

UNIVERSITÉ DE MONTRÉAL

ANIMATION PAR L'EXEMPLE DE TEXTURE VECTORIELLE

THOMAS JAN MAHAMAD
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
JUN 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ANIMATION PAR L'EXEMPLE DE TEXTURE VECTORIELLE

présenté par : JAN MAHAMAD Thomas

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. OZELL Benoît, Ph. D, président

M. HURTUT Thomas, Ph. D., membre et directeur de recherche

Mme THOLLOT Joëlle, Ph. D, membre et codirectrice de recherche

M. VERGNE Romain, Ph. D, membre et codirecteur de recherche

M. PESANT Gilles, Ph. D., membre

DÉDICACE

« Ce que j'aimerais être, mais alors vraiment, c'est un soc de charrue. Je ne sais pas en quoi ça consiste, mais ça me paraît une existence qui laisse une trace »
Deux fleurs dans Les Annales du Disque-Monde, Tome 1 : La Huitième couleur. Terry Pratchett.

REMERCIEMENTS

Je tiens à remercier mes directeurs de recherche Thomas Hurtut, Joëlle Thollot et Romain Vergne pour m'avoir donné l'opportunité de travailler sur ce sujet, leurs sages conseils et pour m'avoir évité de m'attarder sur des pistes peu pertinentes.

Je tiens également à remercier ma famille, et mes amis Stanislas, Anne, Adrien, Simon, Mahmoud, Jeanne, François et Raphaël pour leur soutien continu et leurs idées pertinentes pendant toute la durée de cette maîtrise.

Mes remerciements vont également à Albin Pluche et Geoffrey Dromard qui m'ont fait profiter de leur regard d'artiste sur l'utilité de mes travaux.

J'aimerais remercier tous les membres du comité exécutif de l'AÉCSP avec qui j'ai eu le plaisir de travailler et de grandir durant mon mandat de trésorier, ainsi que Vanessa Casanovas i Michel et Amélie Ferland des SEP sans qui je n'aurais pas pu mener ma maîtrise à bout.

Je remercie enfin les membres du personnel de Coopoly pour leur bonne humeur constant m'ayant permis de ne pas voir défiler les heures passées à travailler en leur compagnie.

RÉSUMÉ

La facilité d'accès et la multiplication des écrans pour les particuliers dans les dernières décennies ont favorisé la diffusion de contenus animés, dont la complexité augmente toujours plus en suivant l'augmentation de puissance de calcul des appareils disponibles. Ce phénomène crée l'expression d'un besoin de réaliser des animations de manière intuitive et efficace de la part des utilisateurs d'outils d'animation. Nous nous intéressons dans ce mémoire aux techniques permettant la synthèse des textures animées, c'est-à-dire la création automatique de textures animées guidée par des algorithmes. Les textures animées sont présentes dans de nombreux types de réalisations, allant de la projection murale sur des bâtiments à l'enrichissement de présentations en passant par l'habillage de scènes dans des films ou jeux vidéos. Bien que de nombreux travaux visent à synthétiser des textures animées, peu se penchent sur l'animation de textures créées en amont.

Nous proposons dans ce mémoire un système permettant de répondre à cette absence de solution en nous intéressant à plusieurs aspects clés du processus d'animation et des textures. En s'appuyant sur les propriétés statistiques des distributions dans les textures, la méthode proposée permet d'animer de manière intuitive toute la texture d'entrée à partir d'un petit exemple. L'utilisateur est invité à spécifier la structure spatiale de son exemple d'animation, pour former ce que nous appelons des groupes de mouvement, afin de guider une recherche gloutonne des dispositions de couleur et spatiales similaires. L'animation des éléments statiques dans la texture d'entrée ne peut être réalisée en appliquant simplement les mouvements de l'animation d'exemple à cause de la variabilité de disposition spatiale des éléments inhérente aux propriétés statistiques des textures. Une analyse de la hiérarchie et de la structure des mouvements spécifiés par l'utilisateur en amont permet d'adapter au mieux les mouvements aux différentes dispositions spatiales. Plusieurs contraintes sont établies dans la méthodologie afin de permettre de synthétiser des animations à partir d'une déformation rigide minimale des mouvements d'exemple.

La méthode proposée dans ce mémoire donne des résultats prometteurs aussi bien sur la partie de recherche de similarités que sur l'adaptation des mouvements, néanmoins le résultat est dépendant de la spécification de groupes de mouvements suffisamment différents les uns des autres par l'utilisateur.

