primitives pour le rendu expressif de scènes animées et le rendu en demi-tons

préparée par

David VANDERHAEGHE

soutenue publiquement le 24 novembre 2008

Composition du jury

Victor OSTROMOUKHOV	Président	Professeur associé à l'Université de Montréal
Pascal GUITTON	Rapporteur	Professeur à l'Université de Bordeaux 1
Frédo DURAND	Rapporteur	Professeur associé au MIT
François X. SILLION	Directeur	Directeur de recherche à l'INRIA Rhône-Alpes
Joëlle THOLLOT	Co-Directrice	Maître de conférence à l'INPG

Résumé

Cette thèse s'intéresse à la distribution de primitives graphiques dans les domaines du rendu expressif et des systèmes d'impressions. Une distribution est le placement d'objets dans un espace, dans notre cas un espace à deux dimensions. Cette distribution suit une fonction d'importance décrivant la densité de primitives souhaitée en chacun des points de l'espace. La qualité de la distribution se mesure par l'observation de la position des objets placés, notamment au moyen d'analyses spectrales.

Dans le domaine du rendu expressif, nous nous intéressons à la distribution de points dans le cadre du rendu animé à base de marques. Le rendu à base de marques permet d'obtenir des images mimant des médias traditionnels tels que la peinture. L'animation est obtenue en déplaçant les marques d'une image à l'autre. Pour obtenir un rendu visuellement satisfaisant, les marques doivent suivre un mouvement en entrée et remplir l'image suivant une fonction d'importance pour correctement représenter le style visé. De plus, les changements brutaux sont à éviter. Nous appelons cohérence temporelle le compromis entre ces trois contraintes. Nous développons une approche complète pour le rendu animé à base de marques proposant plusieurs méthodes de distribution de marques ainsi qu'un mécanisme de contrôle du style.

Dans le cadre des systèmes d'impressions, nous proposons une méthode de distribution de points d'encre avec de bonnes qualités d'un point de vue perceptif, supprimant les répétitions présentes dans les travaux précédents. Cette méthode est suffisamment peu coûteuse en temps de calcul pour être utilisée par les imprimantes actuelles.

Mot clés

Rendu expressif, rendu non-photoréaliste, rendu animé stylisé, rendu à base de marques, rendu peinture, distribution dynamique. Rendu en demi-tons, distribution de points.

Coherent distribution of primitives for animations' stylisation and halftoning

Abstract

This thesis deals with graphic primitives distributions in two different domains: expressive rendering and printing system. To distribute is to position objects into a space. We focus on two dimensional space distributions. Moreover, a distribution follow an importance function describing the density of objects locally wanted for each point of the considered space. The quality of the distribution is judged by the observation of the position of the object, using spectral analysis for instance.

In the field of expressive rendering, we focus on distribution for animated stroke-based rendering. Stroke-based rendering is used to obtain images that mimic some natural medias like painting. In our approach, we obtain the animation by moving the stroke from one frame to the other. When we do that, we need to follow an input motion, a given importance function and to correctly represent the targeted style. We call temporal coherence the trade-off made between this three conflicting constraints. We propose a complete system for animated stroke-based rendering with several distribution algorithms following the input motion and the importance function, and also with control of the rendering style.

For printing system, we propose a distribution method for dot of ink. The method produces perceptually pleasant distribution while staying sufficiently efficient to be implemented in printers available nowadays.

Keywords

Expressive rendering, non photo-realistic rendering, animated stylized rendering, stroke-based rendering, painterly rendering, dynamic distribution. Halftoning, point distribution.

Remerciements

Cette thèse n'aurait pas pu ce faire sans l'aide, le dévouement, le soutien et l'encadrement de mes directeurs François X. SILLION et Joëlle THOLLOT. Merci pour le temps et les encouragements nécessaires à la finalisation de mon travail.

De plus durant ces trois années je me suis épanoui dans l'équipe-projet ARTIS. J'ai pu partager avec chacun des membres de cette équipe, et je tiens à remercier plus particulièrement Pascal BARLA pour ses inépuisables idées, Cyril SOLER pour la rigueur mathématique, Xavier DÉCORET pour la soif de communication et la volonté de bien faire. Julien DIENER, Lionel BABOUD et Lionel ATTY, mes *camarades* de classe de Master 2, finissez votre thèse en beauté (et vite)!

Lors de ma visite au LIGUM, j'ai découvert Victor Ostromoukhov qui m'a apporté beaucoup du point de vue de l'expérimentation, merci. Je tiens aussi à remercier les membres du lab qui m'ont facilité la vie québécoise : Pierre POULIN, Fabrice ROUSSELLE, Yann ROUSSEAU, Marie-Élise CORDEAU.

Je tiens tout particulièrement à remercier les relecteurs qui ont permis à ce manuscrit d'être lisible, et avec peu de fautes d'orthographe et de grammaire. Notamment, l'ajout de virgules donne un certain souffle à la lecture. Merci à vous Pierre BÉNARD, Estelle DUVEAU, Pierre-Edouard LANDES, Henri LASSERRE, Adeline PIHUIT, Maxime TOURINER et Dominique MOUNIER.

Pour finir, je tiens à remercier ma famille et mes amis qui m'ont soutenu durant ces trois années, et plus particulièrement les courageux qui sont venus assister à la soutenance.

*À Myriam et Noam,
mes amours*

Table des matières

1	Introduction	13
I	Rendu expressif de scènes animées	17
2	Rendu animé stylisé	19
2.1	Utilisations de l'animation	20
2.2	Une histoire de style	22
2.2.1	Animation traditionnelle sur celluloïd	23
2.2.2	Détournement des techniques traditionnelles	24
2.2.3	Remplissage de régions pour l'animation	25
2.3	Les problèmes que nous voulons résoudre	26
2.3.1	Place de l'artiste	26
2.3.2	Cohérence temporelle	27
2.4	Contributions	28
3	État de l'art	29
3.1	État de l'art sur la cohérence temporelle	29
3.1.1	Textures de matière en espace image	29
3.1.2	Plaquage de textures en espace objet	30
3.1.3	Éléments visuels et points d'attache	32
3.1.4	Rendu à base de marques	33
3.2	Étude du rendu à base de marques pour l'animation	33
3.2.1	Modèle informatique	34
3.2.2	L'animation à base de marques	35
3.2.3	Problèmes à considérer	35
3.2.4	Travaux précédents	37
	Placement des marques	37
	Scène 3D	38
	Vidéo	39
	Évolution de l'ordre de dessin	40
	Utilisation de calques	42
	Dessin des marques	42
	Contrôle de l'artiste	44
	Récapitulatif	45

4	Placement des points d'attache	47
4.1	Qualité de distribution et compromis	47
4.1.1	Distribution 2D de points	49
4.2	Distribution en espace objet	52
4.2.1	Hierarchie dynamique de points d'attache	52
4.2.2	Sélection du nombre de points d'attache	53
	Subdivision par test d'occlusion	54
	Subdivision par prédiction de densité	54
4.2.3	Résultats	55
4.2.4	Problème de densité	56
	Projection de densité 2D/3D	56
4.3	Distribution en espace image	57
4.3.1	Algorithme de distribution	57
	Distribution statique	58
	Distribution dynamique	59
4.3.2	Cohérence temporelle	60
4.3.3	Résultats	61
	Évaluation statistique	62
	Évaluation visuelle	63
5	Pipeline de rendu animé stylisé	65
5.1	Pipeline d'utilisation	65
5.2	Le style comme une fonction de transfert	66
5.2.1	Fonction de transfert	67
	Propriétés de la scène	67
	Attributs des marques	67
	Lien entre propriétés et attributs	68
5.2.2	Exemples de fonctions de transfert	69
5.2.3	Implémentation	71
5.2.4	Pré et post-traitements	71
5.3	Dessin des marques	72
5.3.1	Utilisation d'image de références, dans la pratique	73
5.3.2	Lissage temporelle des attributs	74
5.3.3	Ordre	75
5.3.4	Fidélité	75
5.3.5	Apparition et disparition par seuillage	76
5.4	Résultats	77
6	Résultats et Discussion	81
6.1	Mesure de la cohérence du rendu final	81
6.2	Contrôle du style	82
7	Simulation des méthodes manuelles en peinture animée	85
7.1	Analyse des méthodes manuelles	85
7.2	Travaux précédents en simulation et animation de peinture	87
7.3	Notre système de peinture animée	89
7.4	Modèle de simulation de peinture	90
7.4.1	Entité de dessin et support	90
	Dessin le long d'un chemin	91

Quantité de matière et calcul de couleur	91
7.4.2 Définition d'un pinceau	93
Quantité de peinture et pression	93
7.4.3 Effet d'épaisseur	94
7.5 Algorithme de peinture automatique	94
7.6 Corrections et modifications d'une peinture	95
7.6.1 Étalement de peinture	95
7.6.2 Corrections de la peinture	96
7.7 Contrôle du style et du rendu	98
7.8 Résultats et perspectives	99
II Rendu en demi ton	101
8 Rendu en demi-tons	103
8.1 Ton et distribution	103
8.2 Système visuel humain	104
8.3 Fonctionnement des imprimantes	105
8.3.1 Principe de base	105
8.3.2 Types d'imprimantes	105
Imprimantes jet d'encre	105
Imprimantes laser	106
8.3.3 Dispersion et groupement de points	107
8.3.4 Traitement de la couleur	107
8.3.5 Traitements particuliers	108
9 État de l'art du rendu en demi-tons	111
9.1 Matrice de seuillage	111
9.1.1 Matrice de seuillage pour éléments de trame groupés	112
9.1.2 Matrice de seuillage pour éléments de trame dispersés	113
9.1.3 Supertrame	117
9.1.4 Discussion et analyse spectrale	118
9.2 Diffusion d'erreur	119
9.3 Optimisation du rendu en demi-tons	119
10 Pavage non-periodique et optimisation de trames	121
10.1 Pavage du plan	121
10.2 Utilisation d'éléments de trame	123
10.3 Optimisation des valeurs de seuil	124
10.3.1 Initialisation des prototypes de bords	126
10.3.2 Initialisation des prototypes des intérieurs	127
10.3.3 Assignation des valeurs de seuil	128
10.3.4 Ajout des nouveaux points	128
11 Résultats, applications et perspectives	129
11.1 Distributions de points bruit bleu	129
11.2 Applications : couleur et supertrame	130
11.3 Améliorations	130

Table des figures	135
Listes des tableaux	137
Listes des algorithmes	137
Listes des programmes	137
Bibliographie	139

Synthèse d'image et artéfacts

LA RECHERCHE en synthèse d'image a pour but d'améliorer la conception, la fabrication et la reproduction des images et des animations avec l'aide des outils de calculs que sont les ordinateurs. La synthèse d'image comprend beaucoup de sous-domaines tels que la modélisation, le traitement du signal, la compression vidéo et le rendu. Nous nous intéressons à ce dernier sous-domaine centré sur le calcul des images par l'ordinateur en fonction des données qui lui sont fournies. Les techniques de rendu donnent de nouveaux outils aux artistes et graphistes pour produire des images et des animations. La montée en puissance des ordinateurs, et notamment celle des cartes graphiques, permet, aujourd'hui, d'obtenir des images époustouflantes de réalisme et d'expression. Les buts des images ainsi créées sont très variés, la communication visuelle — au moyen d'images et d'animations — étant un moyen simple et efficace pour transmettre une information ou une émotion.

La création d'images au moyen des outils informatiques nécessite le développement de logiciels qui permettent aux graphistes de s'exprimer. Développer ces logiciels nécessite de concevoir les algorithmes qui conviennent aux graphistes en apportant des solutions techniques à leurs besoins. Par exemple, lorsqu'un directeur artistique demande à son équipe de concevoir une scène de synthèse composée d'immeubles en verre, il faut être en mesure de calculer les réflexions lumineuses entre les diverses surfaces pour obtenir l'image finale. C'est aujourd'hui possible grâce aux travaux de recherche, mais il reste de nombreuses améliorations à apporter. Tant en termes de temps de calcul qu'en termes d'outils de contrôle du rendu, le domaine de la synthèse d'image est en pleine effervescence depuis le début des années quatre-vingt et continue à mobiliser une grande communauté de chercheurs.

Perception des images

Les images sont faites pour être vues par l'être humain. Notre perception visuelle du monde est traitée par notre système visuel, et c'est l'outil de mesure par

excellence dans notre domaine. Le système visuel humain peut juger de la qualité d'une image, de la vraisemblance d'un rendu et de la justesse d'une animation. Le problème est que notre système visuel n'est pas absolu et qu'il est personnel : on ne voit pas tous la même chose. C'est pourquoi, dans le cadre de la recherche scientifique, d'autres outils de mesure plus absolus sont développés, donnant un résultat numérique, quantitatif. Ces outils de mesure permettent de comparer les travaux entre eux, cependant notre système visuel reste le juge final. Dans le domaine de la synthèse d'image, si deux méthodes ne présentent pas de différences visuelles perceptibles et ont des performances équivalentes, l'une ne peut être déclarée meilleure que l'autre, peu importent les autres mesures.

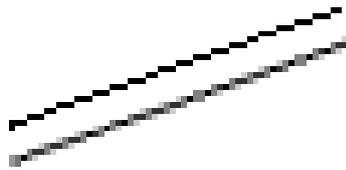


Fig. 1.1 Exemple d'artéfact de rendu provenant de l'utilisation de pixels carrés. La ligne du dessus est rendu sans palliatif, celle du dessous avec un anti-crênelage qui limite l'artéfact.

L'utilisation de l'informatique produit des artéfacts visuels¹ lors de la création d'images. Ces artéfacts sont fortement perçus par notre système visuel. Par exemple, une image est généralement représentée par un tableau, ou grille 2D, de couleurs. Chaque cellule de ce tableau contient des valeurs représentant une couleur appelée pixel (comme *picture element*, élément d'image). Suivant la taille du pixel, nous percevons plus ou moins sa forme (carrée ou rectangulaire sur un écran par exemple), alors que ce n'est pas l'information que nous souhaitons représenter. En fait nous voulons représenter le stimulus à fournir à notre oeil pour que notre système visuel forme une image. Une grille régulière de pixels n'est pas le meilleur moyen pour représenter une image continue, du point de vue de notre système visuel, car nous percevons très fortement les alignements et les frontières, et particulièrement quand ils sont en lignes horizontales ou verticales. Mais les limitations techniques et les avantages d'utiliser cette structure de données ont fait du tableau 2D la référence *de facto* pour représenter une image. Pour limiter les artéfacts de perception, la taille du pixel est rétrécie — ce qui a été rendu possible par l'amélioration des moyens technologiques — et de palliatifs tels que l'anti-crênelage (figure 1.1) sont développés.

Nous nous intéresserons dans cette thèse à deux types d'artefacts : ceux présents dans une animation stylisée, et ceux présents dans les systèmes d'impression.

Lors d'une animation, le système visuel humain perçoit les changements et associe une signification au mouvement des éléments visuels constituant l'image. Tout changement brutal ou rapide attire notre regard, c'est un mécanisme de défense pour nous permettre de détecter rapidement le danger. Ainsi, un des artéfacts majeurs dans l'animation de synthèse est le manque de cohérence temporelle. Intuitivement, il s'agit de changements brutaux dans l'animation qui ne sont pas liés avec le mouvement et la scène que nous voulons représenter. Dans le cadre de notre travail sur le rendu expressif d'animation, nous donnerons une définition plus complète de la cohérence temporelle en section 2.3.2.

Le système visuel perçoit très bien les alignements et les regroupements d'éléments visuels. Pour positionner des éléments visuels (que nous appelons généralement primitives de dessin), de manière équitable, bien répartie, et sans exhiber

1. Un artéfact est un élément indésirable, perturbateur, créé par la méthode de synthèse d'image et dû aux limitations techniques ou à un problème de l'approche choisie.

de groupes, ils doivent être uniformément placés, sans direction privilégiée et sans être trop proches. *Distribuer* est le fait de positionner des primitives ; la *distribution de primitives* ainsi que la *qualité de distribution* sont les termes utilisés pour l'étude d'une méthode particulière.

Outre notre système visuel, les scientifiques ont développé des outils mathématiques de mesures pour juger de la qualité de distribution des différentes méthodes. Nous donnerons plus de détails sur les différents types de distributions, les méthodes et les outils de mesures à la section 4.1. Dans les systèmes d'impression, on est amené à considérer la distribution des points d'encre sur le papier. Cette distribution peut provoquer de nombreux artéfacts et l'on doit prendre un soin particulier lorsque l'on développe les algorithmes de distribution.

Contributions

Dans le cadre du rendu expressif d'animation, nous proposons des travaux sur l'animation de synthèse stylisée. Ces travaux traitent du contrôle par l'artiste et de la gestion des artéfacts liés à l'animation et à la stylisation. Nous présentons l'ensemble de ces travaux en partie I. Nous commençons par introduire le contexte de travail (chapitre 2), puis nous faisons l'état de l'art des travaux précédents s'intéressant à la cohérence temporelle des animations stylisées (chapitre 3) et une étude détaillée du rendu à base de marques (section 3.2), domaine dans lequel s'inscrit notre travail. Nous présentons de nouvelles méthodes pour la distribution dynamique de marques lors d'une animation. Ensuite, nous présentons notre système de rendu à base de marques (chapitre 5). Ce système permet d'obtenir un rendu à base de marques d'une animation donnée sous forme d'une scène 3D [VBTS06a, VBTS06b, VBTS07a, VBTS07b] ou d'une vidéo [VBTS07a]. Notre système permet un contrôle du style et nous proposons des traitements pour obtenir la cohérence du rendu [VBTS06c]. Enfin nous présentons l'analyse de nos travaux au chapitre 6. Nous terminons la première partie par une méthode alternative simulant les techniques manuelles pour obtenir une peinture animée (chapitre 7).

Dans la partie II de cette thèse, nous parlons du rendu en demi-tons qui est notamment utilisé pour les imprimantes. Les systèmes d'impression et les principes de fonctionnement du rendu en demi-tons sont présentés chapitre 8. Le chapitre 9 présente l'état de l'art des techniques de rendu en demi-tons. Nous proposons ensuite une méthode de distribution de points avec de bonnes propriétés visuelles pour le rendu en demi-tons. Pour cela nous utilisons un découpage du plan et une optimisation en précalcul (chapitre 10), ce qui nous permet d'obtenir une technique d'impression pouvant être utilisée dans les imprimantes disponibles aujourd'hui [VO08]. Nous présentons enfin les résultats chapitre 11.

Première partie

Rendu expressif de scènes animées

Chapitre 2

Rendu animé stylisé

LE RENDU EXPRESSIF a été défini initialement par opposition au rendu réaliste. Le but du rendu réaliste est de produire des images provoquant un stimulus visuel équivalent au monde réel comme ce que font les photographies. Au contraire, le rendu expressif¹ visait, historiquement, la simulation informatique des méthodes traditionnelles de création artistique, comme par exemple la peinture ou le dessin. Cette simulation peut se dérouler de deux manières :

- interactivement : l'artiste dessine, au moyen d'une souris ou d'un stylet, et le système interprète cette entrée pour produire l'image résultat ;
- automatiquement : l'artiste donne une entrée (typiquement une photo ou une scène 3D) et le système produit une image simulant le processus de création dans le médium² voulu, représentant cette entrée dans un style donné.

Dans la suite de ce manuscrit, *artiste* désigne la personne qui contrôle la création de la scène et du rendu ; des termes comme concepteur, graphiste ou directeur pourraient aussi convenir. Le *spectateur* ou *utilisateur* désigne la personne qui regarde ou interagit avec la création de l'artiste.

Au travers des travaux de recherche en rendu expressif, d'autres axes plus fondamentaux sont apparus. En effet, les simulations automatiques de méthodes traditionnelles posent la question du contrôle de l'artiste. L'artiste n'exécutant pas directement la création, il lui faut un moyen de donner son propre style au système de rendu. Les voies de recherche en rendu expressif sont guidées par la création d'images. L'un des objectifs fondamentaux d'une image est de communiquer un message, sa force réside donc dans son efficacité de communication. L'abstraction des détails et la suggestion sont des moyens utilisés par les artistes pour transmettre efficacement leur message. Les travaux en rendu expressif ont ainsi naturellement évolué vers ces questions : comment communiquer efficacement ? Comment définir une bonne abstraction ? Quels moyens de contrôle donner aux artistes pour qu'ils arrivent à leurs fins ? En effet, dans le cas où l'artiste dessine interactivement

1. Aussi appelé rendu non-photoréaliste, *non-photorealistic rendering*, NPR.

2. La matière utilisée pour les créations artistiques manuelles est constituée de pigments pour la couleur et de médium, ou liant. Les différents types de médium (eau, huile de lin, cire, etc.) ne réagissent pas de la même manière lors de l'application sur un support. Ils donnent une grande partie de l'apparence d'une œuvre (peinture à l'huile, pastel, aquarelle, etc.).

l'image, il gère directement l'abstraction et la suggestion par son exécution artistique. Dans le cadre d'un système automatique, il faut définir de nouveaux moyens de contrôle pour le faire.

Dans cette thèse nous nous intéressons au cas de l'animation. Outre les questions communes à tous les travaux en rendu expressif que nous venons de citer, l'animation introduit la question de la cohérence temporelle, c'est-à-dire comment le système de rendu garantit un résultat exempt d'artéfacts nuisant à l'interprétation de l'animation, et comment l'artiste peut contrôler l'évolution de l'animation. Nous définirons plus précisément notre vision de la cohérence temporelle en section 2.3.2.

Dans la suite du chapitre, nous présentons l'utilisation d'animation pour la communication et l'interaction. Nous décrivons les styles et les méthodes de création d'une manière générale, puis, plus particulièrement pour les styles qui nous intéressent, c'est-à-dire ceux offrant une matière complexe. Enfin, nous introduisons les problèmes liés à la création d'animation dans des styles où la matière est perçue et les motivations de l'utilisation de l'informatique pour les résoudre.

2.1 Utilisations de l'animation

Une animation est une suite d'images construites pour être vues les unes à la suite des autres et donner l'illusion d'un mouvement. En effet, nous percevons une série d'images fixes (défilant suffisamment rapidement) comme un mouvement continu. C'est le principe qui régit la fabrication de films, mais aussi des jeux vidéo et de la plupart des systèmes d'interaction utilisant un écran. L'effet Beta³ et l'aptitude du système visuel humain à *combler les vides* permettent de reconstruire le mouvement représenté par une série d'images fixes.

Nous distinguons deux types de représentation de mouvement sous forme de suite d'images : les films et les animations. Dans le cas du film, l'action est capturée en direct, généralement avec des moyens photographiques tels que les caméras. Une animation est faite image par image, et l'artiste concentre sa création sur le découpage du mouvement qu'il effectue. Cette frontière est moins nette avec l'utilisation des outils informatiques. En effet, on continue de parler de film lorsque des images de synthèse sont intégrées dans une séquence filmée, et d'animation lorsque la capture de mouvement de personnages réels est utilisée pour l'appliquer à des modèles de synthèse. Bien que cette définition ne soit pas absolue, nous entendons par animation une représentation du mouvement comprenant intrinsèquement une stylisation et abstraction de la scène. L'artiste oriente par ses choix le point de vue du spectateur. Il lui est aussi possible de proposer plusieurs points de vue, le spectateur prenant ainsi un rôle plus important dans l'œuvre en choisissant le point de vue qu'il veut visionner. Pour aller encore plus loin, l'artiste peut vouloir proposer au spectateur d'interagir avec l'animation. L'exemple le plus probant est le domaine du jeu vidéo. L'utilisateur interagit avec un monde virtuel : il peut généralement déplacer le point de vue, déplacer des objets et modifier l'état de ce

3. L'effet Beta est la sensation visuelle de mouvement provoquée par l'apparition d'images perçues successivement, susceptibles d'être raccordées par un déplacement ou une transformation [Wika].



FIGURE 2.1 – Images extraites de jeux vidéo montrant différents styles de rendu. (a) Rendu hyperréaliste tiré de *Tom Clancy's Ghost Recon*. (b) Rendu toon tiré de *Dragon Ball Z Budokai III*. (c) Rendu encre de chine tiré d'*Okami*.

monde. Il prend part à l'histoire créée par l'artiste et participe à l'action proposée au sein du jeu. L'animation n'est plus le choix des images mais la manière dont l'artiste prévoit l'interaction utilisateur.

Historiquement — pour des raisons techniques mais aussi de jouabilité — les images des jeux vidéos étaient dessinées à la main, offrant un aspect épuré et abstrait. Aujourd'hui ce type de rendu est encore présent, *a contrario* des jeux dont l'aspect visuel tend vers le réalisme pour favoriser l'immersion de l'utilisateur dans le monde virtuel. Lorsque le réalisme visuel n'est pas l'objectif, plusieurs styles sont mis en œuvre dans les jeux. Le plus utilisé est le rendu *toon* reproduisant l'apparence d'un dessin animé. Ce type de rendu est choisi par exemple quand le jeu s'inspire d'un dessin animé ou d'une bande dessinée, mais aussi pour des jeux voulant créer une ambiance imaginaire forte. D'ailleurs certains jeux, pour marquer leur originalité, utilisent des rendus plus stylisés et plus abstraits. La figure 2.1 montre des extraits de jeux vidéos utilisant différents types de rendu.

Le rendu interactif peut aussi servir dans le cadre d'installations artistiques. Il existe des exemples d'œuvres d'art contemporaines utilisant les outils informatiques et électroniques. Selon le vouloir de l'artiste, l'utilisateur devient partie de l'œuvre. Du monde interactif vers d'autres installations où par exemple le spectateur est incorporé dans une œuvre en étant stylisé de manière analogue à celle-ci, les exemples d'utilisation par les artistes sont nombreux. La figure 2.2 présente certaines de ces installations.

Nous pensons que la force de communication, la liberté d'expression et l'ouverture vers l'imaginaire sont les éléments qui donnent aux animations leur caractère intéressant et attirant. En effet l'imagination et l'interprétation du spectateur sont mises à contribution au travers du travail réalisé par l'artiste. L'intention de l'artiste est portée par le style et par l'interaction laissée à l'utilisateur. Les outils informatiques donnent aux artistes de nouveaux moyens de création et de communication qu'ils ont déjà su intégrer dans le processus de création. Néanmoins, les questions du contrôle du style et de la cohérence temporelle restent ouvertes et sont les clés à apporter pour offrir aux artistes une liberté de création encore plus étendue.



(a) Aura, de Shawn Lawson, 2002



(b) Faces, d'Adrian Goya, 2006

FIGURE 2.2 – Installations artistiques utilisant l'interaction et le rendu expressif.

2.2 Une histoire de style

« Manière personnelle de pratiquer un art, définie par un ensemble de caractères »
 « Manière particulière à un genre, à une époque, notamment en matière d'art, définie par un ensemble de caractères formels. »[Lar03]

« Manière particulière (personnelle ou collective) de traiter la matière et les formes en vue de la réalisation d'une œuvre d'art ; ensemble des caractères d'une œuvre qui permettent de la classer dans un ensemble constituant un type esthétique. »[Rob02]

Les sens donnés au mot style sont très variés. Dans le cadre artistique, le style est l'ensemble des caractéristiques propres à un artiste pour travailler son médium. On emploie le terme style pour classer les œuvres artistiques selon le type de médium utilisé, suivant l'utilisation de ce médium et suivant les courants artistiques. C'est par ses créations que l'on définit le style d'un artiste. Dans le cadre de nos travaux, nous utiliserons une définition plus formelle que nous introduirons au chapitre 5. Nos travaux portant sur le rendu animé, nous allons faire l'analyse des styles habituellement utilisés par les artistes dans les animations.



FIGURE 2.3 – Exemples de décors et de colorisations utilisés dans le dessin animé.
Gauche : *Tonari no Totoro* d'Hayao Miyazaki, droite : *Droopy* de Tex Avery.

2.2.1 Animation traditionnelle sur celluloïd

Considérons tout d'abord la technique du dessin animé. Sans entrer dans le détail des différents styles de dessins animés, il y a généralement une approche commune à l'ensemble des créations. L'usage classique est de séparer le décor, fixe, et les éléments animés, comme par exemple les personnages. Les éléments animés sont dans un style simple et abstrait, avec quelques lignes pour les contours et les traits distinctifs. La colorisation est constituée d'aplats de couleurs, pouvant suggérer les ombrages (voir figure 2.3). Les décors sont dans un style plus riche, peinture gouache ou acrylique, utilisant des éléments plus ou moins abstraits.

Deux raisons sont à l'origine de cette différence entre parties fixes et parties animées. Elle permet une lecture très rapide de la scène en faisant une distinction entre les éléments importants (les personnages) et le contexte (le décor). Cette différence émane également du processus de production des dessins animés. Les décors sont utilisés, et réutilisés, pour des scènes de plusieurs images, tandis que les dessins de l'animation sont utilisés pour quelques images seulement. Pour les éléments animés, l'artiste dessine chaque image constituant l'animation, en commençant par des poses clés qui décomposent le mouvement. Le temps de dessin étant critique pour une production (il y a entre cinq et quinze dessins par seconde d'animation), il faut pouvoir les produire suffisamment rapidement. Les artistes utilisent plusieurs calques de celluloïd (transparents) pour dessiner un personnage, ce qui permet de garder certains éléments du personnage statiques, alors que d'autres sont en mouvement. Ainsi, ils peuvent animer seulement la bouche d'un personnage lorsqu'il parle, laissant le reste inchangé. Ces dessins sont ensuite superposés aux décors.

La plupart des studios d'animation ont transposé une partie du processus sur ordinateur. Il existe d'ailleurs de nombreux logiciels pour produire une animation : de la conception au montage, en passant par le dessin et la colorisation. On retiendra par exemple Toon Boom Studio ©, Plastic Animation Paper ©Krogh Mortensen Animation ou encore FlipBook ©DigiCel Inc.



(a) *Impression de montagne et d'eau* de Te Wei, aquarelle animée.



(b) *Le moine et le poisson* de Dudok de Wit, décor aquarelle, animation classique.



(c) *Un trou à la place du coeur* de Mathieu Lozinguier, animation de sable.



FIGURE 2.4 – Extraits d'animations expérimentales en aquarelle et en sable.

2.2.2 Détournement des techniques traditionnelles

Plusieurs artistes ont produit des animations dans un style plus riche que le dessin animé classique. Considérées comme expérimentales, ces animations offrent divers styles comme la peinture animée⁴, l'aquarelle animée et l'animation sur sable, comme le montre la figure 2.4.

La richesse des styles provient des effets de matière dus à l'interaction entre le médium et le support. On peut voir les traces des fibres du pinceau et le grain de la toile sur la forme de la peinture déposée par l'artiste. La forme des marques faites par l'artiste donne aussi de la matière et fait partie du style. La figure 2.5 met en évidence certains des effets de matière caractéristiques des styles qui nous intéressent.

L'esthétique et l'expressivité de ces animations ont amené les studios de production à les tester à des fins commerciales (spot publicitaire, long métrage ...) mais il reste difficile de produire ces animations suffisamment rapidement, et il subsiste des artéfacts de rendu. En effet, lorsque l'on veut produire, à la main, une animation dans un style riche, il faut dessiner chacune des images de l'animation. Cela peut être très long et fastidieux. Nous présentons une petite filmographie de courts métrages en peinture animée page 147. Le temps de création d'une peinture, par exemple, est bien plus important que celui d'un dessin simple. À titre

4. Nous détaillerons la technique de peinture animée section 7.1.



(a) « Le sujet de la peinture », Schwitzgebel

(b) « Le Vieil homme et la mer », A. Petrov

FIGURE 2.5 – Effets de matière présents dans les techniques traditionnelles. On note notamment les marques des fibres des coups de pinceau, le mélange et étalement des couleurs et la texture produite par le médium.

de comparaison, A. Petrov a mis plus de deux ans et demi à réaliser les 20 minutes du *Vieil homme et la mer* alors que Akom réalise un épisode des *Simpsons's* en deux mois [Wikib]. Cette différence de temps est à relativiser en fonction du nombre de personnes impliquées dans la création, mais dans le cas d'une peinture animée, il est difficile de séparer le travail (les peintures devant être produites l'une après l'autre). Outre le temps de production, un des effets inhérent à ce genre d'animation est la vibration ou bouillonnement dû au travail manuel de l'artiste. Ce bouillonnement, immédiatement perçu lors de l'observation d'un court métrage en peinture animée par exemple, provient du fait que les différents coups de pinceau sont brusquement ajoutés d'une image à l'autre. Tout en offrant un charme et un côté très vivant à l'animation, ce bouillonnement est incontrôlable (l'artiste peut difficilement le réduire) et il fatigue l'observateur lors d'une observation prolongée, notamment dans le cas de longs métrages ou de jeux vidéo.

2.2.3 Remplissage de régions pour l'animation

Si on compare un dessin animé et une peinture animée, on voit que la richesse graphique de la peinture est apportée par les régions que l'artiste dessine pour remplir les formes. Ces régions sont de simples aplats dans le cas du dessin animé, alors qu'en peinture animée, l'artiste place de nombreux coups de pinceau qui apportent de la matière, suggérant la forme et les ombrages.

Nous voulons offrir des outils permettant la création efficace d'animations dans un style s'inspirant des techniques traditionnelles, en donnant le contrôle de l'évolution de l'animation à l'artiste. Nous devons donc permettre à l'artiste d'animer des régions dans un style plus riche que les aplats utilisés en dessin animé, mais avec la même aisance de manipulation.

2.3 Les problèmes que nous voulons résoudre

L'objectif de notre travail est d'affranchir l'artiste des contraintes inhérentes au rendu pour lui permettre de se concentrer sur l'essentiel : la création et le message qu'il veut transmettre. Nous devons donc déterminer les moyens d'actions à donner à l'artiste et trouver comment le système doit animer les régions, et donc la matière, pour supprimer les effets incontrôlables.

2.3.1 Place de l'artiste

Dans le cadre d'une animation, la création de l'ensemble des images prend beaucoup de temps à l'artiste. De plus, nous pensons que c'est une phase qui suit un certain systématisme et que nous pouvons donc l'automatiser.

Dans le cadre de l'utilisation d'outils informatiques, l'artiste utilise un logiciel, ou système, qui produit les images. Nous allons maintenant décrire le système de rendu d'animation stylisé *idéal*, sans entrer dans le détail, mais en soulignant les points importants que nous avons considérés dans nos travaux.

L'artiste donne en entrée la scène à rendre et une description du style ; le système produit en sortie un rendu stylisé de l'animation. Le système gère les phases qui prennent le plus de temps et qui demandent le plus d'efforts à l'artiste. Par contre, il est utile pour l'artiste de pouvoir modifier, localement, le rendu obtenu, soit pour corriger ce que le système a produit, soit pour ajouter un détail qui n'est pas du ressort de l'automatisation. En concevant les algorithmes, nous devons garder cet esprit d'ouverture et laisser l'artiste intervenir *a posteriori* sur le rendu. Toutefois, dans le cas d'interaction avec l'utilisateur, l'artiste ne peut pas faire de telles modifications car il n'a pas le contrôle sur les choix du spectateur.

Le contrôle du style est un point fondamental que nous devons définir. Ce contrôle permet à l'artiste de donner sa touche personnelle. Comme l'a souligné Hertzmann [Her98], pour que le contrôle du style soit facilement utilisable, il doit être :

- *intuitif* : offrir un contrôle dont l'impact sur le rendu final est prévisible ;
- *consistant* : produire la même apparence pour deux entrées semblables, par exemple pour deux images successives d'une animation ;
- *robuste* : produire des résultats exempts de cas dégénérés ;
- *indépendant* : offrir des contrôles identifiables, qui influencent chacun une partie définie du rendu, par exemple un moyen de contrôle de la couleur et un moyen de contrôle du grain de papier.

De plus, nous pensons que le style doit être :

- *flexible* : offrir de larges possibilités de styles ;
- *transposable* : être applicable à différentes scènes sans perdre son essence.

Les artistes travaillent souvent par essais et corrections : ils donnent un paramètre, regardent son impact, le modifient, et ce jusqu'à trouver ce qui leur convient. Pour offrir un tel comportement, le système doit être suffisamment performant pour permettre cette interaction⁵, même si le résultat temporairement visualisé n'est qu'une version approximative du rendu final.

5. Ici performant signifie rapide : une image doit être produite rapidement pour permettre l'interaction, c'est-à-dire moins d'une seconde par image.

2.3.2 Cohérence temporelle

Les éléments visuels caractérisant les styles présents dans les créations manuelles peuvent être simulés informatiquement. Lors de l'animation traditionnelle, il n'y a pas de lien intrinsèque entre les différentes images de l'animation : en particulier, ces éléments visuels provoquent des artéfacts de rendu. Par contre, pour une animation faite par ordinateur, ces effets peuvent être simulés et liés d'une image à l'autre. Nous pouvons ainsi espérer supprimer les artéfacts des méthodes manuelles.

Nous n'avons pas encore défini clairement ce que nous entendons par éléments visuels caractérisant le style. Il s'agit de la matière, ou encore de la texture, formée par l'interaction du médium, de l'outil et du support. Un exemple simple est le grain de papier qui apparaît dans un dessin. Le grain agit à plusieurs niveaux : sur une feuille blanche, le grain est perçu suivant l'interaction entre la surface du papier et la lumière. En effet, suivant le grain, la surface du papier n'est pas parfaitement plane mais présente des petits reliefs. Aussi, le grain est visible lors de l'application du médium ; le dépôt du médium se faisant, suivant la pression de l'outil, d'abord sur les parties saillantes avant de venir remplir les vallées dans le relief du papier. D'autres éléments visuels appartiennent à notre définition, comme par exemple le relief donné par la forme des brins constituant un coup de pinceau. Un autre exemple est la complexité des mélanges des diverses couches de médium qui sont appliquées sur le support. Nous appelons éléments visuels de matière, ou de texture, les parties distinctes, identifiables, dans ces effets. Bien sûr, ils n'ont de signification que lorsque leur composition est regardée, mais chacun d'eux est plus ou moins identifiable pour une œuvre donnée.

Si cette matière est simplement fixée sur les images constituant l'animation, on obtient ce qu'on appelle l'effet *rideau de douche* : c'est comme si on regardait l'animation à travers une vitre texturée. Les éléments fixes perturbent alors notre perception de l'animation. Une meilleure solution est d'animer cette matière : il faut alors déterminer quel mouvement appliquer aux éléments visuels et comment conserver leurs caractéristiques. Le mouvement qu'ils suivent est, par exemple, le mouvement présent dans la scène (mouvement des objets constituant la scène), mais d'autres mouvements sont possibles, si bien que nous considérons le mouvement comme une entrée du système de rendu. Si ces éléments suivent un mouvement quelconque, avec des changements d'échelle (zoom) et des déformations, la taille et la densité de ces éléments changent et leur aspect originel est perdu. En effet, les éléments de matière créant la stylisation se trouvent en espace image, alors que le mouvement provient de l'espace objet⁶.

Leur taille et leur densité dépendent du type de support, du médium, de la taille de l'outil et de la distance du spectateur au support et non des objets que l'on représente. Pour maintenir cet aspect, on doit, en plus du déplacement, mettre à jour la matière présente (ajout ou suppression des éléments visuels), et ce sans introduire de discontinuités temporelles⁷. Nous définissons ainsi la *cohérence temporelle* comme le compromis entre trois contraintes :

6. Dans le processus de rendu d'une scène 3D, on différencie l'espace image, qui est l'espace 2D de l'écran, de l'espace objet, qui est l'espace 3D de la scène.

7. Changement brutal dans l'animation, faisant apparaître un élément directement d'une image à l'autre.

- le mouvement à suivre ;
- la juste représentation de la matière, c'est-à-dire la ressemblance du rendu de synthèse avec le médium traditionnel ;
- la continuité temporelle.

Les artéfacts, potentiellement présents dans un rendu expressif animé, correspondent à la violation de ces contraintes. Si les éléments de matières ne suivent pas fidèlement le mouvement donné en entrée, cela crée une impression de glissement de matière sur l'animation et de conflit entre le mouvement de la matière et le mouvement voulu (perception de mouvements fantômes, non présents dans l'entrée). La mise à jour des éléments de matière pour obtenir la juste représentation provoque, sans traitements particuliers, des changements brutaux. Or, la présence de changements brutaux nuit à la continuité temporelle. Il faut donc adoucir les changements, sans briser les autres contraintes.

2.4 Contributions

Nos travaux sur le rendu expressif apportent les contributions suivantes.

Tout d'abord, nous proposons une étude approfondie du rendu à base de marques dans le cadre de la production d'animation (chapitre 3.2). Nous mettons notamment en avant les différents choix possibles pour gérer le compromis de cohérence temporelle.

Puis, nous développons une technique de distribution de primitives en espace objet pour le rendu à base de marques (section 4.2). Cette technique est suffisamment rapide pour permettre une liberté d'interaction totale avec la scène. Elle est donc particulièrement adaptée pour le rendu peinture dans le cadre d'applications interactives comme les jeux vidéo.

Nous proposons aussi une méthode hybride de distribution de primitives en section 4.3. Cette technique effectue la distribution en espace image, avec des garanties sur la qualité de la distribution, et suit le mouvement de la scène. On offre ainsi la possibilité de faire un compromis entre la fidélité au mouvement, la continuité temporelle et la fidélité à la densité de primitives voulue.

Nous présentons l'ensemble du pipeline que nous proposons chapitre 5. En particulier, nous spécifions notre vision du style et le contrôle offert à l'artiste. De plus, nous explicitons les méthodes que nous avons développées pour la continuité temporelle et la juste représentation de la matière.

Nous présentons aussi, chapitre 7, une méthode alternative au rendu à base de marques. Cette approche de simulation automatique des techniques traditionnelles offre la qualité visuelle de la peinture animée. Cette méthode offre un contrôle utilisateur de haut niveau concernant le style. Une forme de cohérence temporelle est proposée, en utilisant les mêmes moyens que dans les méthodes traditionnelles, c'est-à-dire mélange et déplacement du médium déjà en place sur le support.

Nous allons maintenant faire un rapide survol des diverses méthodes s'intéressant à la cohérence temporelle d'animations stylisées. Puis dans la section suivante nous rentrerons dans le détail des techniques se rapportant à nos travaux.

3.1 État de l'art sur la cohérence temporelle

Dans un premier temps, nous discuterons des approches basées texture de matière, c'est-à-dire que la texture porte plusieurs éléments visuels de matière. Ces textures sont soit composées en espace image (section 3.1.1) soit plaquées sur les objets 3D (section 3.1.2). D'autres techniques consistent à utiliser une représentation plus fine des éléments visuels attachés à des points (section 3.1.3). Nos travaux entrent dans ce cadre et plus particulièrement dans le rendu à base de marques (section 3.1.4 et section suivant).

3.1.1 Textures de matière en espace image

Pour gérer le compromis de la cohérence temporelle, les éléments visuels voulus peuvent être représentés par une texture en espace image. Dans ce type d'approche, la texture est une image 2D que l'on applique par dessus le rendu (par composition) pour ajouter les éléments visuels de stylisation. Cette texture porte une partie de la stylisation et elle sera animée en fonction de la scène en entrée puis mélangée avec d'autres calques (de couleur par exemple) pour obtenir le rendu final. Diverses méthodes existent pour permettre d'animer la texture, en répondant aux critères de cohérence temporelle. Elles sont souvent spécifiques pour un type de mouvement ou d'éléments de matière.

Pour que le mouvement de la texture suive celui du point de vue (de la caméra), Cunzi et coll. [CTP⁺03] montrent que l'on peut décomposer le mouvement 3D d'une caméra en utilisant des transformations purement 2D. Ils appellent leur méthode *dynamic canvas* car il s'agit d'une métaphore de représentation du support de manière dynamique en fonction du point de vue. Les travellings avant ou arrière

sont approchés par un zoom 2D dans la texture et les rotations sont décomposées en translations et rotations de la texture. Pour cela, Cunzi et coll. représentent la caméra comme une sphère roulant sur un plan, les travellings changeant le rayon de la sphère, ce qui change l'échelle de la texture, et produit donc un zoom. Pour obtenir une liberté totale de zoom et dé-zoom, sans nuire à la bonne représentation des effets visuels (en conservant constante la taille des éléments visuels), ils utilisent plusieurs copies de la texture de matière, à différentes échelles, qu'ils mélangent pour obtenir celle utilisée dans le rendu final. Cela permet d'obtenir un zoom infini, en introduisant et supprimant de manière douce les différentes échelles lors des zooms. Néanmoins, le fait de mélanger plusieurs textures introduit un flou et peut briser certaines structures présentes dans la texture originelle, notamment les hautes fréquences. Cette technique est adaptée pour suivre le mouvement de la caméra, mais ne permet pas de représenter un mouvement complexe tel que celui des objets ou de leurs déformations.

Dans le cadre de la stylisation d'une vidéo, Bousseau et coll. [BNTS07] proposent d'utiliser une texture advectée par le flux optique¹ [BB95]. L'advection d'une texture provoque des compressions et dilatations ; les effets de matière sont donc déformés. Ils remarquent que si l'on advecte la texture en jouant la vidéo à l'envers, les déformations ne se trouvent pas aux mêmes endroits². Ils utilisent ainsi deux textures, advectées respectivement avec le flux optique originel et le flux optique de la vidéo jouée à l'envers. Ils calculent alors la déformation en chaque pixel de l'image, dans chacune des deux textures, et choisissent la version la moins déformée.

Ces techniques sont adaptées pour des textures représentant des éléments petits, peu structurés, tels que du bruit multi échelle (type bruit de Perlin [Per85]), ou des textures qui représentent le grain de papier et les pigments présents dans le médium. Par contre, des problèmes se posent pour des éléments plus gros tels que les coups de pinceau, car la structure est cassée par l'advection et le mélange des textures.

3.1.2 Plaquage de textures en espace objet

Dans le cas d'une scène 3D, des textures sont habituellement plaquées en 3D sur les objets pour décrire leur matériau. On peut utiliser des textures présentant les caractéristiques visuelles des média traditionnels. On obtient alors une certaine stylisation. Les caractéristiques des effets visuels sont liées à la position des objets et la déformation des objets (qui changent suivant l'animation). En définitive, cela revient à prendre une photo d'une scène où les objets sont peints, et non à peindre une scène sur un support.

Pour pallier les problèmes de changements d'échelle, Klein et coll. [KLK⁺00] utilisent une adaptation des *mip-maps* [Wil83], qu'ils appellent *art-maps*, construites pour respecter la taille des éléments visuels en espace image. Les *mip-maps* sont

1. Le flux optique est une approximation du mouvement des objets, calculé à partir de l'évolution des couleurs dans la vidéo.

2. En fait, les déformations les plus importantes se produisent aux désocclusions, et lorsque la vidéo est jouée à l'envers, les désocclusions deviennent des occlusions et inversement.

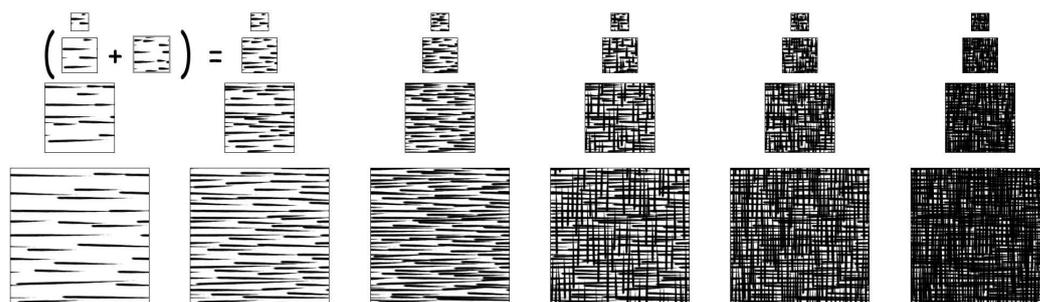
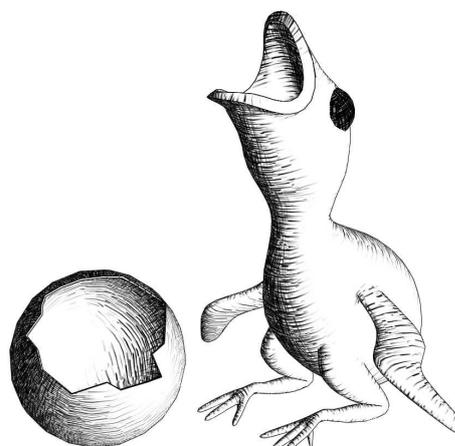


FIGURE 3.1 – Les *tonal art-maps* sont une pyramide de textures de différentes tailles et de différentes intensités. Chaque niveau est construit à partir des niveaux précédents en taille et en intensité.