ABSTRACT

Easy access to multiple screens for people and the growth in computational power in the last few decades promoted the emergence of more and more complex animated contents. This phenomenon creates a need for intuitive and efficient animation tools. In this thesis, we focus on methods allowing synthesis of animated textures, i.e. algorithms guided, or computer assisted technic to create animated textures. Such textures are widely used in numerous applications : wall projections on buildings, presentation enhancements, videogames, movies, etc. Although many research aims to synthesize animated textures, they focus mainly on creation and animation at the same time. Few are addressing the animation of already existing textures.

We propose a system to address the problem of animating static textures by focusing on several key aspects of the textures and the animation process. Based on properties of the statistical distributions, the proposed approach allows intuitive animation of all elements in an input texture from a small example set of animated elements. The user has to specify the spatial structure of the example animation to form what we call the motion groups to guide a greedy search for similar dispositions based on spatial and color similarities. The animation of the static elements in the input texture can't be achieved by simply applying the same motion as a given example because of small variations in the spatial arrangement allowed by the statistical distributions of textures. An analysis of the user specified motion hierarchy and structure allows us to adapt the motion to the differences in spatial arrangements. Several constraints are established in the methodology in order to synthesize animations with a minimum rigid deformation of example motions.

The proposed method shows promising results in both the similarity search and the adaptation of motions. However the result is dependent on the specification of the motion groups with enough differences from one another.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	ix
LISTE DES FIGURES	x
LISTE DES SIGLES ET ABRÉVIATIONS	xii
CHAPITRE 1 INTRODUCTION	1
1.1 Contexte	1
1.1.1 Texture	1
1.1.2 Animation de textures vectorielles	5
1.2 Approche proposée	5
1.3 Objectifs	6
1.4 Hypothèses	6
1.5 Plan du mémoire	6
CHAPITRE 2 REVUE DE LITTÉRATURE	8
2.1 Outils d'animation vectorielle	8
2.2 Textures animées	9
2.2.1 Textures animées paramétriques	9
2.2.2 Textures animées non paramétriques	10
CHAPITRE 3 SPÉCIFICATION DES GROUPES DE MOUVEMENT	12
3.1 Groupes de mouvements	12
3.2 Hiérarchie de mouvement dans les groupes.	13
3.2.1 Mouvements et transformations	13
3.2.2 Hiérarchie des mouvements	13

CHAPITRE 4	RECHERCHE DE SIMILARITÉS DANS LA TEXTURE	15
4.1	État de l'art	15
4.2	Méthodologie	16
4.2.1	Nombre d'occurrences d'un groupe	17
4.2.2	Complexité d'un groupe	18
4.2.3	Éléments isolés	18
4.2.4	Détermination des combinaisons dans un voisinage	18
4.2.5	Mesures de similarité d'un voisinage à un groupe	19
4.2.6	Fonction de coût	21
4.2.7	Seuils d'abandon d'une combinaison	22
4.3	Expérimentations	22
4.4	Discussion et limitations	30
CHAPITRE 5	ADAPTATION DU MOUVEMENT	32
5.1	Hierarchie des mouvements.	32
5.2	Contraintes et perception.	33
5.3	Méthode d'adaptation du mouvement	35
5.3.1	Cas particuliers	37
5.4	Expérimentations	40
5.4.1	Données	40
5.4.2	Méthodes appliquées	40
5.4.3	Résultats	41
5.5	Discussion et limitations	41
CHAPITRE 6	CONCLUSION	45
6.1	Synthèse des travaux	45
6.2	Limitations de la solution proposée	48
6.3	Améliorations futures	48
RÉFÉRENCES	RÉFÉRENCES	50

LISTE DES TABLEAUX

Tableau 4.1	Variation de σ_n pour Exemple 1	24
Tableau 4.2	Variation des σ_n pour Exemple 2	24
Tableau 4.3	Variation des σ_n pour Exemple 3	29
Tableau 4.4	Variation des σ_n pour Exemple 4	29

LISTE DES FIGURES

Figure 1.1	Différence entre une image et une texture. Si on déplace une fenêtre dans les images (a) et (b) les différentes régions de la texture (b) auront l'air similaires contrairement à celles de l'image (a). Source : [17].	2
Figure 1.2	Continuum des types de textures. La figure illustre les deux types extrêmes de textures régulières et stochastiques, ainsi que tout le spectre des exemples intermédiaires. Source : Wikipédia.	2
Figure 1.3	Illustration de la différence entre une texture raster et vectorielle pour la magnification.	3
Figure 1.4	Pipeline de la méthode proposée. En entrée, une texture dont la trajectoire des éléments de l'animation d'exemple est affichée. Après spécification des groupes de mouvement et recherche des groupes similaires dans la texture, tous les éléments sont animés en sortie.	7
Figure 3.1	Élément animé d'une translation. Chacun des points représente la position de l'élément à une image t_i . La courbe formée par ces points est appelée la trajectoire.	13
Figure 3.2	Composition de mouvements.	14
Figure 4.1	Représentation centroïdale	16
Figure 4.2	Résultat attendu de la recherche de similarités. Dans la texture (a), deux groupes de mouvement sont précisés par l'utilisateur. (b) présente le résultat que nous attendons en sortie de l'étape de recherche de similarité des groupes.	17
Figure 4.3	Voisinage d'un élément. Avec e_0^c comme racine, tous les éléments entre les deux cercles sont considérés comme candidats au voisinage n_1 avec $\sigma_n = 0.5$	19
Figure 4.4	Dissimilarité de premier ordre. La différence de distance à la racine est donnée par la différence entre les distances $d(e_0^c, e_i^c)$ et $d(e_0^r, e_i^r)$ dans l'image.	20
Figure 4.5	La racine du groupe de référence est alignée sur e_0^c (a). e_1^c est aligné sur chacun des points du groupe de référence (b) (c) (d)	21
Figure 4.6	Perturbation d'une texture. Pour une perturbation de 20%, les cercles représentent la zone dans laquelle se trouvera chacun des éléments après la perturbation	24
Figure 4.7	Texture Exemple 1	25

Figure 4.8	Texture Exemple 2	26
Figure 4.9	Texture Exemple 3	27
Figure 4.10	Texture Exemple 4	28
Figure 4.11	Exemples de mauvais résultats de la recherche de groupes similaires .	31
Figure 5.1	Pertes de cohérence de l’animation.	33
Figure 5.2	Dynamique du mouvement. Une transition de A vers B conserve la dynamique du mouvement, une transition de A vers C ne la conserve pas. Source : [17]	34
Figure 5.3	Processus d’adaptation de la trajectoire	36
Figure 5.4	Adaptation de la trajectoire de référence lorsque $\mathbf{p}_0(e_i^r) = \mathbf{p}_{min}$	38
Figure 5.5	Adaptation de la trajectoire de référence lorsque $\mathbf{p}_0(e_i^r) = \mathbf{p}_{min} = O$. .	39
Figure 5.6	cas général.	42
Figure 5.7	$\mathbf{p}_{min} = O$	43
Figure 5.8	$\mathbf{p}_0(e_i^r) = \mathbf{p}_{min}$	43
Figure 5.9	$\mathbf{p}_0(e_i^c) = \mathbf{p}_{min}$	44
Figure 5.10	$\mathbf{p}_0(e_i^r) = \mathbf{p}_{min} = O$	44
Figure 6.1	Toutes les trajectoires adaptées pour la texture d’exemple avec une perturbation de 30%.	46
Figure 6.2	Extraits de l’animation synthétisée sur la texture utilisée au début du Chapitre 4.	47

LISTE DES SIGLES ET ABRÉVIATIONS

MRF	Markov Random Field
PCA	Analyse en Composantes Principales
LDS	Système Linéaire Dynamique
CIE	Commission International de l'Eclairage

CHAPITRE 1 INTRODUCTION

Dans ce mémoire nous nous intéressons à l'animation de textures. Pour ce faire il est nécessaire en premier lieu de définir ce que nous entendons par les termes de texture et d'animation. Dans ce chapitre nous définissons en premier lieu ce qu'est une texture vectorielle, comment elles sont réalisées à la main ou synthétisées automatiquement, nous précisons ensuite ce que signifie d'animer ces textures, pour enfin présenter la méthode détaillée dans les chapitres à suivre.

1.1 Contexte

1.1.1 Texture

Définition

Suivant le domaine dans lequel il est utilisé, le terme de texture fait référence à des phénomènes différents. Par exemple, dans l'industrie textile il réfère à l'état tactile d'un matériau tissé. Dans le domaine informatique, les textures sont des images 2D contenant des variations de couleurs et sont souvent ensuite plaquées sur des objets 3D pour y ajouter des détails. Nous nous intéresserons par la suite aux textures visuelles.

Les textures visuelles décrivent un phénomène organisé ayant des distributions spatiales locales se répétant à des échelles plus grandes. Cette caractéristique de répétition, homogène à une échelle donnée, permet de distinguer une texture d'une image quelconque (Figure 1.1).

Classification des textures

La notion de répétition est formalisée par la stationnarité statistique. Il est possible de classer les textures suivant leur degré de régularité. Une texture régulière est composée d'une distribution d'éléments présentant un motif facilement identifiable répété à de nombreuses reprises. Une texture stochastique, au contraire, ne présentera pas de motif simplement identifiable, mais sera plutôt perçue comme du bruit. La plupart des textures présentes dans le monde réel ne sont pas classifiables clairement dans une catégorie ou une autre et forment un continuum, comme illustré dans la Figure 1.2. Par la suite, nous nous intéresserons plus particulièrement aux textures régulières dont les éléments sont clairement identifiables les uns des autres.