(a) *Art-maps* [KLK⁺00].



(b) *Tonal art-maps* [PHWF01, WPFH02].

FIGURE 3.2 – Images rendues en utilisant des textures plaquées sur les objets 3D. Un certain effet de stylisation est obtenu, mais la forme des objets reste directement définie par la forme 3D sous-jacente.

utilisées pour gérer les problèmes de crénelage. Il s'agit un ensemble de textures de différentes tailles, représentant la même image de base à différentes résolutions. Au moment du rendu, on choisit celle (ou un mélange de celles) dont la résolution se rapproche le plus de la résolution de la région à dessiner. Les *art-maps* sont construites sur le même principe. Elles permettent de conserver une taille quasiment constante des éléments visuels dans l'image obtenue, comme on peut le voir figure 3.2.

Praun et coll. [PHWF01, WPFH02] utilisent une pile d'*art-maps* multi tons. En plus de choisir la bonne résolution, il peuvent choisir la texture en fonction du ton (niveau de gris de la scène) à représenter. Les différentes textures sont construites pour avoir le moins de différences possibles quand on change de ton ou d'échelle, voir figures 3.1 et 3.2.

Dans les techniques basées sur des textures plaquées sur la surface des objets, le mouvement des objets est fidèlement suivi, par contre il n'est pas possible de donner un autre mouvement à suivre³. L'utilisation des *art-maps* assure une continuité temporelle lors des changements d'échelle, mais les effets de matière présentent

3. En effet, le mouvements objets est une possibilité parmi d'autre. Par exemple, les éléments visuels pourraient suivre une direction de vent dans une scène représentant une tempête en forêt.

certains problèmes. Les effets de stylisation ne peuvent altérer la forme des objets, les silhouettes sont donc parfaitement nettes et sont collées à la surface des objets. Enfin, des artéfacts de projection sont incontournables, notamment sur les faces ayant un angle rasant par rapport au point de vue, la texture étant déformée par la perspective.

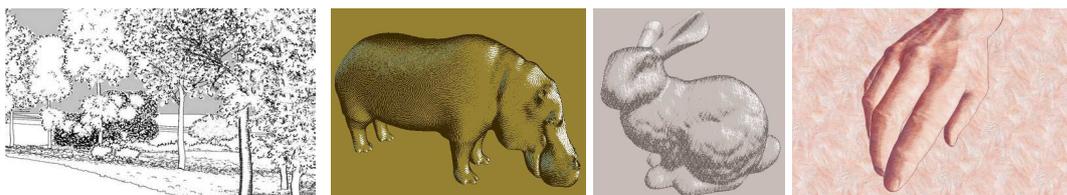
3.1.3 Éléments visuels et points d'attache

Pour obtenir les effets de stylisation en espace image et rompre avec l'aspect trop parfait des scènes 3D, on peut utiliser des méthodes hybrides 3D/2D. Ces méthodes utilisent généralement des points d'attache sur les objets 3D qui vont permettre de calculer le mouvement 2D d'éléments de matière.

Kaplan et Cohen [KC05] ainsi que Bousseau et coll. [BKTS06] utilisent des textures dessinées à partir de points d'attache placés sur les objets 3D de la scène pour le représenter le grain de papier. Les points d'attache suivent le mouvement et les déformations des objets, et le dessin des textures se fait en espace image. Ceci supprime les artéfacts dus à la projection 3D et au plaquage de textures. Chaque texture suit le mouvement d'un point sur le modèle 3D, généralement lié au centre de la texture. Le reste de la texture glisse sur le modèle, ce qui induit quelques artéfacts de mouvement. Kaplan et Cohen [KC05] dessinent à partir de chaque point d'attache une petite texture représentant une fibre du support. Ils utilisent donc beaucoup de points d'attache et proposent une mise à jour dynamique du nombre de points d'attache en fonction du point de vue. La structure des fibres est donnée par la distribution des points d'attache, qui n'est pas facile à contrôler. Bousseau et coll. [BKTS06] utilisent des textures dynamiques, dans l'esprit de *dynamic canvas*, chaque texture étant attachée à un point d'attache sur le modèle 3D. Quelques points d'attache sont utilisés pour chaque modèle (six dans leur implémentation). Chaque texture couvre une zone plus grande que dans l'approche de Kaplan et Cohen, ce qui permet d'avoir une meilleure structure dans le grain représenté, par contre on peut percevoir certains glissements entre les textures et le mouvement des objets, notamment à la frontière entre deux textures. Celles-ci sont adoucies en faisant varier la transparence aux bords des textures, mais cela adoucit les effets visuels (impression de flou) et l'on perçoit distinctement le mouvement de chacune des textures.

Dans ces deux techniques, une texture suit un point d'attache. Pour faire un meilleur lien entre le mouvement des points d'attache et le mouvement des textures, Breslav et coll. [BSM⁺07] approximent le déplacement d'un ensemble de points d'attache par une transformation affine 2D (translation, rotation et échelle). Ils appliquent ensuite cette transformation à des textures attachées aux objets. Coconu et coll. [CDH06] proposent une méthode similaire pour le rendu dans un style hachurage : ils utilisent une paramétrisation de la texture en espace image, dépendante du point de vue et de la géométrie des objets, en englobant les objets dans des sphères. Ensuite, ils utilisent les sphères pour produire des paramétrisations en espace image qu'ils reprojettent sur les objets afin de plaquer des textures de hachures.

Les techniques basées sur des points d'attache placés sur les objets permettent de faire un compromis entre le mouvement à suivre et la bonne représentation des

(a) Rendu *encre* [CDH06]

(b) Rendu demi-ton, hachures et peinture [BSM+07]

FIGURE 3.3 – Différents styles obtenus avec les techniques hybrides entre espace objet et espace image.

éléments visuels de stylisation. En effet, le mouvement des objets est suivi par les points d’attache, puis le mouvement des points d’attache permet de déplacer (plus ou moins directement) des morceaux de texture qui représentent les éléments visuels de stylisation dans l’espace image. Ces techniques sont particulièrement adaptées pour l’utilisation de grandes textures portant par exemple l’information liée au grain du papier mais peuvent aussi s’appliquer à de plus petites textures avec des éléments de grain unitaire. Les grande textures introduisent des glissements par rapport au mouvement de la scène sous-jacente. Lorsque les éléments représentés correspondent à des traces laissées par l’outil de dessin, il est plus avantageux d’utiliser une texture par trace (du point de vue des glissement). Chaque texture est alors une marque.

3.1.4 Rendu à base de marques

Dans notre cadre de travail, nous nous intéressons au rendu faisant apparaître des traces d’outils comme celles laissées par un pinceau que l’on voit sur une peinture à l’huile. Le rendu à base de marques utilise des primitives graphiques (marques) pour effectuer le rendu stylisé d’une animation. Pour faire le lien avec la section précédente, il s’agit de représenter chaque marque par une texture, liée par exemple à des points d’attache sur les objets d’une scène 3D. D’une image à l’autre, les primitives sont déplacées suivant le mouvement à suivre, comme par exemple le mouvement des objets. Nous faisons une étude plus approfondie du rendu à base de marques à la section suivante.

3.2 Étude du rendu à base de marques pour l’animation

Nous présentons dans cette section un pipeline standard permettant l’étude le rendu à base de marque pour la stylisation d’animations. Ensuite nous comparons les différents travaux précédents en se basant sur ce pipeline.

Le rendu à base de marques s’appuie sur l’observation du processus de création d’une peinture. Quand un artiste peint, il applique de la peinture (plus généralement un médium) avec un pinceau (un outil) sur une toile (un support). Une peinture est donc par construction un ensemble de coups de pinceau placés sur un support. Le rendu à base de marques est un modèle informatique qui représente une image par un support et une liste ordonnée de marques. Ces marques représentent les coups de pinceau que doit appliquer le système pour obtenir la peinture.

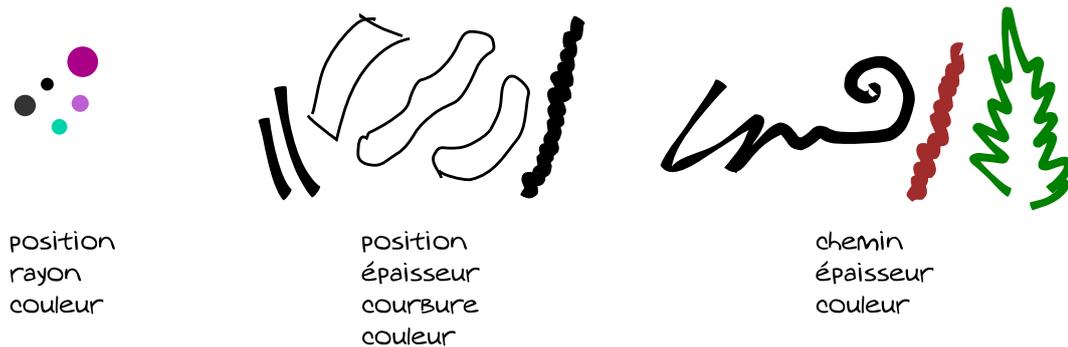


FIGURE 3.4 – Différents modèles de marques et les attributs associés.

Le principal but du rendu à base de marques est de fournir des outils informatiques qui automatisent une partie de la création d'images, sans remplacer l'artiste, mais en lui donnant de bons moyens de contrôle.

Dans les méthodes manuelles, en fonction du type de peinture, les coups de pinceau sont plus ou moins visibles, et leur impact sur le support est plus ou moins direct. Dans la peinture à l'aquarelle par exemple, les coups de pinceau donnent la forme des objets de la scène, mais lors de leur superposition, sans séchage intermédiaire, les pigments se mélangent et on ne distingue pas individuellement les touches de peinture. Au contraire de l'acrylique où, suivant le style voulu par l'artiste, la contribution de chacun des coups de pinceau est bien plus visible. Le rendu à base de marques peut servir dans tous les styles de peinture, mais il est particulièrement adapté quand on veut modéliser un style où les coups de pinceau sont individuellement perçus. En effet, il permet de mettre en valeur la contribution de chacune des marques car chaque marque est connue du système.

3.2.1 Modèle informatique

Une marque est représentée informatiquement par une structure de données. Cette structure décrit l'ensemble des informations nécessaires au dessin d'une marque sur le support. Dans les travaux précédents, on trouve deux types de structure de données pour les marques ; celle dont la position de la marque est définie par un point et celle dont la position est définie par un chemin c'est-à-dire une courbe en 2D qui donne la forme principale suivie par la marque. De plus, un ensemble d'attributs permet de générer la forme complète et l'apparence de la marque. La figure 3.4 présente quelques marques et les attributs correspondants.

Suivant la flexibilité du système, les marques peuvent avoir une largeur (qui est simplement le rayon dans le cas de style pointillé ou *stippling*), une longueur (si la marque est définie en un point, sinon la longueur est le chemin qui décrit la marque) et une couleur qui donne la teinte principale de la marque. De plus, pour obtenir la contribution de la marque sur la région qu'elle couvre sur le support, on peut appliquer des modulations de la couleur et de la transparence au moyen d'une texture par exemple.

3.2.2 L'animation à base de marques

Le rendu à base de marques a initialement été développé pour la synthèse d'images fixes. Hertzmann [Her03] fait un tour d'horizon de ces techniques et des enjeux, notamment du contrôle du style, et présente le rendu à base de marques comme un problème d'optimisation. En effet, le système de rendu doit faire un compromis entre la juste représentation de l'image de référence en entrée et le style voulu par l'artiste, ce qui peut se modéliser par une fonction de coût à minimiser.

Si on applique directement les techniques adaptées à une image fixe sur une animation, en considérant les images de l'animation indépendamment les unes des autres, on obtient un résultat présentant les mêmes effets de bouillonnement incontrôlable que les méthodes manuelles. Nous nous concentrons dans la suite sur les méthodes visant à résoudre le problème de la cohérence temporelle pour le rendu à base de marques.

3.2.3 Problèmes à considérer

Avant de faire l'étude des travaux précédents dans la prochaine section, nous allons décrire notre plan d'étude pour comparer les techniques existantes. Ceci permet d'une part de localiser où se trouvent les problèmes, et d'autre part de mettre en avant quelles options sont indépendantes, ou au contraire intimement liées. Ces options sont guidées par la variété des styles visés, la cohérence temporelle voulue, et les limitations techniques. L'objectif commun de tous ces travaux est d'obtenir un rendu stylisé, expressif, contrôlable, automatisé, et sans artéfacts de rendu.

Nous proposons un pipeline général (voir figure 3.5) pour le rendu à base de marques qui permet de comparer les méthodes précédentes. Les données sont représentées par des rectangle bleus et les actions par ovoïdes orangés. Ce pipeline permet d'explicitier les choix faits dans les travaux existants, même s'ils sont implicites dans certaines approches. Notre pipeline suit la même approche que celle présentée par Willats et Durand [WD05]. Ceux-ci différencient l'espace objet, celui de la scène à représenter, de l'espace image, celui où se trouve la représentation. Ils montrent qu'un système de rendu (automatique ou manuel) est défini par la manière dont sont liés ces deux espaces, et plus particulièrement comment les propriétés des éléments de la scène sont rattachées aux attributs des marques (il s'agit d'éléments quelconques en espace image dans leur définition).

Les données et actions de notre pipeline sont les suivants :

Entrée Le rendu animé à base de marques prend une scène animée en entrée, et produit en sortie une suite d'images formant l'animation. La scène en entrée peut être une vidéo ou une scène 3D, accompagnée d'informations complémentaires : flux optique, segmentation pour les vidéos, *G-buffers* [ST90] pour les scènes 3D (normales, profondeur, identifiants d'objets etc.). L'animation est soit prédéterminée, soit interactive. Les animations prédéterminées peuvent être une vidéo enregistrée ou une scène 3D où le mouvement de la caméra et l'animation des objets sont déterminés à l'avance. Pour le rendu interactif, il peut s'agir d'une caméra filmant en direct, et la stylisation s'effectuant à la volée, ou d'une scène 3D avec laquelle un utilisateur interagit directement, comme dans le cas des jeux vidéo.

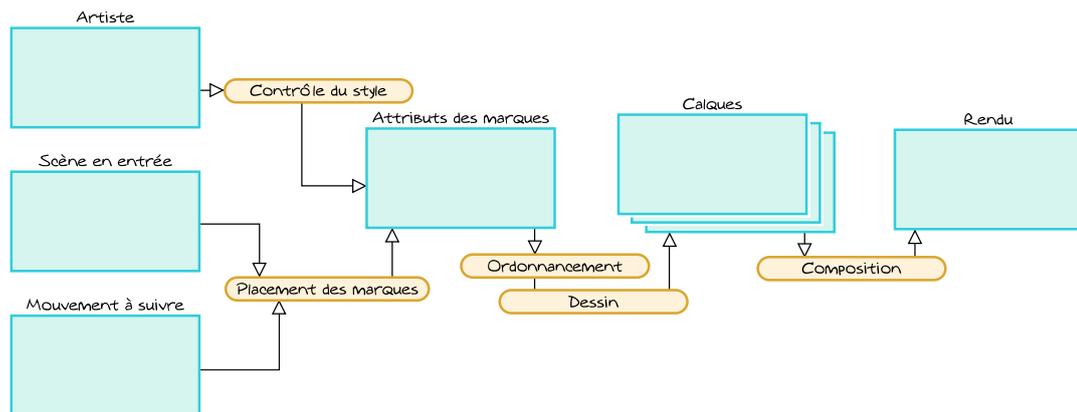


FIGURE 3.5 – Pipeline général de rendu

Contrôle du style L'artiste est l'élément central dans la conception d'un système en rendu expressif. Il faut choisir le niveau de contrôle et d'automatisation que le système lui fournit. Ceci dépend du rendu et du style recherché mais aussi du niveau d'expérience de l'artiste considéré et du temps qu'il veut passer pour la création de l'animation.

Mouvement à suivre On choisit généralement de faire suivre aux marques le mouvement des objets constituant la scène. C'est en effet le choix le plus adapté car il permet de décrire au mieux le mouvement présent dans l'animation. Néanmoins, d'autres mouvements sont possibles, le mouvement des marques est donc une donnée à part entière. Par exemple le mouvement des marques peut suggérer le mouvement du vent dans le cas d'une scène naturelle, ou comme dans l'exemple donné par Hays et Essa [HE04], suivre le flux d'air chaud produit par un feu. Les méthodes existantes sont généralement adaptées à un type de mouvement précis (celui de la caméra, celui des objets) et peuvent plus ou moins être étendues à d'autres types de mouvement.

Placements des marques Pour dessiner les marques sur le support, il faut déterminer où elles se trouvent. Ce placement dépend du style recherché. Seules les marques visibles dans l'image finale comptent, et donc, si les marques sont opaques, c'est la dernière couche de marques qui détermine l'apparence du rendu. Par contre, si les marques sont semi transparentes, ou si l'on veut laisser volontairement de l'espace libre entre les marques, la distribution des marques sur le support joue un rôle important. Par exemple dans un rendu *stippling*, la densité des marques donne le ton dans l'image finale. Le placement prend en compte le mouvement à suivre en considérant la position des marques de l'image précédente de l'animation.

Attributs des marques Une marque est définie par un ensemble d'attributs permettant de la dessiner sur le support. Ces attributs peuvent évoluer au cours de l'animation. La position de la marque est un attribut particulier qui joue un rôle important dans la gestion du compromis de cohérence temporelle. Les autres attributs décrivent la forme et la manière de dessiner la marque sur le support. Aussi,



FIGURE 3.6 – Influence de l’ordre de dessin des marques. Ici trois ordres différents sont utilisés pour dessiner le même ensemble de marques. On voit les effets indésirables qui peuvent apparaître comme l’empilement (à gauche).

l’évolution des attributs peut créer des changements brutaux ou une mauvaise représentation des effets de matière.

Ordre Une fois que l’on a déterminé la position et les attributs de chacune des marques, on doit définir dans quel ordre les dessiner sur le support. L’ordre de dessin influe donc sur l’apparence de l’image finale comme le montre la figure 3.6. De plus, au cours de l’animation, l’ordre peut être amené à changer d’une image à l’autre et donc produire des changements brutaux.

Dessin Suivant la liste ordonnée de marques, on doit calculer la couleur de chacun des pixels de l’image au moyen d’une fonction de mélange. C’est cette étape qui permet d’avoir les effets de matière correspondant au style voulu, par exemple en considérant l’épaisseur du médium sur le support et en calculant les reflets (très fréquents en peinture à l’huile).

Calques Lorsque l’on rend une animation, il peut être pratique de faire plusieurs rendus que l’on compose pour obtenir l’image finale. Ceci peut servir pour réaliser des traitements spéciaux pour telle ou telle composante du rendu, pour améliorer la cohérence temporelle, pour séparer le détail du fond, etc.

3.2.4 Travaux précédents

Nous allons, dans un premier temps, présenter les travaux en rendu à base de marques pour l’animation suivant le plan d’approche introduit dans la section précédente. Puis nous ferons le bilan comparatif de ces approches.

Placement des marques

Généralement, les travaux précédents proposent de suivre le mouvement des objets constituant la scène, c’est-à-dire le champ de mouvement ou *motion field*. Dans le cas d’une scène 3D, cela revient à suivre le mouvement des objets de la scène, dans le cas d’une vidéo à suivre le mouvement présent dans la vidéo, habituellement approximé par le flux optique [BB95].

Scène 3D Daniels et coll. [Dan99, DLK01] proposent de laisser l'artiste dessiner librement par dessus une scène 3D les marques représentant cette scène pour une vue donnée. Dans cette approche, les marques sont définies par les chemins tracés par l'artiste. Durant la procédure de dessin, le système enregistre l'ensemble des informations entrées par l'artiste. Pour chaque chemin dessiné, ils stockent la pression le long de ce chemin, l'ordre de dessin, la couleur et l'épaisseur choisies. Puis le système projette les chemins sur les objets constituant la scène. Au cours de l'animation, chaque chemin suit le mouvement et la déformation de la surface sur laquelle il a été projeté, puis est re-projeté en espace image pour pouvoir dessiner les marques. Kalnins et coll. [KMM⁺02] étendent cette approche avec le système WYSIWYG NPR. Ils proposent à l'utilisateur d'entrer des niveaux de détails, c'est-à-dire qu'ils lui permettent de faire varier le nombre de chemins utilisés pour le rendu, en fonction de la distance à l'objet représenté.

Ces approches permettent un très bon contrôle de bas niveau du placement des marques. En contrepartie, elles demandent beaucoup de travail et d'expertise à l'artiste, qui doit placer chacune des marques individuellement. De plus, la déformation 3D du chemin peut produire des formes de marques incohérentes avec les chemins dessinés au départ, à cause de la projection perspective.

Meier [Mei96] propose de placer un ensemble de points d'attache sur les objets. Ces points d'attache suivent le déplacement des objets auxquels ils sont attachés. Pour rendre les marques, les points d'attache sont projetés en espace image, la forme de la marque est ensuite calculée à partir de ses attributs, directement en espace image, s'affranchissant ainsi du problème de la projection du chemin. Les points d'attache sont distribués sur les objets 3D dans un pré-processus, leur nombre ne varie pas au cours de l'animation et donc si on zoome sur un objet, les points d'attache présents sur cet objet s'écartent de plus en plus. La distribution des points d'attache (leur densité en espace image) varie donc avec la distance des objets au point de vue.

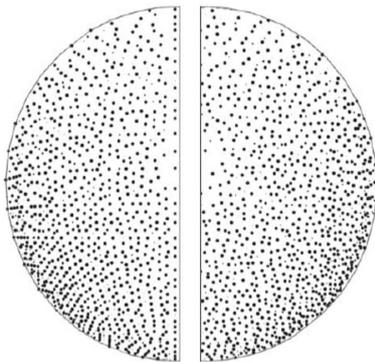


FIG. 3.7 Rendu stippling tiré de Meruvia-Pastor [PFS03]. À gauche sans randomisation, à droite avec.

Dans le cas d'un rendu peinture avec des marques relativement opaques, la qualité de la distribution des marques n'est pas fondamentale. Par contre, certains styles comme le *stippling* ont besoin d'avoir de meilleures propriétés de répartition des marques en espace image. Meruvia-Pastor et coll. [PFS03] proposent une méthode de *stippling* animé en temps réel. Ils se basent sur une hiérarchie de points d'attache, construite en précalcul suivant des techniques de simplification et de subdivision de maillage, comme celles présentées par Hoppe [Hop96] et Luebke [Lue01]. Cela leur permet de choisir un nombre de marques suivant le ton de la scène en entrée. Comme les points d'attache sont distribués sur les objets, la projection fait apparaître des artefacts d'alignement (voir figure 3.7)

même dans le cas où les positions des points sont randomisés. En effet, notre perception fait que nous groupons les points alignés, et que si cet alignement n'a pas de lien avec la scène représentée, nous construisons des informations qui peuvent être contradictoires avec l'intention de l'artiste.

Dans le même ordre d'idée, Haller et Sperl [HS04] utilisent une hiérarchie pré-

calculée de points d'attache, et n'en sélectionnent que le bon nombre au moment du rendu. Ceci permet de meilleures performances pour le rendu des objets lointains (moins de marques sont dessinées) et un zoom plus important. Cependant, la restriction du point de vue subsiste car le nombre maximal de points d'attache est prédéterminé, tout comme dans l'approche Chi et Li [CL06] qui utilise une hiérarchie précalculée de sphères englobantes sur un objet 3D pour conduire la sélection des points d'attache.

Vidéo Dans le cas d'une vidéo, on ne peut pas suivre directement le mouvement des objets 3D car l'information n'est pas disponible. Si l'on veut suivre les mouvements des objets, on peut utiliser le flux optique [BB95]. En fait, les techniques qui utilisent le flux optique peuvent facilement s'étendre pour d'autres mouvements. Comme nous l'avons souligné précédemment, l'application aux marques d'un mouvement non linéaire en espace image crée des accumulations et des trous dans la distribution de celles-ci. Le système doit donc fournir un moyen de mise à jour de la distribution de marques pour obtenir un rendu satisfaisant.

Hertzmann et Perlin [HP00] effectuent un rendu à base de marques d'une vidéo avec des marques définies par un chemin. Les chemins utilisés sont caractérisés par un ensemble de points de contrôle. Pour animer les marques, ils déplacent ces points de contrôle suivant le flux optique de la vidéo. Les résultats obtenus offrent une certaine forme de cohérence temporelle. Mais la déformation d'une marque par le flux optique ne correspond pas forcément à la déformation de la scène sous-jacente. On perçoit alors l'animation comme visqueuse, notamment une marque peut rapidement se trouver étirée sur de larges régions de l'image. Cette viscosité est principalement due aux imprécisions du mouvement présent dans le flux optique. Elle donne l'impression que les marques sont fluides, ce qui est en contradiction avec l'aspect du rendu qui suggère un médium épais tel que la peinture à l'huile.

Comme dans le cas des scènes 3D, l'utilisation de points d'attache est possible. Ceci permet de laisser une certaine marge entre le mouvement des marques et le flux. En effet, le flux contrôle le mouvement des points d'attache et les marques se déplacent suivant ces points d'attache. Chaque marque se déforme alors plus subtilement et ne provoque pas l'effet de viscosité que nous venons de discuter.

Litwinowicz [Lit97] construit, après le déplacement des points d'attache déjà présents, une triangulation de Delaunay [Del34, dBCvKO08]. Il met à jour la distribution en considérant deux critères : pour chaque triangle, si son aire est supérieure à un seuil, un nouveau point d'attache est placé en son centre. Ensuite, pour supprimer les points d'attache, il regarde la longueur maximale des arêtes de la triangulation autour de chaque point d'attache. Si cette longueur est inférieure à un certain seuil, le point d'attache est supprimé. Cette technique permet de garantir une densité minimale de points d'attache tout en supprimant les accumulations. Par contre, elle peut induire de fortes variations locales de densité, comme le montre la figure 3.8.

Hays et Essa [HE04] étendent cette approche en utilisant plusieurs calques. Chaque calque contient un ensemble de marques d'une taille donnée. Ils distribuent les points d'attache indépendamment dans chaque calque et utilisent une

fonction d'importance⁴ pour sélectionner la région où placer les points d'attache. Cette région correspond au niveau de détail défini par la taille des marques associées aux points d'attache du calque. Les points d'attache suivent le flux optique de la vidéo. Dès qu'une zone est non couverte par les marques déjà distribuées, ils ajoutent une marque dans cette zone. De plus, ils suppriment une marque lorsqu'elle est recouverte ou qu'elle sort de la zone de détail à laquelle elle est attachée. Si on considère uniquement les points d'attache, la distribution ne présente pas de bonnes propriétés car les points peuvent être arbitrairement proches. Ils utilisent cette distribution pour faire du rendu peinture où les coups de pinceau sont opaques, ce qui masque la distribution, mais leur méthode n'est pas extensible à d'autres styles où la distribution de marques serait plus visible.

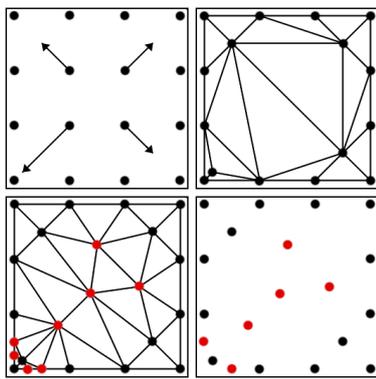


FIG. 3.8 Image tiré de Litwinowicz [Lit97] montrant la mise à jour des points d'attache suivant le flux optique.

Smith et coll. [SLK05] animent des marques pour obtenir une mosaïque en mouvement. Leur méthode de distribution se base sur une généralisation de la relaxation de Lloyd [Llo82, LBG80] à des éléments quelconques. L'originalité de leur travail vient du choix fait pour le mouvement des points d'attache. Ils travaillent sur une animation vectorielle en entrée : l'animation est constituée de formes à remplir. Chaque forme est définie par un contour vectoriel animé et une couleur de remplissage. Leur objectif est de remplir les formes avec des petits éléments de mosaïque en les adaptant aux déformations et mouvements de la forme lors de l'animation. Ils proposent deux schémas de déplacement des marques distribuées suivant le mouvement du contour. Dans les deux cas, ils translatent les marques suivant le mouvement du centre de gravité du contour. Puis,

pour les autres déformations, ils proposent : soit de laisser les points d'attache en place, en ajoutant des points d'attache dans les trous et en supprimant ceux sortant de la forme, soit de déplacer les marques en suivant la déformation du point sur le contour le plus proche de chacun d'eux. Cela permet de grouper les ajouts et suppression, la première approche appuie le coté statique de la forme alors que la deuxième renforce la perception de la déformation du contour. Enfin, ils appliquent la relaxation de Lloyd pour améliorer la qualité de la distribution. Cette approche offre une grande qualité de distribution mais le mouvement suivi ne peut que difficilement être contrôlé. Par exemple si les marques sont déplacées suivant le mouvement des objets d'une scène 3D, les trous créés ne seraient plus centralisés et les ajouts et suppressions ne seraient plus groupés.

Évolution de l'ordre de dessin

Dans les travaux précédents, on trouve deux solutions simples pour déterminer l'ordre de dessin des marques : utiliser soit un ordre fixe (arbitraire), soit un ordre fourni par la scène comme la distance des objets représentés par rapport au point de vue. Si l'ordre utilisé est trop « parfait », comme par exemple suivre strictement

4. La fonction d'importance associée à chaque point p de l'espace, image dans les cas que nous considérons, une valeur $I(p) \in [0, 1]$. Cette valeur détermine la variation du nombre de points d'attache en espace image.

la profondeur lue sous le point d'attache, alors le rendu des marques fait apparaître un effet d'empilement (figure 3.6).

Dans le cas d'une distribution dynamique sur une vidéo, Litwinowicz [Lit97] insère aléatoirement les nouvelles marques dans la liste des marques déjà présentes. Il utilise cet ordre pour ne pas avoir d'empilement. Par contre, cela crée certains changements brutaux quand une marque apparaît par dessus les autres.

L'approche de Hays et Essa [HE04] consiste à ajouter les nouvelles marques toujours sous les marques déjà présentes, et de conserver cet ordre au cours de l'animation. Par contre, ils utilisent une liste de marques par calque, cela réduit l'effet d'empilement qu'ils auraient avec un seul calque, mais peut produire des changements brutaux.

Si on utilise un ordre arbitraire, on peut percevoir des formes fantômes, non présentes dans la scène d'origine, lors de l'animation. C'est particulièrement visible quand on utilise des points d'attache sur les objets 3D car le mouvement des points d'attache est plus précis que lorsqu'ils suivent le flux optique d'une vidéo. Par exemple, considérons un plan recouvert de quelques marques, dessinées dans un ordre fixe. Lors de la rotation du plan, une marque représentant une partie lointaine peut se déplacer devant une marque représentant une partie du plan proche du point de vue. Nous percevons alors une contradiction entre le mouvement des marques et leur profondeur, si bien que nous déformons le plan sous-jacent et que nous avons l'impression qu'il s'agit d'une forme plus ou moins creusée. De plus, si on utilise un ordre arbitraire, les objets du fond peuvent passer devant les objets du premier plan. Enfin, il faut prendre soin d'éliminer les marques sur les surfaces des objets qui ne sont pas visibles depuis la caméra. Concrètement cela peut poser d'autres problèmes, car la visibilité d'un point d'attache n'est pas forcément celle de la marque correspondante (qui prend plus de place en espace image).

Dans le cas d'une scène 3D, avec des points d'attache sur les objets, on peut utiliser l'algorithme du peintre⁵. C'est l'approche choisie par Meier [Mei96] qui trie les marques en profondeur, c'est-à-dire suivant la distance des points d'attache par rapport à la caméra. Cette technique permet de dessiner les parties arrières avant les éléments du premier plan et assure une bonne représentation de la scène. Par contre, l'ordre trop parfait nuit à la bonne représentation des effets de peinture et produit un empilement des marques. Pour pallier cette limitation, Meier utilise des marques dont le bord est flou et dont la forme n'apparaît pas clairement dans le rendu. De plus, elle offre la possibilité de dessiner les marques sur plusieurs calques (en sélectionnant un sous ensemble des marques distribuées pour chaque calque), ce qui réduit l'effet d'empilement. L'utilisation de calques implique d'enlever les marques sur les faces cachées. Meier utilise un simple test de face arrière et, pour ne pas avoir de changements brutaux, elle détermine la transparence des marques suivant l'angle entre la normale et le vecteur de vue⁶. En utilisant la même technique que Meier, si l'on souhaite des styles différents, notamment avec des marques

5. L'algorithme du peintre consiste à trier les éléments qui doivent être dessinés suivant leur distance au point de vue. Ensuite, les éléments sont dessinés dans l'ordre du plus lointain au plus proche (sans test de profondeur, comme c'est l'usage dans le pipeline graphique actuel). Ainsi le dessin obtenu correspond aux éléments visibles depuis le point de vue.

6. Quand la normale fait un angle de $\pi/2$ avec le vecteur de vue, la marque est totalement visible, si cet angle diminue, la marque est rendue de plus en plus transparente.

dont les bords sont plus marqués, l'apparition de changements brutaux lors des modifications d'ordre est inévitable, et l'aspect d'empilement plus flagrant.

Daniels [Dan99] propose une technique hybride, utilisant l'ordre de dessin par l'artiste et la profondeur pour déterminer l'ordre d'affichage. Si deux marques sont proches en profondeur, c'est l'ordre de dessin qui est choisi, sinon elles sont dessinées en suivant l'algorithme du peintre. Les changements d'ordres sont adoucis en utilisant une fonction de mélange qui permet de faire passer une marque devant l'autre de manière continue.

Utilisation de calques

Comme nous l'avons déjà indiqué, Meier propose d'utiliser des calques pour rendre des sous parties de la scène, et de composer les différents calques pour obtenir l'image finale. Par exemple, Meier utilise un calque différent pour les parties spéculaires et pour les ombres. Elle place le calque spéculaire par dessus les autres calques, ce qui met en avant la forme des objets. Pour cela, elle sélectionne une sous-partie des marques distribuées dans la zone correspondante, et fait le rendu dans un calque séparé. Le gain en qualité de rendu se fait au détriment de la continuité temporelle. En effet, lorsqu'un point d'attache change de zone (lors d'un mouvement de la lumière par exemple), le rendu de la marque change de calque et donc change d'ordre.

Hays et Essa, dans l'esprit d'Hetzmann [Her98], utilisent plusieurs calques pour les différents niveaux de détails. Chaque calque comporte des marques d'une taille donnée, et ils composent les calques dans l'ordre de la taille des marques correspondantes. Cela leur permet de faire ressortir les détails (les marques les plus petites ne sont pas recouvertes par les plus grosses), par contre cela introduit des changements brutaux lorsqu'ils enlèvent une marque d'un calque au premier plan.

Dessin des marques

Chaque marque couvre une certaine partie de l'image finale. Pour chaque pixel couvert, il faut déterminer la contribution de la marque. Pour cela, on utilise les paramètres de la marque et l'état du support au moment du dessin de cette marque. Lors de cette étape, l'objectif est d'obtenir les éléments visuels caractéristiques du style et du médium visés, sans nuire à la continuité temporelle du rendu, c'est-à-dire sans introduire de changements brutaux.

Daniels [Dan99] utilise un modèle paramétrique, la marque étant entièrement déterminée par un ensemble de données qui décrivent aussi bien sa forme, sa couleur, son épaisseur et sa transparence, comme le montre la figure 3.9. Une fonction de remplissage de la marque donne sa contribution pour chacun des pixels qu'elle recouvre. La fonction utilisée permet d'obtenir un effet estompé sur les bords de la marque, mais pour obtenir plus de matière, il faudrait utiliser une fonction plus complexe. Il se pose alors le problème de la paramétrisation de cette fonction, notamment lors de l'animation. En général, la paramétrisation dépend du chemin considéré : lors de l'évolution de la taille du chemin, la paramétrisation

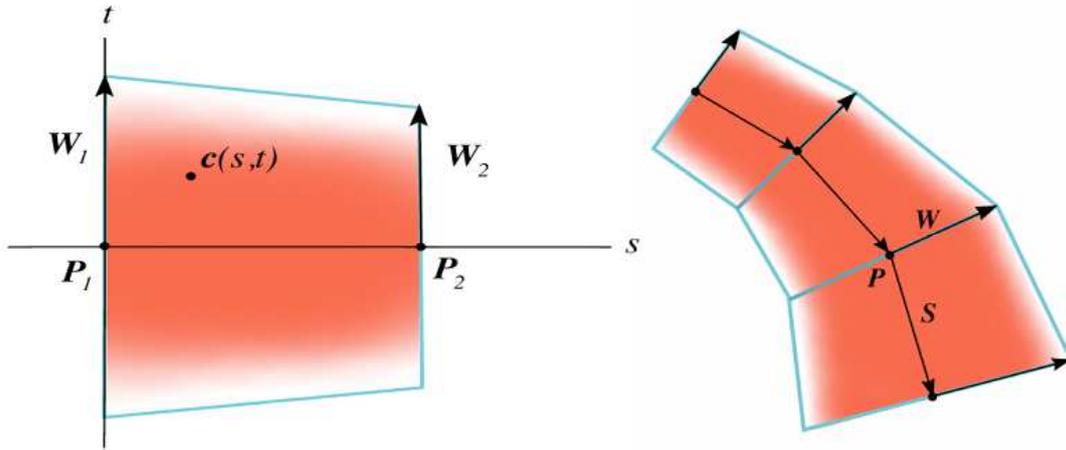


FIGURE 3.9 – Modèle de marque utilisé dans deep canvas [Dan99]

change. Par exemple, si le chemin s’allonge en espace image et que l’on garde la même paramétrisation, les éléments visuels s’allongent de la même manière. Kalnins et coll. [KDMF03] proposent une méthode pour obtenir une paramétrisation cohérente, en espace image, d’un chemin défini en espace 3D. Certains sauts de paramétrisation sont cependant inévitables, et des glissements de matière se produisent aux extrémités des marques.

Dans le cas de marques définies par un point, l’ensemble de la paramétrisation et du dessin se fait en espace image. Il n’y a alors plus de problème lié à la projection. Une solution souvent choisie pour simuler la matière est l’utilisation d’une texture décrivant les variations de transparence et de couleur pour la marque [Mei96, Lit97, KMM⁺02, MT02, MT03, HS04, CL06]. Cette texture est plaquée sur la forme de la marque. Pour obtenir les différentes tailles de marques, la texture est étirée, ce qui provoque des inconsistances dans la taille des éléments visuels constituant la matière.

Hertzmann [Her02] propose d’utiliser une texture de relief (*bump map*) pour obtenir un champ de hauteur de matière présente sur le support. Ce champ de hauteur permet alors de moduler la couleur des marques en calculant l’illumination correspondante. Cette technique peut facilement être appliquée aux animations.

Une fois que la contribution de toutes les marques couvrant un pixel donné est établie, on doit déterminer comment utiliser cette information pour obtenir la couleur finale en ce pixel. L’approche utilisée dans les travaux précédents est de faire de l’*alpha blending*. À chaque pixel couvert par une marque est associé une valeur de transparence, la couleur finale est alors obtenue par une combinaison des couleurs et des transparences des marques. Les méthodes que nous avons présentées utilisent toutes le mélange suivant :

$$c_f = \alpha_n c_n + (1 - \alpha_n (\alpha_{n-1} c_{n-1} + (1 - \alpha_{n-1} * (\alpha_{n-2} \cdots + c_1))))$$

Où c_f est la couleur obtenue, et les couples (c_i, α_i) sont les couleurs et transparences des n marques à dessiner, $i \in [1, n]$. Ce type de mélange ne fait pas de simulation physique pour obtenir les effets visuels du médium : c’est la transparence utilisée sur les marques qui donne la qualité du résultat. En effet, si des marques

semi-transparentes sont utilisées, même si chaque marque a une texture de coup de pinceau, le résultat est flou et la matière présente sur chacune des textures est perdue. Les valeurs des α_i sont prises dans une texture, elles peuvent aussi être modulées pour faire apparaître et disparaître une marque de manière douce.

Contrôle de l'artiste

Dans de nombreux systèmes [Lit97, PFS03, HS04, HE04] que nous présentons, le contrôle du style s'effectue par un ensemble de paramètres que l'artiste peut spécifier ou qui sont définitivement fixés dans le système. Ces paramètres permettent de calculer les attributs des marques et leur évolution au cours de l'animation.

L'artiste doit pouvoir donner son style à l'animation. Pour cela, Daniels [Dan99], comme Kalnins et coll. [KMM⁺02], lui laisse un contrôle de bas niveau. L'artiste place tous les coups de pinceau à la main sur le modèle 3D, indique les paramètres de style et les niveaux de détails nécessaires. Dans le cas du système WY-SIWYG NPR de Kalnins et coll., le système facilite certaines tâches en travaillant par l'exemple. L'artiste dessine un exemple de marque dont le style est appliqué à un ensemble de marques déjà placées. Kalnins et coll. définissent le style d'une marque par des propriétés statistiques sur l'écart du chemin de la marque par rapport à une ligne droite et par la texture utilisée pour faire le dessin de la marque.

Ces techniques permettent un bon contrôle du rendu obtenu, mais demandent beaucoup de travail et d'expertise à l'artiste qui produit l'animation. De plus, dans le cas du rendu interactif, il n'est pas possible de prévoir le mouvement de l'utilisateur, et donc de dessiner tous les cas de figure possibles (tous les niveaux de détails par exemple), le système trouve alors rapidement ses limites.

Dans les approches de plus haut niveau, l'artiste contrôle les attributs des marques, indépendamment de celles-ci⁷. Meier [Mei96] utilise ainsi des images de référence. Ces images sont obtenues en effectuant le rendu de la scène 3D de manière classique pour la couleur, ou avec des textures spécifiques pour d'autres paramètres. Par exemple, l'artiste peut utiliser une texture en niveaux de gris pour indiquer l'épaisseur des coups de pinceau ; puis, le système rend la scène avec cette texture pour obtenir une image de référence. Au moment du dessin d'une marque, la valeur de l'image de référence se trouvant sous le point d'attache associé donne l'épaisseur de la marque. Un soin particulier est à apporter dans l'utilisation des images de référence pour garantir la pertinence de l'information lue sous un point d'attache. En effet, si le point d'attache est au bord de l'objet, les erreurs numériques du calcul du pixel à lire peuvent conduire à un pixel en dehors de l'objet. De plus, si le point d'attache est caché, la valeur lue correspondra à l'objet le plus proche. Meier résout ce problème en ne rendant qu'un objet à la fois dans l'image de référence, et en dilatant celui-ci pour garantir que la valeur lue est bien sur l'objet. Ceci entraîne néanmoins des limitations quant aux scènes utilisables, et ne résout pas le problème dans le cas d'une auto occlusion.

En utilisant des textures plaquées sur les objets, l'artiste contrôle directement les attributs utilisés pour les marques. Pour faire évoluer les attributs au cours de

7. De la même manière que la description d'un matériau se fait indépendamment des objets auxquels on l'applique. C'est une définition *a priori* des attributs des marques.

l'animation, il pourrait utiliser des textures animées. Par contre, il ne peut pas utiliser les informations de la scène et des objets pour calculer les attributs, notamment la position de l'objet en espace objet et en espace image.

Récapitulatif

L'analyse que nous faisons des travaux précédents est synthétisée dans le tableau 3.1. Ce tableau reprend les travaux que nous avons cités et récapitule les cinq axes majeurs que sont le contrôle par l'artiste, le type d'entrée, le type de placement, l'ordre utilisé et la technique de dessin de marques. Nous donnons maintenant les conclusions que nous tirons de l'analyse des travaux précédents.

La distribution des points d'attache influe sur le rendu final et la liberté par rapport à la scène d'entrée. Dans le cas d'une scène 3D, les distributions statiques, en espace objet, peuvent être précalculées, et le temps de rendu en est alors réduit. Mais cela limite le point de vue et les déformations possibles des modèles. De plus, la qualité de la distribution peut être très mauvaise. Dans le cas des vidéos, les méthodes existantes distribuent les points d'attache en espace image, ce qui permet une meilleure qualité de distribution. Le mouvement suivi est alors facilement extensible à tout mouvement donné en entrée.

Nous notons que le placement et le mouvement des marques jouent un rôle particulier qui est relativement indépendant des autres choix. En effet, la qualité du placement des marques est perçue dans certains styles, comme le *stippling* et lorsque les marques sont transparentes, mais n'influence pas la manière dont les attributs des marques sont calculés et évoluent lors de l'animation. De plus, les mécanismes mis en œuvre pour améliorer la continuité temporelle sont généralement des lissages temporels des changements présents dans les différentes techniques.

D'une manière générale, une méthode donnée peut être étendue à l'utilisation de plusieurs calques si la distribution de marques peut être effectuée pour une partie de la scène seulement. Ensuite, pour la composition des calques, il faut être capable de les ordonner et d'avoir un masque de transparence par calque pour effectuer le mélange. Deux cas de figure se présentent. Soit la densité des marques n'est pas perçue et il suffit alors d'offrir la possibilité de faire le rendu sur une partie de l'espace image seulement. Soit la densité des marques est clairement perçue et il faut alors que la distribution se fasse en tenant compte des calques pour que la composition de ceux-ci ne perturbe pas la qualité de la distribution.

Nous avons vu comment les techniques précédentes offrent le contrôle à l'artiste. Dans le cas d'animations prédéterminées avec un contrôle de haut niveau, le contrôle local de l'artiste peut être facilement ajouté. En effet, il suffit pour cela de conserver l'ensemble des structures de données générées par le système lors d'une première passe de rendu et de laisser l'artiste modifier les images de l'animation qu'il souhaite. Bien qu'il soit préférable pour l'artiste de ne pas avoir à modifier les images obtenues (et donc d'offrir un bon contrôle de haut niveau), certaines marques peuvent avantageusement être dessinées en post-processus. On pense notamment aux marques qui n'ont pas seulement un rôle de remplissage, mais qui ont une signification propre, comme par exemple une marque représentant le sourcil d'un visage.

Ref	Style	Placement		Ordre			Dessin	
		dyn/stat	2D/3D	fixe	prof	hybride	proc	tex
[HP00]	Sliders	dyn	2D	✓			✓	
[Lit97]	Fixé dans le système	dyn	2D	✓				✓
[SLK05]	Distribution	dyn	2D	✓				✓
[Dan99]	Définit manuellement	stat	3D			✓	✓	
[PFS03]	Sliders	dyn	3D		✓		✓	
[KMM ⁺ 02]	Définit manuellement	dyn	3D		✓			✓
[Mei96]	Images de reference	stat	3D		✓			✓
[HS04]	Globale	dyn	3D		✓			✓
[CL06]	Sliders	dyn	3D		✓			✓

TABLE 3.1 – Récapitulatif des différents choix faits dans les travaux précédents. *Sliders* désigne un contrôle par le biais d'une interface où l'artiste peut modifier certains paramètres.

Chapitre 4

Placement des points d'attache

L'ANALYSE du rendu à base de marques montre que la distribution des points d'attache est indépendante du reste du système de rendu. Nous décrivons dans ce chapitre la partie distribution de notre système, et nous explicitons le fonctionnement complet au chapitre suivant. Les méthodes que nous proposons pour la distribution adaptative des points d'attache offrent une liberté totale quant à la position du point de vue. Par ailleurs, les temps de calcul obtenus permettent de générer plusieurs images par secondes. Elles sont donc particulièrement adaptées pour le rendu interactif, comme les jeux vidéo mais aussi dans le cadre de la production d'animation, lorsque l'artiste met au point son style de rendu et qu'il a besoin d'avoir un retour visuel rapide sur les différents changements des paramètres de styles et placements de caméra.

Nous proposons deux méthodes de distribution. La première étend l'approche de Meier [Mei96] en offrant des performances adaptées pour le rendu interactif au prix d'une qualité de distributions limitant les styles possibles. La seconde permet une meilleure distribution des points d'attache en utilisant l'espace image, ce qui permet d'éliminer les problèmes liés à la projection perspective que nous avons discuté section 3.2.4. Elle permet donc une plus grande gamme de style, notamment ceux où la distribution des primitives est fortement perçue comme le *stippling* et le pointillisme.