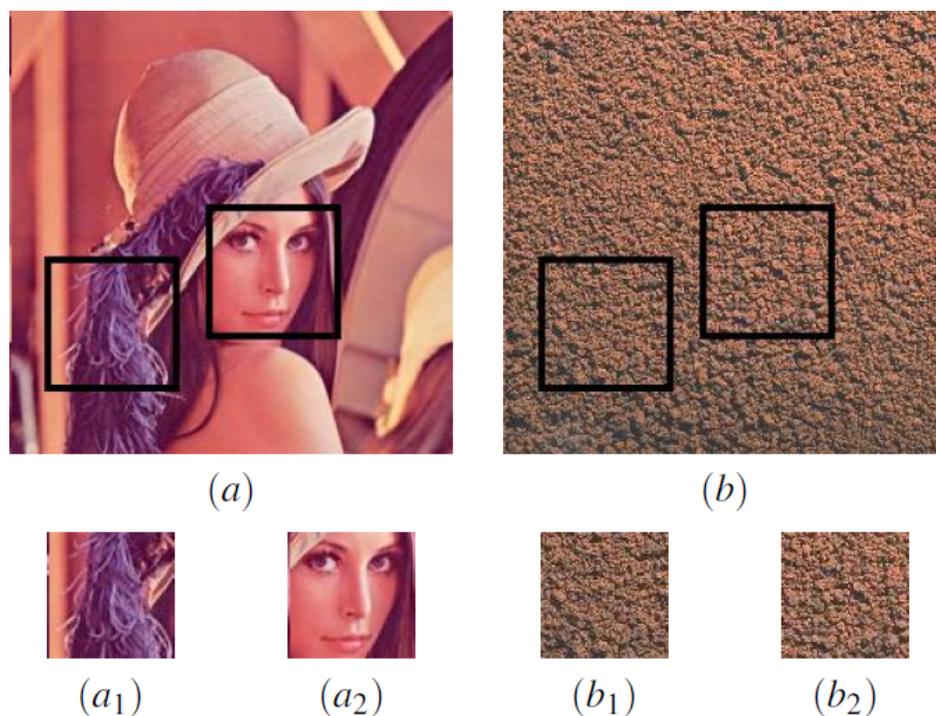


Figure 1.1 Différence entre une image et une texture. Si on déplace une fenêtre dans les images (a) et (b) les différentes régions de la texture (b) auront l'air similaires contrairement à celles de l'image (a). Source : [17].

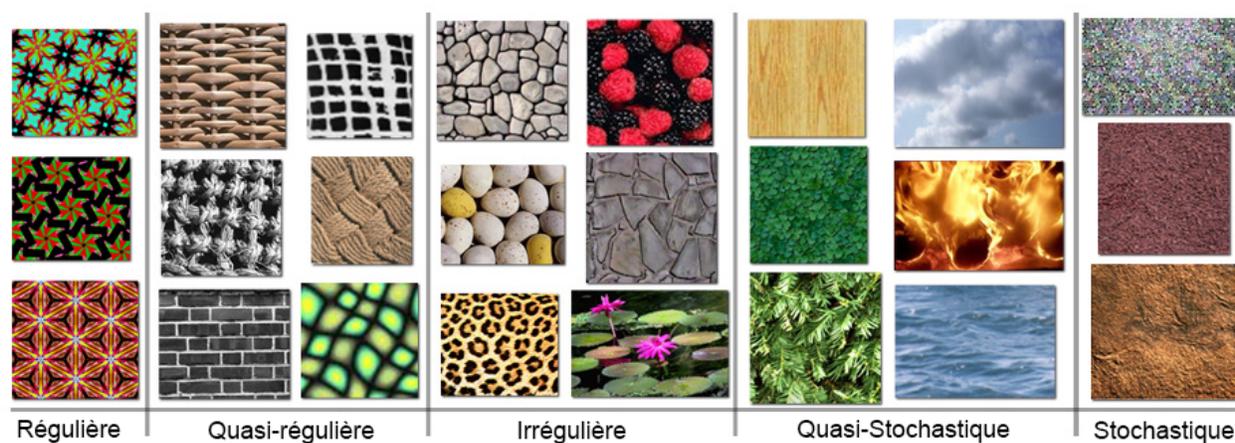


Figure 1.2 Continuum des types de textures. La figure illustre les deux types extrêmes de textures régulières et stochastiques, ainsi que tout le spectre des exemples intermédiaires. Source : Wikipédia.

Représentation des textures

Deux méthodes de représentation existent. Les textures raster sont composées d'une grille de pixels contenant chacun un triplet permettant de représenter une intensité de couleur. Par exemple, les textures naturelles ayant pour but de donner une représentation précise du monde réel sont souvent créées à base de photos. Le haut degré de détail qu'il est possible d'inclure dans l'image rend ces textures préférées pour le photo-réalisme. Le principal inconvénient de la représentation raster est qu'en cas de changement d'échelle de l'image, il n'est pas possible de créer simplement de l'information. Les pixels seront étirés ou compressés créant des artefacts. De plus, éditer les éléments distincts d'une texture raster se fait principalement avec des méthodes de filtrage. Dans une texture vectorielle, les éléments sont composés de courbes et primitives mathématiques permettant de représenter les motifs à différentes échelles sans perte de précision. Obtenir un niveau de détail comparable à une texture raster demande une quantité importante d'éléments rendant la texture coûteuse à stocker ce qui fait préférer leur utilisation pour les rendus non photo-réalistes. L'avantage majeur présenté par les textures vectorielles dans notre cas est la séparation des éléments présents en un ou plusieurs tracés. Il sera possible de traiter les éléments d'un motif séparément sans passer par des étapes de segmentation et de filtrage indispensables pour séparer les objets d'une texture raster. Nous nous limitons par la suite aux textures vectorielles, car elles ont l'avantage de faciliter l'édition rapide des éléments distincts les composant.

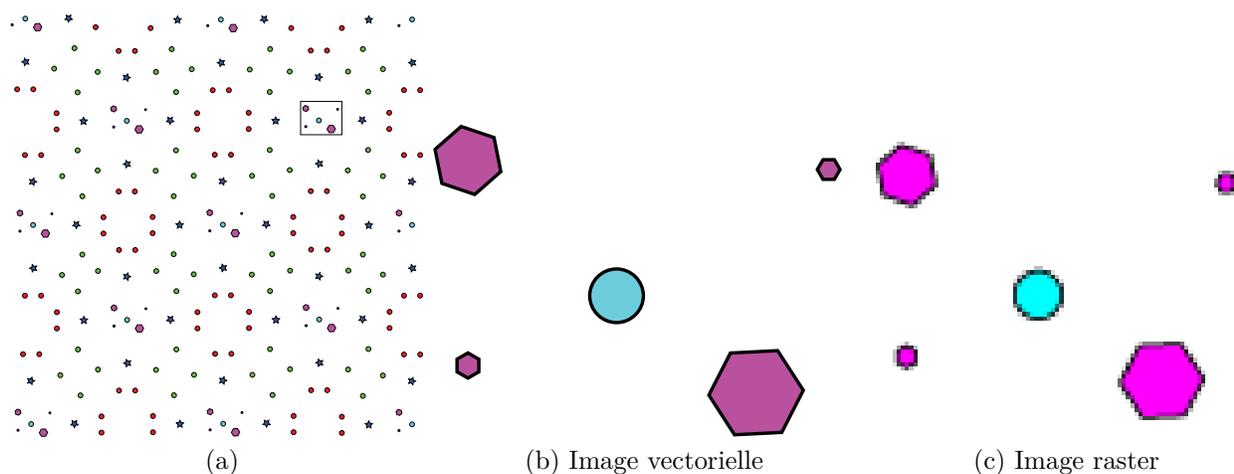


Figure 1.3 Illustration de la différence entre une texture raster et vectorielle pour la magnification.