4.1 Qualité de distribution et compromis

La distribution et le déplacement des points d'attache jouent un rôle important pour la cohérence temporelle. En effet, une bonne distribution aide à bien représenter les effets de matière présents dans le style. Le choix du déplacement des points d'attache est lié au mouvement à suivre, il faut donc mettre à jour la distribution pour représenter la densité des points d'attache voulue. La mise à jour de la distribution peut alors créer des discontinuités temporelles si on ne prête pas une grande attention à réaliser des insertions et suppressions douces. Ce sont dans nos choix quant au déplacement et la mise à jour des points d'attache que se joue la majeure

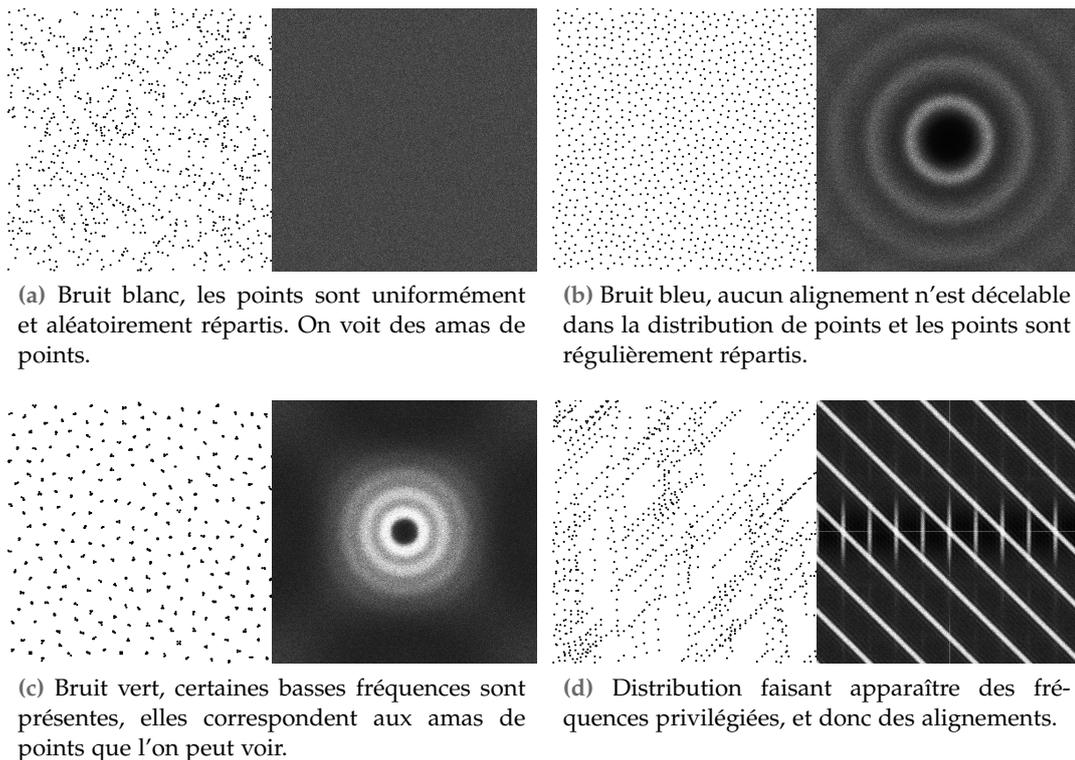


FIGURE 4.1 – Exemples de distributions de points avec les spectres de Fourier correspondants.

partie du compromis entre les trois contraintes de cohérence temporelle que nous avons introduites section 2.3.2.

Se pose tout d'abord la question de la qualité de la distribution des marques. Comme nous l'avons déjà vu dans l'analyse du rendu à base de marques chapitre 3.2, tout dépend de l'application. Pour un rendu avec des marques opaques, nous voulons une distribution qui recouvre le support. Comme le temps de rendu de chaque marque compte, nous voulons un minimum de marques pour remplir l'objectif. Enfin, lorsque la distribution des marques est directement observée, nous voulons qu'elle soit *naturelle*; c'est-à-dire qu'elle ne présente pas d'artéfact visible tel que des alignements. Cette notion n'est pas très bien définie, néanmoins l'analyse spectrale, au moyen de la transformée de Fourier et du spectre de puissance, permet de faire quelques observations. Les agglomérats de points répétés se traduisent par la présence de basses fréquences dans le spectre de Fourier et les alignements par des pics de fréquence. Si une direction est plus marquée elle apparaît sous la forme d'une anisotropie dans le spectre. Enfin, la bonne répartition des points correspond à une fréquence privilégiée uniformément dans toutes les directions. Ces contraintes correspondent aux distributions appelées *bruit bleu* qui n'ont aucune basse fréquence, pas de direction privilégiée (anisotrope) et possèdent un anneau de fréquences plus marquées. La figure 4.1b montre une distribution de point bruit bleu et le spectre de Fourier correspondant, d'autres types de spectres sont montrés avec les distributions de points correspondantes figure 4.1.

Pour spécifier la densité des points, il est commode d'utiliser une fonction d'importance, ou fonction de distribution de probabilité (*PDF*). Cette fonction est définie dans l'espace où l'on veut distribuer les points (l'espace image dans notre cas) et

donne la densité locale de points que nous voulons obtenir.

Nous cherchons donc à produire une distribution suivant une *PDF* donnée, et ayant les propriétés de bruit bleu. Notons que les propriétés de bruit bleu sont définies pour une distribution uniforme, mais que visuellement, on peut juger de la bonne qualité des distributions non-uniformes selon les mêmes critères que la mesure de bruit bleu.

4.1.1 Distribution 2D de points

Les distributions de points forment un domaine de recherche à part entière ; il existe beaucoup de méthodes de distribution et de mesures de la qualité de ces distributions. Les distributions de points servent à évaluer des fonctions, notamment les intégrales de fonctions, comme dans les méthodes Monte-Carlo [DBB06]. Dans notre cas, nous distribuons des points d'attaches pour dessiner des marques. Suivant le type de marque, la distribution est plus ou moins perçue, mais dans tous les cas, nous observons directement cette distribution, contrairement aux méthodes Monte-Carlo où seul le rendu obtenu indirectement à partir de cette distribution est observé.

Commençons par un rapide panorama des techniques de distribution de points 2D. Le moyen le plus simple de placer des points est de distribuer aléatoirement ces points dans la région à couvrir. C'est une distribution uniforme de points (bruit blanc) dont le spectre de Fourier correspondant comprend toutes les fréquences dans toutes les directions. Pour distribuer des points de manière uniforme, on peut tirer aléatoirement les coordonnées de points jusqu'à obtenir le nombre de points correspondant à la *PDF* voulue. Dans ce type de distribution, il n'y a ni distance minimum entre les points (donc on peut avoir des regroupements de points), ni distance maximum (donc des trous relativement larges dans la distribution).

Une meilleure distribution, du point de vue des contraintes que nous avons citées, est ce qu'on appelle une distribution suivant le critère de disque de Poisson. Une distribution qui respecte le critère de disque de Poisson garantit une distance minimum entre les différents points. Ce type de distribution a été introduite par Yellot [Yel82, Yel83], qui étudie le placement des photo-récepteurs dans l'oeil humain et remarque qu'ils suivent une distribution aléatoire, tout en respectant une certaine distance typique. Ce type de distribution permet de réduire les artéfacts d'échantillonnage tels que le bruit et les moirés.

Lagae et Dutré [LD08] ont réalisé une étude approfondie des différentes techniques de distributions de disque de Poisson. Dans leur étude, Lagae et Dutré comparent les différentes techniques suivant plusieurs critères. Un premier est le temps d'exécution des différentes méthodes, en considérant notamment la complexité de l'algorithme. Ils proposent aussi la comparaison du spectre de puissance et du rayon relatif de chacune des méthodes. Rappelons rapidement comment ces deux mesures sont calculées.

Spectre de puissance Cette mesure donne des informations importantes sur le contenu fréquentiel d'une distribution de points. Elle est calculée en faisant la

moyenne des carrés des magnitudes de spectres de Fourier (périodogramme) obtenues pour plusieurs distributions créées avec une méthode donnée. Pour qualifier la qualité d'une méthode, on regarde si le spectre de puissance correspond aux critères de bruit bleu. Autour du pic central (que nous enlevons sur les figures pour plus de clarté), il y a une zone de basse énergie suivie par un anneau plus puissant, et enfin les hautes fréquences sont équitablement représentées. Ulichney [Uli87] décrit comment calculer ce spectre de puissance pour des distributions de points, et recommande d'utiliser dix distributions indépendantes. Les spectres de puissance présents dans cette thèse suivent la même méthode de calcul que celle utilisée par Lagae et Dutré.

Rayon relatif La seconde mesure représente, intuitivement, la qualité de l'arrangement de la distribution. Le moyen de placer le maximum de disques dans le plan est de placer leur centre sur une grille hexagonale (voir figure 4.2). Le rayon maximum r^* , obtenu pour la distribution de N points suivant une telle grille, est donné par

$$r^* = \sqrt{\frac{1}{2\sqrt{3}N}}$$

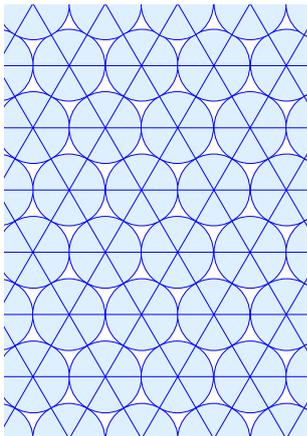


FIG. 4.2 Disques placés sur une grille hexagonale. C'est l'arrangement le plus compact de disques dans le plan.

Ainsi, le rayon relatif d'une distribution de disque de Poisson est défini comme la fraction $\rho \in [0, 1]$ du rayon maximum tel que $r = \rho r^*$. Le rayon relatif est, contrairement au rayon absolu, indépendant du nombre de points et de la taille du domaine de distribution, ce qui facilite la comparaison des méthodes. Selon Lagae et Dutré, une distribution de disque de Poisson doit avoir un rayon relatif élevé ($\rho \geq 0.65$) sans être trop élevé ($\rho \leq 0.90$) car les configurations trop régulières sont à proscrire.

Le critère de disque de Poisson est un critère de distance minimale r entre deux points distribués. Une méthode simple est le lancer de flèche [Coo86] (*dart throwing*), aussi appelé distribution par rejet. Cette méthode consiste à générer aléatoirement des points candidats (comme pour faire une distribution uniforme) et à tester chaque nouveau candidat avec les points déjà placés. Si la distance minimale entre le candidat et les autres points est supérieure à r , le candidat est conservé, sinon il est rejeté. L'algorithme de distribution s'arrête après n rejets consécutifs, n permet ici de faire un compromis entre temps d'exécution et qualité de la distribution. Cette méthode converge vers une distribution bruit bleu, mais les multiples tests de distance prennent beaucoup de temps : l'algorithme est en $O(n^2)$. Plusieurs améliorations ont été proposées pour accélérer la distribution, notamment celle de Dunbar et Humphreys [DH06] et celle de Jones [Jon06]. Ces deux méthodes maintiennent, lors de la distribution, une structure représentant une région où un nouveau point serait directement accepté. Le calcul de cette région après chaque ajout de points est relativement peu coûteux, et les temps de distribution sont moindres qu'avec la méthode de rejet. Ces deux approches sont en $O(n \log n)$ pour des résultats similaires. De plus Dunbar et Humphreys proposent aussi une technique en $O(n)$ au prix d'une perte de qualité de distribution que l'on peut voir dans le spectre de puissance.

McCool et Fiume [MF92] utilisent une approche basée sur le lancer de flèche, mais font varier le rayon du test de rejet. Les points sont initialement distribués avec un rayon large $r = r_{max}$. Une fois qu'aucun nouveau point n'est accepté après n essais, le rayon est multiplié par un facteur $\alpha \in [0, 1[$. L'algorithme s'arrête dès que $r \leq r_{min}$. La qualité de la distribution peut être contrôlée par les paramètres n et α , ce qui permet d'obtenir de bons temps de calcul pour la prévisualisation et le rendu interactif ($n = 100$, $\alpha = 0.9$), ou d'avoir une distribution de haute qualité pour le rendu final ($n = 1000$, $\alpha = 0.99$). La distribution obtenue respecte alors le critère de disque de Poisson pour un rayon de r_{min} . De cette manière McCool et Fiume obtiennent une meilleure distribution que le lancer de flèches classique, notamment le rayon relatif est plus élevé. De plus, ils proposent de faire une relaxation de Lloyd [Llo82] pour encore améliorer le résultat. En effet, la relaxation de Lloyd permet d'espacer des points déjà distribués. Elle consiste à construire le diagramme de Voronoï des points distribués et à déplacer les points au centre de leur région de Voronoï. En itérant ce processus, les points se déplacent pour former une distribution bruit bleu.

Pour accélérer le calcul de distribution, il est possible de faire une partie du travail en précalcul. Il suffit de calculer des morceaux de distribution à l'avance, puis de les assembler pour obtenir une distribution suffisamment grande pour les besoins de l'application. Pour cela, Lagae et Dutré [LD05] proposent d'utiliser des Wang Tiles. Les tuiles de Wang sont une technique de découpage (ou pavage) du plan : on définit un certain nombre de tuiles et des règles de proximité qui autorisent deux tuiles à se toucher lors du pavage du plan. Lagae et Dutré distribuent les points à l'avance dans chacune des tuiles. Pour garantir la bonne juxtaposition de deux tuiles, ils identifient les coins et les bords qui seront autorisés à se toucher. Ensuite, ils contraignent les coins et les bords, de telle sorte que lorsque l'on recolle les différentes tuiles, les critères de qualité des distributions soient respectés. La division des tuiles en coins, bords et intérieur permet de limiter les répétitions tout en offrant une distribution localement bruit bleu. Kopf et coll. [KCODL06] étendent cette approche en utilisant des tuiles de Wang récursives, ce qui leur permet d'obtenir des distributions non-uniformes. Ostromoukhov [ODJ04] utilise le pavage de Penrose [Gla98] qui est un pavage non-periodique et récursif, c'est-à-dire que l'on peut subdiviser chaque partie du pavage suivant des règles pour raffiner la taille des parties. Il définit un ensemble de tuiles ayant chacune un indice structurel qui représente la position dans le pavage (suivant les n dernières subdivisions). En plaçant un point par tuile du pavage, il optimise leur position pour obtenir une distribution bruit bleu.

Les méthodes que nous avons étudiées montrent comment distribuer des points en espace image 2D avec les propriétés de bruit bleu. Dans le cadre de l'animation en rendu à base de marques, nous voulons déplacer les points distribués tout en conservant ces propriétés. Nous allons maintenant voir deux méthodes que nous avons développées pour obtenir une distribution dynamique.

Si l'on veut offrir un mouvement non contraint de l'utilisateur dans la scène 3D en entrée, nous avons besoin de pouvoir adapter le nombre de marques, donc de points d'attache, à la volée. C'est-à-dire que le système de rendu doit pouvoir ajouter des marques, à la demande, sur certaines zones, pour assurer une couverture du support, tout en respectant les critères de qualité de la distribution que nous

venons d'énoncer.

Les méthodes que nous proposons tentent de répondre à ces critères. Ce sont des méthodes de distribution dynamique, qui permettent de déplacer et mettre à jour les points d'attache d'une image à l'autre dans l'animation.

4.2 Distribution en espace objet

Meier [Mei96] a introduit l'idée d'utiliser des points d'attache placés sur la surface des objets de la scène en entrée. Cela permet en effet de faire suivre le mouvement des objets aux marques. Le principal inconvénient de cette méthode et de ses extensions [MT02, MT03, HS04, CL06] est que l'espace des points de vue valides est limité par des choix faits lors des précalculs. Ces méthodes ne sont donc pas utilisables dans un cadre de rendu interactif. En effet, on ne peut pas prévoir à l'avance tous les points de vue que l'utilisateur va choisir.

Comme ces approches, nous faisons le choix de suivre très fidèlement le mouvement des objets, tout du moins pour les points d'attache. D'autres mouvements ne peuvent pas être suivis facilement car la position des points d'attache est directement liée à la surface des objets. Nous verrons dans la section 4.3 une autre approche offrant, entre autres, plus de liberté quant au mouvement en entrée.

4.2.1 Hiérarchie dynamique de points d'attache

L'idée, que nous présentons ici, est d'utiliser une hiérarchie dynamique de points d'attache sur les objets. Cette hiérarchie permet d'ajouter et de supprimer des points d'attache à la volée. En gardant à l'esprit les contraintes de cohérence temporelle et de performance, il faut à la fois essayer d'avoir un nombre suffisant de marques, tout en ayant le moins d'ajouts et suppressions possibles.

Les objectifs sont donc d'ajouter des marques pour couvrir la zone à dessiner et de supprimer les marques quand elles ne sont plus nécessaires à cette couverture, les ajouts et suppressions devant être le plus discrets possible.

Pour cela, nous utilisons une hiérarchie sur les objets, initialisée en précalcul, qui permet d'ajouter ou d'enlever localement des points d'attache. Pour construire cette hiérarchie, nous commençons par répartir les points d'attache sur la surface des objets de la scène en plongeant chaque objet dans un *octree* et en plaçant un point d'attache par cellule de *l'octree*. Nous raffinons *l'octree* jusqu'à obtenir le centre de chaque face du modèle dans une seule cellule (voir figure 4.3). Puis, en suivant un parcours de *l'octree* des feuilles vers la racine, nous construisons une hiérarchie de points d'attache en choisissant pour chaque cellule de *l'octree* le point d'attache le plus proche de son centre parmi les points d'attache des fils. Ainsi les points d'attache d'un niveau sont présents dans le niveau supérieur, ce qui réduit les sauts lors d'un changement de niveau dans la hiérarchie. Enfin, à chaque image de l'animation, nous effectuons une sélection des points d'attache suivant différentes méthodes présentées dans les sections suivantes. Si la hiérarchie n'est pas

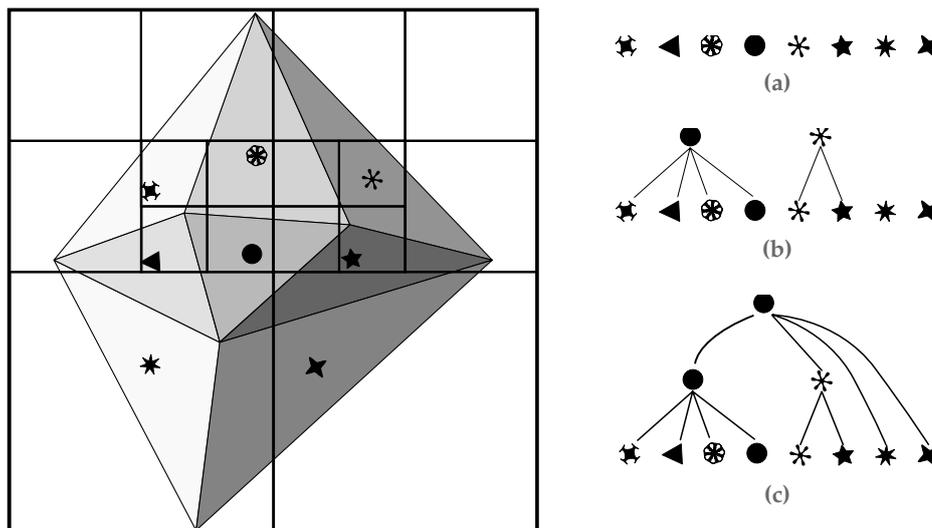


FIGURE 4.3 – Construction de la hiérarchie (cas 2D) : à gauche, l'objet plongé dans un quadtree avec une face (son centre) par cellule ; à droite, (a) les centres correspondant aux feuilles de l'octree; (b) après une itération de la reconstruction ; (c) la hiérarchie correspondant à l'objet.

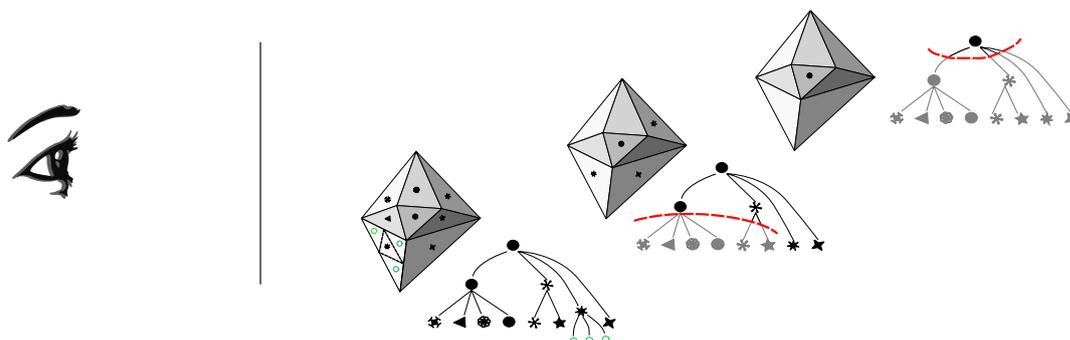


FIGURE 4.4 – Coupe de la hiérarchie de points d'attache suivant le point de vue.

assez dense, nous ajoutons des points d'attache dans la branche concernée en subdivisant la face de l'objet 3D correspondante (voir figure 4.4). Cette méthode ne fonctionne que pour des objets dont la géométrie est statique lors de l'animation, sinon la hiérarchie deviendrait incohérente avec la forme de l'objet.

4.2.2 Sélection du nombre de points d'attache

La sélection du nombre de points d'attache doit se faire de manière conservative par rapport à la couverture de l'image. Il faut donc garantir que le nombre de marques sélectionnées est suffisant, tout en effectuant la sélection le plus rapidement possible.

Nous proposons deux méthodes de sélection des points d'attache. Une première, itérative, nécessitant plusieurs passes de tests d'occlusions et sélectionnant exactement le nombre de points d'attache nécessaire à la couverture du plan image par les marques. La seconde faisant une sélection *a priori* plus performante en terme de temps de calcul, mais sélectionnant plus de points d'attache que le strict nécessaire.

Subdivision par test d'occlusion

Nous utilisons la hiérarchie de points d'attache que nous avons décrite précédemment. Pour chacun des points d'attache nous pouvons dessiner la marque correspondante. Nous voulons déterminer les marques à dessiner pour couvrir un objet donné : nous devons donc sélectionner le nombre de points d'attache nécessaires à cette couverture. Pour cela, nous commençons par dessiner la marque correspondant au noeud père et nous testons s'il reste des faces non couvertes. Si c'est le cas, on passe au niveau suivant de l'*octree* et on recommence les tests d'occlusion dans chacune des branches. Le rendu des marques se termine lorsque toutes les faces sont couvertes. Le test de couverture se fait en utilisant un masque pour les marques. Ce masque représente les parties de la marque que nous considérons comme opaques ; il peut être obtenu soit en faisant un masque sur la texture de la marque, soit en dessinant la marque totalement opaque, ce qui apporte un gain de performance. Dans ce dernier cas, on peut réduire la taille du masque par rapport à la taille de la marque pour prendre en compte les adoucissements présents sur les bords de la texture associée à la marque. L'algorithme suivant résume notre approche :

ALGORITHME 4.1 – Sélection des points d'attache par tests d'occlusion

```

l : file noeuds
enfile(l, pere(hierarchie))
tantque visible(object) faire
    n ← defile(l)
    dessine(marque(tete(n)))
    enfile(l, fils(n))
fin tantque

```

D'une image à l'autre, nous conservons la coupe faite dans l'*octree* comme point de départ de la mise à jour. Ceci permet d'éviter de reparcourir tout l'*octree* à chaque image. En effet, la nouvelle coupe de l'*octree* est en général proche de celle obtenue pour l'image précédente. Dans notre implémentation, nous avons optimisé les tests d'occlusions en envoyant plusieurs demandes à la carte graphique au moyen d'une file d'attente [BWPP04]. Puis nous pré-subdivisons les noeuds pour lesquels nous avons fait la demande d'occlusion¹. Si le résultat du test est positif (c'est-à-dire si au moins une face est visible), les fils pré-subdivisés sont insérés dans la file.

Cet algorithme se déroule en plusieurs passes de rendu. Bien que chacune des passes soit peu coûteuse, le nombre de passes nécessaires pour réaliser la sélection des points d'attache peut être important. Pour pallier ce problème de performance nous proposons de faire un choix *a priori* dans la hiérarchie, en prenant un critère conservatif pour garantir la couverture des objets par les marques.

Subdivision par prédiction de densité

Nous faisons l'hypothèse qu'une marque couvre au moins un disque dont le rayon est déterminé par le style. Nous faisons un choix *a priori* du niveau de subdivi-

1. Cette pré-subdivision est simplement la recherche des fils si le noeud n'est pas une feuille.

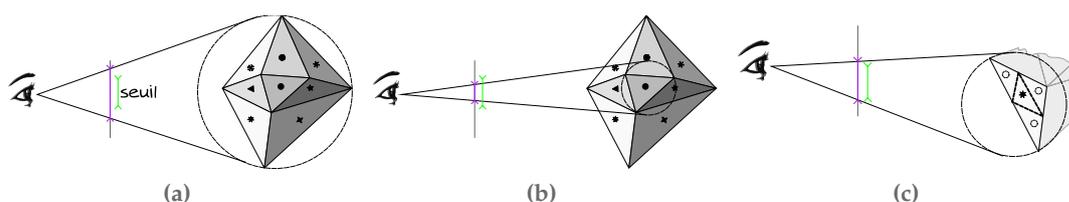


FIGURE 4.5 – Sélection des points d’attache : (a) la projection de l’objet est trop grande pour être représentée par un seul coup de pinceau ; (b) on descend donc dans la hiérarchie jusqu’à atteindre un noeud de taille convenable ; (c) si on atteint une feuille de la hiérarchie trop grande, on la subdivise dynamiquement.

visions en comparant le rayon projeté de la sphère englobante des faces correspondant à un noeud (voir figure 4.5). Si ce rayon est supérieur au rayon de couverture d’une marque, nous subdivisons le noeud ; sinon nous considérons le sous-arbre de la hiérarchie comme couvert.

Ce calcul est plus performant que les nombreuses passes de tests d’occlusions. En effet, pour chaque noeud, le test correspond à la projection du rayon (en faisant la division par la distance entre le centre et le point de vue) et à une comparaison. Le nombre de points d’attache sélectionnés est en revanche plus important que dans la méthode précédente.

Nous avons développé une approche similaire, mais basée sur un *vs-tree* [BHGS06]. Une hiérarchie est construite à partir des faces d’un objet 3D. Dès que les faces d’une cellule peuvent être projetées sans recouvrement sur un plan, nous insérons un noeud *plan* dans la hiérarchie. Lors de la distribution, nous plaçons les points d’attaches sur le plan, au moyen d’une séquence de Halton, puis nous projetons les points du plan vers les faces correspondantes. Pour connaître le nombre de points à distribuer, nous commençons par déterminer l’ensemble des points à distribuer pour l’objet en fonction de sa distance au point de vue. Ensuite nous répartissons le nombre de points à distribuer dans chacune des branches de la hiérarchie en fonction de l’aire projetée des faces contenues dans les feuilles de chaque branches. Une fois que cette répartition atteint un noeud *plan*, nous distribuons le nombre de points nécessaires sur le plan.

4.2.3 Résultats

Les deux méthodes présentées permettent d’obtenir des distributions de points d’attache en espace objet. Ces distributions s’adaptent au point de vue, de telle sorte que lors du dessin des marques, il y ait suffisamment de points d’attache pour couvrir le plan image. De plus, les points d’attache non nécessaires sont retirés, pour réduire le nombre de marques à dessiner (et donc gagner en temps de dessin).

Une partie de la construction de la hiérarchie est faite en précalcul. Le temps de précalcul dépend du nombre n de triangles du modèle 3D. En effet, il faut déterminer dans quel feuille chaque triangle se trouve, ce qui se fait en $O(n \log n)$.

Le temps de distribution dépend du nombre de points sélectionnés et de l’ajout de nouveaux point dans la hiérarchie lors de zoom. Nous avons effectué une mesure sur une animation test comprenant rotation et travelling avant et arrière important

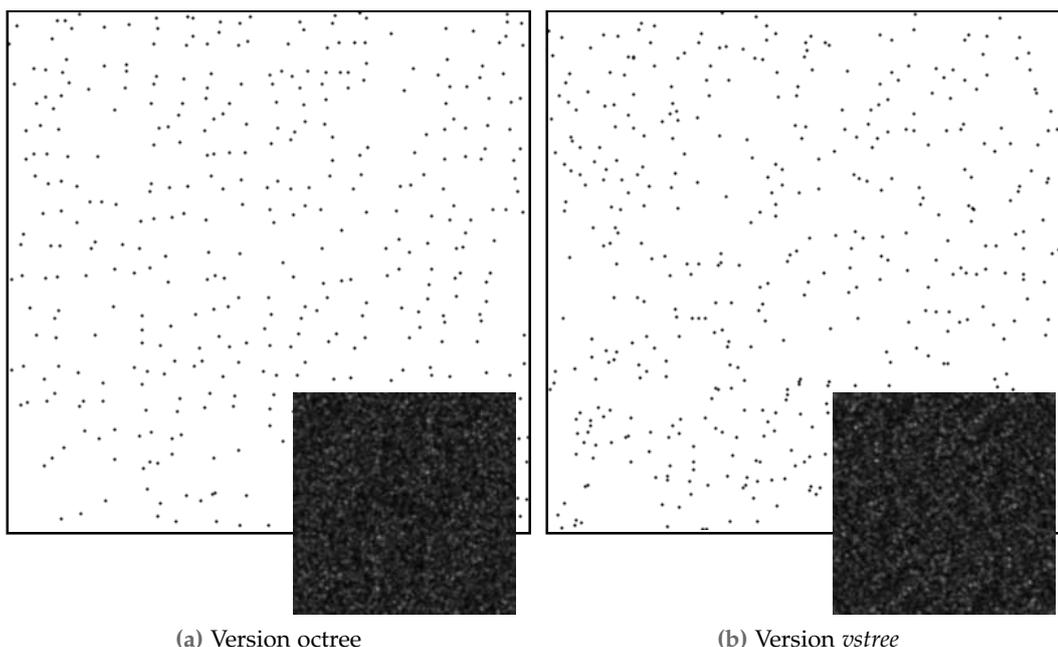


FIGURE 4.6 – Distributions et spectres de Fourier correspondants, calculés pour une distribution couvrant l'espace image.

(changement de la taille projeté de l'objet de huit), en réglant les paramètres de distribution (taille du masque ou seuil) pour obtenir une moyenne de 10 000 points par image de l'animation. Nous obtenons un temps moyen² de 300 milli-secondes pour la subdivision par tests d'occlusion, 69 milli-secondes pour la prédiction de densité par projection de sphère et 8 milli-secondes pour la méthode basée sur un *vs-tree*. Cette dernière offre des temps de distribution suffisamment performants pour être utilisée dans le cadre d'applications interactives.

4.2.4 Problème de densité

Comme le montrent les spectres de Fourier figure 4.6, les distributions obtenues présentent des alignements et ne correspondent pas aux critères de bruit bleu. Cela n'empêche pas de faire des rendus peinture où les marques sont opaques, mais limite l'extension à d'autres styles, où la distribution est plus fortement perçue. De plus, ces deux méthodes ne prennent pas en compte les densités variables en espace image, tout du moins pour un objet donné.

En effet, pour faire varier la densité, il faut descendre dans la hiérarchie, et pour accéder à une densité à un endroit de l'objet, nous devons passer par un certain chemin dans la hiérarchie qui raffine potentiellement d'autres parties de l'objet.

Projection de densité 2D/3D

La densité que nous voulons obtenir est donnée au moyen d'une fonction d'importance. Cette fonction est définie en espace image car c'est l'espace où l'on ob-

² Ces temps sont obtenus sur un PC sous GNU/LINUX Ubuntu 8.04 muni d'un Pentium 4 HT 3Ghz. La carte vidéo utilisée est une nVidia GeForce 7800 PCIE.

serve les points d'attache. En fait, les deux techniques que nous venons de présenter utilisent une fonction d'importance implicitement définie par la couverture des marques pour la première et une fonction d'importance uniforme pour la seconde (le rayon seuil ne change pas). Pour régler les problèmes de densité et de distribution, on pourrait *projeter* la fonction d'importance de l'espace image vers l'espace objet, puis distribuer les points d'attache suivant cette fonction d'importance sur les objets. La projection de la fonction d'importance n'est pas triviale. Mais même une fois projetée, pour obtenir une bonne distribution, par exemple bruit bleu, en espace image, il faudrait faire une distribution anisotrope en espace objet, et nous n'avons actuellement pas de solution pour faire ce genre de distribution.

4.3 Distribution en espace image

Notre deuxième méthode permet de distribuer des points d'attache avec les bonnes propriétés de bruit bleu en espace image. Pour cela nous distribuons les points directement en espace image et nous utilisons la scène sous-jacente (3D ou vidéo) pour déplacer les points. Nous projetons les points sur cette scène et les déplaçons suivant le mouvement donné. Puis nous mettons à jour la distribution pour tenir compte du déplacement des points et des changements de densité qui se sont produits. Au contraire des méthodes basées sur une hiérarchie, cette méthode n'est pas limitée aux objets statiques et peut facilement suivre tout type de déformations 3D.

4.3.1 Algorithme de distribution

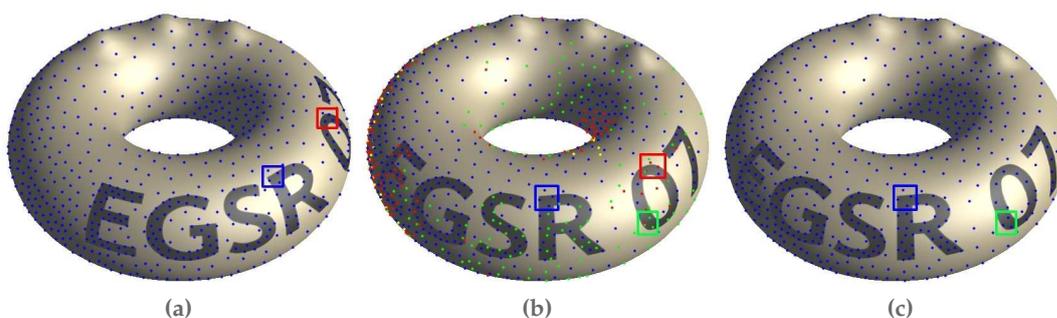


FIGURE 4.7 – (a) Les points sont initialement distribués en espace image, ici en utilisant l'intensité lumineuse pour définir la fonction d'importance. Ils sont ensuite projetés sur les objets 3D. (b) Les points sont ensuite déplacés en suivant le mouvement des objets, et reprojétés pour la frame suivante. Les points rouges ne respectent plus le critère du disque de Poisson, les jaunes sont cachés, les points bleus sont conservés et les points verts sont les nouveaux points ajoutés. (c) Une fois que l'on a fait la mise à jour, les points rouges, verts et bleus sont respectivement retirés, ajoutés ou conservés (mis en valeur par les différents cadres).

Comme nous faisons suivre un mouvement aux points d'attache et que nous voulons garder une bonne distribution, nous devons utiliser un algorithme de distribution qui permet la mise à jour d'un ensemble de points déjà distribués mais aussi la modification des points distribués. Cela élimine les techniques basées sur

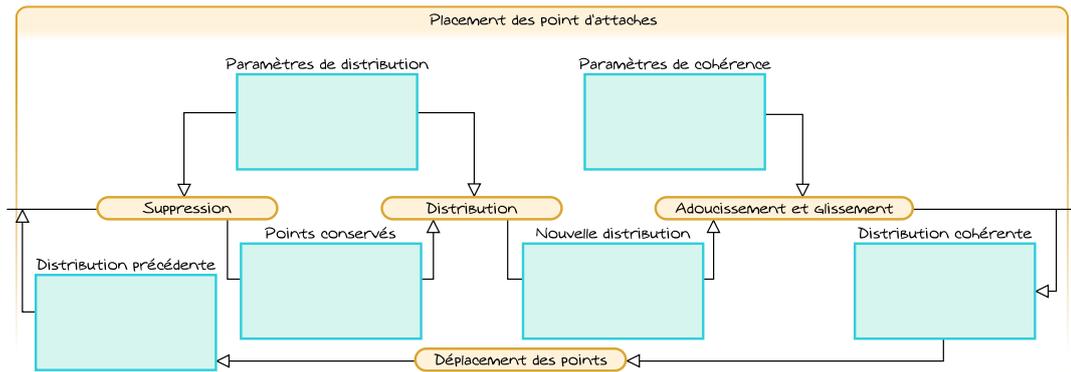


FIGURE 4.8 – Les différentes étapes de notre distribution dynamique : suppression, distribution, lissage et glissement. Nous mettons à jour une distribution dynamique non-uniforme de manière cohérente d’une frame à l’autre, pour tout type de mouvement en entrée : les objets 3D peuvent être déformés, le point de vue est totalement libre, et les mouvements présents dans une vidéo peuvent être suivis grâce à l’utilisation du flux optique.

des précalculs de patch bien distribués car elles n’offrent pas de moyen de mise à jour locale. Les relaxations, qui déplacent les points distribués, introduisent des mouvements parasites. Les méthodes basées sur le lancer de disque semblent, quant à elles, offrir tous les avantages dont nous avons besoin.

Nous avons choisi de fonder notre distribution dynamique sur l’algorithme de distribution de Poisson statique proposé par McCool et Fiume [MF92] qui est le mieux adapté aux contraintes particulières de notre problème.

Dans la suite de cette section, nous présentons comment nous avons étendu cet algorithme pour traiter la partie dynamique. L’idée principale, présentée figure 4.8, est qu’entre chaque image de l’animation, les points d’attache sont déplacés suivant le mouvement donné avec la scène en entrée. Ensuite, les points ne respectant plus le critère de densité sont supprimés et les trous formés lors du déplacement sont comblés en ajoutant de nouveaux points d’attache. Aussi, nous conservons les bonnes qualités de la méthode statique dans cette extension dynamique.

Distribution statique

L’approche de McCool et Fiume [MF92] à l’avantage d’avoir de bonnes propriétés statistiques de distribution (comme nous l’avons vu section 4.1.1) et de plus, d’être simple à implémenter, est bien adaptée à notre problème : elle permet d’avoir des densités non-uniformes et utilise une distribution hiérarchique qui peut être construite incrémentalement. En particulier la densité de points peut suivre une fonction d’importance I arbitraire, définie en espace écran, en utilisant un critère d’importance. Le point p est conservé seulement si $1 - \frac{r(p) - r_{\min}}{r_{\max} - r_{\min}} \leq I(p)$, avec $r(p) \in [r_{\min}, r_{\max}]$ le rayon avec lequel il a été distribué. De cette manière, l’artiste peut contrôler la distribution de points en réglant les paramètres r_{\min} et r_{\max} .

Distribution dynamique

Nous avons expliqué comment les points d'attache sont distribués pour une seule image de l'animation. Nous commençons toute distribution dynamique par une telle distribution sur la première image (figure 4.7a). Ensuite, nous *déplaçons* les points vers l'image suivante. Ceci provoque des agglomérations de points que nous enlevons en *supprimant* les points non désirés ; mais cela crée aussi des trous que nous remplissons en *distribuant* de nouveaux points.

En fonction de l'entrée du système, les possibilités pour le mouvement sont différentes. Pour gérer les mouvements complexes présents dans les scènes 3D, nous projetons les points d'attache distribués en espace image sur la surface des objets 3D, nous animons les objets (en suivant l'animation présente dans la scène, y compris les déformations et le *skinning*) et la caméra, et nous re-projetons les points d'attache de l'espace objet sur l'espace image. En pratique, nous prenons en entrée des maillages triangulés. La projection sur la surface s'effectue par l'intermédiaire d'une texture d'identifiant générée pour chaque image de l'animation. Cette texture donne l'objet, le triangle, et les coordonnées barycentriques dans ce triangle pour chaque pixel de l'espace image. Ensuite, pour projeter un point d'attache distribué en espace image, nous récupérons le contenu de cette texture et pouvons ainsi déterminer la position 3D et l'objet sur lequel il est projeté, par une simple lecture dans cette texture. Pour déplacer le point d'attache à sa nouvelle position dans l'image suivante, nous appliquons au point 3D la déformation appliquée au triangle auquel il est attaché (en interpolant les transformations des sommets suivant les coordonnées barycentriques) et nous projetons le point 3D suivant la transformation perspective donnée par la caméra. Cette approche permet aux points d'attache de rester collés à leur triangle, même s'ils sont très déformés. Pour les vidéos, nous faisons simplement suivre au point le mouvement défini par un flux optique.

Un autre aspect important du mouvement est l'occlusion : pour les vidéos, les points sont considérés comme cachés uniquement lorsqu'ils sortent de l'espace image, mais dans le cas d'une scène 3D, ils peuvent disparaître derrière la surface des objets. Nous faisons un simple test de profondeur pour détecter les occlusions et marquons comme cachés les points qui seront traités de manière spécifique à l'étape suivante. De plus, les points distribués sur le fond (c'est-à-dire sur aucun objet) peuvent être considérés soit cachés pour ne placer des marques que sur les objets, soit visibles pour remplir l'espace image. La figure 4.7b montre les points après déplacement, les points cachés sont en jaunes.

Dans les vidéos et les scènes 3D, les points sont soit visibles, soit cachés après qu'ils aient été déplacés. Dans la suite de ce paragraphe, nous considérons uniquement les points qui sont encore visibles. Pour pouvoir supprimer les agglomérations de points, nous réinsérons les points conservés dans l'ordre où ils ont été distribués précédemment, en utilisant le critère de disque de Poisson pour rejeter les points non voulus. Pour chaque rayon r_L correspondant au niveau L de la distribution hiérarchique, nous testons si *tous* les points précédemment insérés respectent le critère d'importance. Les points qui passent le test avec succès sont réinsérés avec un rayon r_L et nous testons les points restant au niveau $L - 1$. Le procédé s'arrête dès que tous les points ont été réinsérés ou que le niveau le plus bas de la hiérarchie (correspondant à r_{min}) est traité. Les points rejetés sont marqués comme supprimés

(points rouges, figure 4.7), tandis que les points acceptés (en bleu) sont conservés, ce qui rend la distribution cohérente d'une image à l'autre.

Enfin, il peut subsister des trous dans la distribution, dus à des apparitions d'objets et des dilatations du flux optique. Nous devons donc distribuer de nouveaux points pour remplir ces trous, avec les mêmes propriétés de distribution que dans le cas statique. En considérant *tous* les points réinsérés comme faisant partie de la distribution courante, nous réutilisons l'algorithme de McCool et Fiume : chaque niveau L de la distribution hiérarchique est traversé de haut en bas ; des points sont tirés aléatoirement avec un rayon r_L , jusqu'à ce qu'il n'y ait plus de place trouvée après n essais, et nous ne conservons pas les points qui ne respectent pas le critère d'importance. Le procédé s'arrête quand le niveau le plus fin, correspondant à $r_L \leq r_{\min}$, a été traité. Cette approche permet aux nouveaux points d'être placés le plus haut possible dans la hiérarchie, et donc d'offrir une meilleure répartition.

4.3.2 Cohérence temporelle

L'algorithme que nous avons décrit garde la plupart des points d'une image à l'autre. Cependant nous ne pouvons pas l'empêcher d'insérer et de supprimer *quelques* points, le mouvement et la densité étant des contraintes contradictoires. C'est pourquoi nous proposons deux mécanismes permettant d'améliorer la cohérence temporelle : les glissements et l'adoucissement.

Adoucissement: Ajouter ou supprimer des points, et donc les marques correspondantes, dès que notre algorithme de distribution le demande, provoque des changements brutaux. Comme dans les travaux précédents, nous adoucissons ces transitions en faisant apparaître ou disparaître progressivement les marques associées aux points d'attache. Nous associons une durée de transition à chaque point d'attache qui vient d'apparaître ou de disparaître. Nous utilisons alors le pas de temps courant, le pas de temps d'apparition et le temps de transition pour obtenir une valeur d'adoucissement. Cette valeur est utilisée dans l'étape de dessin (voir section 5.3.5) pour assurer des transitions cohérentes dans le temps.

La principale différence avec les approches précédentes est que nous calculons uniquement le temps de transition, et l'utilisation de cette valeur est laissée à la partie rendu, qui peut l'utiliser de différentes manières pour faire apparaître et disparaître les marques en conservant les aspects visuels propres aux différents styles. De plus, comme nous détectons différents types d'apparition et de disparition, nous offrons la possibilité de régler différemment les temps de transition selon si l'évènement est dû au critère de densité ou à l'occlusion. Pour les événements d'occlusion, les temps de transition peuvent être plus courts, dans le cas contraire, on risque de laisser des blancs lorsqu'un objet dévoile une partie cachée, et des traces fantômes lorsqu'un objet disparaît. En pratique, trouver les points qui sont occultés est facile (avec le tampon de profondeur), mais pour les points qui apparaissent par désocclusion, nous devons stocker la scène (c'est-à-dire la position 3D des triangles, la caméra ainsi que le tampon de profondeur) dans l'état de l'image précédente. Ainsi nous pouvons calculer la position qu'un point d'attache nouvellement inséré aurait eu dans l'image précédente s'il était déjà présent, et faire le test de profondeur pour

savoir s'il aurait été caché. Concernant les points insérés et supprimés par densité, les transitions peuvent être plus longues sans produire ni sauts ni trous dans l'animation. L'artiste doit cependant les régler en fonction du style et de la quantité de mouvement présente dans l'animation.

Glissement : Un autre problème peut apparaître lorsque l'on supprime un point qui ne respecte plus le critère de disque de Poisson, et qu'il laisse suffisamment de place pour ajouter un autre point dans son voisinage : même avec des transitions douces, cela peut introduire des clignotements durant l'animation. Nous gérons ce problème en construisant une paire (p_s, p_i) constituée d'un point supprimé et un point ajouté, p_s étant le point supprimé le plus proche de p_i en espace image, et on utilise une valeur de seuil définie par l'utilisateur pour la distance maximale. Ensuite, au lieu de supprimer p_s , nous le faisons glisser linéairement vers la position du point nouvellement inséré. Nous réalisons cette opération seulement pour les ajouts et suppressions dus à la densité, sinon les points cachés auraient tendance à glisser au travers des frontières d'occlusions.

En pratique, nous calculons l'évolution des positions des points p_i et p_s sur la surface des objets, et interpolons linéairement entre leurs positions projetées. Ceci permet d'obtenir une distance $d_p(t)$ entre le point en train de glisser et sa position cible, avec $t \in [0, t_g]$, où t_g est la durée du glissement fournie par l'utilisateur.

Cette approche fonctionne bien pour des mouvements simples, mais des mouvements plus complexes comme des déformations ou des rotations peuvent augmenter la distance en espace image entre p_i et p_s et donc augmenter $d_p(t)$ ce qui est l'opposé de l'effet recherché. C'est pourquoi nous restreignons $d_p(t)$ à la distance qui serait obtenue avec l'interpolation linéaire $d_p(0)(t - t_g)/t_g$. De cette manière le point interpolé avance progressivement vers p_i tout en suivant le mouvement de l'objet auquel il est attaché.

4.3.3 Résultats

La figure 4.9 montre les distributions obtenues avec notre algorithme sur une image fixe. Le temps de distribution des points d'attache est indépendant de la complexité de la scène, et n'a pas besoin de précalcul. Les performances dépendent uniquement des dimensions de l'image et des choix pour r_{min} , r_{max} , n et α qui déterminent la densité de point. Comme dans la méthode originelle, nous utilisons une grille uniforme en espace image pour accélérer les tests de distance durant la distribution. Pour une manipulation interactive ($n = 100$, $\alpha = 0.9$) nous obtenons entre 10 et 30 images par secondes, tandis que pour du rendu off-line ($n = 1000$, $\alpha = 0.99$), le temps de distributions est d'environ une seconde. Ces paramètres de qualité sont ceux donnés par McCool et Fiume, et dans les deux cas, nous utilisons des résolutions de 740×540 et un rayon allant de 50 à 2 pixels.

Nous présentons maintenant les évaluations statistiques et visuelles de notre algorithme de distribution en comparaison des travaux précédents.

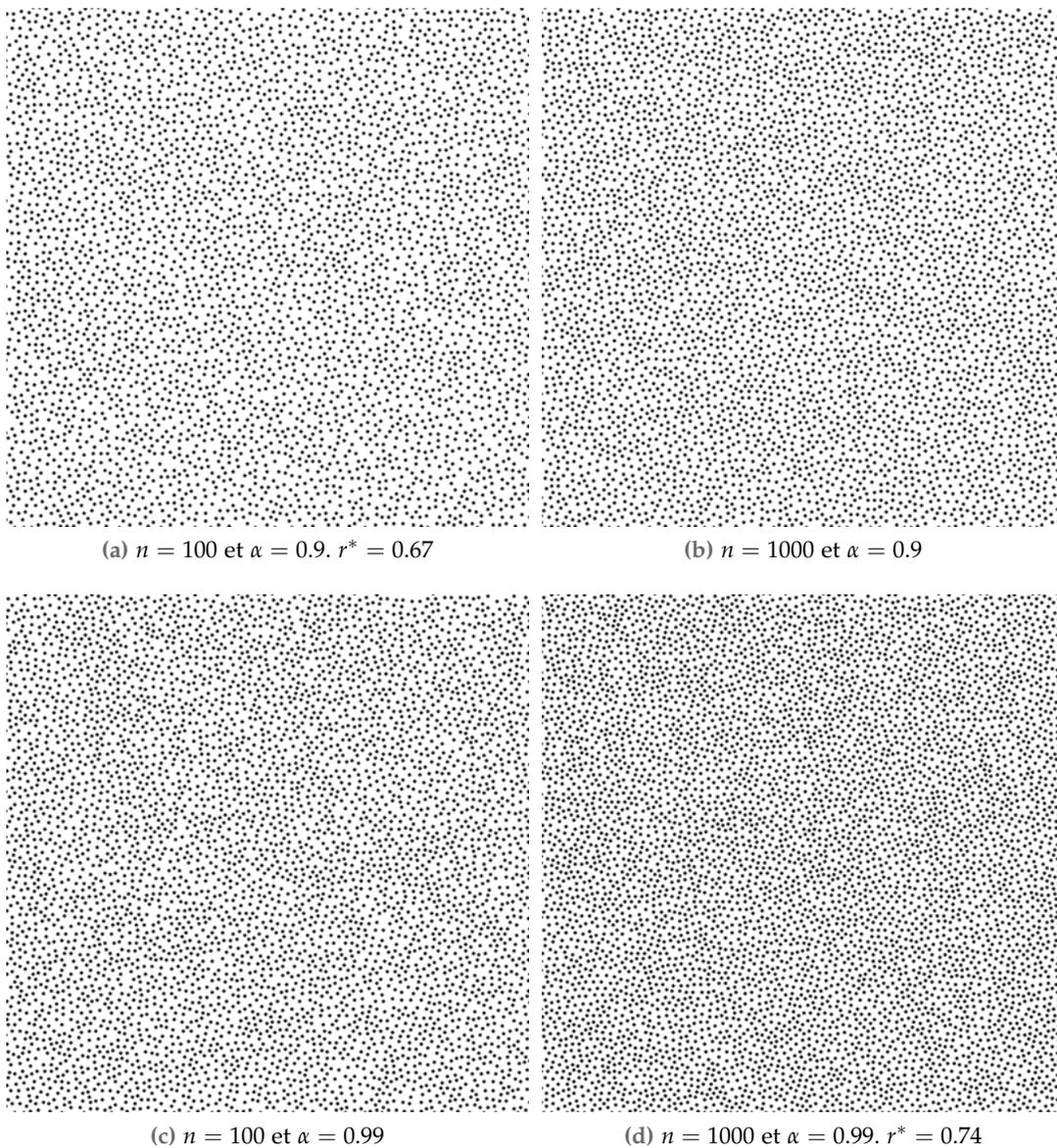


FIGURE 4.9 – Distributions de points d'attache obtenus avec notre algorithme. Notons l'influence des paramètres n et α sur la qualité de la distribution, notamment sur le rayon relatif r^* .

Évaluation statistique

Dans un premier temps, nous vérifions que nous obtenons les mêmes qualités de distribution que l'algorithme de McCool et Fiume.

Suivant les mesures données par le rayon relatif et le spectre de puissance — qui sont uniquement valides pour des densités *uniformes* de distribution — la méthode de McCool et Fiume est l'une des meilleures méthodes disponibles. Nous n'avons pas trouvé d'évaluation similaire pour les distributions non-uniformes.

Nous avons mesuré le spectre de puissance et le rayon relatif des images d'une animation de test en utilisant le modèle de la figure 4.7 et une distribution dynamique uniforme. Nous nous attendions à obtenir un rayon relatif moins bon, et des fréquences moins bien réparties, en raison de l'ajout de la cohérence temporelle.

Mais nous n'avons finalement pas noté d'influence significative : le spectre de puissance (figure 4.10) fait clairement apparaître les caractéristiques de bruit bleu et le rayon relatif est légèrement en dessous de la distribution statique.

Ces mesures ne sont pas meilleures que celles obtenues dans le cas de distributions de Poisson statiques, mais montrent que notre méthode permet une qualité comparable de distribution, tout en respectant d'autres contraintes : le suivi du mouvement et la cohérence temporelle. La qualité de distribution, en comparaison avec les méthodes objet, est bien meilleure, les propriétés de bruit bleu sont respectées et nous n'avons pas les problèmes d'alignements et d'anisotropies dues à la projection perspective.

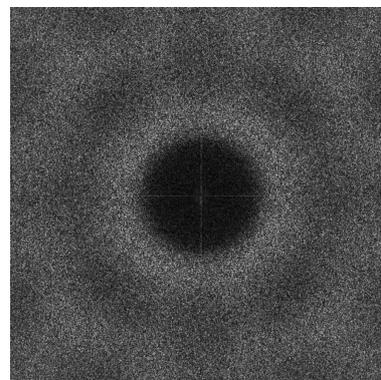


FIG. 4.10 Spectre de Fourier moyenné sur une animation complète

Évaluation visuelle

La section précédente compare notre approche aux techniques de distributions uniformes et statiques. Cependant, une telle comparaison n'est pas valable pour tous les travaux précédents, notamment pour des distributions non-uniformes ou pour les méthodes qui traitent de la cohérence temporelle. Nous nous basons donc sur une évaluation visuelle pour nous comparer avec les méthodes objets et illustrer le comportement avec des distributions non-uniformes. Le travail de Meier [Mei96] produit des distributions remarquablement stables. Cela vient du fait que la distribution est statique et ne s'adapte ni au point de vue ni à l'animation des objets. La distribution est réalisée une fois pour toute en précalcul. Une approche similaire en espace image en utilisant une distribution 2D statique produirait l'effet *rideau de douche*. La méthode de Meier donne de bien meilleurs résultats, bien que l'espace de point de vue valide — pour une distribution ni trop compacte ni trop dispersée — soit limité.

Cependant, dès que l'on veut ajouter ou supprimer les points d'attache pour enlever les contraintes de déplacement de la caméra et d'animation des objets, on perd la stabilité et on doit adoucir les changements brutaux. Ce problème est commun aux approches espace-objet et espace-image. Dans le premier cas, les points sont distribués à la surface des objets et sélectionnés suivant une fonction de raffinement, tandis que dans le deuxième cas, les points sont ajoutés ou supprimés suivant un critère de qualité de distribution (critère de disque de Poisson). Bien que les deux approches partagent des problèmes similaires, la fonction de raffinement est primordiale pour la qualité de la technique. Comme nous l'avons expliqué section 4.2.4 et comme le montre la figure 3.7, il n'y a pas de moyen simple d'obtenir une bonne distribution 2D en partant d'une distribution 3D, même après une randomisation comme celle proposée par Meruvia-Pastor et coll. [PFS03]. Par contre dans notre méthode en espace image, comme le montre la figure 4.11, nous supprimons le problème en utilisant directement une distribution en espace image.

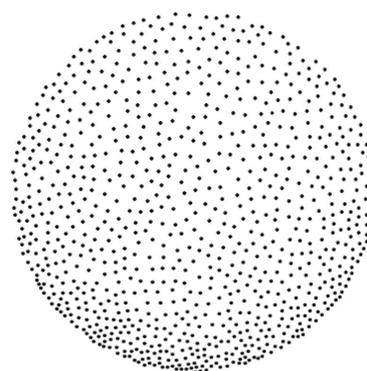


FIG. 4.11 Rendu stippling obtenue avec notre méthode, à comparer avec la figure 3.7.

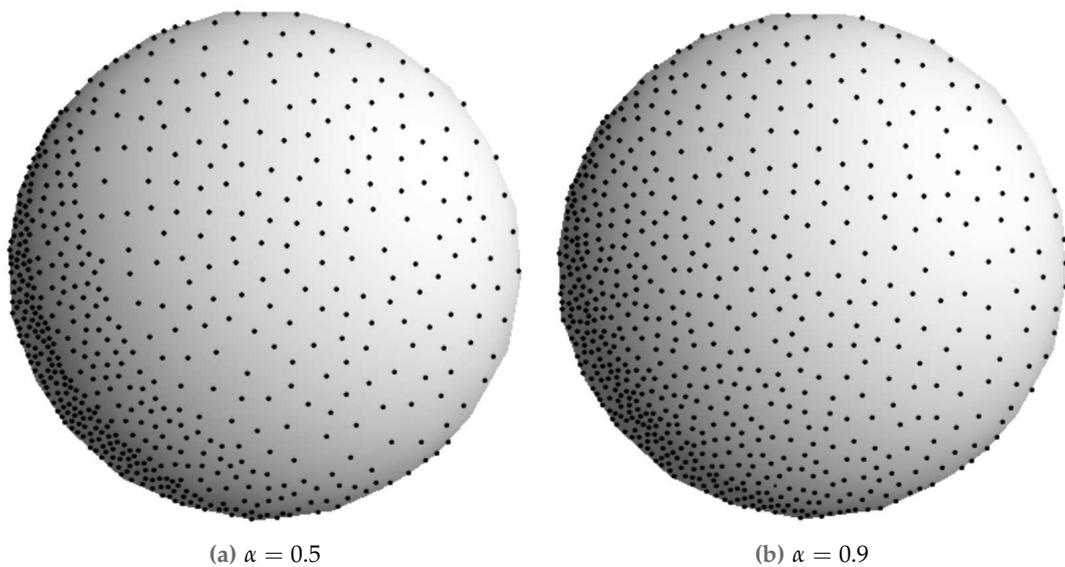


FIGURE 4.12 – Distribution non-uniforme, on remarque l'influence de α sur l'effet de quantification des niveau de densité

Les distributions non-uniformes créées avec notre approche dépendent des valeurs de r_{\min} , r_{\max} et α . r_{\min} et r_{\max} contrôlent la densité des régions où l'importance est respectivement 0 et 1, la valeur de α influence la continuité de la distribution non-uniforme. Comme le montre la figure 4.12, une petite valeur pour α produit un effet de quantification, alors qu'une valeur plus grande adoucit les changements de densité. Dans la pratique, une valeur d' $\alpha > 0.9$ donne de très bons résultats dans la majorité de cas (nous n'avons pas trouvé de cas où cela n'est pas suffisant). Contrairement à l'approche de McCool et Fiume, nous ne réalisons pas de relaxation de Lloyd sur les points distribués. Cela diminue légèrement la qualité du rayon relatif, mais nous gagnons du point de vue du suivi du mouvement en entrée (sinon tous les points glisseraient sur la surface des objets). Nous avons, en effet, préféré localiser les glissements lorsque cela permet d'éviter un ajout et une suppression de points d'attache.

Pipeline de rendu animé stylisé

Nous présentons ici le fonctionnement général de notre approche pour le rendu d'animations à base de marques. Nous décrivons, dans un premier temps, le pipeline que nous utilisons (section 5.1). Celui-ci est une extension de celui présenté dans le chapitre précédent. Nous définissons, ensuite, notre vision du style dans le cadre du rendu d'animation à base de marques (section 5.2), ainsi que les moyens de contrôle que nous offrons à l'artiste pour personnaliser le rendu effectué par notre système (section 5.3). Enfin nous montrons quelques résultats qui illustrent les possibilités de notre définition du style (section 5.4).

5.1 Pipeline d'utilisation

Le pipeline général pour le rendu à base de marques que nous proposons reprend la structure principale de celui présenté figure 3.5 mais adaptés en fonction de nos choix. Ce pipeline permet d'identifier les différentes actions constituant le rendu à base de marque et donc d'explicitier les points d'entrée sur lesquels l'artiste va pouvoir agir.

La figure 5.1 présente une vue d'ensemble du pipeline. nous détaillons les différents moyens de contrôle dans les sections suivantes. À partir de la scène, les propriétés sont extraites de la scène en effectuant les calculs *ad-hoc*; elles sont projetées en espace image dans des *G-buffers*. Ceux-ci sont traités, par exemple par filtrage, avant de devenir les entrées de la fonction de transfert \mathcal{S} . L'application de \mathcal{S} permet d'obtenir les attributs des marques, sous forme d'images de référence (comme dans l'approche de Meier [Mei96]). Ces images sont, elles aussi, traitées pour obtenir les attributs finaux qui seront utilisés par la fonction de dessin des marques. Un autre processus utilise la scène et l'attribut d'importance calculé par \mathcal{S} et traité, pour placer ou déplacer les points d'attache (typiquement selon une des méthodes décrites au chapitre 4). Ensuite, les attributs sont extraits des images de référence aux positions des points d'attache. Ceci permet d'obtenir les marques avec leurs attributs, qui sont ordonnées, puis, dessinées sur le support.

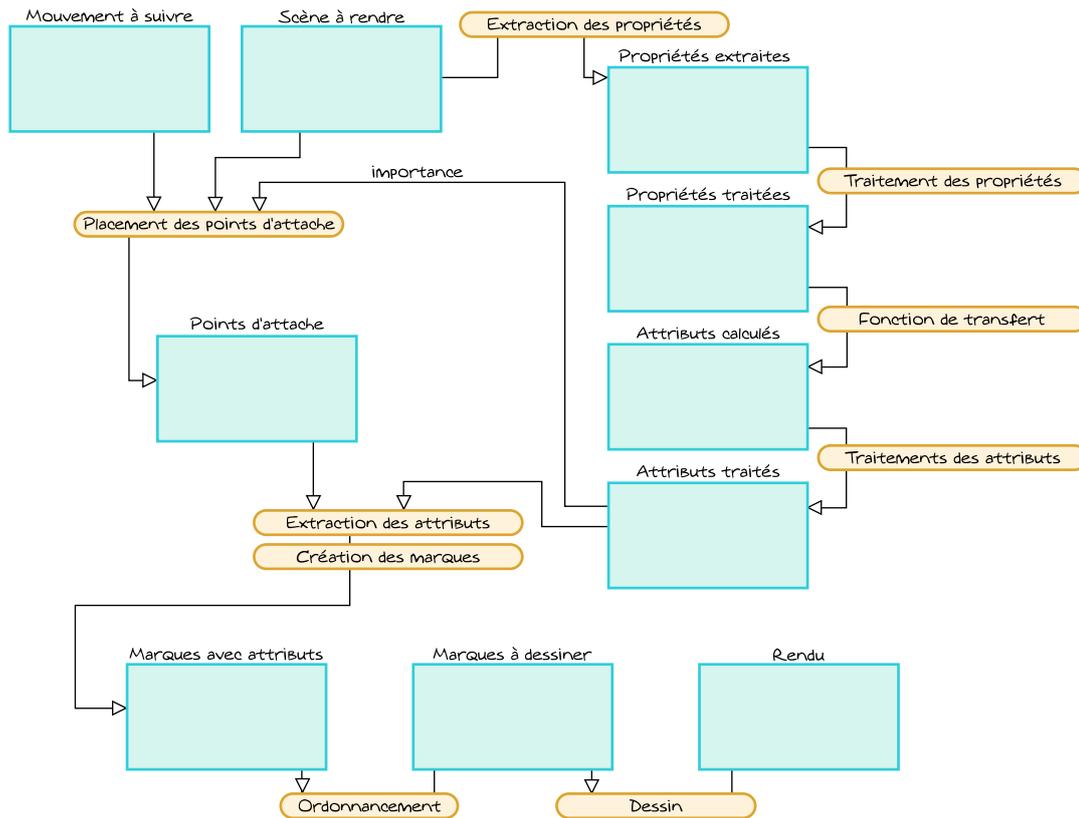


FIGURE 5.1 – Pipeline de rendu

Nous avons implémenté l'intégralité de ce pipeline. De cette manière nous avons pu tester notre approche pour le contrôle du style et pour la distribution des marques pour la réalisation d'un court métrage, présenté section 5.4.

Nous allons décrire dans un premier temps les actions s'effectuant sans la connaissance des positions des points d'attache (section 5.2), puis celle prenant en compte les points d'attache pour dessiner les marques (section 5.3).

5.2 Le style comme une fonction de transfert

Pour qu'un logiciel de rendu soit utilisable dans le cadre de production d'image, l'artiste doit pouvoir contrôler le rendu obtenu. Pour le rendu réaliste par exemple, le contrôle se fait au moyen de la définition des matériaux et de l'éclairage. Dans le cas du dessin au trait, pour une image fixe, Grabli et coll. [GTDS04] proposent une approche programmable pour la stylisation de lignes. Un style est présenté sous la forme d'un petit programme qui décrit comment obtenir certains paramètres du rendu des lignes. L'attrait principale de cette approche est que les styles possibles ne sont pas fortement contraints par les choix des concepteurs et que les artistes peuvent expérimenter de nouveaux styles pour chaque projet. D'ailleurs, les logiciels commerciaux comme Maya et RenderMan utilisent ce type d'approche pour la définition des matériaux des objets 3D. Une fois qu'un style est écrit, il est alors facile de le réutiliser pour d'autres scènes. Nous avons choisi de suivre nous aussi une approche programmable. Pour cela, nous donc devons définir quelles sont les

entrées et sorties du contrôle de style pour le remplissage de régions dans le cadre d'animations stylisées.

Pour nous, le style est l'ensemble des actions qui permettent de déterminer le dessin des marques à partir de la scène en entrée, et en particulier les attributs des marques. Notre hypothèse est que ces attributs sont liés aux propriétés de la scène en entrée. Nous représentons ce lien par une fonction de transfert \mathcal{S} des propriétés de la scène vers les attributs des marques.

5.2.1 Fonction de transfert

La fonction de transfert \mathcal{S} est l'association que l'artiste définit entre les propriétés de la scène et les attributs des marques.

Propriétés de la scène

Plusieurs propriétés peuvent être extraites d'une scène animée. Tout d'abord, les images constituant l'entrée donnent la propriété de couleur. Ensuite, il y a le mouvement présent dans l'animation en espace image (flux optique, flux de mouvement) ou en espace objet, c'est-à-dire le mouvement de la surface des modèles 3D. Enfin, plus particulièrement dans le cas d'une scène 3D, les propriétés géométriques des objets peuvent être extraites de la scène, telles que la normale des surfaces, la courbure, la distance au point de vue, et la position des objets dans l'espace image.

Si l'entrée est une vidéo, ces propriétés sont naturellement définies en espace image. Dans le cas 3D, les propriétés peuvent être soit définies en espace objet, soit projetées en espace image dans des textures appelées *G-buffers*. Chaque pixel des *G-buffers* représente alors les valeurs des propriétés de la scène se trouvant sous ce pixel. Les *G-buffers* ont été introduit par Saito et Takahashi [ST90] pour appliquer des traitements en espace image sur un rendu 3D (ajout de contour par filtrage de la profondeur par exemple). Dans notre implémentation, nous avons choisi d'utiliser des *G-buffers*, ainsi la différence entre vidéo et scène 3D n'est pas fondamentale.

Attributs des marques

Une marque est définie par un ensemble d'attributs de bas niveau permettant de la dessiner lors du rendu. Dans nos travaux, nous avons considéré les marques basées sur des points d'attache avec les attributs suivants : couleur, transparence et texture pour l'apparence ; longueur, épaisseur, orientation et courbure pour la forme. La figure 5.2 montre la construction géométrique d'une marque avec ces attributs : les attributs définissent un squelette que nous couvrons avec une texture (suivant l'épaisseur) pour donner l'apparence de la marque.

Outre les attributs de bas niveau, on utilise un attribut de haut niveau permettant de guider le placement des marques : la densité des points d'attache. Cette densité est définie comme une fonction d'importance et donne localement la densité de marques souhaitée pour l'étape de distribution.

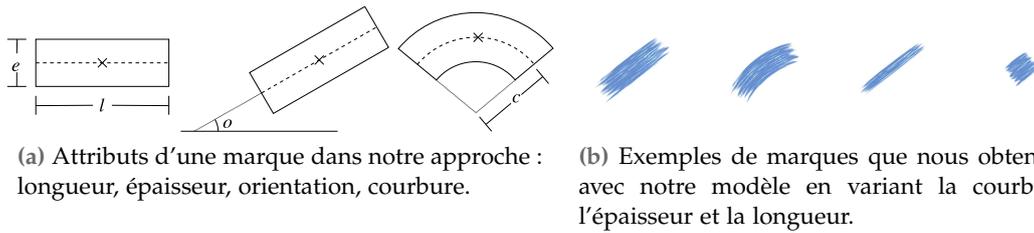


FIGURE 5.2

Pour le dessin des marques, nous proposons un attribut de fidélité. La fidélité représente l'écart que l'artiste autorise entre les valeurs des propriétés de la scène sous le point d'attache et les valeurs de propriétés couvertes par le squelette de la marque. Dans notre implémentation, une fidélité est associée à la propriété de couleur et de profondeur. Par exemple, l'artiste peut autoriser ou non une marque dont le point d'attache est placé sur une région bleue à « dépasser » sur une région rouge. Nous définissons la fidélité comme la différence maximum autorisée. Nous présentons plus en détail comment nous avons implémenté la restriction d'une marque par la fidélité section 5.3.4.

Lien entre propriétés et attributs

La fonction de transfert \mathcal{S} entre les n propriétés de la scène et les m attributs des marques est définie par :

$$\mathcal{S} : \begin{cases} \mathbb{R}^n & \rightarrow \mathbb{R}^m \\ p & \mapsto a \end{cases}$$

où $p = \{p_c, p_z, p_r, p_n, p_f\}$ est le vecteur des propriétés (couleur, profondeur, courbure, normale, flux), $a = \{a_c, a_\alpha, a_l, a_e, a_o, a_c, a_l, a_f\}$ le vecteur des attributs de la marque (couleur, transparence, longueur, épaisseur, orientation, courbure, importance, fidélité). Cette fonction est définie *a priori* sans savoir où sont placés les points d'attache.

La fonction \mathcal{S} donne l'importance qui permet le placement des points d'attache. Puis pour chaque point d'attache, nous récupérons les valeurs de \mathcal{S} pour les positions effectivement utilisées. Dans nos travaux, nous avons choisi d'utiliser l'espace image pour le calcul des attributs, mais notre définition du style reste valide si on veut travailler en espace objet dans le cas d'une scène 3D.

La fonction de transfert \mathcal{S} permet par exemple de définir l'épaisseur d'une marque suivant la distance de l'objet au point de vue. Pour cela il suffit de lier l'épaisseur e à la propriété de profondeur z , par exemple :

$$a_e = 5(1 - p_z) + 3 \text{ pixels}$$

pour une épaisseur variant entre trois et huit pixels, z variant de zéro pour le plan le plus proche à 1 pour le plan le plus lointain. a_e représente la coordonnée de l'épaisseur du vecteur d'attributs, et de la même manière p_z est la propriété de profondeur.

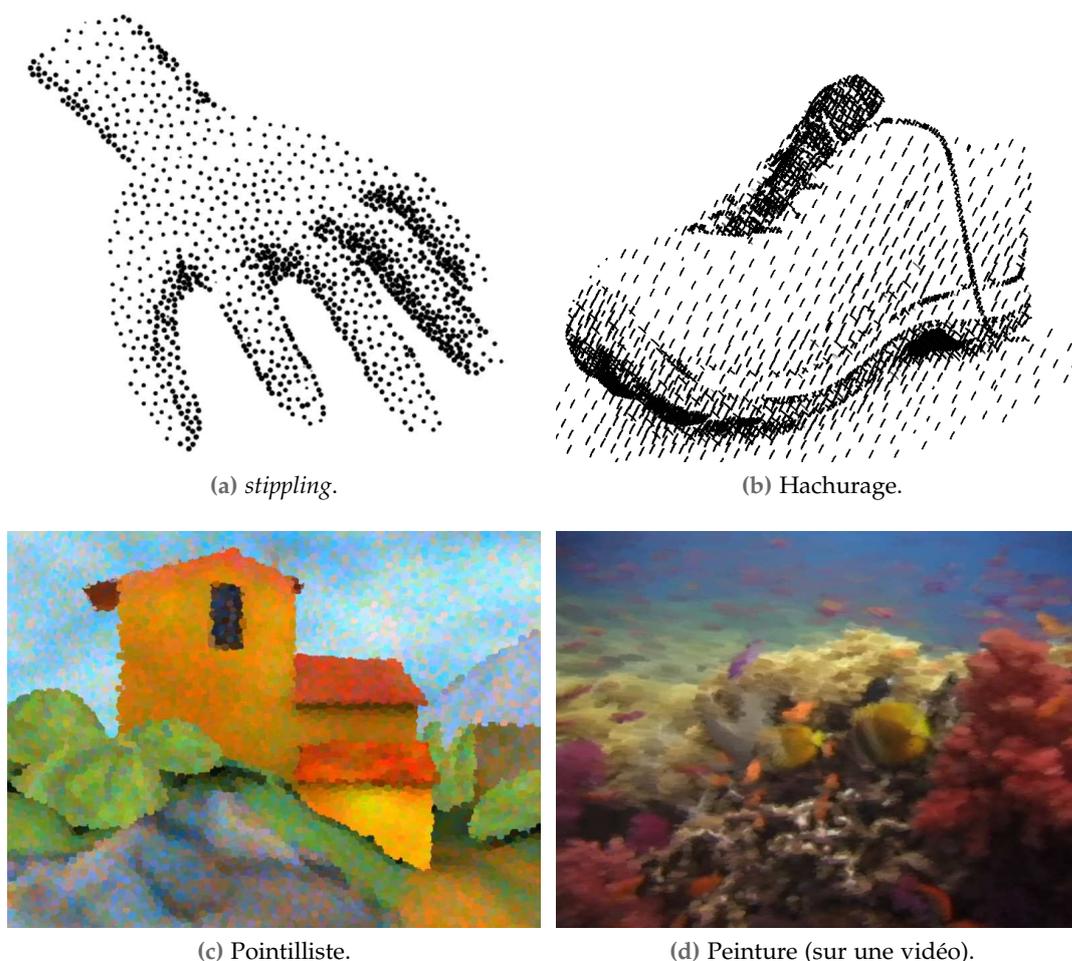


FIGURE 5.3 – Exemples de rendu avec notre définition du style sous forme de fonction de transfert. Les attributs des marques sont liés aux propriétés extraites de la scène (3D ou vidéo) par la fonction de transfert.

5.2.2 Exemples de fonctions de transfert

Lorsque l’artiste écrit une fonction de transfert, donne le calcul des attributs des marques en fonction des propriétés de la scène. Pour pouvoir faire la mise au point interactive du style, il est pratique de laisser des variables *accessibles*. L’artiste ajuste ensuite ces variables avec un retour visuel immédiat.

Nous montrons maintenant l’étendue de la palette des styles possibles avec cette définition. Comme nous l’avons introduit avec les figures 5.2, les marques que nous pouvons produire sont variées. Regardons maintenant des exemples de fonctions de transfert entre la scène et la marque. La figure 5.3 propose un rendu pour chacune de ces fonctions.

Stippling : Nous utilisons une approche similaire à celle présentée par Meruvia-Pastor et coll. [PFS03]. Nous associons à chaque point d’attache une marque formant un point d’encre. Les points d’encre sont simplement des disques noirs dont la taille varie de $TailleMin^1$ à $TailleMin + EchelleVariation$, $EchelleVariation$ donnant l’influence de l’intensité lumineuse $Intens \in [0, 1]$. La densité et la taille des points suivent l’intensité lumineuse de la scène.

1. Les noms en italique sont des variables accessibles, défini par l’artiste.

$$\begin{aligned}
 a_I &= 1 - \text{Intens}(p_c) \\
 a_e &= \text{TailleMin} + \text{EchelleVariation} \times (1 - \text{Intens}(p_c)) \\
 a_l &= \text{TailleMin} + \text{EchelleVariation} \times (1 - \text{Intens}(p_c)) \\
 a_c &= \text{noir}
 \end{aligned}$$

Pointillisme : Pour obtenir l'effet pointilliste, nous ajoutons la couleur au rendu *stippling* mais avec une fonction d'importance constante. La couleur provient de la scène sous-jacente et la densité des marques est cette fois-ci constante. La fonction de style perturbe la couleur des marques en ajoutant un terme aléatoire sur les trois composantes en espace de couleur Lab². Cet espace de couleur permet de moduler la variation aléatoire suivant la composante de luminosité et les composantes colorimétriques, tout en ayant une certaine consistance sur la force de la modulation.

$$\begin{aligned}
 a_I &= \text{Densite} \\
 a_e &= \text{Taille} \\
 a_l &= \text{Taille} \\
 a_c &= p_c
 \end{aligned}$$

Hachurage : Dans cet exemple nous avons choisi une longueur fixe, et deux orientations en espace image. La densité dépend de l'intensité de la scène et nous avons utilisé deux distributions de marques liées de telle sorte que les marques se croisent dans les zones les plus foncées. La première distribution place des marques dans les régions très sombre alors que la seconde couvre les partie claire, elle atteint sa densité maximale quand la première commence à apparaître.

première distribution	deuxième distribution
$a_I = 1 - \max(2\text{Intens}(p_c) - 1, 0)$	$a_I = \max(1 - 2\text{Intens}(p_c), 0)$
$a_e = 3$	$a_e = 3$
$a_l = 40$	$a_l = 40$
$a_O = 5\pi/6$	$a_O = \pi/3$
$a_c = \text{noir}$	$a_c = \text{noir}$

Peinture : Le rendu peinture tire parti de tous les attributs définis pour nos marques. Il est possible de faire varier ces attributs en fonctions des objets, de leur position et de la position des marques en espace image. La figure 5.3 montre un exemple rendu peinture que nous pouvons obtenir. L'entrée est une vidéo sous-marine, pour accentuer le détails sur les poissons et le fond marin, nous utilisons une importance en fonction de la couleur bleu, majoritairement présente sur le fond. De la même manière, les attributs de longueur et d'épaisseur sont choisis pour produire des coups de pinceau plus petits sur les régions d'intérêt que sur le fond. Enfin l'orientation suit le flux de mouvement en entrée, qui est le flux optique

2. L'espace utilisé est CIE L*a*b* 1976, abrégé Lab. Cet espace est construit expérimentalement pour définir des différences de couleurs (distances) consistantes. Pour plus d'informations sur les espaces de couleurs, on pourra se reporter au le livre de Fairchild [Fai05] par exemple.

```

vec3 propC = texture2D(proprieteCouleur, positionMarque).rgb;

float intens(vec3 c){
    return (0.299*c.r+0.587*c.g+0.114*c.b);
}

float importance(){
    return clamp(1.0-intens(propC), 0.0,1.0);
}

float epaisseur(){
    return TailleMin+EchelleVariation*(1-intens(propC));
}

float longueur(){
    return epaisseur();
}

vec3 couleur(){
    return vec3(0.0,0.0,0.0);
}

```

LISTING 5.1 – style *stippling*

```

// modifie par rapport au stippling
float longueur(){
    return 40;
}

float angle(){
    return PI/3.0;
}

```

LISTING 5.2 – style hachurage

dans cet exemple. La couleur est simplement prise directement de l'entrée.

$$\begin{aligned}
 a_I &= 1 - \text{Bleu}(p_e) \\
 a_e &= \text{Epaisseur} \times (0.6\text{Bleu}(p_e) + 0.4 \times \text{EpaisseurConst}); \\
 a_l &= \text{Longueur} \times (0.6\text{Bleu}(p_e) + 0.4 \times \text{LongueurConst}); \\
 a_O &= \arccos\left(\frac{p_f}{\|p_f\|}\right) \\
 a_C &= p_e
 \end{aligned}$$

5.2.3 Implémentation

La fonction de transfert est implémentée sous la forme d'un programme décrivant le calcul à effectuer pour chacun des attributs des marques. Les listings 5.1 et 5.2 donnent une idée de la forme de ces programmes.

5.2.4 Pré et post-traitements

Le point crucial du style est la définition de la fonction de transfert S . En effet, c'est cette fonction qui permet de calculer les attributs des marques et par consé-

quent donne en grande partie l'apparence des marques obtenues. Notre pipeline offre néanmoins d'autres points d'entrée pour l'artiste. En particulier, il peut ajouter un pré-traitement des propriétés et un post-traitement des attributs pour tenir compte de leur contexte spatio-temporelle.

La fonction de transfert calcule les attributs en tout point de l'image, mais elle ne connaît pas le contexte local (espace-temps) du vecteur p des propriétés de la scène sur lequel elle est appliquée. Pour tenir compte de ce contexte, l'artiste peut filtrer les propriétés de la scène (*G-buffers*), ainsi que les attributs calculés (images de référence).

Les traitements applicables sur les images de référence et les *G-Buffers* sont principalement des filtrages spatio-temporels. Suivant les choix de style de l'artiste, il peut être utile de lisser les propriétés de la scène ou les attributs (flou Gaussien), de les segmenter en conservant les discontinuités (filtre bilatéral) ou encore de réduire les valeurs possibles (quantification).

Ces filtres peuvent prendre en compte la dimension temporelle, ce qui améliore la continuité des propriétés traitées au cours de l'animation, et donc des attributs des marques. Lors de l'application de ces filtres, nous devons prendre en compte le type de données à filtrer. En effet, un flou ne s'applique pas de la même manière sur une couleur que sur un angle d'orientation par exemple. Bien que nous n'ayons pas exploré l'étendue complète des traitements possibles, nous présentons ici quelques résultats.

Prenons l'exemple d'une quantification de l'attribut de couleur des marques. Nous pouvons l'appliquer en soit en pré-traitement sur la propriété de couleur de la scène et lié cette couleur à la couleur de marque, soit par la fonction de style, soit en post-traitement sur les attributs calculé. Dans le premier cas la fonction de style n'a pas connaissance de la couleur originelle de la scène, les attributs dépendant de la couleur sont alors *tous* quantifier (par exemple si l'attribut de longueur est défini suivant la couleur, il sera aussi quantifier). Dans le cas où c'est la fonction de style qui effectue la quantification, il faut choisir si on considère la propriété de la scène ou l'attribut de marque comme donnée à quantifier. Enfin si la quantification est effectuée en post-traitement, c'est l'attribut de couleur qui est quantifier. Nous perdons alors les variations aléatoires qu'aurait pu introduire la fonction de transfert. L'intérêt d'utiliser un pré ou un post-traitement est de pouvoir prendre en compte la dimension temporelle. En effet il est plus pratique de filtrer temporellement une suite d'images (comme les propriétés et les attributs sont représenté dans des tableaux 2D de valeurs) que la fonction de transfert.

Nous n'avons pas particulièrement exploré l'étendue des possibilité, mais l'artiste peut utiliser tous les filtres répondant à son besoin : par exemple, pour supprimer les détails dans une vidéo, Bousseau et coll. [BNTS07] proposent une adaptation du filtre morphologique. Un tel filtre est facilement applicable aux propriétés extraites de la scène (de couleur par exemple).

5.3 Dessin des marques

Les étapes du contrôle du style et du calcul des attributs des marques s'effectuent sans savoir où se trouvent les marques. Ces étapes fournissent en particulier

la fonction d'importance qui guide le placement des points d'attache. Une fois que nous avons distribué les points d'attache, nous affectons les attributs aux marques correspondantes (une marque est attachée à chaque point d'attache).

5.3.1 Utilisation d'image de références, dans la pratique

Il faut prendre un certain soin pour lire les attributs des marques dans les images de référence. En effet, dans le cas d'une scène 3D, nous voulons que lorsqu'un point d'attache est déplacé suivant le mouvement d'un objet 3D, les valeurs lues après déplacement correspondent bien à celles calculées à partir de l'objet auquel il est attaché. Or, les problèmes d'aliasage et de précision numérique font que la nouvelle position du point d'attache ne tombe pas forcément sur un pixel où sa face est dessinée. Nous garantissons que les propriétés utilisées sont valides pour le calcul des attributs des marques : soit elles sont effectivement lues dans les images de référence si le pixel correspond bien à l'objet sur lequel le point d'attache est placé, soit les valeurs sont conservées d'une image précédente.

Pour cela, nous stockons à quel objet et quelle face correspond un point d'attache. Lors de la mise à jour des valeurs, nous vérifions que le point d'attache est toujours projeté sur la même face. Si ce n'est pas le cas, nous cherchons dans un voisinage de deux pixels si on retrouve la face dans le *G-buffer* d'identifiants. Si on ne trouve pas la face correspondante, et que le point n'est pas occulté, soit le point se trouve tout de même sur l'objet auquel il est attaché (la face n'est pas affichée pour des raisons d'aliasage), il faut alors mettre à jour la face et les coordonnées barycentriques avec les données lues sous le point d'attache, soit l'objet n'est plus le même (l'objet n'est plus affiché pour les mêmes raisons³) et les anciennes valeurs d'attributs sont alors conservées pour l'affichage de la marque. L'algorithme 5.1 résume l'opération de lecture des attributs.

ALGORITHME 5.1 – Lecture des attributs des marques

```

p : position du point d'attache
si p est occulté alors
    conserve anciennes valeurs d'attributs
    assigne p comme caché
sinon
    si face en p = face de distribution alors
        lit en p nouvelles valeurs d'attributs
    sinon
        si face ou objet de distribution présente dans voisinage en p' alors
            lit en p' nouvelles valeurs d'attributs
        sinon
            conserve anciennes valeurs d'attributs
    finsi
finsi
finsi

```

3. Pour pallier les problèmes d'aliasage, il faudrait utiliser des niveaux de détails (pour un objet, mais aussi entre les objets) en mettant à jour les données du point d'attache. Ainsi, nous serions certains que les les identifiants recherchés lors de la lecture seraient présents dans l'image de référence.

5.3.2 Lissage temporelle des attributs

Les attributs que nous affectons aux marques peuvent être lissés pour prendre en compte les attributs que la marque avait à l'image précédente. Pour cela, nous avons développé un mécanisme d'interpolation en conservant les différents attributs d'une marque pour une image de l'animation au temps t que nous mettons à jour en utilisant les valeurs obtenues au temps $t + 1$. Nous utilisons un temps $t \in \mathbb{N}$ représentant le nombre d'images traitées depuis le début de l'animation.

Nous nous plaçons dans le cadre où les positions et attributs des marques sont connus uniquement pour les images précédemment générées. Une autre possibilité serait de faire une première passe de calcul pour avoir l'ensemble des points d'attache et les valeurs de \mathcal{S} associées, puis de faire une deuxième passe de lissage. Notre cadre offre l'avantage de pouvoir traiter les données à la volée et donc de pouvoir modifier le style plus rapidement.

Une possibilité est de borner le changement maximum par pas de temps pour chaque attribut, soit $B \in \mathbb{R}^m$ le vecteur contenant la borne pour chaque attribut. Considérons une marque avec les attributs $a(t)$ au temps t . Au temps $t + 1$, la fonction de style permet d'obtenir les nouveaux attributs pour la marque suivant la position de son point d'attache soit $\mathcal{S}(p(t + 1))$. Pour calculer les nouveaux attributs de la marque, nous faisons

$$a(t + 1) \leftarrow \min(a(t) + B, \mathcal{S}(p(t + 1)))$$

Une autre possibilité est d'utiliser une vitesse variable. La vitesse est alors calculée suivant la différence entre $a(t)$ et $a(t + 1)$. L'idée est que, si la différence est grande, le changement peut être plus rapide, et que lorsque la différence diminue, le changement ralentit pour éviter que l'évolution ne s'arrête net. Le calcul des nouveaux attributs s'écrit alors

$$a(t + 1) \leftarrow (1 - \Delta)a(t) + \Delta a(t + 1)$$

Δ représente un facteur de changement, par exemple 0.25 revient à réduire d'un quart la différence entre les deux valeurs à chaque pas de temps.

D'après nos expérimentations, la couleur, la longueur et l'épaisseur peuvent varier rapidement pour conserver une bonne représentation de la scène. Nous utilisons généralement le lissage à vitesse variable avec $\Delta = 0.1$. Par contre l'orientation est plus délicate à gérer. En effet, l'orientation joue un rôle important dans la manière dont nous interprétons l'animation. Nous avons remarqué que si l'orientation change en espace image, nous interprétons ce changement comme une rotation de l'objet sous-jacent. Si cette rotation ne correspond pas à la rotation de l'objet, nous percevons alors un effet de glissement. Il est alors préférable que l'orientation des marques ne change pas ou peu, pour limiter cet effet plutôt que d'avoir des imprécisions sur la rotation.

L'utilisation du lissage temporel provoque un effet indésirable : lorsque l'animation en entrée fait une pause, nous avons un effet de retard, le temps que les attributs des marques prennent la valeur donnée par le style (qui n'évolue plus).

Les changements des attributs des marques n'ont alors aucune relation avec le mouvement dans la scène et cela peut être gênant. Pour contrôler cet effet, nous proposons de lier la vitesse de changement à la norme du mouvement en entrée et ainsi ne mettre à jour que les attributs de marques se trouvant dans les zones où il y a effectivement du mouvement.

L'artiste a encore des choix possibles : Finalement, les choix d'ordonnancement et de dessin sont plus des points techniques que des actions artistiques, mais l'artiste peut néanmoins faire varier l'apparence finale en modifiant ces processus.

5.3.3 Ordre

En tirant parti des observations que nous avons faites dans l'analyse des travaux précédents, nous proposons trois méthodes de tri des marques :

- en profondeur pour avoir la cohérence entre l'ordre de rendu et la surface représentée par la marque ;
- un ordre arbitraire, suivant la date de création de la marque : l'effet d'empilement est supprimé mais certaines incohérences apparaissent lors du rendu ;
- en profondeur modulée par un décalage : c'est une version simplifiée de l'ordre de Deep Canvas [Dan99] car nous n'avons pas implémenté d'adoucissement lors du changement d'ordre.

Selon le type de scène, l'artiste pourra choisir l'ordonnancement qui lui convient.

5.3.4 Fidélité

En plus de la forme des marques, l'artiste peut vouloir contrôler le niveau d'abstraction la scène. Pour cela, il peut agir en amont, c'est-à-dire sur les données qu'il fournit à la fonction de transfert. C'est ce que nous avons présenté section 5.2.4 en parlant des traitements que l'on peut faire sur la scène en entrée. Il peut aussi travailler au niveau des marques, c'est-à-dire influencer la forme de la marque en fonction de la scène et du détail qu'il veut obtenir. Pour cela, nous proposons un mécanisme de marche le long du squelette de la marque. Dans le même ordre d'idée que Hertzmann [Her98], nous proposons de stopper la marque lorsque la valeur d'une propriété sous le point d'attache (typiquement la couleur ou la profondeur) devient trop différente de la valeur sous le squelette.

Pour cela nous parcourons le squelette de la marque du centre de celui-ci — position du point d'attache — vers les extrémités et déterminons les deux positions (une vers chaque extrémité) à partir desquelles la différence entre la valeur au point d'attache et la valeur sous le squelette dépasse un seuil donné par l'artiste. Ce seuil est défini en espace image ; il est calculé suivant la même approche que les autres attributs (fonction de transfert et traitements). La figure 5.4 reprend schématiquement ces étapes.

Nous avons implémenté la marche le long des squelettes de marque sur le GPU en utilisant deux passes de rendu, voir l'algorithme 5.2. Une première passe parcourt le chemin dans un *pixel shader* pour chaque marque à dessiner. Ce *shader* est appelé pour chaque point d'attache sélectionné. Les paramètres (position le long du chemin) minimum et maximum qui respectent le critère de seuil sont stockés dans une

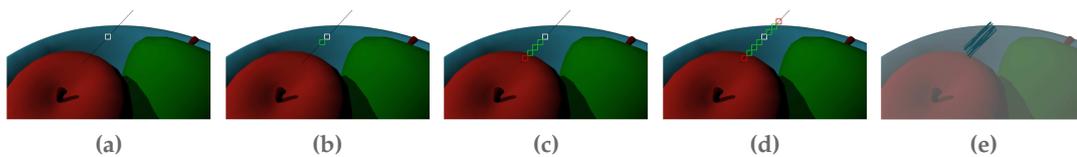


FIGURE 5.4 – Marche le long du squelette d’une marque pour déterminer sa longueur, c’est l’attribut de couleur qui est utilisé ici pour stopper la marche. (a) Squelette de la marque centré sur le point d’attache, la marche démarre de la position du point d’attache. (b) Chaque position p le long du squelette est testée, si la différence de couleur entre la couleur au point d’attache et celle sous p est inférieure au seuil défini par l’artiste, la marche continue. (c) Lorsque le seuil est franchi, la marche s’arrête et la position trouvée sur le squelette est stockée pour l’étape de dessin. (d) Même chose dans l’autre direction. (e) Dessin de la marque sur le squelette, limité suivant les paramètres obtenus dans chaque direction.

texture. Ensuite, lors d’une deuxième passe, chaque marque est dessinée avec un *quad-strip* défini par les attributs de la marque. Les paramètres stockés modifient les coordonnées de texture pour limiter la région couverte par la marque.

ALGORITHME 5.2 – Marche le long des chemin de marques

```

Q : liste points d’attache
p : propriété pour la fidélité (G-buffer)
t : seuil de fidélité
pour tout  $q \in Q$  faire
     $ref \leftarrow p(q)$ 
     $a \leftarrow S(p)$ 
    flottant  $c \leftarrow 0.5$  // position sur le squelette  $\in [0, 1]$ 
    flottant  $c_{min}$ 
    tantque  $\|ref - p(c)\| < t$  et  $c \leq 1$  faire
         $c \leftarrow c + dc$ 
    fin tantque
     $c_{min} \leftarrow c$ 
     $c \leftarrow 0.5$ 
    flottant  $c_{max}$ 
    tantque  $\|ref - p(c)\| < t$  et  $c \leq 1$  faire
         $c \leftarrow c + dc$ 
    fin tantque
     $c_{max} \leftarrow c$ 
    tracer marque en  $q$  restreinte sur  $[c_{min}, c_{max}]$  // en deuxième passe
fin pour

```

5.3.5 Apparition et disparition par seuillage

Comme nous l’avons vu dans le chapitre 3.2, la méthode utilisée par les travaux précédents pour l’apparition des marques est l’*alpha blending*. La figure 5.5a montre un exemple d’apparition en suivant cette approche. C’est un moyen de lisser temporellement le changement brutal dû à l’apparition d’une marque, mais les effets de matières sont alors estompés. Nous proposons d’utiliser une méthode de seuillage pour introduire progressivement une marque dans le rendu. Nous nous



(a) Apparition douce en utilisant l'*alpha blending*.

(b) Apparition douce en utilisant notre méthode, inspirée des travaux de Durand et coll. [DOM⁺01].

FIGURE 5.5 – Apparition douce d’une marque. La ligne du haut de chaque figure montre l’apparition sur un support vide, la ligne du bas montre l’apparition sur un support rempli avec d’autres marques.