Création des textures vectorielles

La création de textures est généralement réalisée par des artistes spécialisés dans ce domaine. Dessiner entièrement une texture à la main permet d’avoir un contrôle total sur le rendu final, mais c’est un processus long. Les artistes ont à leur disposition de nombreux systèmes informatiques permettant la création de textures vectorielles qui présentent chacun un compromis entre le niveau d’optimisation et de contrôle. Les outils de dessin permettent un contrôle poussé, mais peu de possibilités d’automatisation. Les outils de dessin vectoriel tels que Inkscape¹ ou Illustrator² mettent à la disposition des artistes des outils permettant de tracer eux-mêmes les primitives et de les organiser avec un système de calques. Cette méthode proche de la réalisation faite main permet de réaliser une première instance du motif et de le copier/coller à volonté. Des filtres sont fournis afin de perturber la position des calques et ainsi obtenir un rendu plus ou moins aléatoire. Ce processus de création demande du temps, car la distribution spatiale des motifs est en grande partie réalisée à la main. Les méthodes de synthèse automatique proposent quant à elles une automatisation du processus de création, mais réduisent généralement le contrôle de l’artiste sur la texture. La synthèse de textures offre plus d’abstractions dans la création des textures et vise à réduire le temps de création d’une texture tout en proposant des rendus parfois plus simples à modifier. À l’opposé de la création manuelle, la synthèse programmable des textures propose à l’utilisateur un ensemble d’opérateurs contrôlant la création de la texture [9]. Le niveau de spécification des algorithmes réalisant la texture varie suivant les outils. Processing³ est par exemple un langage de programmation basé sur le langage Java adapté à la création graphique statique ou interactive. Dans ce cas, l’utilisateur devra programmer la plus grande partie de son rendu. La spécification avancée des distributions et des algorithmes permet, avec la synthèse programmable, d’obtenir des résultats difficiles, voire impossibles à réaliser à la main. Malgré des travaux pour se rapprocher des utilisateurs ne possédant pas de bagage technique et des artistes, comme dans [9], la synthèse programmable reste peu intuitive et demande une forte abstraction par rapport à la réalisation des textures.

La synthèse automatique de texture par l’exemple demande à l’utilisateur de donner un exemple de ce qu’il souhaite réaliser à plus grande échelle. L’échantillon fourni est ensuite analysé afin d’extraire l’organisation spatiale des éléments constituant le motif. La seconde étape est la synthèse, qui génère une texture à partir de ce qui a été extrait durant l’analyse [5, 1]. Cette approche présente l’avantage d’être intuitive pour l’artiste en offrant un compromis entre les techniques exposées précédemment. La spécification représente un processus créatif

1. <https://inkscape.org/fr/>

2. http://www.adobe.com/ca_fr/products/illustrator.html

3. <https://processing.org/reference/libraries/>

proche des textures faites main, tout en confiant la partie répétitive à l'ordinateur qui en générant la texture retire une partie du contrôle sur sa spécification totale. Nous utilisons ce type de compromis en termes de spécification du souhait de l'utilisateur. Dans le travail proposé dans ce mémoire, l'utilisateur fournit un exemple de mouvements de référence à appliquer à une texture. Notre méthode analyse la texture et les mouvements pour les adapter automatiquement à l'ensemble de la texture statique. Les exemples de mouvements fournis par l'utilisateur sont propagés aux éléments encore statiques dans la texture d'entrée.

1.1.2 Animation de textures vectorielles

L'animation vise à donner l'illusion que des objets ou personnages inanimés sont en mouvement. Elle se compose d'une suite chronologique d'images qui, visionnées à une cadence assez élevée, donnent l'illusion d'un mouvement réel. La spécification du mouvement d'un objet dans une animation demande d'avoir pour chaque image de celle-ci l'état de l'objet. Le nombre d'images par seconde est un facteur clé de l'animation, plus celui-ci est élevé, plus le mouvement sera perçu comme fluide. Ce phénomène a longtemps été attribué à la persistance rétinienne, il est désormais communément accepté qu'il est dû à l'effet phi [15]. Par l'expérience il a été déterminé que la limite basse pour la perception d'un mouvement est de 16 images par seconde, à partir de 24 images par secondes le mouvement commence à être perçu comme fluide [?]. Dans le cas d'animations longues, spécifier l'état du mouvement pour chaque image demande donc de réaliser un nombre élevé d'images. L'animation de textures vectorielles présente des défis comparables à la synthèse. En effet, l'aspect stationnaire des textures nous mène à attendre un mouvement qui le sera aussi. Les outils permettant d'animer les textures, comme pour leur création, sont nombreux et varient suivant le degré de contrôle que l'utilisateur pourra exercer sur la spécification du mouvement.

1.2 Approche proposée

Notre but est de proposer un algorithme permettant d'animer une texture vectorielle composée d'éléments distincts à partir d'un exemple de mouvements spécifiés sur une sous partie de ces éléments. Le travail réalisé comporte trois parties. L'utilisateur doit spécifier les groupes d'éléments à animer et leurs relations dans le but de fournir des informations sur la topologie du motif de la texture. La recherche d'occurrence des sous-ensembles d'éléments dans le reste de la texture permet de déterminer à quels endroits la structure des groupes d'éléments se retrouve afin de mener à la dernière étape qui synthétisera l'animation dans le reste de la texture. Nous parlons alors d'adaptation du mouvement, étant donné que celui-ci pourra être modifié afin de respecter la distribution des éléments. L'ensemble de l'approche est illustrée

dans la Figure 1.4.

1.3 Objectifs

L'objectif de ce travail est de proposer une méthode d'animation par l'exemple de textures vectorielles régulières et irrégulières.