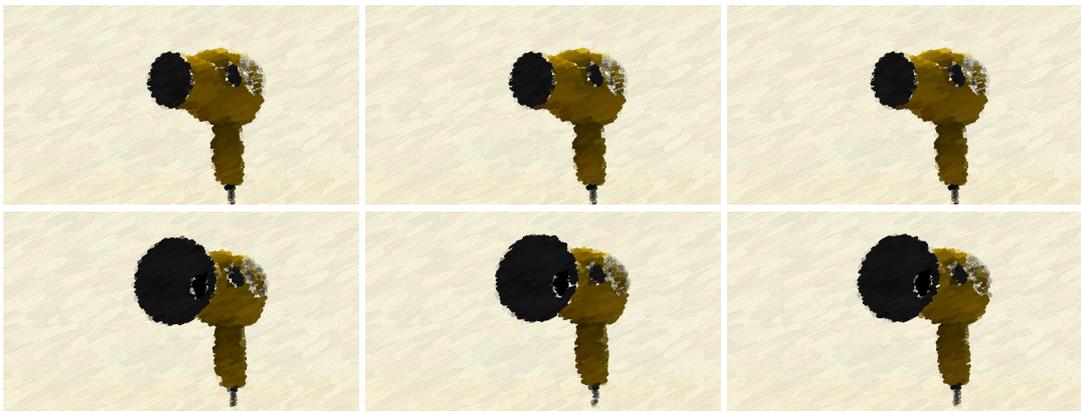
sommes inspiré du travail de Durand et coll. [DOM⁺01] qui proposent une technique de dessin de marques en utilisant deux textures et une valeur de pression. La première texture représente la forme de la marque, tandis que la deuxième introduit des variations dues au médium et au support. Ils utilisent ce modèle pour rendre un dessin (fait par un artiste sur un ordinateur) dans différents styles. Nous avons adapté leur approche pour obtenir une apparition douce et continue des marques, tout en respectant la juste représentation des effets de matière. Pour cela nous utilisons le niveau de pression que nous faisons varier lors de l’apparition d’une marque. La figure 5.5b montre le résultat que nous obtenons en comparaison avec l'*alpha blending*.

En fait, cela revient à faire apparaître la marque comme si elle était redessinée d’une image à l’autre avec une pression croissante, les paramètres intrinsèques tel que le grain du papier et les interactions de la matière étant les mêmes, ce qui n’est bien évidemment pas possible de faire manuellement.

5.4 Résultats

Nous avons mis en pratique notre système lors de la réalisation d’un court métrage intitulé *Dumb Luck* [MBV⁺07]. La réalisation de ce court métrage montre que notre système permet de gérer les mouvements complexes de scènes 3D et que l’utilisation de différents styles permet de porter le message. En effet, les artistes (Florent Moulin et Laurence Boissieux) ont utilisé des styles représentant par exemple la force du mouvement, et suggérant l’état du personnage comme l’illustre la figure 5.6.

Les scènes de ce court métrage ont été rendues en utilisant plusieurs calques : un pour le fond et un pour chaque élément animé indépendant (du point de vue de la profondeur). Nous avons généré les propriétés de la scène directement à partir du logiciel de modélisation (Autodesk® 3ds® Max) pour tirer parti de la qualité du rendu. Puis, nous avons importés ces propriétés dans notre logiciel de rendu. Nous avons utilisé la technique de distribution présentée section 4.3.



(a) Ajout de changements brutaux dans les attributs des marques pour rendre l'augmentation de puissance du sèche cheveux.



(b) Variations du niveau de perturbation aléatoire des couleurs pour rendre un effet de séchage de la peinture. Pour accentuer l'effet, le directeur artistique à aussi diminué

FIGURE 5.6 – Extrait du court métrage « Dumb Luck » montrant comment le style peut porter le message de l'artiste.

```

void main(void){
    // compute stroke position on screen
    vec4 projectedStrokePos = project();
    strokeData.xy      = projectedStrokePos.xy;
    strokeData.xy      = slidePosition(strokeData.xy);

    // set style variables according to input data
    vec4 attr1 = texture2D(strokeAttr1Map, strokePos);
    vec4 attr2 = texture2D(strokeAttr2Map, strokePos);

    vec4 nattr1 = updateParam3(attr1, oldStrokeAttr1);
    vec4 nattr2 = updateParam3(attr2, oldStrokeAttr2);
    wid = nattr1[0];
    len = nattr1[1];
    angle = updateAngle(attr1[2], oldStrokeAttr1[2]);
    nattr1[2] = angle;
    jittering = nattr1[3];
    bending = nattr2.x;

    // prevent from computing and div by 0 if len is null
    if(len == 0){
        paramMin = 0.6;
        paramMax = 0.4;
    }
    else{
        angle = jittering * jitterAngle*6.0 + angle;
        len *= 1.0+jittering * jitterLen;
        bending *= 1.0+jittering * jitterBending;
        rotate = mat2 (cos(angle), sin(angle), -sin(angle), cos(angle));
        // retrieve stroke data at stroke Position
        strokeColor = testDxture2D(colorMap,strokeData.st);
        strokeDepth = texture2D(depthMap,strokeData.st);
        // retrieve old stroke data at stroke Position
        bool newDataInvalid = (minIdDist>=0.5/255.0);
        bool notvisible = !visible;
        // skip invisble stroke or compute min/max param and
        // consereve old strokes attributes
        computeParamMin();
        computeParamMax();
        if(paramMin>0.5-wid/len)
            paramMin = 0.5-wid/len;
        if(paramMax<0.5+wid/len)
            paramMax = 0.5+wid/len;
    }
    // "control commande" update
    strokeData.z = paramMin;
    strokeData.w = paramMax;

    //////////////////////////////////////
    // write output data
    gl_FragData[0] = vec4(strokePos,0,1);// strokeData;
    gl_FragData[1] = attr1;
    gl_FragData[2] = nattr2;
}

```

LISTING 5.3 – Extrait du *shader* pour la première passe de rendu des marques.

Résultats et Discussion

Nous présentons ici les pistes de recherches pour le rendu animé et stylisé à base de marques. Comme nous l'avons vu au chapitre précédent, les points d'entrée possibles sont nombreux pour interagir avec l'artiste. Nous avons plus spécifiquement étudié la distribution de points d'attache et le contrôle du style en utilisant une fonction de transfert. Il reste d'autres points où de nombreuses contributions sont possibles. Nous allons dans un premier temps discuter des limitations et des pistes pour des travaux futurs sur les points que nous avons spécifiquement abordés, puis nous ouvrirons vers d'autres points d'entrée qu'il serait intéressant de travailler.

6.1 Mesure de la cohérence du rendu final

Nous avons utilisé des outils de mesure pour qualifier la qualité des distributions statiques ainsi qu'une évaluation visuelle des distributions dynamiques. Mesurer la qualité du rendu final est bien plus difficile. En effet, à notre connaissance il n'existe pas de moyen de comparer deux méthodes de rendu à base de marque. Outre la liberté de style productible, il serait intéressant de mesurer la cohérence temporelle de l'animation obtenue. Pour cela, nous devons évaluer comment se comporte notre méthode, ou une autre d'ailleurs, suivant les contraintes que nous avons définies. La juste représentation du mouvement demandé pourrait se faire en le comparant avec le flux optique de l'animation obtenue.

Mais il faudrait pour cela avoir une mesure de la similitude de deux flux optiques (si tant est que les artistes veulent obtenir le même flux optique). Une telle mesure reste à trouver, mais nous pensons qu'elle devrait se comporter comme le rayon relatif utilisé dans le cas des distributions. C'est-à-dire que nous voulons globalement suivre le mouvement en entrée, mais qu'un suivi trop parfait ne rendrait pas bien l'abstraction inhérente à la stylisation. C'est un reproche que les artistes font souvent au rendu cartoon qui, étant trop parfait, nuit au style recherché.

Concernant les changements brutaux, nuisant à l'interprétation de l'animation et fatiguant le spectateur, nous avons pensé à l'utilisation d'un oculomètre pour

suivre le regard d'un spectateur. En effet, comme nous savons où nous ajoutons et supprimons des marques, nous pouvons déjà repérer si le regard du spectateur est attiré par ces changements. De la même manière, nous pourrions mesurer, à partir de quelle vitesse de changement des attributs, le regard du spectateur est captivé.

Le dernier point à mesurer, le plus délicat sans doute, est la juste représentation des effets visuels, de la matière, donnant le style. Par construction nous essayons d'obtenir certains effets, mais l'évaluation de la réussite, et surtout la comparaison à d'autres méthodes, paraît délicate. En effet, le jugement de la qualité des effets obtenus est très subjective, et chacun peut dire qu'il *préfère* tel rendu (tel aspect visuel) à tel autre.

6.2 Contrôle du style

La spécification du style par l'artiste n'est pas facile. En effet l'artiste a besoin de pouvoir donner sa touche personnelle au rendu. Dans notre type d'approche, où une grande part est automatisée, l'artiste a besoin d'un contrôle de haut niveau, c'est-à-dire qu'il doit pouvoir spécifier, *a priori*, son style, et que le système interprète cette spécification pour calculer le résultat.

Pour cela, nous avons choisi d'utiliser la fonction de transfert présentée chapitre 5. Pour contrôler le calcul des attributs, nous pouvons utiliser des propriétés de plus haut niveau que celles directement extraites de la scène en entrée. Le travail sur l'abstraction de la stylisation de Cole et coll. [CDF⁺06] est un bon exemple de ce que l'on peut faire. Ils proposent divers opérateurs permettant de contrôler le style en fonction d'un point d'intérêt. Les opérateurs suivent une métrique en espace objet ou en espace image pour déterminer le niveau d'abstraction d'une zone de l'espace image lors du rendu. Le niveau d'abstraction sert de contrôle à la stylisation. Dans leur travaux ils donnent l'exemple de lignes dont l'épaisseur varie suivant l'abstraction. Nous pouvons aussi avantageusement utiliser une telle métrique comme entrée de \mathcal{S} pour contrôler les attributs des marques.

Néanmoins, l'approche programmable pour la spécification du style est difficile à prendre en main pour un artiste. Sous réserve que l'on puisse obtenir le style voulu par la fonction de transfert, il serait intéressant d'obtenir cette fonction par un autre moyen qu'un programme descriptif. Une piste prometteuse est la spécification par l'exemple. En effet, un bon moyen pour les artistes de décrire ce qu'ils veulent est de produire un exemple. L'exemple serait ensuite analysé par le système de rendu pour construire la fonction de style. Toute la difficulté est de capter l'essence que constitue l'exemple pour l'appliquer correctement à l'animation.

Le contrôle du style par l'exemple a fait l'objet de recherches pour le dessin au trait, comme nous l'avons vu dans le cas de l'approche de Kalnins et coll. [KMM⁺02]. Jodoin et coll. [JEGPO02] utilisent une approche de contrôle par l'exemple, pour le rendu de hachures. Ils calculent les propriétés statistiques de dessins de lignes pour l'appliquer à un ensemble de lignes. Lum et Ma [LM05] proposent de laisser l'artiste sélectionner les lignes extraites sur un rendu 3D pour le dessin au trait. Ils analysent ensuite la sélection, suivant un ensemble de caractéristiques prises sur les lignes extraites, au moyen d'un réseau de neurones. Ensuite

leur approche permet d'appliquer le même type de sélection sur d'autres objets 3D et d'autres vues. Ces travaux sur le contrôle par l'exemple montrent la puissance de ce type d'approche, mais aussi la difficulté de définir le comportement correct, c'est-à-dire prévisible par l'artiste. Il faut que d'une certaine manière, le système *compre* l'exemple de l'artiste, pour l'appliquer à nouveau.

Sans les avoir expérimentés, nous avons réfléchi à plusieurs moyens de contrôle par l'exemple et aux enjeux que cela représente. L'idée est de laisser l'artiste dessiner quelques marques, suivant notre modèle, par dessus une vue fixe d'une scène 3D. Ensuite, nous devrions pouvoir construire une fonction de transfert à l'aide d'une optimisation, type recuit simulé. Nous avons pensé que, proposant une définition par essais et corrections, l'artiste pourrait obtenir rapidement une fonction de transfert de style satisfaisante. Par exemple, après avoir dessiné quelques marques, le système effectuerait le remplissage de la scène ; ensuite l'artiste viendrait corriger les parties qui ne lui plaisent pas. Après quelques corrections, le rendu devrait correspondre à son attente. Puis, pour contrôler le style tout au long de l'animation, il ajouterait des images clés pour l'exemple. Ceci nous permettrait d'obtenir un contrôle de la fonction de transfert de style, par l'exemple, pour le rendu à base de marques d'animations.

Simulation des méthodes manuelles en peinture animée

CE CHAPITRE présente les travaux que j'ai réalisés en collaboration avec Hugo Robert durant son stage de Magistère 1^{re} année. Nous avons conjointement élaboré les idées et directions de recherche et il a effectué la majeure partie du développement.

Dans les chapitres précédents, notre approche consistait à déplacer et d'adapter des marques pour former l'animation. Les résultats que nous avons obtenus offrent une forme de cohérence temporelle, mais l'aspect de l'animation s'éloigne de ce que les artistes obtiennent manuellement. Effectivement, déplacer une marque déjà appliquée sur le support n'est pas réalisable dans le cadre de créations manuelles. Nous proposons, dans ce chapitre, une alternative simulant le procédé manuel.

Notre objectif est donc de simuler, de manière automatique, ce que font les artistes à la main pour obtenir une peinture animée. Pour simuler ce procédé, nous commençons par identifier les éléments visuels caractéristiques, présents dans une peinture animée, ainsi que les méthodes de travail des artistes. Puis nous définissons un modèle de simulation informatique permettant de reproduire ces éléments ainsi qu'une méthode de peinture automatique pour l'animation. Enfin, nous présentons comment l'artiste peut contrôler le rendu de l'animation dans notre système, à différents niveaux, pour exprimer son propre style.

7.1 Analyse des méthodes manuelles

Comme nous l'avons vu avec la figure 2.5, les effets présents dans la peinture animée sont principalement dus à l'apparence du pinceau, c'est-à-dire la présence d'irrégularités dans le tracé dues aux fibres du pinceau, à la matière donnée par la peinture, et au mélange des coups de pinceau successifs sur le support. Le dispositif traditionnel de peinture animée, présenté figure 7.1, est constitué d'un appareil de capture (caméra) placé au dessus d'un support (plaque de verre) rétro-éclairé. L'artiste utilise un médium constitué de pigments et de liant. Ce dernier est choisi pour



(a) La peinture est produite sur une plaque de verre avec divers outils tel que pinceaux, couteaux ou directement avec les doigts.

(b) Plusieurs plaques de verres peuvent être empilées. Ainsi, l'artiste dispose de plusieurs calques indépendants.

FIGURE 7.1 – Installation pour la création de peinture animée.

permettre de travailler la peinture pendant plusieurs jours car il sèche lentement. Dans la suite de ce chapitre, nous parlerons de *matière* pour désigner le mélange pigment médium (habituellement appelé peinture) pour ne pas créer d'ambiguïté avec le terme *peinture* qui désignera les images obtenues. Le principe de la peinture animée traditionnelle est le suivant : l'artiste peint chacune des images constituant l'animation sur le support. Il donne ainsi l'illusion de mouvement en produisant des peintures qui se *suiwent*. Il réalise la première image comme une peinture classique, en plaçant des touches de matière sur le support. Puis, il crée l'animation en modifiant la matière présente sur le support. Il modifie la peinture en effaçant ou en ajoutant de la matière, ou encore en effectuant des mélanges et étalements de la matière déjà en place. En effet, comme la matière n'a pas le temps de sécher entre deux prises de vue, chaque nouvel ajout se mélange avec la matière déjà présente sur le support. Dans notre approche, nous simulons automatiquement ces actions.

De plus, l'artiste peut varier le nombre d'images par seconde au cours de l'animation. Il utilise généralement entre une et dix images par seconde, suivant le type de mouvement représenté. Il peut aussi, grâce à l'utilisation de plusieurs plaques de verre, obtenir des calques indépendants et n'animer que certains calques. Nous n'aborderons pas ces deux possibilités dans notre approche, car elles peuvent être facilement réalisées avec un logiciel d'édition vidéo.

La réalisation de peinture animée est très longue, et nous pensons qu'une grande part de la création découle d'un certain automatisme de la part de l'artiste. Sans vouloir se substituer à l'artiste, notre méthode permet d'envisager l'utilisation du rendu peinture animée dans le style traditionnel pour des animations plus longues que des courts métrages expérimentaux. La suite de ce chapitre est organisée comme suit : nous présentons les travaux précédents directement liés à notre approche section 7.2, puis le fonctionnement de notre système de simulation de peinture animée et comment nous avons choisi d'automatiser les parties les plus fastidieuses sections 7.3 à 7.6. Nous montrons les possibilités d'interaction section 7.7 avant de discuter les résultats section 7.8.

7.2 Travaux précédents en simulation et animation de peinture

Les travaux sur les animations en peintures animées sont nombreux, mais traitent le plus souvent de la cohérence temporelle que nous avons étudiée chapitre 3 et 3.2. Pour obtenir une peinture animée qui aurait pu être obtenue avec les méthodes traditionnelles, nous avons besoin d'une part d'un modèle de simulation de peinture (interaction entre la matière, le pinceau et le support), et d'autre part d'un algorithme de peinture automatique reproduisant les techniques manuelles pour l'animation. Nous allons donc passer en revue les travaux correspondants à ces besoins.

Un certain nombre de travaux traitent de la simulation informatique des média traditionnels. Baxter et coll. [BSL01, BLL04, BWL04] proposent un modèle d'interaction entre le pinceau et le support, basé sur une simulation de collisions entre les fibres du pinceau et le plan support. L'artiste peint alors avec un ordinateur plutôt qu'avec les outils traditionnels. On peut voir sur la figure 7.2 une installation typique d'utilisation de leur programme. Le pinceau est modélisé par un ensemble de courbes 3D représentant des groupes de fibres. Lors du mouvement du pinceau, Baxter et coll. calculent comment les fibres se déforment suivant le contact avec le support, l'orientation du pinceau et le mouvement que fait l'artiste avec l'appareil de contrôle (par exemple un stylet et une tablette graphique). Ensuite, ils simulent les échanges de matière qui se produisent entre les fibres du pinceau et le support, et calculent ainsi la couleur présente sur le support. Le modèle de simulation de Baxter et coll. est particulièrement adapté pour les matières pâteuses comme la peinture à l'huile. Chu et Tai [CT02, CT04, CT05] proposent le même type d'approche, pour des média fluides tels que l'encre de chine et les encres colorées, et axent leur approche sur la diffusion du médium sur le support.

Les méthodes de peinture automatique utilisent en général un modèle de simulation plus simple : le mélange des couleurs se fait par transparence et on perd alors les effets de matière propres à la peinture. En fait, les travaux se concentrent sur la description de la génération automatique de coups de pinceau. Dans son tour d'horizon, Hertzmann [Her03] présente la génération automatique de peinture comme une optimisation entre deux contraintes : la juste représentation de l'image de référence en entrée (photo) et le style voulu. Les différentes méthodes donnent des sens différents à ces deux contraintes, mais l'objectif est le même : générer une peinture (image virtuelle) représentant l'image de référence avec l'aspect d'une peinture dans un style donné (par les concepteurs ou par l'artiste).

Hertzmann [Her98] génère des coups de pinceau courbes pour peindre automatiquement une photo (image de référence). Le chemin que suit un coup de pinceau est défini à partir du gradient¹ de l'image de référence. Le placement se fait de telle sorte que chaque coup de pinceau réduise la différence de couleur entre l'image de référence et la peinture en train d'être calculée. L'utilisateur du système peut contrôler le style en jouant sur les paramètres de couleur, d'épaisseur, de longueur et la forme du chemin pour mimer par exemple l'expressionnisme ou le pointillisme.

1. Ici le gradient est la dérivée deux dimensions, en espace image, de l'intensité de l'image de référence. Soit $\mathcal{J}(x, y)$ l'intensité de l'image à la position (x, y) , le gradient ∇ est défini par le vecteur $\nabla(\mathcal{J}) = \left(\frac{\delta \mathcal{J}(x)}{\delta x}, \frac{\delta \mathcal{J}(y)}{\delta y} \right)$.

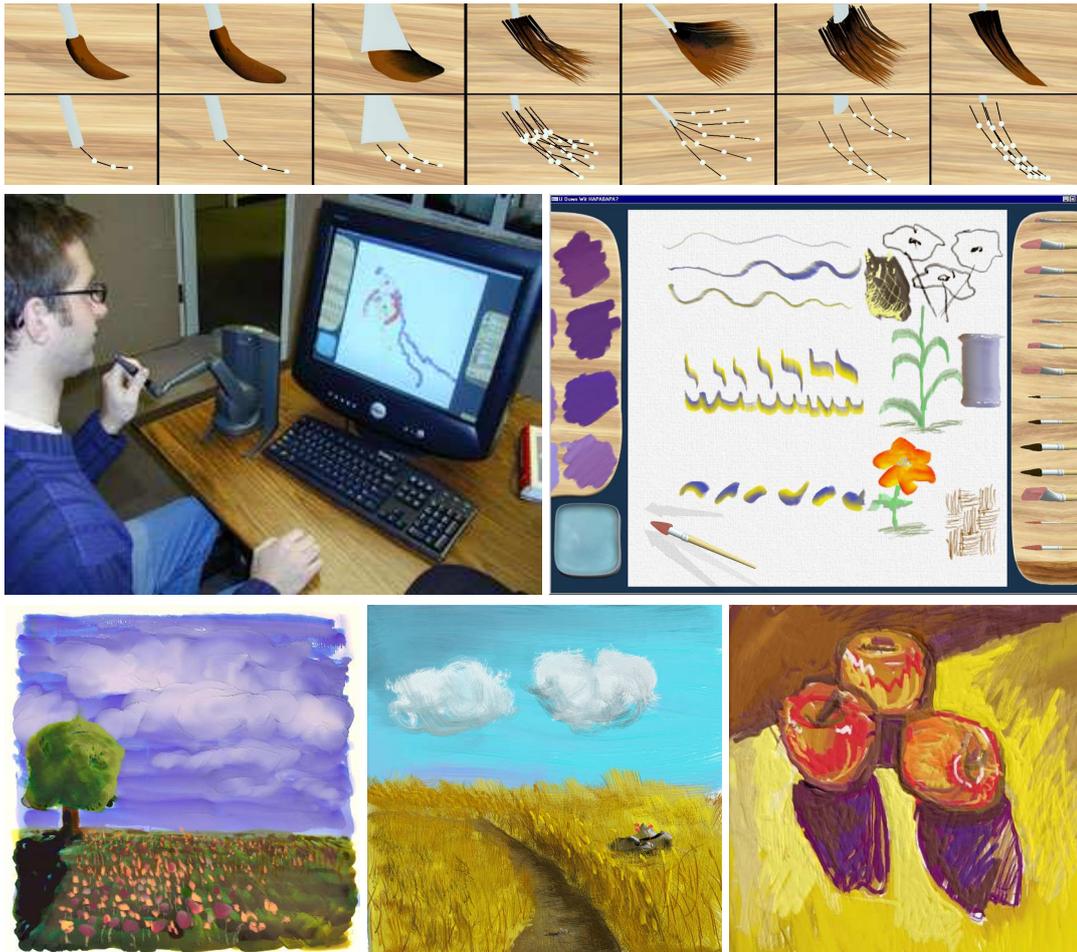


FIGURE 7.2 – Images tirées des travaux de Baxter et coll. [BSL01, BLL04, BWL04]. La première ligne montre le modèle de pinceaux pour différents outils. La seconde ligne présente le système d'interaction ainsi que l'interface du logiciel. La troisième ligne montre quelques résultats.

La figure 7.3 montre quelques exemples de résultats. Nous basons notre création de peinture automatique sur cette approche et nous détaillerons le fonctionnement de la méthode section 7.5.

Hertzmann [Her01] propose une autre approche basée sur une relaxation. L'utilisation d'une relaxation, et donc d'une fonction de coût, permet de définir ce que l'on veut obtenir sans programmer explicitement l'algorithme de placement et de dessin des coups de pinceau. La définition du style est alors l'écriture de cette fonction de coût. L'artiste n'a plus à se préoccuper du fonctionnement interne du programme, par contre, il est difficile de comprendre comment les paramètres de la fonction de coût modifie la peinture produite.

Collomosse et Hall [CH02, CH05] proposent l'utilisation d'algorithmes génétiques et de détection de saillance pour guider le processus de peinture automatique. Ils calculent une image de saillance à partir de filtres imitant le fonctionnement du système visuel humain. La carte de saillance donne le niveau de détails, ou d'importance, des différentes régions de l'image en entrée. Ils utilisent ensuite cette carte pour faire varier les paramètres des coups de pinceau et peindre avec plus ou moins de précision l'image en entrée. Ainsi ils conservent les détails importants



FIGURE 7.3 – Première ligne : image de référence et trois étapes de la création automatique d’une peinture avec l’algorithme d’Hertzmann. Deuxième ligne : image de référence et différents styles possibles: impressionniste, expressionniste et pointilliste. Images tiré du travail d’Hertzmann [Her98].

tandis que les parties moins significatives sont plus abstraites.

Shugrina et coll. [SBC06] utilisent l’analyse émotionnelle, à partir d’une capture en temps réel du visage de l’observateur, pour contrôler le style de la peinture. Ils utilisent un modèle d’émotion avec deux axes (joyeux/triste, excité/fatigué) et une analyse faciale pour déterminer quelle est l’émotion du visage de l’observateur. Ils lient l’émotion à des paramètres de style (couleur, forme, courbure) des coups de pinceau. Ainsi la peinture obtenue s’adapte pour refléter l’humeur de l’observateur.

Kalaidjian [Kal07] utilise un système de *shaders* définissant des procédures de peinture pour réaliser automatiquement des peintures dans le style de Bob Ross. Il est parti de l’observation de la méthode de peinture de Bob Ross² pour décrire des méthodes systématiques de peinture d’arbres ou de montagnes par exemple. Il définit ensuite un ordre et certains espaces de valeurs pour des variables aléatoires modifiant légèrement l’aspect des procédures de peinture. Dans son approche, il utilise un modèle de simulation de peinture plus complexe qu’une simple image permettant de reproduire les effets de mélange de couleurs entre le pinceau et le support. Apportant plus de richesse graphique que la simple utilisation d’une image (comme dans les autres méthodes automatiques), et restant plus performant que les méthodes basées sur l’interaction physique entre le pinceau et le support, son approche produit de bons résultats. Nous avons basé notre système de simulation de peinture sur ses travaux.

7.3 Notre système de peinture animée

Nous avons développé un système de rendu de peintures animées s’inspirant des techniques traditionnelles. Nous prenons en entrée une vidéo accompagnée de son flux optique. La vidéo est obtenue soit avec une caméra, soit en effectuant le rendu d’une scène 3D. Nous appellerons dans ce chapitre *peinture* une image produite par le système et *image de référence* une image de la séquence vidéo d’entrée. Comme présenté figure 7.4, nous calculons une première peinture à partir de la

2. De nombreuses explications et vidéos sont disponibles, notamment sur le site de Bob Ross <www.bobross.com>

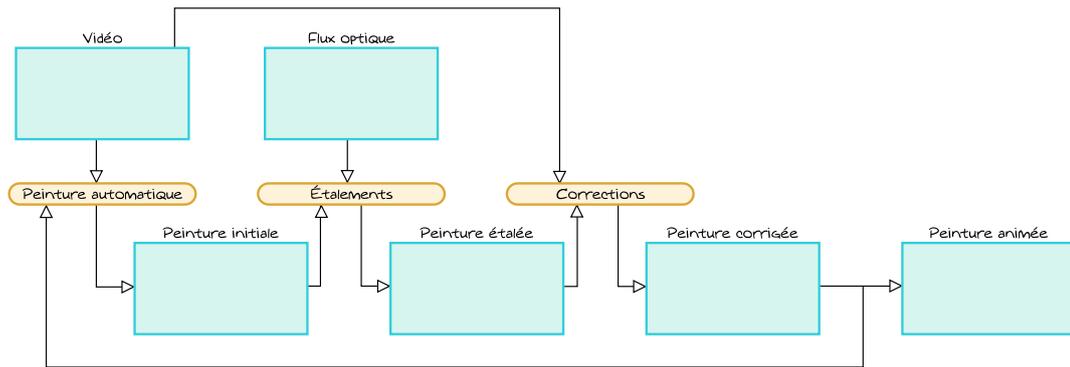


FIGURE 7.4 – Pipeline de notre approche de simulation automatique de peinture animée. L’algorithme de peinture prend la vidéo et la peinture précédente en entrée (peinture vide pour la première image), puis suivant le flux optique, applique des corrections et étalements. La peinture intermédiaire est ensuite corrigée suivant la vidéo de référence. La succession des peintures donne la peinture animée résultant de notre approche.

première image de référence. Une fois la première peinture obtenue, nous passons à la suivante en modifiant la première peinture. Pour cela, nous suivons les observations que nous avons faites à propos des méthodes manuelles, c’est-à-dire que nous produisons des étalements et des ajouts de peinture sur le support.

7.4 Modèle de simulation de peinture

La simulation de la peinture se fait en définissant les interactions (les échanges de matière) entre un pinceau et le support. La création de modèles de pinceau qui utilisent la physique peut être une tâche très compliquée comme nous l’avons vu dans les travaux précédents. Ces modèles sont suffisamment performants pour permettre l’interaction avec l’artiste qui peut dessiner en temps réel les coups de pinceau. Par contre, le rendu d’une peinture, et donc de plusieurs centaines de coups de pinceau, nécessite un temps considérable. Ce type de méthodes prend donc trop de temps d’exécution pour la production d’animations. Nous avons donc choisi d’utiliser un modèle plus simple, dont le temps de calcul est plus court et qui permet d’obtenir des effets visuels suffisants pour notre objectif. Nous utilisons une simplification des phénomènes physiques en partant des travaux de Kalaidjian [Kal07].

7.4.1 Entité de dessin et support

Nous présentons maintenant notre système, en pointant les principales différences avec le travail de Kalaidjian. Bien que notre système ne soit pas basé sur la physique des fluides, nous devons simuler des éléments, comme les mèches de fibres, pour capturer l’apparence qu’un pinceau produit lorsqu’on peint. Lors de l’utilisation d’un pinceau, la matière agglomère plusieurs fibres entre elles, ce qui produit des traînées de couleur sur le support. Nous modélisons les différentes mèches de fibres du pinceau par un ensemble d’entités de dessin (ED). Une ED

est définie par une couleur et une quantité de matière. De plus, la forme de l'entité, c'est-à-dire la trace qu'elle laisse sur le support lors du dessin, est définie par un masque de transparence. Dans notre implémentation, il s'agit d'un tableau de trois par trois valeurs présenté figure 7.5. Cette définition est la même que dans les travaux de Kalaidjian.

Le support est représenté par une grille de pixels, chaque pixel contenant non seulement une couleur mais aussi une quantité, au même titre que les entités de dessin. Au moment du dessin sur le support, les quantités de matière du support et des EDs sont modifiées pour tenir compte des échanges entre le pinceau et le support. Les couleurs présentes sur le support et sur les EDs sont, elles aussi, mises à jour, pour tenir compte des échanges de matière. Dans la méthode de Kalaidjian, le support est simplement une image, ne tenant pas compte de la conservation de la matière, ce qui entraîne des pertes de matière (par exemple, de la matière rouge sur un fond blanc deviendra de la matière rose lors des étalements). En conservant la quantité de peinture sur le support, nous n'avons pas ces problèmes de dilution de la couleur.

[0.01, 0.03]	[0.05, 0.07]	[0.01, 0.03]
[0.05, 0.07]	[0.4, 0.6]	[0.05, 0.07]
[0.01, 0.03]	[0.05, 0.07]	[0.01, 0.03]

FIG. 7.5 Intervalles de valeurs du masque de transparence d'une ED. Ces valeurs correspondent à une gaussienne 2D dont le maximum est sur le point central. Pour une ED, les valeurs sont tirées aléatoirement dans les intervalles présentés.

Dessin le long d'un chemin

La trace d'une ED sur le support est définie par un chemin, une B-spline [Fer64] dans notre implémentation. Pour dessiner cette trace, nous échantillons le chemin, c'est-à-dire que nous définissons un certain nombre de positions à partir desquelles nous calculons les échanges de couleur entre les ED et le support. Nous plaçons les points d'échantillonnage sur le chemin, de telle sorte que chaque ED avance d'un pixel entre deux échantillons. Lorsque l'on passe d'un échantillon à l'autre, on dit que l'on avance d'un *pas* le long du chemin.

Quantité de matière et calcul de couleur

Chaque ED possède sa propre quantité de matière qui est représentée par une valeur $q \geq 0$ et sa propre couleur D . Chaque pixel du support a aussi une quantité de matière, notée r et une couleur C . Lors du tracé, il s'effectue un échange de matière, entre le support et l'ED, qui dépend de la pression d'application $\beta \in [0, 1]$. Nous avons implémenté ces échanges de manière empirique, en ajustant les constantes jusqu'à obtenir un résultat visuellement plausible. Donnons tout d'abord une description intuitive des échanges de couleur : s'il n'y a pas de matière sur le support, celui-ci prend directement la couleur D présente sur l'ED. Au contraire, s'il y a une forte quantité de matière sur le support et peu sur l'ED, le support conserve sa couleur C . Dans le cas où le support et l'ED ont des quantités de matière équivalentes, la couleur sur le support devient majoritairement celle de l'ED.



FIGURE 7.6 – Mélange des coups de pinceau sur le support. Le coup de pinceau rouge est placé en premier, le jaune vient par dessus et la partie supérieure du pinceau prend de la matière sur le support. Les EDs changent alors de couleur sur la fin du coup de pinceau.

À chaque pas de dessin, nous calculons les nouvelles couleurs C' , D' du support et de l'ED ainsi que les quantités de matière r' et q' correspondantes. Pour cela, nous calculons la quantité de matière qui est prise sur l'ED ainsi que celle prise sur le support pour chacun des pixels recouverts par l'ED. Nous prenons sur l'ED une quantité l définie par :

$$l = \min \left[\beta(0.05 + \text{rand}(0.005)) \times \min(q, 0.2) + 0.005, q \right]$$

Les valeurs r , m et C sont différentes pour chacun des neuf pixels couverts par l'ED. Pour ne pas surcharger les formules nous n'avons pas indicé ces symboles, mais il faut lire r_p , m_p et C_p . Sur le support nous prenons la quantité m suivante pour *chacun* des pixels couverts par l'ED :

$$m = \begin{cases} \min(\beta \frac{r}{2}, 0.05r) & \text{si } r > q \\ \frac{\beta \text{clamp}(0.05r, \max((1-q), 0), 0)}{2} & \text{sinon} \end{cases}$$

Nous calculons ensuite la nouvelle couleur sur le support en p suivant la valeur du masque de transparence α de l'ED pour ce pixel :

$$C' = \frac{\alpha l D + (r - m)C}{\alpha l + r - m}$$

La nouvelle quantité de matière en p sur le support est $r' = r - m + \alpha l$.

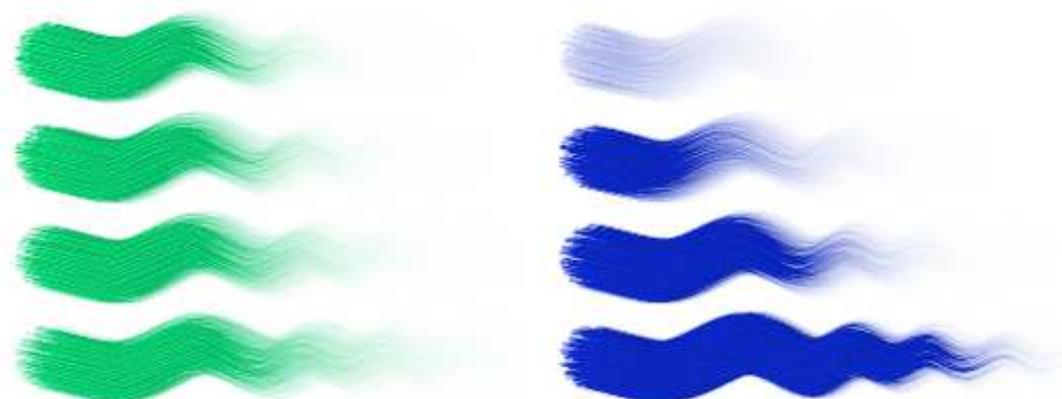
Pour la mise à jour de la couleur de l'ED, nous mélangeons les couleurs de l'ED et la somme des couleurs prises sur le support pour chacun des pixels couverts par l'ED:

$$D' = \frac{\sum_p (mC) + (q - l)D}{\sum_p (m) + q - l}$$

Enfin, nous ajoutons à la quantité de matière de l'ED, la quantité de matière perdue par le support pour chacun des pixels couverts : $q' = q - l + \sum_p m$

La prise de matière par l'ED permet d'obtenir les effets d'étalement présents, par exemple, dans la peinture à l'huile et à l'acrylique. Dans notre approche, il n'y a pas de perte de matière : les sommes de quantités de matière présentes sur le support et sur les EDs *avant* et *après* l'application d'un coup de pinceau sont égales.

Cette approche est très inspirée des travaux de Kalaidjian, que nous avons simplifiés, tout en améliorant la qualité des résultats. Kalaidjian avait notamment défini plusieurs états pour les EDs, changeant leur comportement si elles étaient en train de déposer de la matière ou d'étaler celle déjà présente sur le support. Comme nous conservons la quantité de matière présente sur le support, nous n'avons pas besoin d'utiliser plusieurs états d'ED pour correctement représenter les effets d'étalement. Cela nous permet d'avoir un modèle à la fois plus simple et plus unifié pour simuler les échanges de matière entre le pinceau et le support.



(a) Variation de la pression du coup de pinceau. De haut en bas : pression de 1, 0.8, 0.6, 0.4 (quantité de matière de 0.8).

(b) Variation de la quantité de matière. De haut en bas : quantité de matière de 0.3, 0.5, 0.6, et 1 (pression de 0.8).

FIGURE 7.7 – Influence de la quantité de peinture et de la pression sur le tracé d'un coup de pinceau. La longueur du tracé visible sur le support dépend de la quantité de matière et de la pression.

7.4.2 Définition d'un pinceau

Un pinceau est défini par un ensemble d'ED disposées suivant la forme du pinceau désiré. On peut facilement définir diverses formes de pinceau pour mimer les outils de la peinture traditionnelle. Le placement des ED se fait en distribuant les différentes entités à l'intérieur d'une forme comme un cercle, un ovale ou un rectangle. L'artiste peut aussi placer les ED manuellement, s'il veut plus de contrôle sur la forme de la trace laissée par le pinceau. Notons, toutefois, qu'il est important que les ED aient un placement reflétant l'aléatoire présent dans les vrais pinceaux pour ne pas faire ressortir de formes trop régulières. La figure 7.8 montre la différence entre un placement aléatoire et un placement régulier.



FIG. 7.8 Placement (aléatoire en haut, régulier en bas) des entités de dessin pour construire un pinceau.

Quantité de peinture et pression

Lorsqu'un coup de pinceau est créé, on lui affecte une quantité de peinture. Cette quantité est répartie aléatoirement dans les différentes ED qui le constituent. En fait, si on affecte la même quantité de peinture à chaque ED, le tracé du coup de pinceau s'arrête trop brutalement, à l'instant où toutes les ED n'ont plus de peinture. L'application d'un coup de pinceau se fait avec une certaine pression. Cette pression modifie la quantité de peinture déposée et prise par le pinceau à chaque pas de dessin. La figure 7.7 montre l'influence de la quantité de peinture et de la pression sur le tracé d'un coup de pinceau.

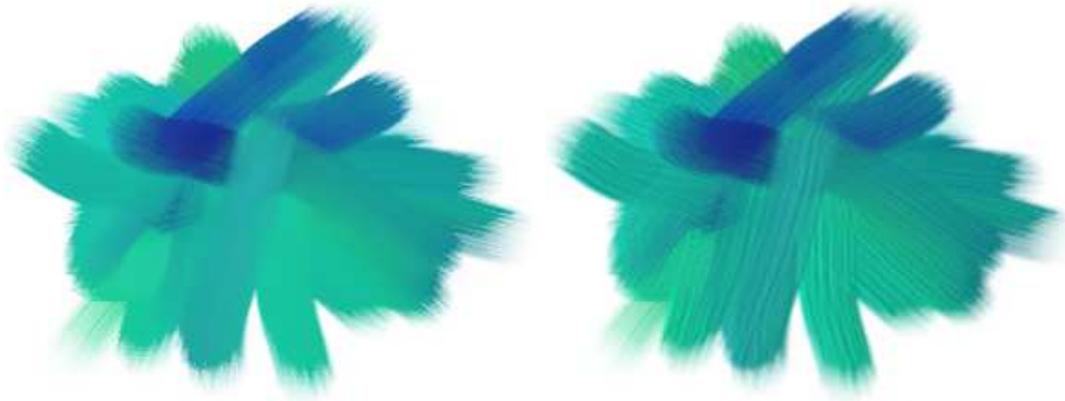


FIGURE 7.9 – Effet de matière produit par l'éclairage du support, en tenant compte de la quantité de peinture sur celui-ci. À gauche, la couleur rendue est celle directement présente dans chaque pixel du support. À droite, la couleur de chaque pixel est modulée par une lumière placée en haut à gauche.

7.4.3 Effet d'épaisseur

Si on observe une peinture traditionnelle, on remarque que s'il y a suffisamment de matière sur le support, la forme de celle-ci est perçue. En effet, la couche de matière sur le support n'est pas complètement plane, mais présente de légères variations de hauteur (quelques millimètres) par rapport au niveau du support. Cela crée des variations d'intensité lumineuse dues à l'éclairage, et aussi des reflets spéculaires. Pour reproduire ces effets, nous construisons une carte de hauteur en fonction de la quantité de matière présente sur le support et appliquons un *bump-mapping* [Bli78], qui calcule la façon dont cette carte de hauteur est éclairée. Ceci permet de mettre en avant les différentes ED et la forme de chacun des coups de pinceau présents. La figure 7.9 montre l'influence de l'utilisation du *bump-mapping* sur le rendu.

7.5 Algorithme de peinture automatique

Ayant décrit notre système de simulation de peinture, nous allons expliciter comment les coups de pinceau sont générés, pour produire l'animation. La première étape de la génération de l'animation est le calcul automatique d'une peinture correspondant à la première image de référence de l'entrée. Pour cela, nous utilisons l'approche d'Hertzmann [Her98] que nous allons maintenant détailler.

Nous décomposons une peinture en une série de couches faites avec des pinceaux³ de différentes tailles : en commençant par un pinceau de grande taille pour remplir le fond, puis en allant vers de plus petites tailles pour peindre les détails. Le placement des coups de pinceau pour chacune des couches, se fait en suivant une grille régulière dont la taille des cellules g est choisie en fonction de la taille du pinceau utilisé r (rayon inscrit) de manière à remplir le support, c'est-à-dire $g = 1/(2\sqrt{2})r$. Pour la première couche, un coup de pinceau est placé aléatoirement par case de la grille. Chaque coup de pinceau est construit suivant un chemin

3. Nous utilisons le terme pinceau pour désigner l'outil de l'artiste, mais rien n'empêche de simuler d'autres outils comme un couteau ou les doigts.

qui suit le gradient de l'image en entrée. La couleur du coup de pinceau est prise dans l'image de référence à la position initiale, et le coup de pinceau s'arrête quand la couleur de l'image d'entrée sous le chemin devient trop différente de celle de départ. On obtient alors une peinture temporaire constituée des coups de pinceau de la plus grande taille.

Pour les couches suivantes, un nouveau coup de pinceau est placé dans une case, seulement si le maximum de la différence de couleur entre la peinture temporaire et l'image en entrée dépasse un certain seuil μ_p . Le point de départ est placé à l'endroit où se trouve la plus grande différence, et la construction du chemin suit le même algorithme que pour la première couche. On continue ainsi jusqu'à la plus petite taille de coup de pinceau.

7.6 Corrections et modifications d'une peinture

Maintenant, nous voulons produire une nouvelle peinture correspondant à l'image suivante de l'animation. Cette peinture est réalisée à partir de la peinture obtenue précédemment, en effectuant des étalements et des modifications. L'algorithme de création d'étalements et de corrections doit produire une peinture en satisfaisant trois contraintes. Il est nécessaire de représenter correctement la nouvelle image de référence, de suivre le mouvement de l'entrée et de ne pas modifier les régions correspondant à des parties fixes des images de références.

Pour l'animation, nous aurions pu simplement utiliser le même algorithme que précédemment, sans tenir compte, ni de la première peinture générée ni du mouvement présent dans la vidéo de référence. Or, en regardant les animations faites à la main, on constate que seules certaines parties de l'image sont modifiées. Elles correspondent au lieu de l'action, et sont dues aux diverses touches que fait l'artiste pour suggérer le mouvement. Nous proposons donc de créer le mouvement de la même manière que les artistes, c'est-à-dire en simulant l'application de coups de pinceau par dessus l'image précédente. Le pinceau peut être sec, sans matière, il étale alors la matière déjà en place sur le support. Il peut aussi être soit chargé de peinture, il modifie alors, en plus de l'étalement, la couleur de la peinture sur le support. Nous avons choisi de procéder en deux temps, en utilisant un pinceau sec pour déplacer la matière en suivant le mouvement de la scène en entrée (donnée par le flux optique) puis en ajoutant de la matière pour représenter correctement la nouvelle image de référence.

7.6.1 Étalement de peinture

Les étalements permettent de donner plus de force aux mouvements présents dans la scène en entrée, tout en conservant les effets de matière (en comparaison avec une simple advection de texture par exemple) et offrent une meilleure cohérence temporelle, que l'utilisation seule des ajouts. En fait, les étalements n'ajoutant pas de peinture, ils modifient plus subtilement la peinture finale que ne font les coups de pinceau classiques. Pour passer d'une image à l'autre, nous construisons



FIGURE 7.10 – Avant et après l'étape d'étalement. On remarque que la peinture est déformée par les étalements suivant le sens du mouvement, et que l'on perçoit la forme des EDs sur les coups de pinceau sec qui ont servi à étaler.

donc des coups de pinceau d'étalement. Ces coups de pinceau sont construits à partir du flux optique de manière à représenter le mouvement qu'il décrit.

Une méthode naïve consisterait à créer un coup de pinceau pour chaque pixel du flux optique, correspondant au vecteur de déplacement représenté par le flux. Cette approche produit de très nombreux petits coups de pinceau qui rendent la peinture obtenue floue, faisant alors disparaître les effets de matières. Il est donc nécessaire d'obtenir de plus grands coups de pinceau d'étalement pour retrouver la cohérence des traces laissées par l'étalement dans la peinture traditionnelle.

Afin d'obtenir de plus grands coups de pinceau, nous segmentons le flux optique en régions et nous plaçons un certain nombre d'étalements plus longs avec un pinceau large dans chacune des régions. Pour ce faire, nous calculons la moyenne du flux optique dans chacune des régions et nous plaçons les coups de pinceau, suivant cette moyenne, aux endroits où le flux optique est maximal.

Dans notre implémentation, nous segmentons l'image suivant un *quadtree*. Nous construisons le *quadtree* pour chaque image du flux optique de l'animation. Pour cela, nous calculons la somme γ des normes des vecteurs du flux optique pour une case. Si $\gamma/A > \mu_s$ alors on subdivise la cellule du *quadtree*, A étant l'aire de la cellule du *quadtree* et μ_s un seuil défini par l'artiste. Une fois que nous avons atteint ce seuil, nous plaçons un ou plusieurs coups de pinceau avec une quantité de matière nulle dans la cellule en suivant le vecteur moyen de la cellule. Afin de ne pas introduire d'étalement sur les parties statiques (ou quasi-statiques), aucun coup de pinceau n'est créé si $\gamma/A < \mu_e$, où μ_e est encore une fois un seuil défini par l'artiste. Cette méthode améliore l'aspect des étalements, et nous verrons dans la section 7.8 les autres pistes que nous projetons d'explorer. L'algorithme 7.1 résume l'étape de création des coups de pinceau d'étalement. Si les étalements permettent d'améliorer la cohérence temporelle, ils ne sont pas suffisants pour représenter fidèlement l'image de référence. En effet, les étalements mélangent la peinture, ce qui amoindrit de plus en plus la saturation des couleurs.

7.6.2 Corrections de la peinture

L'objectif des corrections est de représenter correctement la vidéo en entrée. Pour cela, nous ajoutons des touches de matière là où les différences sont les plus

ALGORITHME 7.1 – Création des coups de pinceau d'étalement

```

F : flux optique (tableau 2D)
L : liste de coups de pinceau
Q : quadtree sur F
T : file de cellules de Q
enfile(Q, racine(Q))
tantque non vide(Q) faire
  c ← tete(Q)
   $\gamma \leftarrow \sum_{v \in F(c)} ||v||$ 
  si  $\gamma > \mu_s$  alors
    si feuille(t) alors
      subdivise(t)
    fin
    enfile(Q, fils(t))
  sinon
    si  $\gamma > \mu_e$  alors
      ajoute(L, coupsPinceauSecs(c))
    fin
  fin
fin tantque

```

importantes entre la peinture obtenue après étalements et la nouvelle image de référence. Il nous suffit de réutiliser l'algorithme décrit pour la création de la première image, en partant simplement de la peinture obtenue après étalement, au lieu de partir d'un support vide. Notons, toutefois, que nous pouvons utiliser la méthode que nous présentons pour les corrections, y compris avec une autre approche de peinture pour la première image.

Le dessin d'un coup de pinceau réduit la différence qu'il y a entre l'image et la peinture, tout du moins au point de départ du coup de pinceau. Cependant, il peut créer de nouvelles différences, qui risquent d'être suffisantes pour induire de nouvelles corrections à la prochaine image. C'est pourquoi nous avons modifié l'algorithme de base. En effet, sans prendre de précaution particulière, les parties statiques de la vidéo (qui n'ont donc pas subies d'étalement) sont corrigées, ce qui produit des bouillonnements là où aucun changement n'est attendu.

Pour résoudre ce problème, nous devons seulement corriger les régions qui sont modifiées dans la vidéo. En nous inspirant de l'approche d'Hertzmann et Perlin [HP00], nous conservons la différence cumulée \mathcal{C} des images de la vidéo ainsi que la norme cumulée \mathcal{F} du flux optique. Ces deux informations sont représentées sous la forme de deux tableaux 2D de même taille qu'une image de la vidéo en entrée. Lorsque nous passons d'une image à l'autre, nous ajoutons dans chaque cellule de ces tableaux la différence de couleur entre les deux images pour le premier et la norme du vecteur de flux optique pour le second.

Nous déclenchons une correction en un pixel lorsque la valeur $\omega\mathcal{C} + (1 - \omega)\mathcal{F}$ dépasse un seuil μ_c en ce pixel.

Lorsque la somme dépasse un seuil, nous plaçons un coup de pinceau pour corriger la peinture et la somme cumulée est remise à zéro sur la région repeinte. Cela



FIGURE 7.11 – Influence du délai d de ré-initialisation après déclenchement d'une correction. La première ligne à un délai de zéro images, nous notons les traces restant après une corrections. La deuxième ligne à un délai de trois images, les traces restantes après une corrections sont supprimées dans les images suivantes.

permet de réduire les changements dans les zones statiques, tout en corrigeant la peinture en cas de changements doux. Cette méthode élimine une partie des corrections non désirées, mais n'évite pas tous les bouillonnements. En pratique, le bruit présent dans le flux optique de la vidéo⁴ provoque l'augmentation de la somme cumulée, même dans les régions statiques. Pour tenir compte de ce phénomène, nous définissons un niveau de bruit, et nous accumulons la norme seulement si elle est supérieure à ce niveau. Les changements supérieurs au niveau de bruit sont considérés comme importants.

Lorsqu'une correction est déclenchée, nous avons remarqué qu'elle ne réduit pas toujours suffisamment la différence entre l'image de référence et la peinture. Pour améliorer la correspondance, nous proposons d'appliquer les corrections sur un petit nombre d d'images successives. Pour cela, nous réinitialisons \mathcal{C} et \mathcal{F} seulement d images après correction. Ainsi, pendant ce délai, d'autres corrections peuvent être générées (voir figure 7.11).

7.7 Contrôle du style et du rendu

Nous avons présenté comment fonctionne notre système pour produire une peinture animée à partir d'une vidéo. Nous expliquons maintenant quels moyens de contrôle nous laissons à l'artiste. Concernant l'algorithme de peinture automatique, nous offrons les mêmes libertés que l'algorithme originel d'Hertzmann [Her98] : nous avons testé avec ses paramètres pour l'expressionnisme, l'impressionnisme et le pointillisme, en retrouvant des effets similaires à sa méthode. En permettant à l'artiste de spécifier un grand nombre de paramètres, il est en mesure de donner sa touche personnelle au rendu obtenu, par exemple, en contrôlant

4. le bruit se définit par des petites variations qui ne sont pas nécessairement présentes dans le signal d'origine, mais qui apparaissent lors de la capture, notamment à cause des capteurs.

la longueur des coups de pinceau générés. De plus, en jouant sur les seuils μ_p et μ_c , il contrôle le niveau de justesse de la peinture et de l'animation par rapport aux images de référence, et peut ainsi choisir le compromis qu'il souhaite entre une bonne cohérence temporelle (peu d'ajout de coups de pinceau) et une bonne représentation (peinture proche de l'image de référence). Cela lui permet aussi de contrôler la précision de la peinture obtenue.

Une faible valeur pour μ_c donnera beaucoup de corrections tandis qu'une valeur plus importante laissera apparaître plus d'étalements. Le délai de mise à jour d permet aussi de jouer sur l'équilibre entre corrections et étalements. Un délai court entraînera peu de corrections, alors qu'un délai plus long produira des corrections sur plusieurs images de l'animation.

Concernant les coups de pinceau d'étalement, nous avons identifié plusieurs paramètres sur lesquels l'artiste peut jouer. Le premier est le seuil de déclenchement d'un étalement, c'est-à-dire la quantité de mouvement à partir de laquelle le système déplace la matière. Il peut ainsi trouver un compromis entre fidélité au mouvement et cohérence temporelle. En effet, en l'absence d'étalement, les parties les plus fixes ne sont pas modifiées au cours de l'animation et, dans les cas extrêmes, elles seront uniquement corrigées (avec l'ajout de coups de pinceau chargés de peinture) sans être étalées. Au contraire, si ce seuil est faible, l'animation est plus agressive et on perçoit plus distinctement les mouvements, même dans le cas de légères variations. Nous lui laissons aussi la possibilité de définir la taille et le nombre de coups de pinceau d'étalement. Une fois la segmentation des zones à étaler effectuée, le système place un nombre des coups de pinceau d'étalement plus ou moins important, avec une pression plus ou moins forte. Cela permet à l'artiste de donner différents effets d'étalement, soit très suggéré et léger, soit plus fort, de telle sorte que la trace laissée par le coup de pinceau d'étalement soit visible.

7.8 Résultats et perspectives

Notre modèle de simulation de peinture, qui est basé principalement sur le modèle de Kalaidjian, donne des résultats concluants pour notre objectif. Il permet de mimer une grande partie de l'apparence de la peinture traditionnelle telle que le mélange des couleurs et les éléments visuels de matière produite par le pinceau. Il permet également de créer de nombreuses formes de pinceau et donc de reproduire les effets d'une large gamme de styles.

Du point de vue de la génération automatique de peinture, nous obtenons là aussi des résultats prometteurs offrant une richesse graphique à nos animations. Nous avons cependant remarqué que certains coups de pinceau sont trop longs par rapport à la zone qui méritait d'être corrigée ; il serait intéressant de développer une autre technique pour la génération de coups de pinceau de correction, en prenant en compte le flux optique par exemple.

Une technique utilisée par certains artistes est de prendre plusieurs photos de la peinture à différentes étapes de sa création. Aussi nous avons proposé d'utiliser une image intermédiaire entre étalements et corrections pour recréer le même genre d'effet. L'utilisation d'images intermédiaires, par exemple en n'appliquant

qu'une partie des étalements et des corrections, donne l'impression que le rendu se construit petit à petit et que cette construction fait partie de l'animation.

Nous avons, en outre, remarqué que lorsque le style utilisé perturbe la couleur prise au point de départ dans l'image de référence, lors des corrections, il ne peut plus y avoir de cohérence entre la nouvelle couleur choisie pour corriger et celle déjà présente dans la peinture. En effet, pour corriger une région qui se déplace, les nouvelles couleurs utilisées ne sont pas les mêmes que celles utilisées pour la peinture précédente. Ce qui produit des changements brutaux que nous souhaiterions éviter.

Du point de vue du contrôle de l'artiste, il est possible d'ajouter un contrôle local, c'est-à-dire que pour une image donnée, l'artiste pourrait supprimer un certain nombre de coups de pinceau générés (et donc les enlever dès leur création) ou encore ajouter manuellement des coups de pinceau et des étalements. Nous n'avons pas implémenté ces options, mais puisque nous conservons l'ensemble des données définissant les coups de pinceau, c'est une extension facilement envisageable.

Le contrôle des étalements n'est qu'un premier essai ; il serait intéressant de pouvoir donner un sens à leur création, par exemple en faisant varier la pression et la taille des étalements en fonction du mouvement, tout en laissant l'artiste suggérer une émotion à travers les étalements. De plus, nous pensons qu'il est possible de trouver une segmentation du flux optique plus intelligente que le simple *quad-tree*. L'idéal serait d'obtenir une segmentation hiérarchique de l'image de référence prenant en compte le flux optique et la couleur. Il faudrait, par exemple, qu'un personnage soit identifié comme tel et décomposé (bras, jambes, mains, pieds) ; il serait alors possible de plus ou moins descendre dans la hiérarchie suivant le mouvement de chacune des parties.

La comparaison avec les méthodes qui déplacent les marques pour obtenir la cohérence temporelle n'est pas évidente. En effet, l'aspect obtenu n'est pas le même. D'un point de vue esthétique, cette deuxième approche est plus fidèle au rendu obtenu dans la peinture animée. Par contre, il y a plus de changement que dans les méthodes basées sur le déplacement des marques. Nous pensons qu'il serait intéressant d'utiliser les deux approches combinées : d'une part d'avoir l'aspect et le mélange des couleurs propres à la peinture animée et d'autre part déplacer certains coup de pinceau pour réduire les corrections.

Deuxième partie

Rendu en demi ton

Rendu en demi-tons

CETTE PARTIE présente le travail que j'ai réalisé au cours d'un séjour au laboratoire d'informatique graphique de l'université de Montréal (LIGUM) sous la direction de Victor Ostromoukhov. Le sujet sur lequel s'est porté notre intérêt est la distribution de points pour les systèmes d'impression. Nous présentons tout d'abord les motivations de notre travail en introduisant le domaine et les travaux précédents (chapitre 9), puis nous détaillons nos contributions (chapitres 10 et 11).

8.1 Ton et distribution

Une image numérique est un ensemble de pixels représentant les couleurs à reproduire pour l'affichage. La couleur peut être définie comme l'association d'une teinte et d'un ton¹: la teinte donne la couleur *brute* et le ton donne sa *force*. En faisant varier le ton, la couleur passe de la teinte au blanc. Généralement, les dispositifs d'affichage (écran, imprimantes etc.) ne peuvent représenter qu'un nombre limité de tons pour chaque teinte. Dans la suite du document, nous exprimons le ton en pourcentage. Celui-ci représente l'intensité lumineuse du ton entre la teinte brute, par exemple le noir, à 0 % et le blanc à 100 %.

Le rendu en demi-tons (*halftoning*) est un procédé de représentation d'une image en ton continu (ou avec un grand nombre de tons possibles) sur un dispositif d'affichage ne pouvant afficher qu'un nombre limité de tons. Une imprimante, noir et blanc par exemple, ne peut produire que du noir, le blanc de la feuille constituant le second ton. Un écran ne peut généralement produire que 256 tons différents pour chacune des trois teintes : rouge, vert et bleu. Pourtant, il est possible d'utiliser de tels dispositifs pour obtenir un résultat présentant des variations de tons plus fines. Le rendu en demi-tons tire parti du fonctionnement de notre système visuel, en particulier de ses limites, pour permettre aux dispositifs d'affichage de représenter plus de tons *perçus* que ceux effectivement *affichables* par le dispositif.

1. en peinture traditionnelle on parle de teinte, saturation et clarté.

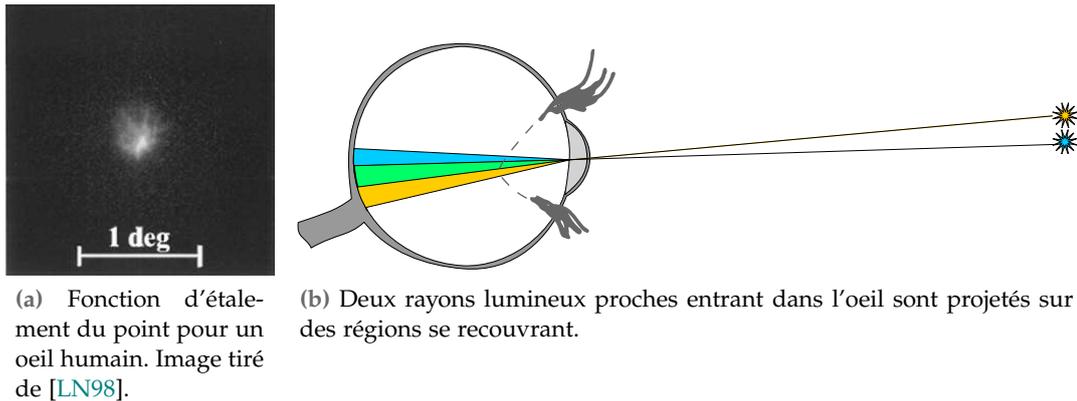


FIGURE 8.1 – Lorsqu'un rayon lumineux passe dans l'œil, l'image de celui-ci sur la rétine est légèrement étalée. Cela provoque le mélange de deux rayons proches lorsqu'ils arrivent sur la rétine.

8.2 Système visuel humain

L'ensemble du processus de la vision et de la perception humaine n'est pas encore clairement identifié par les neurologues et les biologistes, mais plusieurs pistes expliquent certaines propriétés. L'étude du système visuel humain montre que nous *intégrons* (mélangeons) les tons et les teintes présents dans une petite région. Cela est dû en particulier à la fonction d'étalement du point de l'œil qui est la manière dont un point lumineux est étalé sur la rétine après être passé dans l'œil [LN98], voir figure 8.1a. L'étalement s'explique par la perturbation du chemin de la lumière lors du passage dans la cornée, la pupille, le cristallin et le corps vitré avant de toucher la rétine. Deux points lumineux proches vont créer deux régions qui se recouvrent sur la rétine, comme présenté schématiquement figure 8.1b.

Une autre propriété importante est la sensibilité à l'orientation du système visuel humain à l'orientation de ce qu'il perçoit. Ceci est mis en évidence en regardant une série de lignes parallèles. En effet, si on arrive à distinguer des lignes noires sur fond blanc, horizontales ou verticales jusqu'à une certaine densité, cette densité est plus faible lorsqu'elles sont à 45° .

Le cerveau détecte aussi les répétitions et les alignements pour former des groupements. Nous percevons des structures dès lors que l'on voit des alignements d'entités ou des groupes répétés d'entités.

Ces observations sont les clés du fonctionnement des algorithmes de rendu en demi-tons. Dans les systèmes d'impression, l'obtention des différents tons résulte du placement des points d'encre (d'une teinte et d'un ton fixe) sur le papier. Notre système visuel perçoit la proportion d'encre sur le fond blanc comme un ton ayant cette même proportion. La répartition des points, leur distribution, doit tenir compte des caractéristiques du système visuel humain et être adaptée aux contraintes techniques du matériel d'impression. Du point de vue du système visuel humain, la distribution des points d'encre idéale est de type proche du *bruit bleu*, que nous avons introduit section 4.1, avec de légères anisotropies dans la répartition des fréquences pour tenir compte de la sensibilité à l'orientation. Pour plus de détails se référer aux livres traitant de la vision, notamment celui de Marr [Mar82] et de Palmer [Pal99].

8.3 Fonctionnement des imprimantes

Nous présentons dans un premier temps le fonctionnement des imprimantes ainsi que le matériel principalement utilisé de nos jours. Nous détaillons dans le chapitre suivant les méthodes existantes pour produire des rendus en demi-tons.

8.3.1 Principe de base

Une imprimante transforme une image numérique, donc virtuelle, en un dessin, physique, sur une feuille de papier. Pour cela, elle dépose de l'encre sur le papier pour représenter l'image. Pour des raisons techniques et économiques, une imprimante possède peu d'encres différentes, c'est-à-dire peu de teintes différentes. Typiquement, une imprimante noir et blanc a seulement de l'encre noire, et une imprimante couleurs dispose de trois à huit couleurs (généralement noir, cyan, magenta et jaune). De plus, elle ne peut produire que très peu de tons pour chacune de ces teintes. En effet, la plupart des imprimantes ne peuvent produire que des régions couvertes par une ou plusieurs encres ou laissant le papier apparent : soit deux tons pour chaque encre. Certains dispositifs avancés arrivent à produire un ton intermédiaire.

Pour produire les régions couvertes par l'encre sur le papier, on peut représenter l'imprimante comme une grille de cellules *adressables*. Ce sont ces cellules que l'imprimante peut remplir. Le programme de contrôle de l'imprimante est exécuté lorsqu'un document est imprimé, il effectue la transformation des données numériques en un contrôle mécanique de l'imprimante pour produire les positions et teintes des cellules à imprimer. Les cellules sont généralement représentées par des points, mais leur forme, une fois imprimée, dépend du type d'encre, du papier, et du type d'imprimante utilisés. Dans la suite du document, nous parlerons de points adressables et de distribution de points. La figure 8.2 montre un exemple d'impression numérisée et grossie, ce qui permet de distinguer la forme des régions couvertes par l'encre sur le papier.

8.3.2 Types d'imprimantes

Les fabricants d'imprimantes développent différentes technologies pour s'adapter aux marchés grand public et professionnel (tant les industriels de l'image que les entreprises de tout secteur d'activité). Deux grandes familles d'imprimantes existent aujourd'hui sur le marché, nous en présentons le fonctionnement.

Imprimantes jet d'encre

Une imprimante jet d'encre est capable de propulser une goutte d'encre pour chacun des points adressables. Chaque goutte est alors absorbée par le papier. Actuellement, les imprimantes jet d'encre sont capables d'imprimer à 600 DPI², voire

2. DPI signifie *dot per inch*, ou point par pouce. 600 DPI correspond à des points adressables espacés de 42.33 μm

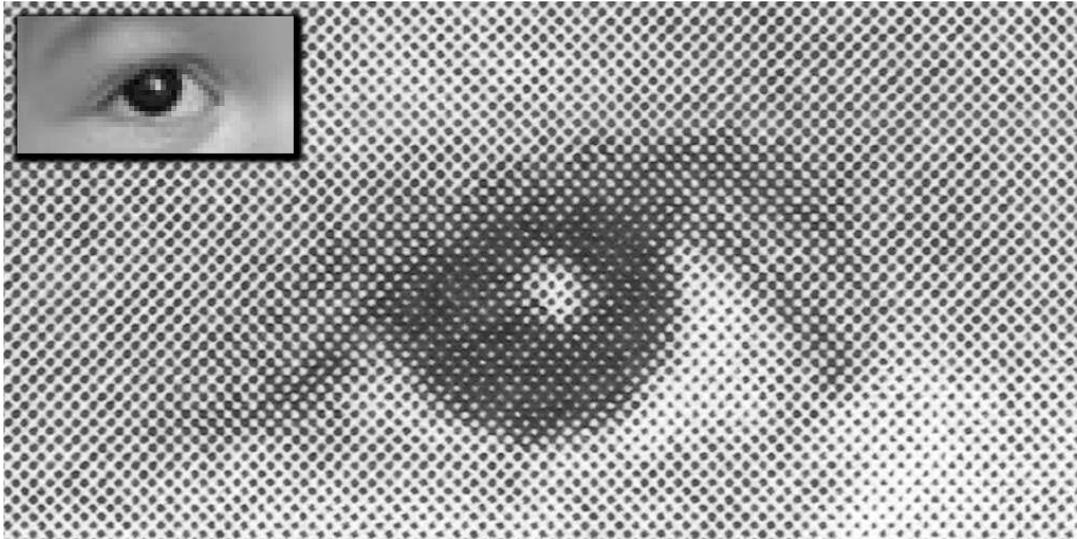


FIGURE 8.2 – Numérisation d’une impression obtenue avec une imprimante laser, grossie dix fois. En encart, les données numériques envoyées à l’imprimante.

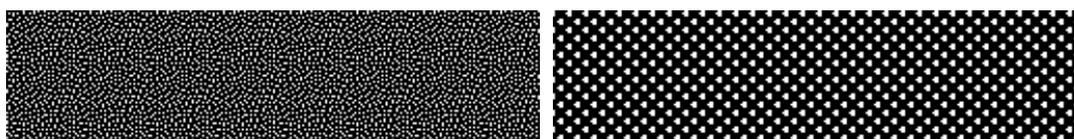
beaucoup plus (on trouve notamment des imprimantes 4200×1200 DPI). Le volume des gouttes d’encre produites varie entre quatre et vingt pico litres, ce qui revient à des gouttes de $50 \mu\text{m}$ de rayon environ. En fait, les valeurs de DPI font références à la finesse de la cellule adressable et non à la taille réelle des gouttes d’encre imprimées, ce qui est en fait un argument plus commercial que véritablement représentatif des performances d’une imprimante. En effet, la qualité de l’imprimante est liée à la taille des gouttes d’encre qu’elle peut propulser, à la précision avec laquelle cette goutte d’encre est placée sur le papier ainsi qu’à la qualité de l’encre et du papier qui régit le dessin formé par cette goutte d’encre.

Du point de vue économique, cette technologie est bon marché (on trouve des imprimantes jet d’encre à partir de 50€) mais l’encre est chère : souvent une recharge en encre — qui comprend la tête d’impression — coûte le même prix que l’imprimante neuve. Ce type d’imprimantes est destiné au grand public, qui n’imprime que peu de pages par an.

Imprimantes laser

L’autre grande famille est celle des imprimantes laser. Contrairement aux imprimantes jet d’encre, les encres utilisées ne sont pas liquides mais sous forme de toner : une poudre fine constituée d’un mélange de résine et de pigment à l’état solide. L’image est dessinée par un laser sur un tambour et celui-ci se charge magnétiquement suivant le dessin du laser. Le toner est polarisé inversement au tambour et vient se placer sur celui-ci, sur ses zones polarisées. Puis, le toner est déposé sur une feuille par le tambour. La feuille est ensuite chauffée pour faire fondre la résine du toner et fixer les pigments. Le procédé électromagnétique oblige à prendre une certaine quantité minimum de toner, ce qui implique que les points ne peuvent pas être trop petits. Par contre, l’utilisation du laser permet d’obtenir des tracés très précis (supérieurs à 1200 DPI)

Ce type d’imprimante est vendu beaucoup plus cher que les imprimantes jet d’encre, de 300€ à 1000€ pour une imprimante de qualité correcte. Par contre, le



(a) Modulation de fréquence, le ton est représenté par une densité de points de même taille.

(b) Modulation d'amplitude, le ton est représenté par la taille des éléments de trame, disposés à des emplacements fixes

FIGURE 8.3 – Exemple de ton à 20% obtenu en plaçant des points suivant les deux approches principalement employées en rendu en demi-tons.

coût d'impression à la page est bien moins important car le toner est bon marché (65€ pour 16 000 pages). Cette catégorie d'appareil est donc destinée aux professionnels qui impriment un grand nombre de pages par mois.

8.3.3 Dispersion et groupement de points

Comme nous l'avons vu, les contraintes matérielles ne sont pas les mêmes suivant le type d'imprimante. Aussi, il existe plusieurs types d'algorithmes de rendu en demi-tons, adaptés aux différents matériels. Deux catégories de méthodes sont identifiables : la modulation d'amplitude et la modulation fréquence. La modulation d'amplitude suit le principe suivant : des points de trame sont placés à des positions fixes, et leur taille varie pour représenter le ton. En fait, les points adressables imprimés sont groupés pour former un point de trame. La trame groupée est définie par la répétition des points de trame : ces points sont disposés avec une certaine fréquence de répétition, et s'alignent suivant un certain angle avec l'horizontale.

La modulation de fréquence consiste, elle, à distribuer des points de même taille avec une fréquence représentant le ton. En d'autres termes, le ton est traduit par la densité des points. Le terme trame dispersée est utilisé pour désigner la répartition des points imprimés, même si, comme nous le verrons au chapitre suivant, il n'y a pas forcément répétition d'éléments de trame.

8.3.4 Traitement de la couleur

Nous n'avons pas encore différencié niveaux de gris et couleur. La couleur est obtenue en *superposant* les différentes encres, c'est-à-dire en faisant une passe par encres du système d'impression. Pour cela, on calcule une composante par encres à partir de l'image à imprimer, puis on effectue un rendu en demi-tons pour chaque composante, sur la même feuille.

Dans le cas de la modulation de fréquence, il suffit de faire des passes *indépendantes* les unes des autres pour chacune des encres. La modulation d'amplitude est plus délicate. En effet, l'utilisation de la même trame pour toutes les encres peut provoquer trop de recouvrement et donc une perte de saturation. D'un autre côté, si les trames sont différentes pour chacune des encres, l'apparition d'effets de moiré [AHO94] est difficile à contourner. L'effet de moiré est l'apparition de basses fréquences lorsque des trames de fréquence proche sont superposées, la figure 8.4

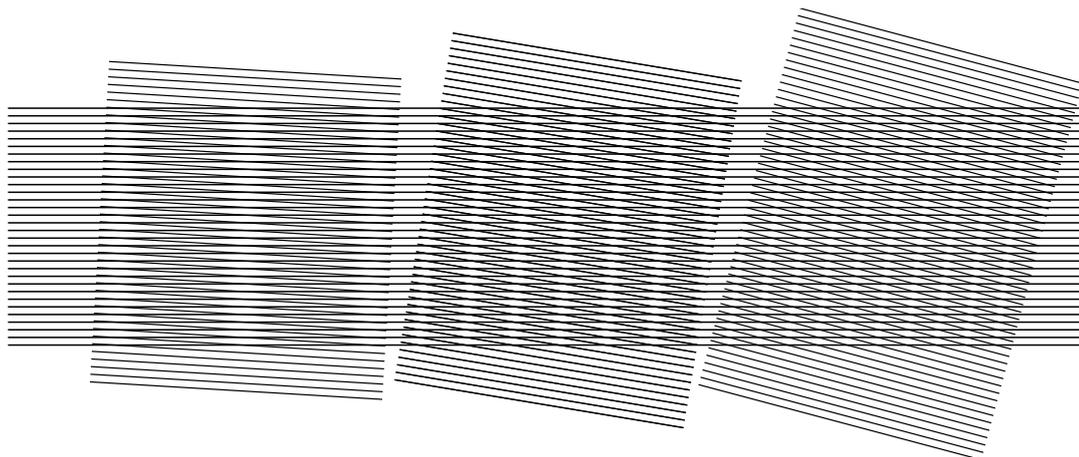


FIGURE 8.4 – L’effet de moiré se produit lorsque deux motifs de même fréquence sont superposés avec un angle légèrement différent par rapport à l’horizontale. Un motif basse fréquence apparaît alors. Plus la différence est faible, plus la fréquence du motif parasite est basse.

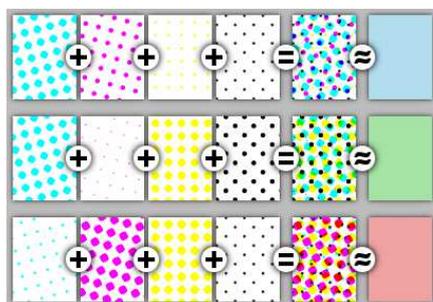


FIGURE 8.5 – Rosettes obtenues avec des trames groupées, l’angle de chaque trame étant différent pour chacune des encres. Suivant les propriétés du système visuel humain, l’angle choisi pour l’encre noire est de 45°, 15° pour le magenta, 75° pour le cyan (une différence de 30° minimise les moirés pour trois trames), la trame jaune est à 0°. Image du domaine public.

illustre le principe. En fait, il faut essayer de prendre des trames dont la répartition forme des rosettes stables (écart de 30° entre chaque trame, voir figure 8.5). Comme notre œil n’est pas sensible de la même manière aux différentes couleurs, les directions et fréquences des différentes trames sont choisies en conséquence. Par exemple, notre système visuel étant plus sensible au noir qu’au jaune par exemple, un alignement à 45° est utilisé pour le noir et à 0° pour le jaune.

8.3.5 Traitements particuliers

Sans entrer dans les détails du fonctionnement complet d’un système d’impression, nous voulons simplement mentionner qu’il y a plusieurs étapes pour obtenir le rendu en demi-tons d’une image. Ces étapes sont mises en place pour tenir compte des différentes contraintes du système, comme par exemple l’élargissement du point de trame.

L’élargissement du point de trame désigne un phénomène déterminant pour la qualité tout au long du processus d’impression. C’est la déformation que subit le point de trame au cours des différentes étapes de l’impression. Formellement, c’est l’écart mesuré entre la valeur initiale du point que l’on veut imprimer et celle mesurée sur la feuille imprimée. Pour donner un ordre de grandeur, l’élargissement moyen varie, pour un groupe de points donnant un ton de 50 %, entre 8 % et 25 % (c’est-à-dire 50 % devient 58 % à 75 %). Il est donc crucial de maîtriser l’élargissement du point de trame lors de l’impression. Ce contrôle est fait par les

constructeurs qui définissent une correction de ton en fonction de l'élargissement attendu. Il existe plusieurs modèles d'élargissement et de recouvrement des points de trame. Pour plus de détails, on pourra se référer à l'article de Yule et Nielsen [YN51]. Dans nos travaux nous nous concentrons sur la création d'éléments de trame. Les livres de Sharma [Sha02] et de Kang [Kan99] présentent plus en détail l'ensemble du processus d'impression.

État de l'art du rendu en demi-tons

LES CONTRAINTES techniques et la connaissance du système visuel humain guident la création des rendu en demi-tons. Nous allons maintenant décrire les principales approches pour obtenir un rendu en demi-tons, en mettant en avant les limitations de celles-ci. Pour plus de détails, Hersch et Ostromoukhov [HO95] proposent une introduction à la génération d'images en demi-tons.

9.1 Matrice de seuillage

Prenons l'exemple d'une imprimante noir et blanc : comme le système visuel humain mélange l'information perçue sur une petite région et que l'imprimante peut tracer uniquement des points noirs, pour obtenir le rendu en demi-tons, il faut définir un moyen de distribuer les points sur une petite région en conservant la proportion correspondant au ton voulu.

Une méthode est de découper la grille de points adressables en petites régions de même taille et de même forme, puis de choisir le nombre de points à imprimer suivant le ton demandé, pour chacune des régions. Pour cela, une matrice de seuillage, ayant la forme de cette région, est utilisée pour choisir comment placer les points et produire un élément de trame (le dessin sur une région). Cette matrice de seuillage est remplie avec les valeurs de l'intervalle $[1, n]$ où n est le nombre de points adressables présents dans une région. Le remplissage est fait de telle sorte que chaque valeur apparaisse une seule fois dans la matrice. En conservant uniquement les points correspondant à des valeurs de seuil supérieures ou égales au ton

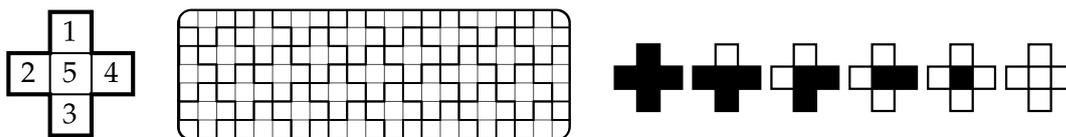


FIGURE 9.1 – Une matrice de seuillage et le pavage de la grille de points adressables par cette matrice. À droite, les éléments de trame associés pour les six intensités productibles par cette matrice.

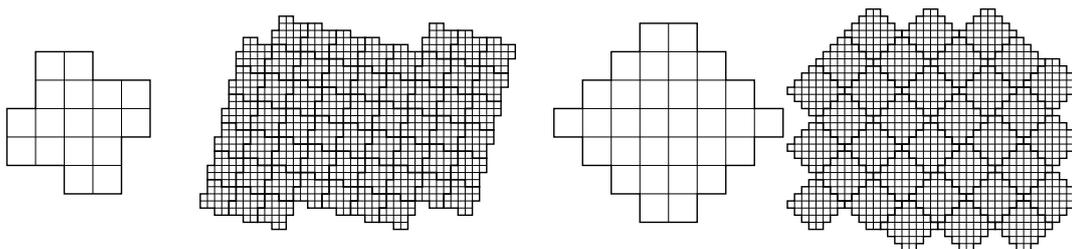


FIGURE 9.2 – Deux formes de matrice de seuillage (17 et 32 valeurs de seuil) et le pavage de la grille adressable associé.

en entrée, le nombre de points obtenus correspond bien au ton voulu. La figure 9.1 illustre ce fonctionnement pour un élément de trame de taille cinq.

Pour imprimer une image, la matrice de seuillage est répétée pour couvrir la grille de points adressables de l'imprimante. La figure 9.2 montre des formes de matrices de seuillage de différentes tailles, ainsi que leur répétition pour couvrir la grille adressable. La position des valeurs dans la matrice de seuillage donne l'ordre dans lequel on dessine les points adressables pour produire les éléments de trame pour chacun des tons. Suivant l'organisation des valeurs, l'élément de trame produit sera une trame groupée (modulation d'amplitude donnant des points de trames) ou une trame dispersée (modulation de fréquence).

Les éléments de trame produits respectent une contrainte d'empilement, c'est-à-dire que, si un point est présent dans l'élément de trame pour un ton donné, il sera présent dans tous les tons plus foncés. De la même manière, si un point n'est pas sélectionné pour un ton, il ne le sera pas dans tous les tons plus clairs. La valeur de seuil dans la matrice donne le ton qui provoque le changement de sélection du point dans l'élément de trame.

9.1.1 Matrice de seuillage pour éléments de trame groupés

Les matrices de seuillage produisant des éléments de trame groupés (ou matrice groupée) sont généralement construites à la main. Lorsque l'on veut obtenir des éléments de trame groupés, il faut que le passage d'un ton à l'autre (par exemple en allant du blanc vers le noir) se fasse en ajoutant le nouveau point de façon juxtaposée avec ceux déjà sélectionnés dans le ton précédent. Dans la mesure du possible, le motif pour un ton inférieur à 50 % doit être le négatif du ton complémentaire supérieur à 50 % (par exemple le motif à 25 % est le négatif du motif à 75 %, voir figure 9.3).

Les matrices groupées sont utilisées pour les imprimantes laser. La taille de la matrice de seuillage joue un rôle important : cette taille donne le nombre de tons imprimables, autrement dit elle détermine la quantification faite par le processus de rendu en demi-tons (marques vertes sur la figure 9.3). L'idéal est d'avoir au moins cent valeurs de sortie. Mais la taille de la matrice influe aussi sur la finesse des détails imprimables comme le montre la figure 9.4. Plus la matrice est petite, mieux les détails sont représentés.

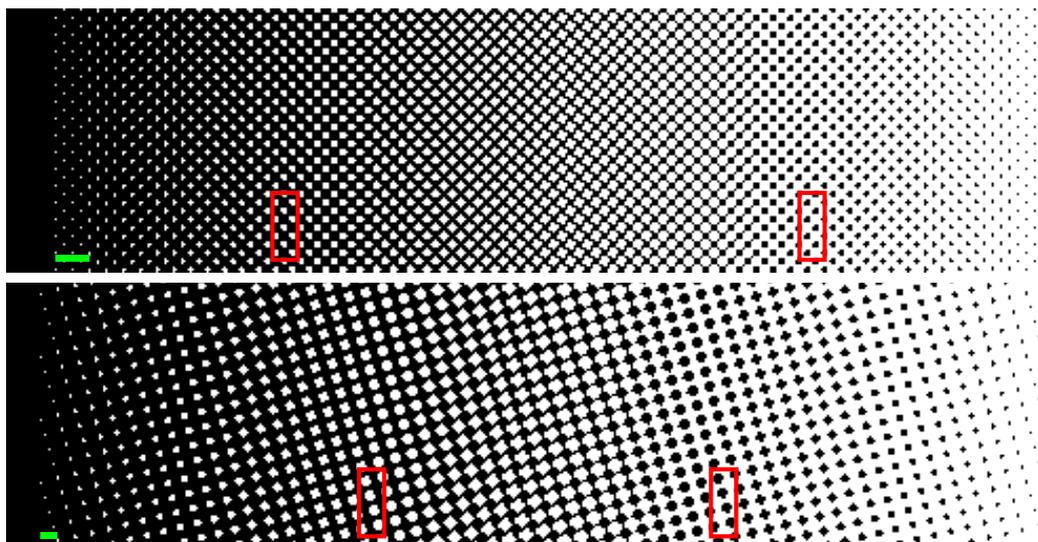


FIGURE 9.3 – Rampes (ton variant de 0 à 100 %) rendues avec des matrices de seuillage à trame groupée de 32 et 60 valeurs. En rouge sont encadrés deux tons complémentaires dont les motifs sont négatifs l'un de l'autre. La marque verte donne la largeur d'une zone où la matrice de seuillage produit le même élément de trame et définit donc la taille de la quantification faite par le rendu en demi-tons.



FIGURE 9.4 – Deux images rendues avec des matrices de seuillage à trame groupée de taille 32 et 60. Les détails et le nombre de tons varient suivant cette taille.

9.1.2 Matrice de seuillage pour éléments de trame dispersés

Suivant le même principe de comparaison à un seuil, les matrices de seuillage peuvent être construites pour disperser les points imprimés (matrice dispersée) pour chaque ton en entrée. Un nouveau point est alors placé le plus loin possible des points déjà sélectionnés. Ces matrices de seuillage sont généralement construites en suivant une optimisation du placement des valeurs de seuil. La question est alors d'éviter les alignements perçus, que l'on peut mesurer avec le spectre de puissance. Notre contribution se situe dans cette classe d'approche; nous allons par conséquent détailler quelques une des techniques classiques.

Bayer [Bay73] utilise un schéma de subdivision pour construire des matrices dispersées carrées, de taille $2^n \times 2^n$, ayant une grande distance entre les points sélectionnés pour tous les niveaux. La figure 9.5 montre la construction d'une telle matrice. Elle présente des alignements très fortement perçus qui se retrouvent dans le spectre de puissance, figure 9.7, sous forme de pics de fréquences.

Pour casser ces alignements, et se rapprocher d'une distribution bruit bleu des points sélectionnés, Ulichney [Uli93] propose d'utiliser une méthode d'optimisation qu'il appelle *void-and-cluster*. L'objectif est de déterminer les valeurs de seuil

d'une matrice de seuillage rectangulaire. Nous basons notre travail sur l'approche d'Ulichney, nous allons donc détailler son fonctionnement. *Void-and-cluster* part d'un élément de trame prototype ayant les propriétés de bruit bleu pour un niveau de départ. Ensuite, la méthode consiste à assigner des valeurs de seuil à chacun des points présents dans l'élément de trame prototype. Enfin, les valeurs de seuil sont assignées aux positions qui n'avaient pas encore été assignées, pour remplir la matrice de seuillage.

L'idée principale est d'utiliser un filtre qui permet de déterminer la place du plus grand *void* (le plus grand trou) et celle du plus petit *cluster* (l'amas de points le plus serré) dans un élément de trame. Il définit ces places par le maximum et le minimum de la trame prototype \mathcal{T} filtré par un noyau gaussien :

$$J_G(x, y) = \sum_{(x', y') \in \mathcal{T}} \text{vacKernel}(r) = e^{-\frac{r^2}{2\sigma^2}}$$

où r est la distance entre la position considérée et les positions avoisinantes et σ la distance de mi-hauteur du filtre gaussien. Ulichney utilise $\sigma = 1.5$ cellules, ce qui présente les meilleurs résultats d'après ses expérimentations. Lors du filtrage, l'élément de trame est considéré comme un espace torique, ce qui permet de tenir compte de sa réplcation lors de la production du rendu en demi-tons, comme le montre la figure 9.6. Le filtre gaussien tient compte non seulement de la distribution locale de points, car les points proches ont une influence plus forte que les points éloignés, mais aussi de la distribution globale car sa taille est *infinie*. Dans la pratique, il suffit que le support du filtre soit suffisant pour couvrir deux fois l'élément de trame.

Lors de l'assignation des valeurs de seuil, deux tableaux sont mis à jour : le premier est un tableau binaire, l'élément de trame prototype (dont on ne connaît pas encore la matrice de seuillage), le second est la matrice de seuillage à construire. L'algorithme 9.1 détaille les étapes d'initialisation, c'est-à-dire l'obtention d'une distribution de points dans l'élément de trame prototype. L'algorithme 9.2 explicite le processus d'assignation des valeurs de seuil aux points présents dans l'élément de trame prototype.

L'élément de trame prototype initial sert de contrainte pour l'optimisation. Le choix du niveau initial influence la matrice de seuillage obtenue, Ulichney préconise de prendre un niveau de 10 % (un niveau de 0 % produisant une matrice de

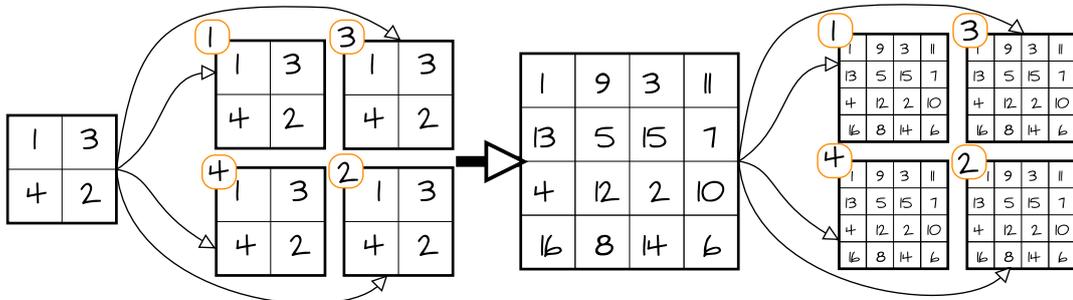


FIGURE 9.5 – Construction d'une matrice de seuillage suivant le schéma de Bayer. Pour passer d'une étape à l'autre, la matrice est dupliquée quatre fois avec un déphasage pour répartir les valeurs de seuil.

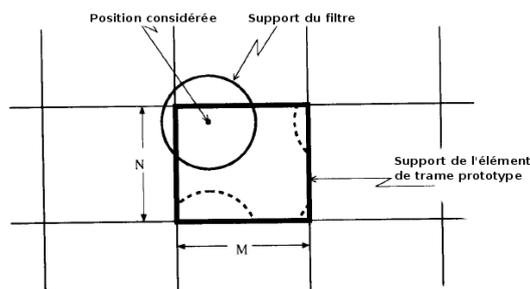


FIGURE 9.6 – Application du filtre gaussien sur l’élément de trame prototype. L’élément de trame est considéré comme torique, ce qui revient à prendre en compte sa réplification pour couvrir la grille adressable de l’imprimante. Image tiré de [Uli93].

ALGORITHME 9.1 – *Void-and-cluster*, initialisation du motif initial.

ETP : tableau de $n \times m$ cellules (« 0 » ou « 1 »)

initialise ETP à « 0 »

met 10% de « 1 » dans ETP

répéter

$v \leftarrow \text{position}(\min(G(\text{ETP})))$

$\text{ETP}(v) \leftarrow \text{« 1 »}$

$c \leftarrow \text{position}(\max(G(\text{ETP})))$

$\text{ETP}(c) \leftarrow \text{« 0 »}$

jusqu’à c égal v

seuillage correspondant à l’approche de Bayer). Ulichney [Uli95] montre que la précision numérique utilisée joue un rôle dans le choix du minimum (*void*) et que le paramètre σ influence les éléments de trame produits [Uli94]. Lin [Lin94] améliore l’approche d’Ulichney en prenant un filtre qui imite mieux le système visuel humain que le filtre gaussien. Ancin et coll. [ABS99] améliorent la distribution des valeurs de seuil en modifiant l’approche d’Ulichney ; ils utilisent notamment une relaxation de Lloyd pour l’initialisation et différents critères pour le choix du plus grand *void* et du plus petit *cluster*.

Mitsa et Parker [MP91] proposent la méthode de *blue noise mask*. Ils construisent une matrice de seuillage dont les valeurs de seuil sont choisies de telle manière qu’ils obtiennent une distribution de type bruit bleu des points pour tous les niveaux d’entrée possibles. Pour cela, ils partent d’une distribution pour le niveau 50 % pour un élément de trame, puis ils appliquent un *filtre bruit bleu* sur la transformée de Fourier (ce filtre creusant les basses fréquences du spectre). Ils appliquent ensuite la transformée inverse, pour obtenir un nouvel élément de trame, non plus binaire, mais à valeurs réelles. Ils cherchent alors la position du maximum dans cet élément de trame pour déterminer le point à enlever dans le niveau suivant. Puis, ils stockent la valeur de seuil correspondante dans la matrice de seuillage et recommencent le processus. Ils obtiennent ainsi une matrice de seuillage donnant une bonne répartition des valeurs de seuil. Travailler dans l’espace fréquentiel leur permet de partir de l’objectif, c’est-à-dire obtenir une distribution bruit bleu, pour construire la matrice de seuillage, alors que les autres approches utilisent le domaine fréquentiel seulement comme mesure.

Allebach et Lin [AL96] construisent une matrice de seuillage par optimisation. Comme l’approche d’Ulichney [Uli93], ils commencent par optimiser un élément

ALGORITHME 9.2 – *Void-and-cluster*, numérotation.

```

M : matrice de seuillage à construire ( $n \times m$  cellules)
initialise M à « 0 »
ETPtemp  $\leftarrow$  ETP
pour  $i=k-1$  à 0 faire
  c  $\leftarrow$  position(max(G(ETPtemp)))
  ETPtemp(c)  $\leftarrow$  « 0 »
  M(c)  $\leftarrow$  « i »
fin pour
ETPtemp  $\leftarrow$  ETP
pour  $i=k$  à  $n \times m/2$  faire
  v  $\leftarrow$  position(min(G(ETPtemp)))
  ETPtemp(v)  $\leftarrow$  « 1 »
  M(v)  $\leftarrow$  « i »
fin pour
inverse « 0 » et « 1 » dans ETPtemp
pour  $i= n \times m/2+1$  à  $n \times m - 1$  faire
  v  $\leftarrow$  position(min(G(ETPtemp)))
  ETPtemp(v)  $\leftarrow$  « 1 »
  M(v)  $\leftarrow$  « i »
fin pour

```

de trame prototype pour un ton initial. Allebach et Lin construisent ce niveau par l'approche qu'ils appellent *direct binary search*. Cette méthode consiste à minimiser une fonction de coût représentant la juste représentation du ton initial suivant un modèle du système visuel humain. Ils définissent deux types d'opérations : le changement d'état (un point noir devient blanc et inversement) et l'échange d'état (un point échange son état avec l'un de ces huit voisins de l'élément de trame). Leur approche est itérative et à chaque itération ils choisissent l'opération qui minimise la fonction de coût.

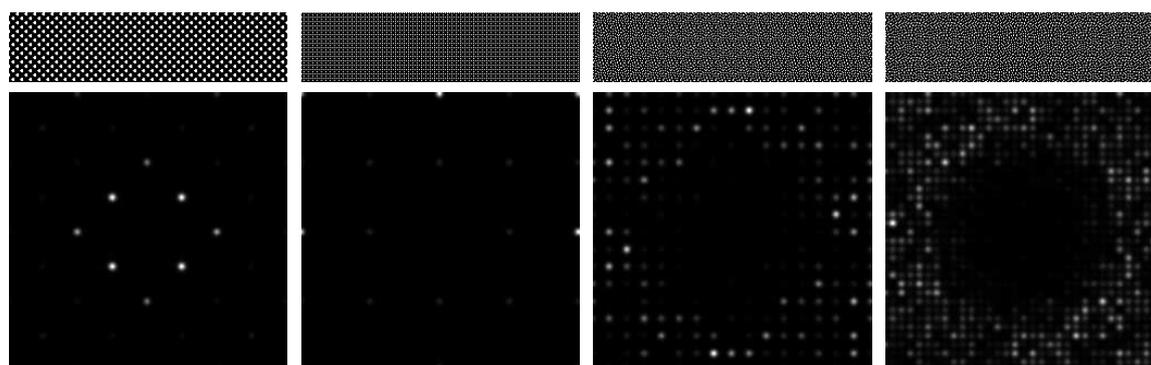
L'algorithme d'Allebach et Lin converge vers une bonne distribution respectant le ton initial et ayant les propriétés de bruit bleu (suivant le modèle de système visuel utilisé). Ensuite, pour construire le niveau suivant (plus clair par exemple), Allebach et Lin suppriment aléatoirement autant de points noirs que nécessaire, puis appliquent la même optimisation. Mais ils ne permettent pas le changement d'état (pour conserver le ton) et les échanges d'état ne peuvent intervenir qu'entre les points noirs et ceux qui viennent d'être supprimés (pour satisfaire la contrainte d'empilement). Une fois que les éléments de trames sont obtenus pour tous les niveaux possibles, ceux-ci forment une fonction discrète $p[x, y; a]$, où x et y sont les coordonnées des points adressables et a est le ton. Cette fonction respecte la contrainte d'empilement :

$$p[x, y; a] = 1 \Rightarrow p[x, y; b] = 1 \quad \forall a > b$$

La matrice de seuillage m correspondante est alors

$$m[x, y] = \{t | p[x, y, t - 1] = 0 \text{ et } p[x, y, t] = 1\}$$

Li et Allebach [LA00] proposent une extension de cette méthode d'optimisation, en faisant un compromis sur la contrainte d'empilement imposé par l'utili-



(a) groupée, 32 valeurs (b) Bayer 16 valeurs (c) *void-and-cluster* 256 (d) *void-and-cluster* 1024

FIGURE 9.7 – Spectres de Fourier pour les techniques utilisant des éléments de trame.