- \mathcal{O}_1 : Proposer à l'utilisateur une méthode de spécification des particularités de son animation ;
- \mathcal{O}_2 : Établir une méthode permettant de retrouver des dispositions spatiales similaires dans la texture fournie par l'utilisateur ;
- \mathcal{O}_3 : Généraliser l'animation d'exemple fournie par l'utilisateur dans toute la texture ;

1.4 Hypothèses

- \mathcal{H}_1 : Il est possible de proposer à l'utilisateur de spécifier les particularités de son animation de manière intuitive ;
- \mathcal{H}_2 : Il est possible de retrouver des dispositions spatiales similaires dans la texture à l'aide d'algorithmes gloutons ;
- \mathcal{H}_3 : La variation des dispositions spatiales peut être capturée dans l'animation par transformation rigide ;

1.5 Plan du mémoire

Ce mémoire est divisé en 6 chapitres. Dans le Chapitre 2, nous présentons une revue des travaux existants dans la littérature. Le Chapitre 3 introduit la spécification des groupes de mouvement. Il introduit la notion de groupe de mouvements, au cœur du reste des travaux. Dans le Chapitre 4 la méthode de recherche de voisinage pour la recherche du motif spécifié dans la texture au complet est présentée. Le Chapitre 5 traitera de l'adaptation du mouvement. Enfin, le Chapitre 6 conclut les travaux réalisés.

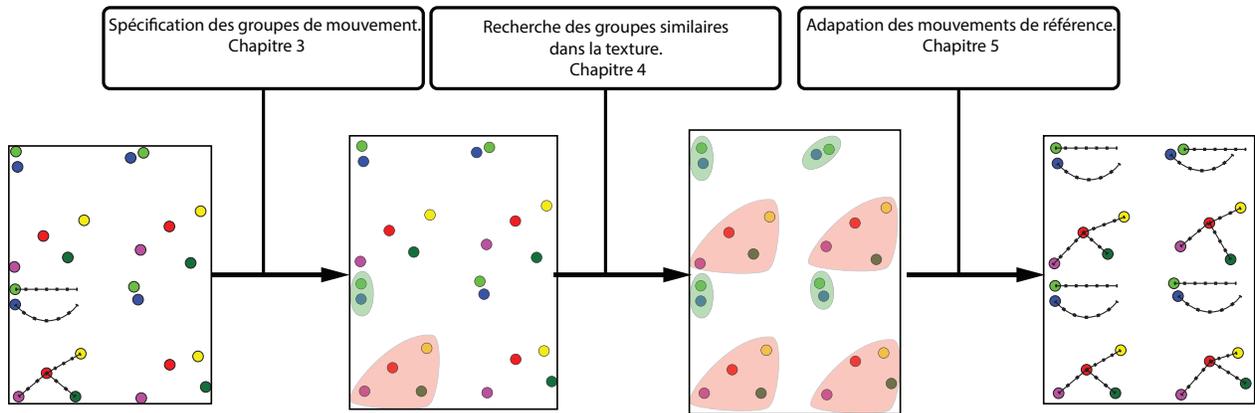


Figure 1.4 Pipeline de la méthode proposée. En entrée, une texture dont la trajectoire des éléments de l'animation d'exemple est affichée. Après spécification des groupes de mouvement et recherche des groupes similaires dans la texture, tous les éléments sont animés en sortie.

CHAPITRE 2 REVUE DE LITTÉRATURE

Ce mémoire vise à proposer un système permettant d’animer une texture vectorielle à partir d’un exemple spécifié par l’utilisateur. Ce chapitre a pour but de présenter les outils d’animation de textures vectorielles existants, ainsi que des méthodes de synthèse d’animation existantes afin de justifier nos choix méthodologiques.

2.1 Outils d’animation vectorielle

L’animation de chaque élément individuellement est rendue possible dans des logiciels tels qu’After Effects¹, qui proposent une spécification par trois types de méthodes : par images clés, par l’esquisse ou procédurales. Néanmoins, bien qu’il soit possible de copier/coller des mouvements, animer une texture composée de plusieurs centaines d’éléments est un travail long et répétitif. De plus, deux éléments identiques peuvent varier légèrement d’un motif à l’autre à cause de faibles variabilités autorisées par la nature stationnaire de la texture, demandant une adaptation du mouvement. La démocratisation des outils d’animation, qui effacent progressivement la séparation entre les animateurs et les artistes a poussé l’arrivée de méthodes de spécification plus intuitives. Les méthodes de spécification du mouvement basées sur les esquisses que l’on retrouve notamment dans le logiciel K-Sketch [2] permettent à l’utilisateur de dessiner à l’aide d’une tablette graphique ou d’une souris le mouvement souhaité [16, 12].