Dans le cas d'une trame groupée et de Bayer, les répétitions et les alignements apparaissent sous forme de pics dans les basses fréquences. Même dans le cas de *void-and-cluster* la réplication de l'élément de trame provoque des spectres où un petit nombre de fréquences est effectivement présent.

sation d'une matrice de seuillage. En effet, plutôt que de prendre une matrice de seuillage, Li et Allebach utilisent directement la fonction discrète p pour réaliser le seuillage. Le niveau de sortie (imprimer ou non un point) est donné par cette fonction par une simple lecture (la fonction est stockée sous la forme d'une matrice 3D). Li et Allebach optimisent chaque niveau en suivant l'approche *direct binary search*. Outre la prise en compte d'un modèle du système visuel humain, la fonction de coût a une composante représentant la contrainte d'empilement. En effet, le non respect de la contrainte d'empilement crée du grain (haute fréquence) lors de changement de ton (sur une rampe par exemple). C'est pourquoi ils tiennent compte de la perception du grain dans la fonction de coût et une pondération entre la contrainte d'empilement et le modèle du système visuel humain permet de faire un compromis entre bonne distribution et respect de la contrainte d'empilement.

Li et Allebach obtiennent ainsi de meilleurs résultats qu'avec un simple matrice de seuillage, ils montrent en effet qu'un peu de grain est parfois préférable à une mauvaise distribution, notamment lors des changement de tons doux.

9.1.3 Supertrame

Le nombre de niveaux représentables par une matrice de seuillage correspond à sa taille; il faut donc des matrices suffisamment grandes pour permettre d'avoir une finesse suffisante de tons de sortie. Or, dans le cas de la production de trames groupées, la taille de l'élément de trame devient vite perceptible. Il est possible de construire une matrice de seuillage de telle sorte qu'elle produira plusieurs groupements de points, légèrement différents. Cette matrice de seuillage, appelée supertrame, peut être construite par la *convolution* d'une matrice dispersée et d'une matrice groupée, ce qui crée des copies, déphasées, de la matrice groupée. La figure 9.8 montre la construction d'une matrice de seuillage 6×6 à partir d'une matrice groupée 3×3 et d'une matrice dispersée 2×2 .

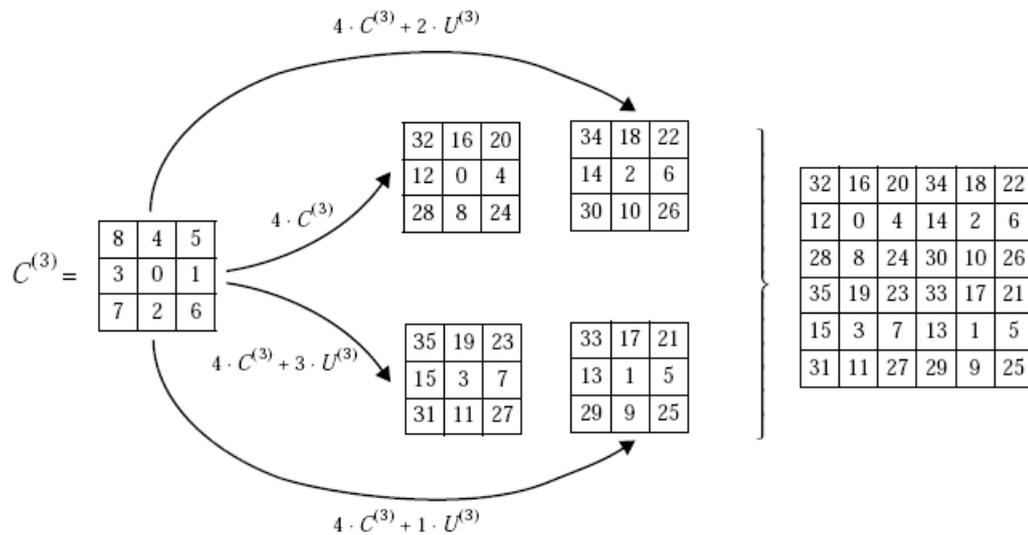


FIGURE 9.8 – Construction d'une supertrame image tiré de [HO95].

9.1.4 Discussion et analyse spectrale

L'utilisation d'une matrice de seuillage est très performante à l'exécution. En effet, toutes matrices de seuillage peuvent se représenter sous forme rectangulaire [Hol80]. Trouver la cellule correspondant à un point adressable se fait alors en deux opérations « modulo », et le ton de sortie est déterminé par une comparaison. Les calculs effectués pour chaque point adressable sont indépendants et peuvent donc être évalués en parallèle. L'apparence du rendu en demi-ton est directement liée à la distribution des valeurs de seuil dans la matrice de seuillage. Ceci permet de contrôler le niveau de dispersion et de groupement. Par contre, les petits détails et les faibles contrastes sont perdus.

Regardons comment se comporte le rendu en demi-tons avec des matrices de seuillage du point de vue spectral. Pour cela nous calculons les spectres de puissance obtenus pour un niveau constant. La figure 9.7 présente ces spectres pour une trame groupée et diverses trames dispersées. La fréquence de répétition d'une trame, même dispersée, apparaît clairement dans le spectre. En effet, les fréquences non nulles du spectre de puissance se trouvent uniquement sur les multiples de la fréquence de répétition. Sur la distribution produite, cela correspond à des répétitions visibles. Pour contourner ce problème il faudrait une période de répétition infinie, donc une matrice de seuillage infiniment grande (ou au moins suffisamment grande pour couvrir entièrement la grille de points adressables).

L'optimisation d'une matrice de seuillage prend un certain temps (plusieurs jours) pour une taille correcte (de 256×256 par exemple). Faire l'optimisation pour une page entière (de 9000×13000 valeurs) prendrait un temps trop important aux constructeurs, sans compter le stockage de la matrice sur la mémoire de l'imprimante (il faudrait environ 900 Mo). Cette solution n'est donc pas envisageable.

9.2 Diffusion d'erreur

L'utilisation de matrices de seuillage est une possibilité de rendu en demi-tons. L'autre approche couramment utilisée est la diffusion d'erreur. Floyd et Steinberg [FS76] proposent l'algorithme de diffusion d'erreur. Il consiste à parcourir chaque point adressable et de faire le choix du niveau de sortie (noir ou blanc) par simple comparaison à un seuil de 50 %. L'erreur (la différence entre le niveau de sortie et le niveau d'entrée) est alors diffusée sur les prochains pixels de l'entrée. Floyd et Steinberg ont choisi un ordre de parcours en *scanline*, c'est-à-dire de gauche à droite, de haut en bas, et ils utilisent les poids présentés figure 9.9 pour diffuser l'erreur aux quatre pixels voisins. Cette méthode produit des rendus en demi-tons avec modulation de fréquence ; elle permet notamment de faire ressortir les faibles contrastes. Il y a cependant des artéfacts d'alignement (serpentin) et de motifs (dans les tons proches de 50 %). Ceux-ci sont dus aux poids de diffusion et à l'ordre de parcours proposé par Floyd et Steinberg. Diverses améliorations ont été proposées, nous allons maintenant en présenter les plus importantes.

Ostromoukhov [Ost01] fait varier le seuil suivant le niveau d'entrée (avant correction). Il choisit les valeurs de seuil dans une phase d'initialisation pour casser les motifs et les alignements. Sa méthode est étendue par Li et Allebach [LA02] qui ajoutent une variation aléatoire du seuil en fonction du niveau d'entrée. Ceci permet de supprimer les structures encore présentes dans la méthode d'Ostromoukhov.

L'approche de diffusion d'erreur est difficilement parallélisable. En effet les points à traiter dépendent du résultat des points précédents. Or le calcul parallèle permet d'accélérer grandement le temps d'impression. Metaxas [Met99, Met03] montre un moyen de paralléliser au mieux la méthode de Floyd et Steinberg. De plus Chang et Allebach [CA03] définissent un ordre de parcours et une méthode spécifique pour faire de la diffusion d'erreur par blocs parallélisables.

Les techniques de diffusion d'erreur donnent de très bons résultats pour la modulation de fréquence ; elles permettent notamment de faire ressortir les faibles contrastes qui sont masqués avec les matrices de seuillage. Par contre, l'obtention de groupements de points est très délicate, et les quelques tentatives existantes [Knu87, LAG98] ne sont pas concluantes.

9.3 Optimisation du rendu en demi-tons

Comme nous l'avons vu, un processus d'optimisation peut être utilisé pour construire une matrice de seuillage. En fait, il est possible d'utiliser le même type

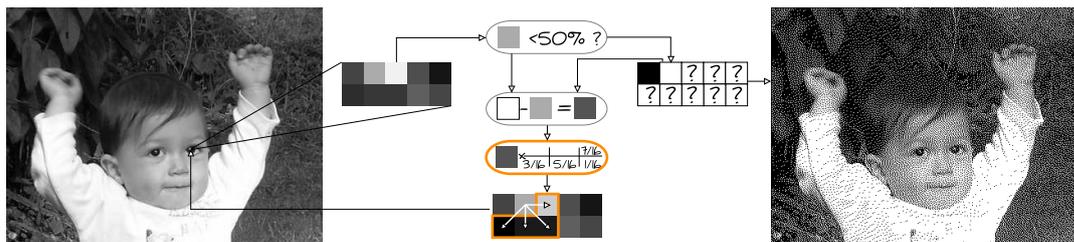


FIGURE 9.9 – Principe de la diffusion d'erreur.

de processus pour optimiser un rendu en demi-tons dans sa globalité.

L'approche de *direct binary search* que nous avons présentée dans le cas de la construction d'une matrice de seuillage, à été initialement développée par Allebach [AA92, LA97] pour obtenir le rendu en demi-tons d'une image complète. Le fonctionnement est le même, mais, plutôt que d'optimiser les différents niveaux de gris pour un élément de trame, l'image en entrée est directement prise comme contrainte dans l'optimisation. Les résultats obtenus ne font apparaître ni motif ni alignement, et représentent très fidèlement le ton. Cependant les temps de calcul sont prohibitifs, le rendu d'une page A4 à 1200 DPI (soit 9921×14030 points) se compte en heures, voire en jours.

Pang et coll. [PQW⁺08] présentent la méthode *structure aware halftoning*. Ils commencent par distribuer aléatoirement (bruit blanc) un nombre de points qui correspondent au niveau moyen de l'image à représenter. Ils définissent alors une fonction de coût représentant l'erreur entre cette distribution de points et l'image en entrée. Puis par échange successif de points noirs et points blancs, ils réduisent la fonction de coût jusqu'à stabilisation. Leur contribution réside dans le fait que leur fonction de coût tient compte de la structure (contraste, arrêtes), présente dans l'image en entrée, pour les souligner dans le rendu en demi-tons. Plus performante que *direct binary search*, il faut tout de même compter plusieurs heures pour le rendu d'une page A4.

Ces approches prennent un temps trop important pour être adaptées aux systèmes d'impression actuels, mais elles permettent d'obtenir une qualité de rendu en demi-tons qui tient compte d'un modèle du système visuel humain et d'un modèle de reproduction. C'est le but à atteindre en terme de qualité, mais avec des performances suffisantes pour être utilisable dans les imprimantes.

Pavage non-periodique et optimisation de trames

L'ANALYSE spectrale que nous avons réalisée dans le chapitre précédent montre que l'utilisation d'une seule matrice de seuillage, répliquée pour couvrir la grille de points adressables, fait ressortir certaines fréquences et laisse une grande partie du spectre nulle. Cela se traduit par des répétitions et des alignements perceptibles des points imprimés. Nous résolvons ce problème en utilisant non pas une seule matrice mais plusieurs matrices de seuillage pour couvrir la grille. Nous tirons parti des techniques de pavage non-périodique du plan pour définir un ensemble fini de matrices de seuillage qui préviendra la création de répétitions. Pour cela, nous définissons des matrices de seuillage comme les éléments d'un pavage non-periodique, en prenant soin de placer correctement les valeurs de seuil afin d'obtenir une bonne distribution de points pour tous les niveaux d'entrée. Les résultats obtenus (voir chapitre 11), ne présentent pas de répétition et les spectres de puissance correspondants sont très proches des méthodes de distribution bruit bleu (qui ne partagent pas les contraintes liées à la constructions d'une matrice de seuillage).

10.1 Pavage du plan

Un pavage est une division en parties distinctes d'un ensemble (le plan image dans notre cas) tel que l'union des parties corresponde à l'ensemble pavé et l'intersection de deux parties différentes du pavage soit l'ensemble vide.

Nous nous intéressons ici aux pavages où chacun des éléments de la division a la même forme et, plus particulièrement, une forme composée de carrés. L'utilisation du carré comme forme de base est importante pour pouvoir par la suite placer des matrices de seuillage sur le pavage. En effet, les matrices de seuillage forment un ensemble de valeurs que nous devons associer aux points adressables du système d'impression. Ces points se trouvent, eux, sur une grille régulière.

Nous avons choisi le pavage par polyominos qui, comme nous le verrons, présente de nombreux avantages pour notre problème. Les polyominos sont une classe



FIGURE 10.1 – Différents polyominos.

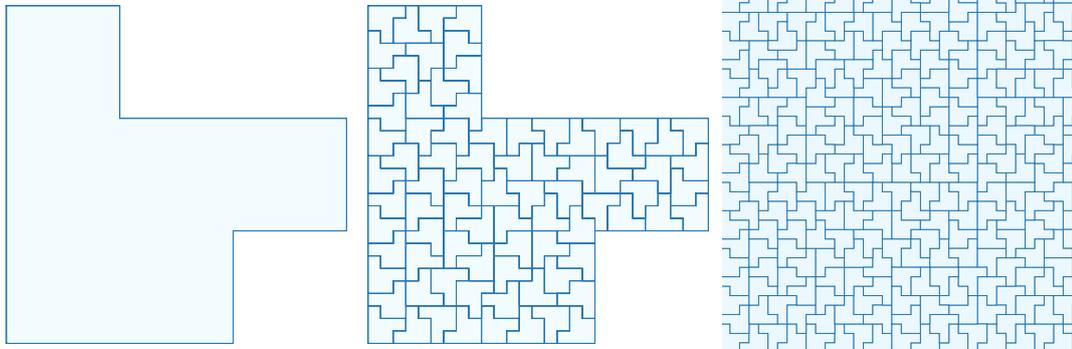


FIGURE 10.2 – Pavage du plan avec des G-hexaminos. À droite la règle de subdivision d'un G-hexamino. À gauche le pavage obtenu par la forme du voisinage.

de formes géométriques 2D constituées de la concaténation de carrés juxtaposés bords à bords (voir figure 10.1). Certains polyominos sont dits auto-similaires, c'est-à-dire que l'on peut les paver avec un nombre fini de répliques plus petites du même polyomino. Nous appellerons règles de subdivision la manière de remplir le polyomino initial par des répliques plus petites. Plus de détails sur les polyominos et leurs utilisations sont disponibles dans le livre de Golomb [Gol96].

Les polyominos auto-similaires peuvent être utilisés pour paver le plan. Il suffit pour cela d'utiliser la règle de subdivision et de mettre à l'échelle le résultat de la subdivision. Après chaque subdivision une partie plus grande du plan est pavée.

Grünbaum et Shephard [GS86] prouvent qu'un pavage à base de polyominos utilisant une seule règle de subdivision non triviale est non-périodique. C'est-à-dire qu'il n'existe aucune translation ni rotation d'un pavage donné permettant de retrouver ce même pavage. Cette propriété est intéressante dans notre cas, car elle prévient la création de motifs répétés par translations ou rotations.

Ostromoukhov [Ost07] utilise les polyominos pour effectuer efficacement l'échantillonnage d'une fonction d'importance avec les propriétés de bruit bleu. Pour cela, il identifie dans un pavage chacun des polyominos par un indice structural représentant son voisinage. On peut en effet remarquer sur la figure 10.2 que, lors du pavage, une multitude de voisinages se forment. Pour construire les indices structuraux, Ostromoukhov utilise des labels placés autour de chaque point des polyominos (voir figure 10.3) et parcourt le contour d'un polyomino en regardant l'ensemble des labels entourant les points de ce contour. Deux polyominos ayant

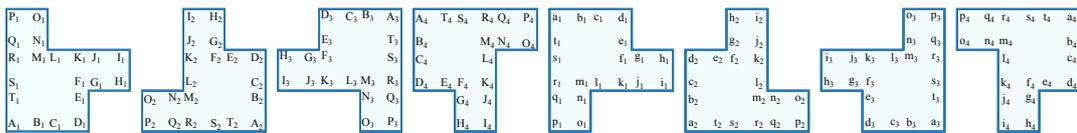


FIGURE 10.3 – Nous plaçons des labels différents autour des sommets d'un G-Hexamino suivant son orientation et sa rotation, ce qui définit huit séries de labels différentes.

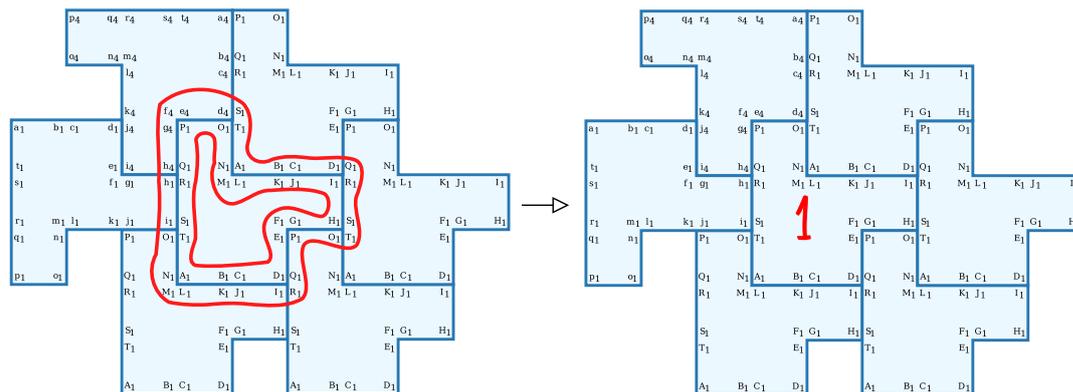


FIGURE 10.4 – L'énumération de labels (à gauche entourée en rouge) défini de manière unique un indice structurel (en rouge à droite).

des voisinages différents n'auront pas la même énumération de labels, cette énumération représente donc la forme du voisinage d'un polyomino dans le pavage. Ostromoukhov définit ensuite un indice structurel par énumération de labels possibles (voir figure 10.4). Pour trouver l'ensemble des labels possibles, il subdivise un polyomino tant que de nouveaux labels sont trouvés. Après un nombre fini de subdivisions, aucun nouveau label n'est trouvé et donc aucun nouveau voisinage n'apparaîtra dans les nouvelles subdivisions.

Une fois l'ensemble des indices structurels déterminé, il est facile de les attribuer aux polyominos constituant un pavage donné. Il suffit en effet de regarder l'énumération de labels le long du chemin de chaque polyomino et de trouver à quel indice il correspond. Nous appelons tuile un polyomino d'un indice structurel donné. Nous pouvons construire une règle de subdivision par tuile donnant les tuiles résultantes de cette subdivision et donc identifiant de manière déterministe quels voisinages sont créés par toutes les subdivisions futures.

Ostromoukhov utilise ensuite cette subdivision en plaçant un point par tuile, et profite ainsi des bonnes propriétés de pavage du plan par les polyominos. En effet, il montre qu'un simple placement des points aléatoire dans la région centrale de chaque tuile donne une meilleure distribution que l'utilisation de carré pour le pavage (ce qui est souvent utilisé pour faire de l'échantillonnage hiérarchique). Pour améliorer la distribution, il optimise itérativement la position du point pour chacune des tuiles en fonction des points des tuiles voisines. Ostromoukhov peut alors rapidement échantillonner une fonction d'importance en subdivisant ce pavage, dans le même esprit que la méthode utilisant le pavage de Penrose [ODJ04] que nous avons présenté section 4.1.

10.2 Utilisation d'éléments de trame

Dans notre cas, nous voulons produire une matrice de seuillage pour chacune des tuiles, les contraintes sont alors différentes de l'échantillonnage. Nous ne voulons pas seulement placer un point par tuile, mais un nombre fixe de valeurs de seuil pavant la grille de points adressables du système d'impression. Nous prenons le G-hexamino comme forme de base de la matrice de seuillage. En effet, le pavage avec des G-hexominos suivant l'approche d'Ostromoukhov offre un nombre

intéressant d'indices structurels lors de la subdivision (1744). Cependant, notre approche est valide pour d'autres formes de polyominos.

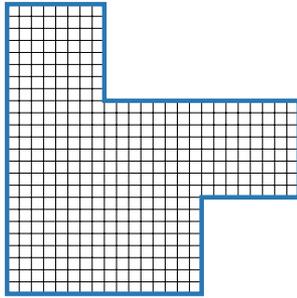


FIG. 10.5 Définition d'une matrice de seuillage sur un G-hexomino. Ici la taille de segment est de $S = 8$.

Nous définissons une matrice de seuillage ayant la forme d'un G-hexomino de la manière suivante : nous construisons une grille de valeurs sur un G-hexomino en définissant une taille de segment de base S . Cette taille de segment permet de définir une matrice de $6 \times S^2$ valeurs sur un G-hexomino (voir figure 10.5). Ensuite, pour obtenir le pavage de la grille adressable avec une correspondance univoque entre les valeurs de la matrice de seuillage et les points de la grille, nous utilisons la règle de subdivision des G-hexominos. Nous plaçons la grille dans un G-hexomino suffisamment grand pour l'englober totalement, et nous le subdivisons jusqu'à obtenir des tuiles suffisamment petites pour avoir la correspondance. Pour trouver exactement une correspondance univoque, il suffit de prendre une tuile de départ ayant une base

de $S \times A^n$ points, où $n = \lceil \log_A(I_s / (2S)) \rceil$ est le nombre de subdivisions requis, I_s est la taille de la grille (le maximum entre la hauteur et la largeur) et A est le facteur d'échelle de la subdivision. Le facteur d'échelle est le rapport entre la taille de segment de base des tuiles avant et après une subdivision, dans le cas du G-hexomino $A = 9$, c'est-à-dire qu'un G-hexomino se subdivise en $A^2 = 81$ G-hexominos.

10.3 Optimisation des valeurs de seuil

Nous allons maintenant détailler la manière dont nous calculons les valeurs de seuil des matrices, pour obtenir de bonnes propriétés de répartition des points pour tous les tons en entrée.

Comme nous l'avons vu section 9.1.2, il existe plusieurs méthodes pour l'optimisation d'éléments de trame. Nous avons basé notre approche sur *void-and-cluster*. En effet, cette approche a l'avantage d'être relativement peu coûteuse à calculer et d'offrir, déjà dans le cas d'une matrice carrée, de très bons résultats.

Nous suivons le pipeline de *void-and-cluster* présenté section 9.1.2, mais en l'adaptant à l'utilisation des G-hexominos. L'optimisation de *void-and-cluster* est basée sur un filtrage Gaussien d'un élément de trame prototype pour déterminer la position du plus grand *void* et du plus petit *cluster*. Lors de ce filtrage, nous avons besoin du contexte local à l'élément de trame pour tenir compte de la distribution de points autour de celui-ci. Dans le cas classique, une seule matrice de seuillage est utilisée pour couvrir le plan, donc nous avons un seul contexte pour le filtrage de l'élément de trame (l'élément de trame peut alors être vu comme un espace torique lors du filtrage).

Le pavage que nous utilisons est constitué de plusieurs G-hexominos et, dans l'esprit de *void-and-cluster*, nous plaçons un élément de trame sur chaque G-hexomino. Or, en regardant le pavage, nous notons que plusieurs entourages de G-hexominos apparaissent. Si nous utilisons un unique élément de trame pour tous les G-hexominos du pavage, l'élément de trame aurait plusieurs contextes locaux. En effet, dans le pavage, si nous regardons un bord du G-hexomino donné,

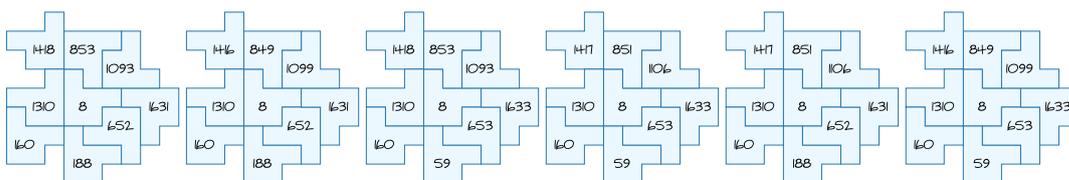


FIGURE 10.6 – Les six voisinages de la tuile d’indice structurel huit. Chaque voisinage à la même forme (la même géométrie). Par contre, aucun de ces voisinages n’a le même ensemble de tuiles (les ensembles d’indices structurels entourant la tuile diffèrent tous d’au moins un indice).

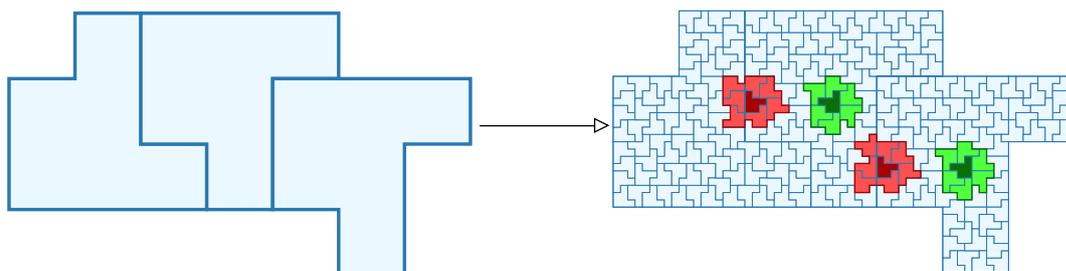


FIGURE 10.7 – Les deux G-hexominos en vert foncé ont la même forme de voisinage, ce sont donc deux même tuiles. Les G-hexominos rouge foncé ont des voisinages différents, ce sont donc des tuiles différentes. On remarque que pour deux G-hexominos orientés de la même manière, les tuiles créées au centre sont les mêmes alors que sur les bords de nouvelles tuiles apparaissent.

nous remarquons qu’il est juxtaposé avec tous les autres bords du G-hexomino (suivant sa position dans le pavage). Un simple espace torique ne suffit plus pour tenir compte des différents contextes locaux qui apparaissent sur le pavage.

Par contre, les G-hexominos du pavage définissent des tuiles, chaque tuile a un voisinage dont la géométrie (la forme) est unique. Nous pouvons donc utiliser la définition de tuile pour construire le contexte local. Pour cela, nous associons un élément de trame prototype différent (et donc, par extension, une matrice de seuillage différente) à chaque tuile. Le contexte local est alors défini par les tuiles entourant une tuile donnée.

Puisque le pavage est non-périodique, une tuile donnée peut être entourée par différentes tuiles voisines, bien que la géométrie de ce voisinage soit la même (par définition d’une tuile). En d’autres termes, le voisinage de chaque tuile voisine d’une tuile donnée n’est pas unique. En fait, pour chaque tuile, il y a un ensemble fini de voisinages (nous avons entre un et cinquante-huit voisinages par tuile dans notre pavage). La figure 10.6 montre les six voisinages possibles pour la tuile définie par l’indice structurel huit.

Le bord d’une tuile donnée peut donc être juxtaposé avec plusieurs tuiles différentes. Sans précaution particulière, chaque bord devient sur-contraint et l’optimisation ne donne pas de bons résultats : les nouveaux points sont dans un premier temps accumulés au centre pour compenser la sur-contrainte des bords.

Pour pouvoir garantir le bon assemblage d’un bord d’une tuile avec l’ensemble des tuiles voisines pouvant lui être juxtaposées, nous imposons un bord unique que nous déterminons avant de calculer l’intérieur (c’est une approche similaire aux tuiles de Wang [LD05] que nous avons introduites section 4.1.1). Les bords

sont caractérisés par les labels à leurs extrémités (voir figure 10.8). Une tuile est alors entourée par un ensemble unique de bords (douze bords dans le cas du G-hexamino) et le même bord se retrouve entre plusieurs tuiles différentes.

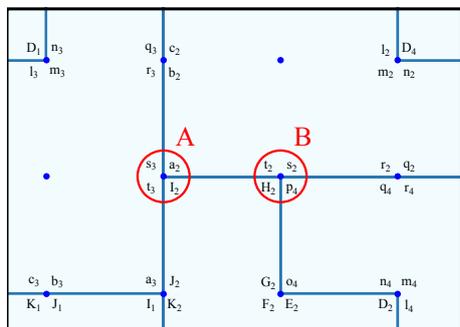


Fig. 10.8 Chaque bord est défini par deux labels, ici le bord (A, B) est défini par les labels $((s_3, a_2, t_3), (t_2, s_2, p_4, H_2))$.

Comme dans l'approche *void-and-cluster*, nous déterminons les valeurs de seuil des matrices par le filtrage d'éléments de trame prototypes. Les éléments de trame prototypes associés à chacune des tuiles sont donc découpés en bords, qui seront partagés par plusieurs tuiles, et intérieurs, propres à chacune des tuiles (voir figure 10.9). Ceci permet d'obtenir le bon contexte local pour l'optimisation de l'intérieur des tuiles (voir section 10.3.1). Les prototypes de bords vont contraindre les prototypes des intérieurs à bien s'assembler sur le pavage. En effet, *void-and-cluster* tient compte à la fois de la densité¹ globale et de l'arrangement local des points pour trouver les positions

d'intérêt. Lors de l'optimisation d'un intérieur, l'arrangement local est donné par le bord et la densité globale est donnée par les intérieurs des tuiles voisines.

Nous commençons par calculer des prototypes de bords avec une distribution de points bruit bleu, en prenant en compte les juxtapositions possibles de bords. Puis nous effectuons le calcul des prototypes des intérieurs (section 10.3.2). Dans cette seconde opération, les prototypes de bords ne sont pas mis à jour.

Une fois l'ensemble des éléments de trame prototypes déterminés, nous attribuons un seuil à chacun des points présents dans les prototypes, que nous stockons dans la matrice de seuillage associée, voir section 10.3.3.

Ensuite nous ajoutons des points dans les prototypes de bords et d'intérieurs et nous recommençons l'attribution des valeurs de seuil pour les nouveaux points (voir section 10.3.4). Nous recommençons ce processus jusqu'à remplir toutes les matrices de seuillage. Nous allons, dans les sections suivantes, détailler chacune de ces étapes.

10.3.1 Initialisation des prototypes de bords

Dans notre implémentation, les bords sont définis comme la partie en bas à droite des matrices de seuillage de chacune des tuiles (voir figure 10.9). Pour initialiser les éléments de trame prototype des bords, nous partons d'une bonne distribution de points dans laquelle nous découpons un morceau de la taille du bord pour affecter son prototype. Nous utilisons une relaxation de Lloyd [Llo82] (décrit section 3.2.4) à partir d'un bruit blanc ayant le ton initial voulu : $d_i = \frac{i}{5}$. Pour tenir compte des bords voisins déjà affectés, nous fixons les points correspondants lors du calcul de la distribution. Les points déjà affectés ne perturbent pas la relaxation car ils ne peuvent pas bouger, ils viennent seulement ajouter une contrainte sur le placement des nouveaux points en les repoussant. La figure 10.10 résume l'initialisation des bords.

1. Nous préférons le terme de *densité* au terme de *ton* lorsque nous parlons de la distribution des points.

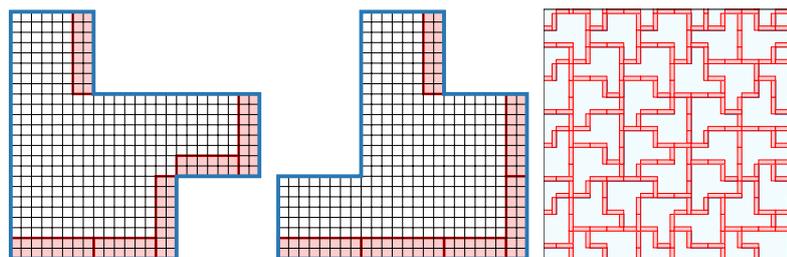


FIGURE 10.9 – Les bords sont définis comme la partie en bas à droite de chaque matrice. Lorsque les matrice pave le plan, deux intérieurs sont séparés par un bord.

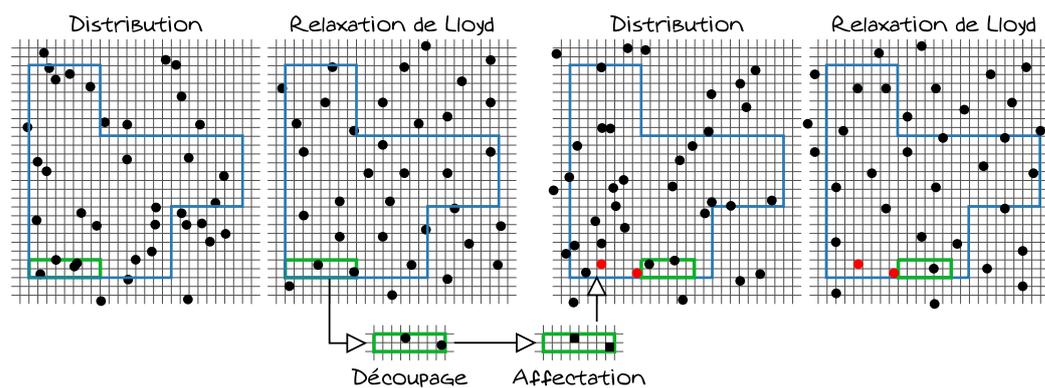


FIGURE 10.10 – L’initialisation d’un bord commence par une distribution aléatoire de points avec la densité d_i . Ces points sont relaxés pour obtenir une distribution bruit bleu. La région du bord est découpée et son élément de trame prototype est affecté. Lors de l’initialisation du prochain bord, les bords déjà affectés sont mis comme contraintes lors de la relaxation.

10.3.2 Initialisation des prototypes des intérieurs

Une fois les bords définis, nous construisons l’élément de trame prototype de l’intérieur de chacune des tuiles. Pour cela, nous plaçons aléatoirement un nombre de points correspondant à la densité d_i , en tenant compte du nombre de points présents dans les bords de la tuile (qui peut être différent d’une tuile à l’autre). Nous effectuons ensuite la même relaxation de Lloyd que pour les bords, les points imposés par les bords restant fixes (voir figure 10.11).

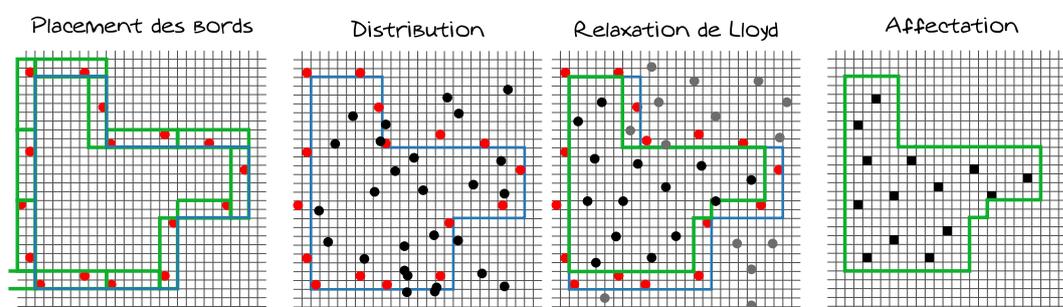


FIGURE 10.11 – Pour l’initialisation des intérieurs, nous commençons par fixer les points des bords et distribuer aléatoirement des points. Puis, nous relaxons les points pour obtenir l’élément de trame prototype.

10.3.3 Assignment des valeurs de seuil

A partir des éléments de trame prototypes, nous attribuons une valeur de seuil à chacun des points distribués, que nous stockons dans la matrice de seuillage correspondante. Pour cela, nous cherchons pour chacune des tuiles, prises dans un ordre aléatoire, la position du plus petit *cluster* dans l'élément de trame prototype (concaténation des prototypes de bords et du prototype de l'intérieur de la tuile). Nous retirons ce point et en même temps nous stockons la valeur de seuil correspondante² dans la matrice de seuillage de la tuile, à la position du point retiré. Notons ici que les matrices de seuillage ne tiennent pas compte explicitement des bords. Ainsi un même point sur un même bord peut avoir deux valeurs de seuil différentes dans deux tuiles différentes.

Pour trouver la position du plus petit *cluster*, nous utilisons l'ensemble des voisinages de la tuile considérée. Chacun de ces voisinages apparaît dans le pavage avec une fréquence différente. En effet, la diversité des voisinages est créée sur les bords des G-hexominos lors de la subdivision. Les G-hexominos produits au centre des tuiles sont toujours entourés de la même manière, c'est donc toujours les mêmes tuiles qui sont créés au centre d'un G-hexomino d'une orientation donnée (nous avons différencié les orientations lors de l'assignation des labels) alors que sur les bords, le voisinage créé dépend de la tuile voisine avant subdivision. La figure 10.7 illustre ce phénomène. Nous faisons donc la somme des éléments de trames prototype des différents voisinages de la tuile, ceux-ci sont pondérés par leur fréquence d'apparition dans un pavage représentatif (ou chacune des tuiles apparaît). Ainsi, nous obtenons un élément prototype qui n'est plus binaire, mais qui représente les points pondérés par leur fréquence d'apparition, entourant une tuile. Nous filtrons ce prototype et nous cherchons la position du maximum dans la région correspondant à la tuile pour déterminer la position du plus petit *cluster*.

10.3.4 Ajout des nouveaux points

Il reste maintenant à combler les valeurs de seuil manquantes, dans les matrices de seuillage. Pour cela, nous procédons par palier de densité. Nous ajoutons suffisamment de nouveaux points pour augmenter la densité de $1/S$. Pour ajouter ces points, nous calculons les éléments de trames prototype filtrés par un noyau gaussien. Ce filtrage se fait en tenant compte du voisinage pondéré de la même manière que présentée section 10.3.3. Nous obtenons alors une position du plus grand *void* pour chacun des bords et intérieurs prototypes. Nous ajoutons un point dans le *void* qui a la valeur minimale (pris sur l'ensemble des prototypes filtrés). Nous mettons alors à jour les prototypes filtrés qui sont affectés par l'ajout du point, et nous itérons jusqu'à ce que nous ayons ajouté suffisamment de points pour le palier de densité.

Nous réalisons alors une nouvelle étape d'assignation de valeurs de seuil comme précédemment, pour tous les nouveaux points ajoutés. Enfin, nous recommençons pour les paliers suivants jusqu'à remplir toutes les valeurs de seuil (soit $S - i$ fois).

2. Cette valeur de seuil est égale au nombre de points dans l'élément de trame prototype au moment de l'affectation.

Résultats, applications et perspectives

POUR analyser la qualité des distributions de points que nous obtenons, nous considérons les spectres de puissance pour tous les tons. Nous avons pris une taille de segment de base $S = 8$, donc des matrices à 384 valeurs, ce qui est suffisant pour les systèmes d'impression. Pour le ton des éléments de trame prototypes initiaux, nous avons utilisé $i = 1$ et donc $d_1 = 100/8 = 12.5 \%$, ce qui correspond à la préconisation d'Ulichney [Uli93].

11.1 Distributions de points bruit bleu

Nous remarquons différents comportements suivant le ton ; ces comportements sont symétriques par rapport au ton 50 % ce qui est une bonne propriété. Dans les tons clairs, nous avons une distribution de points de type bruit bleu comme le montre la figure 11.1a. Lorsque l'on approche des 5 %, on voit apparaître quelques fréquences plus marquées. Après avoir analysé en détail la distribution, nous avons remarqué que certains bords apparaissent plus souvent que d'autres, et lorsque le ton demandé correspond à un multiple de 12.5 % (à chaque pallier de densité), deux même bords produisent le même élément de trame : on retrouve alors dans le spectre la fréquence d'apparition du bord dans la distribution. En fait, à ce niveau la fréquence principale (celle de l'anneau plus marqué) est suffisamment petite pour ne pas être perturbée par ces fréquences marquées : cela devient plus perceptible vers 15 %. Au delà de 30 %, le phénomène de recouvrement sur le spectre de puissance devient trop important pour que celui-ci soit exploitable, voir figure 11.1c. Néanmoins, nous voyons peu de structures répétitives dans les distributions : notamment à 50 % nous obtenons des régions de damiers, brisées par des régions plus aléatoires, comme le montre le figure 11.1d.

La figure 11.2 applique notre rendu en demi-tons sur un dégradé variant doucement d'un ton à 0 % jusqu'à 100 %. Tous les tons d'entrée sont correctement représentés, nous notons la bonne répartition des points dans les tons très clairs et très foncés. Nous remarquons toutefois quelques alignements dont nous n'avons pas identifié l'origine.

11.2 Applications : couleur et supertrame

Les matrices de seuillage que nous avons produites peuvent être directement utilisées pour produire un rendu en demi-tons dispersé noir et blanc. Pour obtenir le rendu avec plusieurs teintes, il suffit d'avoir des distributions « indépendantes ». Pour la figure 11.5, nous avons utilisés trois distributions pour les trois teintes (magenta, cyan et jaune) constituant le rendu en demi-tons. Nous obtenons les trois distributions avec le même ensemble de matrices de seuillage. Pour cela, nous utilisons une première tuile qui est subdivisée pour couvrir la grille de points adressables. Ensuite, nous effectuons deux autres fois la même subdivisions, mais en prenant des tuiles de départ différentes. En outre, pour éviter que deux même tuiles se retrouvent superposées pour les différentes teintes, nous décalons les pavages obtenus de la taille d'un segment de base.

Nous pouvons aussi nous servir de notre distribution de valeurs de seuil comme base de construction d'une supertrame. Pour cela nous convoluons chaque matrice avec une matrice groupée. Les figures 11.3 montrent la comparaison entre les supertrames que nous obtenons et celle obtenues avec la matrice dispersée de Bayer. Nous notons que la structure présente lorsqu'on utilise la matrice de Bayer est supprimée avec notre approche.

Enfin, nous pouvons combiner différentes supertrames obtenues avec notre distribution pour effectuer un rendu en demi-tons couleurs. La figure 11.4 en présente deux exemples.

11.3 Améliorations

Comme nous l'avons souligné, la répétition trop fréquente de certains bords nuit à la bonne distribution des points. Une idée serait de construire le pavage de telle sorte que chaque entité (bords et intérieurs) soit équitablement représentée. Cette piste semble intéressante, mais difficile à mettre en œuvre. Par contre, nous avons commencé quelques expérimentations avec des Monominos. Ceux-ci permettent de contrôler la répartition des indices structurels car la complexité n'est plus intrinsèque à la règle de subdivision (un monomino se divise simplement en quatre monominos) mais à l'information complémentaire que l'on peut adjoindre.

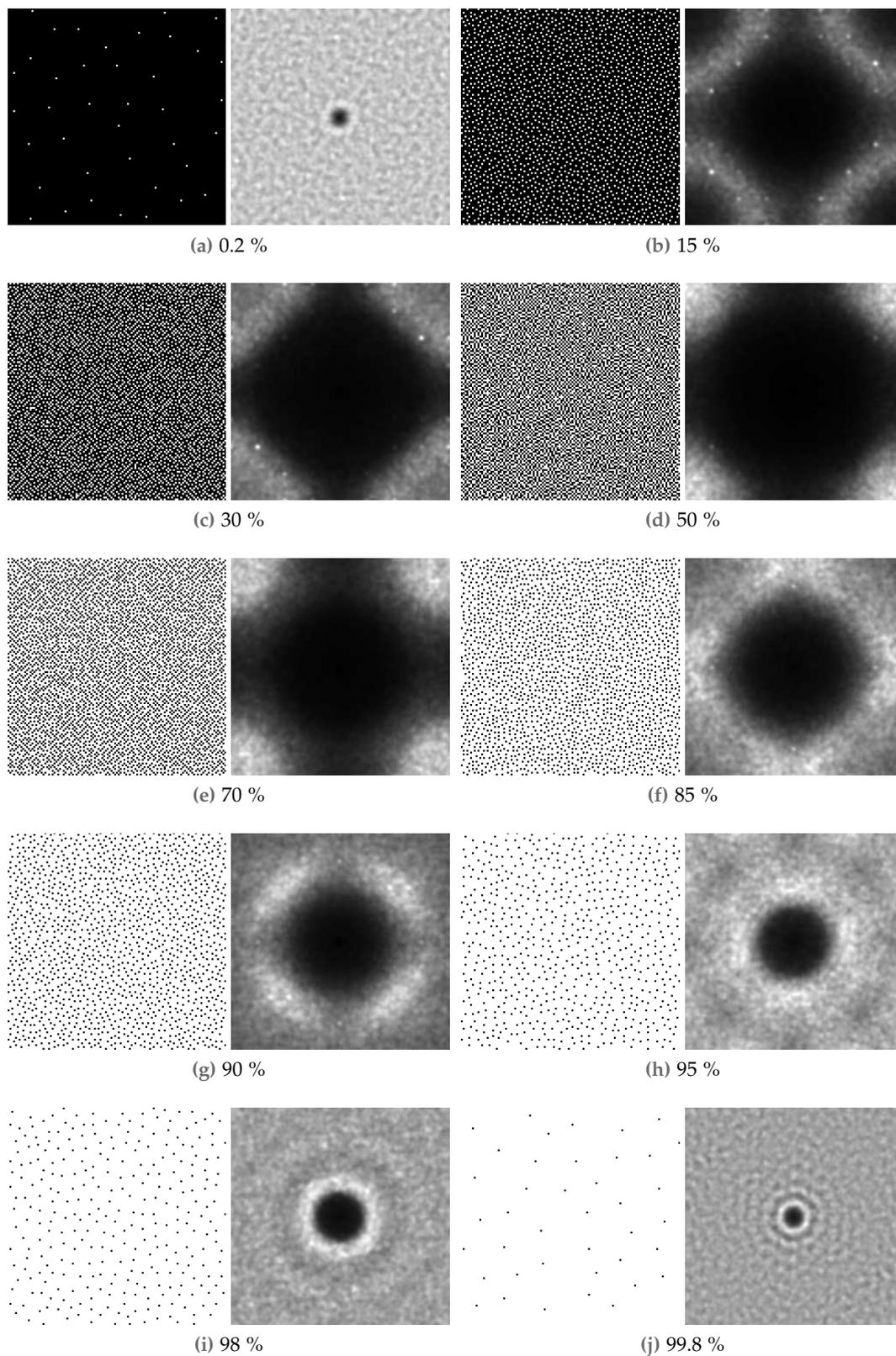


FIGURE 11.1 – Distributions obtenues avec notre méthode et les spectres de puissance correspondants (filtré par un filtre Gaussien).

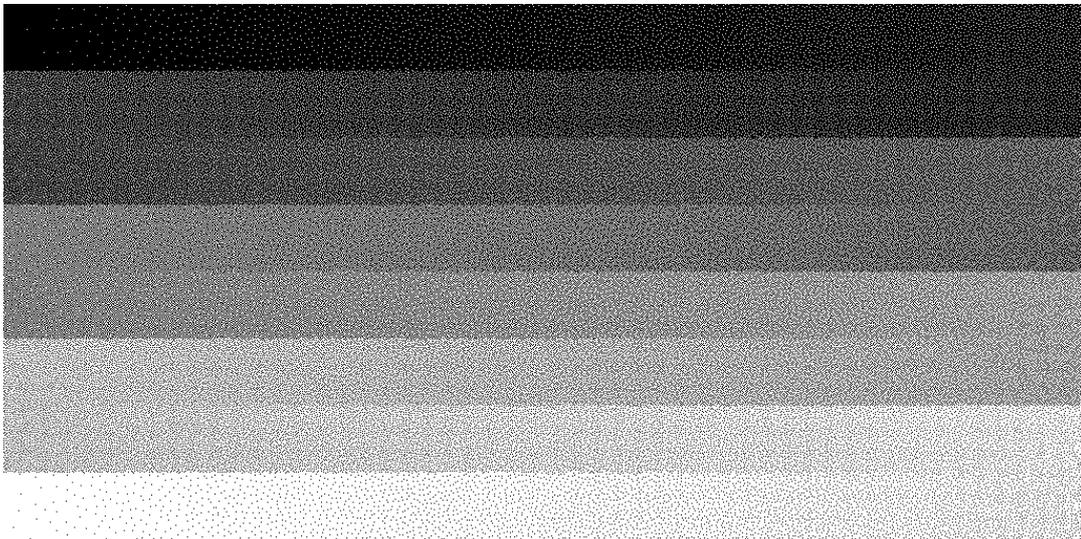
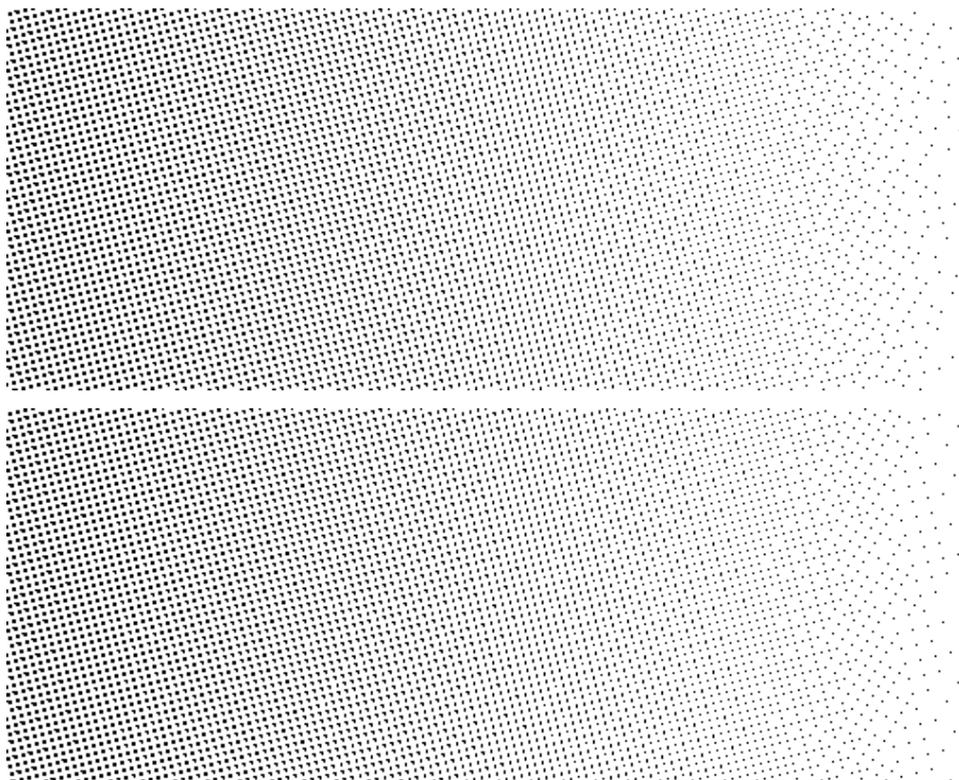
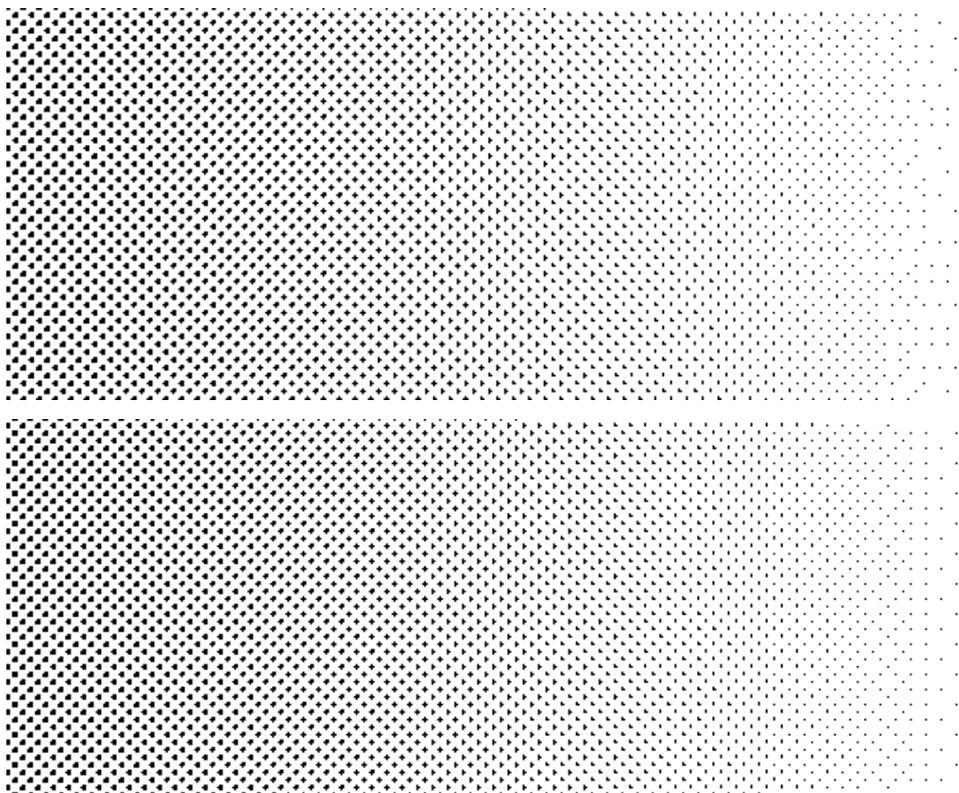


FIGURE 11.2 – Une rampe de 0 % à 100 % montrant la qualité de distribution des points de notre approche. Pour apprécier la distribution de point, il est conseillé de regarder la version électronique du document et de zoomer pour voir individuellement chacun des points.

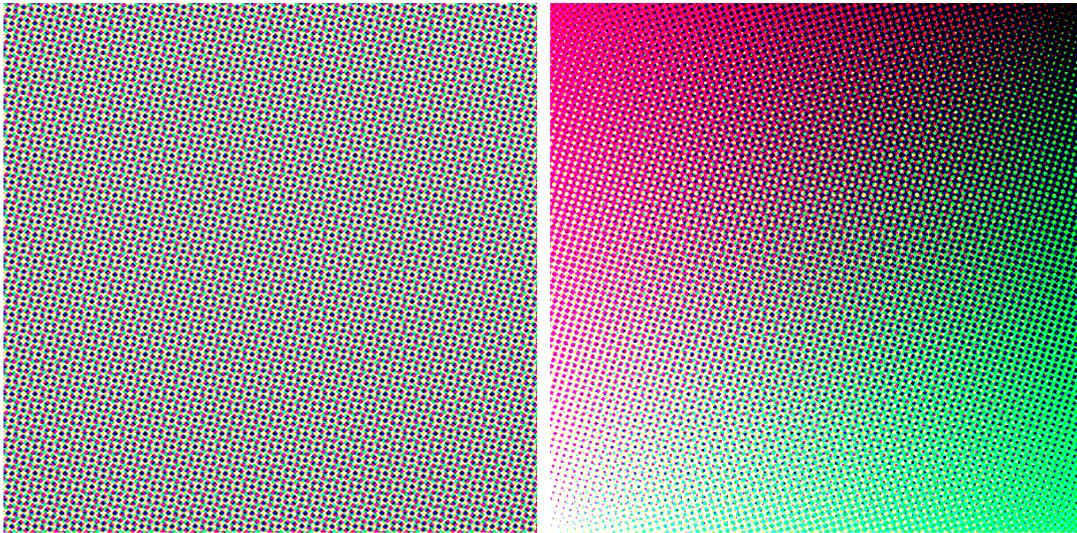


(a) Supertrame construite avec une matrice groupée de 17 valeurs.



(b) Supertrame construite avec une matrice groupée de 32 valeurs.

FIGURE 11.3 – La première ligne de chaque figure présente notre matrice dispersée, la seconde utilise la matrice de Bayer, pour construire une supertrame. Nous notons que les structures présentes lors de l'utilisation de la matrice de Bayer disparaissent avec notre distribution de valeurs de seuils.



(a) Couleur constante, gris à 60 %, avec encre magenta, cyan et jaune

(b) Le ton magenata varie verticalement, le cyan varie horizontalement et le jaune en diagonale.

FIGURE 11.4 – Rendu couleur et avec une supertrame utilisant notre distribution de valeurs de seuils.

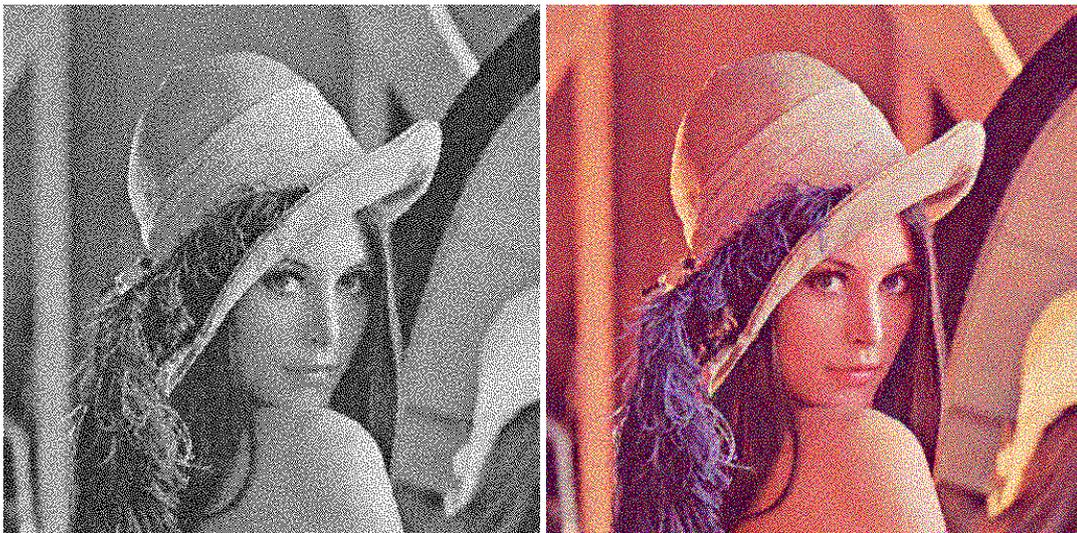


FIGURE 11.5 – Deux images rendues directement avec les matrices de seuillage dispersées que nous avons calculées. La première en noir et blanc, la seconde en couleur en utilisant trois pavages différents pour les trois canaux magenta, cyan et jaune.

Table des figures

1.1	Artéfact de rendu	14
2.1	Jeux vidéos	21
2.2	Installations artistiques	22
2.3	Dessins animés	23
2.4	Animations expérimentales	24
2.5	Effets de matière	25
3.1	<i>tonal art-maps</i>	31
3.2	Textures plaquées sur les objets 3D	31
3.3	Techniques hybrides	33
3.4	Modèles et attributs de marques	34
3.5	Pipeline général de rendu	36
3.6	Ordre de dessin	37
3.7	Rendu <i>stippling</i>	38
3.8	Mise à jour des points d'attaches de Litwinowicz	40
3.9	Modèle de marque	43
4.1	Distribution de points	48
4.2	Grille hexagonale	50
4.3	Hierarchie de points d'attache	53
4.4	Coupe de la hiérarchie	53
4.5	Sélection de points d'attache par prédiction	55
4.6	Résultats : distributions et spectres de Fourier	56
4.7	Principe de fonctionnement	57
4.8	Pipeline de distribution	58
4.9	Résultats	62
4.10	Spectre de Fourier	63
4.11	Rendu <i>stippling</i>	63
4.12	Densité non-uniforme	64
5.1	Pipeline	66
5.2	Attributs et apparence d'une marque	68
5.3	Exemples de styles	69
5.4	Marche le long du squelette d'une marque	76
5.5	Apparition d'une marque	77

5.6	Dumb Luck	78
7.1	Installation pour la création de peinture animée	86
7.2	Simulation de peinture	88
7.3	Peinture automatique	89
7.4	Pipeline de simulation	90
7.5	Masque de transparence d'une entité de dessin	91
7.6	Mélange des couleurs	92
7.7	Influence de la quantité de peinture	93
7.8	Construction d'un pinceau	93
7.9	Effets de matières	94
7.10	Étalements	96
7.11	Influence du délai	98
8.1	Rayon lumineux traversant l'oeil	104
8.2	Numérisation d'une impression	106
8.3	Exemple de ton à 20%	107
8.4	Effet de moiré	108
8.5	Rosette	108
9.1	Principe de la matrice de seuillage	111
9.2	Exemple de forme de matrices de seuillage	112
9.3	Rampes rendues avec des matrices groupées	113
9.4	Image en demi-tons par matrice groupée	113
9.5	Schéma de Bayer	114
9.6	Support du filtre dans <i>void and cluster</i>	115
9.7	Analyse spectrale du rendu en demi-tons	117
9.8	Construction d'une supertrame	118
9.9	Principe de la diffusion d'erreur.	119
10.1	Différents polyominos	122
10.2	Pavage du plan avec des G-hexominos	122
10.3	Labels autour des G-Hexomino	122
10.4	Indice structurel	123
10.5	Matrice sur un G-hexomino	124
10.6	Voisinages possibles pour une même tuile	125
10.7	Création des différents voisinages lors de la subdivision.	125
10.8	Labels d'un bord	126
10.9	Bords de matrice	127
10.10	Initialisation des bords	127
10.11	Initialisation des intérieurs	127
11.1	Distribution de points avec notre approche	131
11.2	Rampe rendue en demi-ton avec notre approche	132
11.3	Supertrame avec notre distribution	133
11.4	Demi-tons en couleur et supertrame avec notre distribution	134

11.5 Demi-tons à trame dispersée avec notre distribution	134
--	-----

Liste des tableaux

3.1 Tableau comparatif des travaux précédents	46
---	----

Liste des algorithmes

4.1 Sélection des points d'attache par tests d'occlusion	54
5.1 Lecture des attributs des marques	73
5.2 Marche le long des chemin de marques	76
7.1 Création des coups de pinceau d'étalement	97
9.1 <i>Void-and-cluster</i> , initialisation du motif initial.	115
9.2 <i>Void-and-cluster</i> , numérotation.	116

Liste des programmes

5.1 style <i>stippling</i>	71
5.2 style hachurage	71
5.3 Extrait du <i>shader</i> pour la première passe de rendu des marques.	79

Bibliographie

- [AA92] Mostafa ANALOUTI et Jan P. ALLEBACH. « Model-based halftoning using direct binary search ». Dans *Proceedings of the 1992 SPIE/IS&T Symposium on Electronic Imaging Science and Technology*, volume 1666, p. 96-108, février 1992. (Cité p. 120)
- [ABS99] Hakan ANCIN, Anoop K. BHATTACHARJYA et Joseph SHU. « New void-and-cluster method for improved halftone uniformity ». *Journal of Electronic Imaging*, 8(1):104-111, January 1999. <dx.doi.org/10.1117/1.482701>. (Cité p. 115)
- [AHO94] Isaac AMIDROR, Roger D. HERSCH et Victor OSTROMOUKHOV. « Spectral Analysis and Minimisation of Moiré patterns in Colour Separation ». *Journal of Electronic Imaging*, 3(3):295-17, 1994. <www.iro.umontreal.ca/~ostrom/publications/abstracts.html#JEI94_Moire>. (Cité p. 107)
- [AL96] J. ALLEBACH et Qian LIN. « FM screen design using DBS algorithm ». *Image Processing, 1996. Proceedings., International Conference on*, 1:549-552 vol.1, Sep 1996. <dx.doi.org/10.1109/ICIP.1996.559555>. (Cité p. 115)
- [Bay73] B. E. BAYER. « An optimum method for two-level rendition of continuous-tone pictures ». *IEEE International Conference on Communications*, 1, 1973. (Cité p. 113)
- [BB95] Steven S. BEAUCHEMIN et John L. BARRON. « The computation of optical flow ». *ACM Computing Surveys (CSUR)*, 27(3):433-66, 1995. <dx.doi.org/10.1145/212094.212141>. (Cité p. 30, 37 et 39)
- [BHGS06] Tamy BOUBEKEUR, Wolfgang HEIDRICH, Xavier GRANIER et Christophe SCHLICK. « Volume-Surface Trees ». *Computer Graphics Forum (Proceedings of Eurographics 2006)*, 25(3):399-06, 2006. <iparla.labri.fr/publications/2006/BHGS06>. (Cité p. 55)
- [BKTS06] Adrien BOUSSEAU, Matt KAPLAN, Joëlle THOLLOT et François X. SILLION. « Interactive watercolor rendering with temporal coherence and abstraction ». Dans *NPAP'2006: Fourth International Symposium on Non-Photorealistic Animation and Rendering*, p. 141-49, New York, NY, USA, 2006. ACM Press. <dx.doi.org/10.1145/1124728.1124751>. (Cité p. 32)
- [Bli78] James F. BLINN. « Simulation of wrinkled surfaces ». *Computer Graphics (Proceedings of SIGGRAPH'78)*, 12(3):286-92, 1978. <dx.doi.org/10.1145/965139.507101>. (Cité p. 94)
- [BLL04] William BAXTER, Yuanxin LIU et Ming C. LIN. « A Viscous Paint Model for Interactive Applications ». *Computer Animation and Virtual Worlds*, 15(3-4):433-41, July 2004. <gamma.cs.unc.edu/viscous>. (Cité p. 87 et 88)
- [BNTS07] Adrien BOUSSEAU, Fabrice NEYRET, Joëlle THOLLOT et David SALESIN. « Video watercolorization using bidirectional texture advection ». *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):104, 2007. <doi.acm.org/10.1145/1275808.1276507>. (Cité p. 30 et 72)
- [BSL01] William BAXTER, Vincent SCHEIB et Ming C. LIN. « dAb: Interactive Haptic Painting With 3D Virtual Brushes ». Dans Eugene Fiume, éditeur, *SIGGRAPH'2001*, p. 461-68. ACM Press, 2001. <gamma.cs.unc.edu/dab>. (Cité p. 87 et 88)

- [BSM⁺07] Simon BRESLAV, Karol SZERSZEN, Lee MARKOSIAN, Pascal BARLA et Joëlle THOLLOT. « Dynamic 2D patterns for shading 3D scenes ». *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):20, 2007. <[dx.doi.org/10.1145/1276377.1276402](https://doi.org/10.1145/1276377.1276402)>. (Cité p. 32 et 33)
- [BWL04] William BAXTER, Jeremy WENDT et Ming C. LIN. « IMPaSTo: A Realistic Model for Paint ». Dans *NPAR'2004: Third International Symposium on Non-Photorealistic Animation and Rendering*, p. 45-6, June 2004. <gamma.cs.unc.edu/impasto>. (Cité p. 87 et 88)
- [BWPP04] Jiří BITTNER, Michael WIMMER, Harald PIRINGER et Werner PURGATHOFER. « Coherent Hierarchical Culling: Hardware Occlusion Queries Made Useful ». *Computer Graphics Forum (Proceedings of Eurographics 2004)*, 23(3):615-24, 2004. (Cité p. 54)
- [CA03] Ti-Chiun CHANG et Jan P. ALLEBACH. « Memory efficient error diffusion. ». *IEEE Transactions on Image Processing*, 12(11):1352-1366, 2003. (Cité p. 119)
- [CDF⁺06] Forrester COLE, Doug DECARLO, Adam FINKELSTEIN, Kenrick KIN, Keith MORLEY et Anthony SANTELLA. « Directing Gaze in 3D Models with Stylized Focus ». *Eurographics Symposium on Rendering*, juin 2006. <http://www.cs.princeton.edu/gfx/pubs/Cole_2006_DGI>. (Cité p. 82)
- [CDH06] Liviu COCONU, Oliver DEUSSEN et Hans-Christian HEGE. « Real-time pen-and-ink illustration of landscapes ». Dans *NPAR'2006: Fourth International Symposium on Non-Photorealistic Animation and Rendering*, p. 27-5, New York, NY, USA, 2006. ACM Press. <[dx.doi.org/10.1145/1124728.1124734](https://doi.org/10.1145/1124728.1124734)>. (Cité p. 32 et 33)
- [CH02] John P. COLLOMOSSE et Peter M. HALL. « Painterly Rendering using Image Saliency ». Dans *EGUK '02: Proceedings of the 20th UK conference on Eurographics*, page 122, Washington, DC, USA, 2002. IEEE Computer Society. (Cité p. 88)
- [CH05] John P. COLLOMOSSE et Peter M. HALL. « Genetic Paint: A Search for Salient Paintings. ». Dans *Proceedings of EvoWorkshops*, volume 3449 de *Lecture Notes in Computer Science*, p. 437-447. Springer, mars 2005. (Cité p. 88)
- [CL06] Ming-Te CHI et Tong-Yee LEE. « Stylized and Abstract Painterly Rendering System Using a Multiscale Segmented Sphere Hierarchy ». *IEEE Transactions on Visualization and Computer Graphics*, 12(1):61-2, février 2006. <[dx.doi.org/10.1109/TVCG.2006.14](https://doi.org/10.1109/TVCG.2006.14)>. (Cité p. 39, 43, 46 et 52)
- [Coo86] Robert L. COOK. « Stochastic sampling in computer graphics ». *ACM Transaction on Graphics*, 5(1):51-2, 1986. <[doi.acm.org/10.1145/7529.8927](https://doi.org/10.1145/7529.8927)>. (Cité p. 50)
- [CT02] Nelson Siu-Hang CHU et Chiew-Lan TAI. « An Efficient Brush Model for Physically-Based 3D Painting ». Dans *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, page 413, Washington, DC, USA, 2002. IEEE Computer Society. (Cité p. 87)
- [CT04] Nelson Siu-Hang CHU et Chiew-Lan TAI. « Real-Time Painting with an Expressive Virtual Chinese Brush ». *IEEE Computer Graphics and Applications*, 24(5):76-5, 2004. <[dx.doi.org/10.1109/MCG.2004.37](https://doi.org/10.1109/MCG.2004.37)>. (Cité p. 87)
- [CT05] Nelson Siu-Hang CHU et Chiew-Lan TAI. « MoXi: Real-Time Ink Dispersion in Absorbent Paper ». *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005)*, 24(3):504-11, août 2005. <visgraph.cs.ust.hk/MoXi/>. (Cité p. 87)
- [CTP⁺03] Matthieu CUNZI, Joëlle THOLLOT, Sylvain PARIS, Gilles DEBUNNE, Jean-Dominique GASCUÉL et Frédo DURAND. « Dynamic Canvas for Immersive Non-Photorealistic Walk-throughs ». Dans *Proceedings of Graphics Interface*, june 2003. (Cité p. 29)
- [Dan99] Eric DANIELS. « Deep Canvas in Disney's Tarzan ». Dans *SIGGRAPH Conference Abstracts and Applications*, page 200, 1999. <[doi.acm.org/10.1145/311625.312010](https://doi.org/10.1145/311625.312010)>. (Cité p. 38, 42, 43, 44, 46 et 75)
- [DBB06] Philip DUTRÉ, Kavita BALA et Philippe BEKAERT. *Advanced Global Illumination (Second Edition)*. A K Peters, 366 p., 2006. (Cité p. 49)

- [dBCvKO08] Mark de BERG, Otfried CHEONG, Marc van KREVELD et Mark OVERMARS. *Computational Geometry: Algorithms and Applications Third Edition*. Springer-Verlag, 386 p., 2008. (Cité p. 39)
- [Del34] Boris N. DELAUNAY. « Sur la sphère vide ». *Bulletin of Academy of Sciences of the USSR*, p. 793-00, 1934. (Cité p. 39)
- [DH06] Daniel DUNBAR et Greg HUMPHREYS. « A spatial data structure for fast Poisson-disk sample generation ». *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)*, 25(3):503-08, 2006. <dx.doi.org/10.1145/1141911.1141915>. (Cité p. 50)
- [DLK01] Eric DANIELS, Anastasios LAPPAS et George T. KATANICS. « Method and apparatus for three-dimensional painting ». USA, Brevet N° 6268865, <www.freepatentsonline.com/6268865.html>, juillet 2001. (Cité p. 38)
- [DOM⁺01] Frédo DURAND, Victor OSTROMOUKHOV, Mathieu MILLER, Francois DURANLEAU et Julie DORSEY. « Decoupling Strokes and High-Level Attributes for Interactive Traditional Drawing ». Dans *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, p. 71-2, London, UK, 2001. Springer-Verlag. (Cité p. 77)
- [Fai05] Mark D. FAIRCHILD. *Color Appearance Models*. John Wiley & Sons, 408 p., 2005. (Cité p. 70)
- [Fer64] James FERGUSON. « Multivariable Curve Interpolation ». *Journal of ACM*, 11(2):221-28, 1964. <dx.doi.org/10.1145/321217.321225>. (Cité p. 91)
- [FS76] Robert W. FLOYD et Louis STEINBERG. « An adaptive algorithm for spatial grey scale ». *Proceedings of the Society for Information Display*, 17:75-7, 1976. (Cité p. 119)
- [Gla98] Andrew GLASSNER. « Penrose Tiling ». *IEEE Computer Graphics and Applications*, 18(4):78-6, 1998. <dx.doi.org/10.1109/38.689670>. (Cité p. 51)
- [Gol96] Solomon W. GOLOMB. *Polyominoes: Puzzles, Patterns, Problems, and Packings*. Princeton University Press, 1996. (Cité p. 122)
- [GS86] Branko GRÜNBAUM et Geoffrey C. SHEPHARD. *Tilings and Patterns*. W.H. Freeman, 1986. (Cité p. 122)
- [GTDS04] Stéphane GRABLI, Emmanuel TURQUIN, Frédo DURAND et François X. SILLION. « Programmable Style for NPR Line Drawing ». Dans Alexander KELLER et Henrik Wann JENSEN, éditeurs, *Eurographics Symposium on Rendering*, p. 33-44, Norrköping, Sweden, 2004. Eurographics Association. <www.eg.org/EG/DL/WS/EGWR/EGSR04/033-044.pdf>. (Cité p. 66)
- [HE04] James HAYS et Irfan ESSA. « Image and video based painterly animation ». Dans *NPAR'2004: Third International Symposium on Non-Photorealistic Animation and Rendering*, p. 113-20, 2004. <dx.doi.org/10.1145/987657.987676>. (Cité p. 36, 39, 41 et 44)
- [Her98] Aaron HERTZMANN. « Painterly rendering with curved brush strokes of multiple sizes ». Dans *SIGGRAPH'98*, p. 453-60, 1998. <mr1.nyu.edu/publications/painterly98/>. (Cité p. 26, 42, 75, 87, 89, 94 et 98)
- [Her01] Aaron HERTZMANN. « Paint By Relaxation ». Dans *CGI '01: Computer Graphics International 2001*, p. 47-4. IEEE Computer Society, 2001. (Cité p. 88)
- [Her02] Aaron HERTZMANN. « Fast paint texture ». Dans *NPAR'2002: Second International Symposium on Non-Photorealistic Animation and Rendering*, p. 91-6, New York, NY, USA, 2002. ACM Press. <http://doi.acm.org/10.1145/508530.508546>. (Cité p. 43)
- [Her03] Aaron HERTZMANN. « A Survey of Stroke-Based Rendering ». *IEEE Computer Graphics and Applications*, 23(4):70-81, juillet 2003. <dx.doi.org/10.1109/MCG.2003.1210867>, Special Issue on Non-Photorealistic Rendering. (Cité p. 35 et 87)
- [HO95] Roger D. HERSCH et Victor OSTROMOUKHOV. « Introduction à la génération d'images en demi-tons ». Dans *Actes du congrès GUTenberg*, volume 21, 1995. (Cité p. 111 et 118)
- [Hol80] T. M. HOLLADAY. « An optimum algorithm for halftone generation for displays and hard copies ». *Proceedings of the Society for Information Display*, 21(2):185-92, 1980. (Cité p. 118)

- [Hop96] Hugues HOPPE. « Progressive meshes ». Dans *SIGGRAPH'96*, p. 99-08, New York, NY, USA, 1996. ACM Press. <dx.doi.org/10.1145/237170.237216>. (Cité p. 38)
- [HP00] Aaron HERTZMANN et Ken PERLIN. « Painterly rendering for video and interaction ». Dans *NPAP'2000: First International Symposium on Non-Photorealistic Animation and Rendering*, p. 7-2, 2000. <dx.doi.org/10.1145/340916.340917>. (Cité p. 39, 46 et 97)
- [HS04] Michael HALLER et Daniel SPERL. « Real-time painterly rendering for MR applications ». Dans *GRAPHITE '04*, p. 30-8, 2004. <dx.doi.org/10.1145/988834.988839>. (Cité p. 38, 43, 44, 46 et 52)
- [JEGPO02] Pierre-Marc JODOIN, Emric EPSTEIN, Martin GRANGER-PICHÉ et Victor OSTROMOUKHOV. « Hatching by example: a statistical approach ». Dans *NPAP'2002: Second International Symposium on Non-Photorealistic Animation and Rendering*, p. 29-6, New York, NY, USA, 2002. ACM Press. <doi.acm.org/10.1145/508530.508536>. (Cité p. 82)
- [Jon06] Thouis R. JONES. « Efficient Generation of Poisson-Disk Sampling Patterns ». *journal of graphics tools*, 11(2):27-36, 2006. (Cité p. 50)
- [Kal07] Alex KALAJDZIAN. « Automated Landscape Painting in the Style of Bob Ross ». Master's thesis, University of Waterloo, School of Computer Science, 2007. (Cité p. 89 et 90)
- [Kan99] Xiao-Kang KANG. *Digital Color Halftoning*. John Wiley & Sons, 1999. (Cité p. 109)
- [KC05] Matthew KAPLAN et Elaine COHEN. « A Generative Model For Dynamic Canvas Motion ». Dans *Proceedings of the First Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, p. 49-6, 2005. <dx.doi.org/10.2312/COMPAESTH/COMPAESTH05/049-056>. (Cité p. 32)
- [KCODL06] Johannes KOPF, Daniel COHEN-OR, Oliver DEUSSEN et Dani LISCHINSKI. « Recursive Wang tiles for real-time blue noise ». *ACM Transactions on Graphics*, 25(3):509-18, 2006. <dx.doi.org/10.1145/1141911.1141916>. (Cité p. 51)
- [KDMF03] Robert D. KALNINS, Philip L. DAVIDSON, Lee MARKOSIAN et Adam FINKELSTEIN. « Coherent Stylized Silhouettes ». *ACM Transactions on Graphics*, 22(3):856-861, juillet 2003. (Cité p. 43)
- [KLK⁺00] Allison W. KLEIN, Wilmot LI, Michael M. KAZHDAN, Wagner T. CORRÊA, Adam FINKELSTEIN et Thomas A. FUNKHOUSER. « Non-photorealistic virtual environments ». Dans *SIGGRAPH'2000*, p. 527-34, 2000. <http://doi.acm.org/10.1145/344779.345075>. (Cité p. 30 et 31)
- [KMM⁺02] Robert D. KALNINS, Lee MARKOSIAN, Barbara J. MEIER, Michael A. KOWALSKI, Joseph C. LEE, Philip L. DAVIDSON, Matthew WEBB, John F. HUGHES et Adam FINKELSTEIN. « WY-SIWYG NPR: Drawing Strokes Directly on 3D Models ». *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2002)*, 21(3):755-62, 2002. <www.cs.princeton.edu/gfx/pubs/Kalnins_2002_WND>. (Cité p. 38, 43, 44, 46 et 82)
- [Knu87] Donald E. KNUTH. « Digital halftones by dot diffusion ». *ACM Transaction on Graphics*, 6(4):245-73, 1987. <doi.acm.org/10.1145/35039.35040>. (Cité p. 119)
- [LA97] David J. LIEBERMAN et J. P. ALLEBACH. « Efficient model based halftoning using direct binary search ». Dans *ICIP '97: Proceedings of the 1997 International Conference on Image Processing (ICIP '97) 3-Volume Set-Volume 1*, p. 775-778, Washington, DC, USA, 1997. IEEE Computer Society. (Cité p. 120)
- [LA00] Pingshan LI et Jan P. ALLEBACH. « Look-up-table based halftoning algorithm. ». *IEEE Transactions on Image Processing*, 9(9):1593-1603, 2000. <dx.doi.org/10.1109/83.862640>. (Cité p. 116)
- [LA02] Pingshan LI et Jan P. ALLEBACH. « Tone dependent error diffusion ». Dans *Proceedings SPIE Cot Electronic Imaging, (San Jose, CA)*, p. 293-01, 2002. (Cité p. 119)
- [LAG98] Daniel L. LAU, Gonzalo R. ARCE et Neal C. GALLAGHER. « Green-noise digital halftoning ». Dans *Proceedings of the IEEE*, p. 2424-444. Marcel Dekker, Inc, 1998. <dx.doi.org/10.1109/5.735449>. (Cité p. 119)

- [Lar03] LAROUSSE, éditeur. *Le Petit Larousse*. 2003. page 969. (Cité p. 22)
- [LBG80] Yoseph LINDE, Andrès BUZO et Robert M. GRAY. « An Algorithm for Vector Quantizer Design ». *Communications, IEEE Transactions on [legacy, pre - 1988]*, 28(1):84-5, janvier 1980. (Cité p. 40)
- [LD05] Ares LAGAE et Philip DUTRÉ. « A procedural object distribution function ». *ACM Transaction on Graphics*, 24(4):1442-461, 2005. <dx.doi.org/10.1145/1095878.1095888>. (Cité p. 51 et 125)
- [LD08] Ares LAGAE et Philip DUTRÉ. « A Comparison of Methods for Generating Poisson Disk Distributions ». *Computer Graphics Forum*, 21(1):114-29, 2008. <dx.doi.org/10.1111/j.1467-8659.2007.01100.x>. (Cité p. 49)
- [Lin94] Qian LIN. « Improving halftone uniformity and tonal response ». Dans *IS&T Tenth International Congress on Advances in Non-Impact Printing*, 1994. (Cité p. 115)
- [Lit97] Peter LITWINOWICZ. « Processing images and video for an impressionist effect ». Dans *SIGGRAPH'97*, p. 407-14, 1997. <doi.acm.org/10.1145/258734.258893>. (Cité p. 39, 40, 41, 43, 44 et 46)
- [Llo82] Stuart P. LLOYD. « Least squares quantization in PCM ». *Information Theory, IEEE Transactions on*, 28(2):129-37, mars 1982. (Cité p. 40, 51 et 126)
- [LM05] Eric LUM et Kwan-Liu MA. « Expressive Line Selection by Example ». *The Visual Computer (Proceedings of Pacific Graphics 2005 Conference)*, 21(8):811-820, septembre 2005. (Cité p. 82)
- [LN98] Angeles LOSADA et Rafael NAVARRO. « Point spread function of the human eye obtained by a dual double-pass method ». *Pure and Applied Optics: Journal of the European Optical Society Part A*, 7:L7-L13, 1998. <http://stacks.iop.org/0963-9659/7/L7>. (Cité p. 104)
- [Lue01] David P. LUEBKE. « A Developer's Survey of Polygonal Simplification Algorithms ». *IEEE Computer Graphics and Applications*, 21(3):24-5, mai 2001. (Cité p. 38)
- [Mar82] David MARR. *Vision - A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H.Freeman & Co Ltd, 397 p., 1982. (Cité p. 104)
- [MBV⁺07] Florent MOULIN, Laurence BOISSIEUX, David VANDERHAEGHE, Joëlle THOLLOT et Pascal BARLA. « Dumb Luck ». Eurographics Animation Theatre, septembre 2007. (Cité p. 77)
- [Mei96] Barbara J. MEIER. « Painterly rendering for animation ». Dans *SIGGRAPH'96*, p. 477-84, New York, NY, USA, 1996. ACM Press. <dx.doi.org/10.1145/237170.237288>. (Cité p. 38, 41, 43, 44, 46, 47, 52, 63 et 65)
- [Met99] Panagiotis Takis METAXAS. « Optimal parallel error diffusion dithering ». Dans *Proceedings of SPIE Color imaging : device-independent color, color hardcopy, and graphic arts IV*, volume 3648, p. 485-494, janvier 1999. (Cité p. 119)
- [Met03] Panagiotis Takis METAXAS. « Parallel digital halftoning by error-diffusion ». Dans *PCK50: Proceedings of the Paris C. Kanellakis memorial workshop on Principles of computing & knowledge*, p. 35-1, New York, NY, USA, 2003. ACM Press. <dx.doi.org/10.1145/778348.778355>. (Cité p. 119)
- [MF92] Michael McCool et Eugene Frume. « Hierarchical Poisson disk sampling distributions ». Dans *Proceedings of the conference on Graphics interface '92*, p. 94-05, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc. (Cité p. 51 et 58)
- [MP91] T. MITSU et K. J. PARKER. « Digital halftoning using a blue-noise mask ». Dans *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, p. 2809-2812, 1991. <dx.doi.org/10.1109/ICASSP.1991.150986>. (Cité p. 115)
- [MT02] Kyoko MURAKAMI et Reiji TSURUNO. « Pastel-like rendering considering the properties of pigments and support medium ». Dans *SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications*, p. 227-27, New York, NY, USA, 2002. ACM Press. <dx.doi.org/10.1145/1242073.1242240>. (Cité p. 43 et 52)

- [MT03] Kyoko MURAKAMI et Reiji TSURUNO. « Polygon-Based Pastel-Like Rendering for Animation: Improved algorithms for pastel-like rendering suitable for animation ». Dans *SIGGRAPH '03: ACM SIGGRAPH 2003 Sketches & Applications*, p. 1-, New York, NY, USA, 2003. ACM Press. <<http://portal.acm.org/citation.cfm?id=965400.1077119>>. (Cité p. 43 et 52)
- [ODJ04] Victor OSTROMOUKHOV, Charles DONOHUE et Pierre-Marc JODOIN. « Fast hierarchical importance sampling with blue noise properties ». *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, 23(3):488-95, 2004. <[dx.doi.org/10.1145/1015706.1015750](https://doi.org/10.1145/1015706.1015750)>. (Cité p. 51 et 123)
- [Ost01] Victor OSTROMOUKHOV. « A simple and efficient error-diffusion algorithm ». Dans *SIGGRAPH'2001*, p. 567-72, New York, NY, USA, 2001. ACM Press. <[dx.doi.org/10.1145/383259.383326](https://doi.org/10.1145/383259.383326)>. (Cité p. 119)
- [Ost07] Victor OSTROMOUKHOV. « Sampling with polyominoes ». *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):78, 2007. <[dx.doi.org/10.1145/1276377.1276475](https://doi.org/10.1145/1276377.1276475)>. (Cité p. 122)
- [Pal99] Stephen E. PALMER. *Vision Science: Photons to Phenomenology*. The MIT Press, Cambridge, Massachusetts., 1999. (Cité p. 104)
- [Per85] Ken PERLIN. « An Image Synthesizer ». *Computer Graphics (Proceedings of SIGGRAPH'85)*, 19(3), 1985. (Cité p. 30)
- [PFS03] Oscar Meruvia PASTOR, Bert FREUDENBERG et Thomas STROTHOTTE. « Real-Time Animated Stippling ». *IEEE Computer Graphics and Applications*, 23(4):62-8, 2003. <[dx.doi.org/10.1109/MCG.2003.1210866](https://doi.org/10.1109/MCG.2003.1210866)>. (Cité p. 38, 44, 46, 63 et 69)
- [PHWF01] Emil PRAUN, Hugues HOPPE, Matthew WEBB et Adam FINKELSTEIN. « Real-time hatching ». Dans *SIGGRAPH'2001*, p. 581-86, New York, NY, USA, 2001. ACM Press. <[doi.acm.org/10.1145/383259.383328](https://doi.org/10.1145/383259.383328)>. (Cité p. 31)
- [PQW⁺08] Wai-Man PANG, Yingge QU, Tien-Tsin WONG, Daniel COHEN-OR et Pheng-Ann HENG. « Structure-aware halftoning ». *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, 27(3):89, 2008. <[dx.doi.org/10.1145/1360612.1360688](https://doi.org/10.1145/1360612.1360688)>. (Cité p. 120)
- [Rob02] ROBERT, éditeur. *Le Nouveau Petit Robert*. 2002. page 2499. (Cité p. 22)
- [SBC06] Maria SHUGRINA, Margrit BETKE et John P. COLLOMOSSE. « Empathic Painting: Interactive stylization through observed emotional state ». Dans *NPAP'2006: Fourth International Symposium on Non-Photorealistic Animation and Rendering*, p. 87-6, New York, NY, USA, juin 2006. ACM Press. <[dx.doi.org/10.1145/1124728.1124744](https://doi.org/10.1145/1124728.1124744)>. (Cité p. 89)
- [Sha02] Gaurav SHARMA. *Digital Color Imaging Handbook*. CRC Press, Inc., Boca Raton, FL, USA, 2002. (Cité p. 109)
- [SLK05] Kaleigh SMITH, Yunjun LIU et Allison KLEIN. « Animosaics ». Dans *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, p. 201-08, New York, NY, USA, 2005. ACM Press. <[doi.acm.org/10.1145/1073368.1073397](https://doi.org/10.1145/1073368.1073397)>. (Cité p. 40 et 46)
- [ST90] Takafumi SAITO et Tokiichiro TAKAHASHI. « Comprehensible rendering of 3-D shapes ». *Computer Graphics (Proceedings of SIGGRAPH'90)*, 24(4):197-06, 1990. <[doi.acm.org/10.1145/97879.97901](https://doi.org/10.1145/97879.97901)>. (Cité p. 35 et 67)
- [Uli87] Robert ULICHNEY. *Digital Halftoning*. The MIT Press, Cambridge, MA, USA, 362 p., 1987. (Cité p. 50)
- [Uli93] Robert ULICHNEY. « The Void-and-Cluster Method for Generating Dither Arrays ». Dans Jan P. ALLEBACH et B. ROGOWITZ, éditeurs, *Proceedings SPIE 1993*, p. 332-343, 1993. (Cité p. 113, 115 et 129)
- [Uli94] Robert ULICHNEY. « Filter Design for Void-and-Cluster Arrays ». *Society for Information Display International Symposium Digest of Technical Papers*, 25:809-812, 1994. (Cité p. 115)

- [Uli95] Robert ULICHNEY. « Spatial Extent of Void and Cluster Finding Filters ». Dans *The Society for IMAGING SCIENCE et TECHNOLOGY*, éditeurs, *IS&T's Eleventh International Congress on Advances in Non-impact Printing Technologies*, p. 430-433, 1995. (Cité p. 115)
- [VBTS06a] David VANDERHAEGHE, Pascal BARLA, Joëlle THOLLOT et François SILLION. « Dynamic Painting of Animated 3D scenes ». NPAR Poster Session, juin 2006. (Cité p. 15)
- [VBTS06b] David VANDERHAEGHE, Pascal BARLA, Joëlle THOLLOT et François SILLION. « Un Système Interactif et Controlable de Peinture pour l'Animation 3D ». Dans *AFIG '06 (Actes des 19èmes journées de l'AFIG)*, novembre 2006. <artis.imag.fr/Publications/2006/VBTS06b>. (Cité p. 15)
- [VBTS06c] David VANDERHAEGHE, Pascal BARLA, Joëlle THOLLOT et François X. SILLION. « A dynamic drawing algorithm for interactive painterly rendering ». Dans *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, page 100, New York, NY, USA, 2006. ACM Press. <[doi.acm.org/10.1145/1179849.1179974](https://doi.org/10.1145/1179849.1179974)>. (Cité p. 15)
- [VBTS07a] David VANDERHAEGHE, Pascal BARLA, Joëlle THOLLOT et François SILLION. « Dynamic point distribution for stroke-based rendering ». Dans *Rendering Techniques 2007 (Proceedings of the Eurographics Symposium on Rendering)*, p. 139-46, 2007. <<http://artis.imag.fr/Publications/2007/VBTS07a>>, acceptance rate 35% (33/94). (Cité p. 15)
- [VBTS07b] David VANDERHAEGHE, Pascal BARLA, Joëlle THOLLOT et François SILLION. « Rendu Peinture, Interactif et Controlable, de Scènes 3D ». *Revue Electronique Francophone d'Informatique Graphique*, 1(1):53-62, jan 2007. <<http://artis.imag.fr/Publications/2007/VBTS07b>>. (Cité p. 15)
- [VO08] David VANDERHAEGHE et Victor OSTROMOUKHOV. « Polyomino-Based Digital Halftoning ». Dans *IADIS Computer Graphics and Visualization 2008 (part of MCCSIS 2008)*, juillet 2008. (Cité p. 15)
- [WD05] John WILLATS et Frédo DURAND. « Defining Pictorial Style: Lessons from Linguistics and Computer Graphics ». *Axiomathes*, 15(2), 2005. (Cité p. 35)
- [Wika] WIKIPEDIA. « Beta movement ». <en.wikipedia.org/wiki/Beta_movement>, consulté le 2/12/08. (Cité p. 20)
- [Wikb] WIKIPEDIA. « Les Simpson ». <fr.wikipedia.org/wiki/Les_Simpson>, consulté le 28/05/08. (Cité p. 25)
- [Wil83] Lance J. WILLIAMS. « Pyramidal parametrics ». *Computer Graphics (Proceedings of SIGGRAPH'83)*, p. 1-1, 1983. <portal.acm.org/citation.cfm?id=801126>. (Cité p. 30)
- [WPFH02] Matthew WEBB, Emil PRAUN, Adam FINKELSTEIN et Hugues HOPPE. « Fine tone control in hardware hatching ». Dans *NPAR'2002: Second International Symposium on Non-Photorealistic Animation and Rendering*, p. 53-8, New York, NY, USA, 2002. ACM Press. <[doi.acm.org/10.1145/508530.508540](https://doi.org/10.1145/508530.508540)>. (Cité p. 31)
- [Yel82] John I. YELLOTT. « Spectral analysis of spatial sampling by photoreceptors: Topological disorder prevents aliasing ». *Vision Research*, 22:1205-210, 1982. (Cité p. 49)
- [Yel83] John I. YELLOTT. « Spectral consequences of photoreceptor sampling in the rhesus retina ». *Science*, 221:382-85, 1983. (Cité p. 49)
- [YN51] John A. C. YULE et W. J. NIELSEN. « The penetration of light into paper and its effect on halftone reproduction ». Dans *Proceedings of the Technical Association of the Graphic Arts (TAGA)*, volume 3, 1951. (Cité p. 109)

Filmographie

Nous présentons ici quelques œuvres d'animation, créées manuellement, dans les styles qui inspirent nos travaux. Le travail de Georges Schwizgebel présente un large panel de peintures animées, en particulier « Perspectives » (1975), « Le sujet du tableau » (1989) et « L'homme sans ombre » (2004). Un DVD regroupant treize courts métrages de Georges Schwizgebel est distribué par Les films du Paradox.

Plus médiatisé, Alexandre Petrov a réalisé plusieurs courts métrages en peinture animée qui frappent par leurs détails et leur matière, nous citerons en particulier « Le Vieil Homme et la mer » (2004, DVD aux Éditions Montparnasse) et « Mon amour » (2006).

Matrine Chatrand a étudié la peinture animée avec Alexandre Petrov, elle est l'auteur de « Âme noire » (2000) et « MacPherson » (à paraître).

Hu Jinqing, Zhou Kequin et Te Wei expérimentent une technique d'aquarelle animée (principalement des aquarelles fixes sur plusieurs calques mobiles de celluloid), avec notamment « Impression de montagne et d'eau » (1988, DVD distribué par Les films du Paradox)

Sans entrer dans le détail de leur filmographie, nous tenons à citer le travail de Marie Paccou (peinture animée), Florence Miaillhe, David Myriam (animation de sable) <art-engage.net>, Caroline Leaf <www.carolineleaf.com>. Leurs divers courts métrages montrent l'expressivité et la variété de styles que permet l'animation.