Animation par images clés. L’utilisation d’images clés est une application directe des scénarimages (*storyboard*) utilisés avant l’animation par ordinateur pour lesquels des animateurs experts définissent et dessinent des images clés de la séquence. Le terme s’est largement étendu afin de définir pour certaines images de l’animation non seulement la position d’un objet, mais également tout attribut variable lui étant associé (couleur, taille, orientation ...). Le calcul des déformations intermédiaires est réalisé par l’ordinateur, et l’animateur est en mesure de préciser les paramètres de l’interpolation. La spécification du mouvement par images clés est la technique la plus commune que l’on retrouve dans des logiciels tels que VPaint², spécialisé dans la réalisation d’animations vectorielles.

1. http://www.adobe.com/ca_fr/products/aftereffects.html

2. <http://www.vpaint.org/>

Animation procédurale L'animation procédurale se base sur la spécification de lois régissant le mouvement au cours du temps pour mettre des corps en mouvement. La simulation physique permet de déterminer le mouvement d'un objet soumis à des lois et contraintes physiques et à des forces internes ou externes. Les moteurs physiques traitent ce type de problématiques : dynamique des corps rigides, flexibles, cheveux, fluides. Les systèmes de particules visent à simuler le comportement d'objets flous (*fuzzy objects*). L'application de systèmes procéduraux (comme ceux proposés dans [13]) à l'animation de textures permet de spécifier des comportements généraux pour tous les éléments présents en créant un résultat pour toute la texture. Généralement, les lois précisées visent à spécifier un comportement naturel. Cette méthode est donc plus utilisée pour de l'animation de textures naturelles de matériaux et demandera une longue étape de paramétrisation des algorithmes. Cette méthode de spécification est peu intuitive du point de vue de l'artiste qui n'aura pas un contrôle direct sur le mouvement des objets.

Les méthodes que nous avons vues précédemment permettent de spécifier un mouvement dans une image sans être spécifiques aux textures. Dans le cas qui nous intéresse, l'artiste doit pouvoir spécifier un mouvement pour tous les éléments de la texture, quelle que soit sa dimension. L'animation procédurale permet d'animer un nombre quelconque d'éléments, et pourrait donc permettre d'animer avec un ensemble de règles une texture au complet comme proposé dans [13]. L'animation par l'exemple, comme pour la synthèse de textures, va à partir d'un échantillon de mouvements spécifié par l'utilisateur permettre d'animer la texture en entier en fournissant un contrôle plus intuitif qu'une approche procédurale.

2.2 Textures animées

Les textures animées sont un sujet actif de recherche depuis de nombreuses années pour leur possibilité d'enrichir des scènes ou documents. L'aspect stationnaire des textures permet de réaliser des analyses statistiques non seulement sur les dimensions spatiales, mais également sur le temps.

2.2.1 Textures animées paramétriques

Un des premiers travaux d'animation automatique de textures raster date du début des années 2000 [14]. Les auteurs définissent ce qu'ils appellent les textures vidéos (*Video Textures*) permettant de créer un extrait vidéo continu à partir d'une vidéo de taille quelconque. Pour y arriver, ils segmentent le flux d'entrée en différentes zones et pour chacune trouvent des séquences similaires à l'intérieur de la vidéo afin de réaliser des sauts en avant ou en arrière

entre les différentes parties et ainsi créer une animation continue. L’objectif est de créer du contenu dynamique pour des pages web, des films ou des jeux vidéos. Dans [3], les auteurs proposent de réaliser une approche plus paramétrique. Les images de la vidéo d’exemple sont exprimées dans un espace réduit par Analyse en Composantes Principales (PCA). Le passage d’une image à l’autre est réalisé par l’apprentissage d’un Système Linéaire Dynamique (LDS), permettant d’exprimer les transitions comme :

$$\begin{aligned} I_i &= Cx_i + w_i, \\ x_{i+1} &= Ax_i + v_i, \end{aligned} \tag{2.1}$$

avec C la matrice de projection des x_i obtenues par PCA, v_i et w_i représentent un bruit gaussien, et A la matrice de passage d’une image à l’autre apprise lors du LDS. Cette formulation permet d’obtenir des animations infinies (et donc bouclées), mais montre ses limites avec des exemples d’entrée plus longs. Dans [18], les auteurs proposent une méthode permettant la création de textures dynamiques à partir de très peu d’exemples. Les textures d’entrée sont décomposées en une série de patches, puis utilisées pour générer des patches temporels à l’aide de Champ aléatoire de Markov (MRF) en s’appuyant sur l’aspect stationnaire des textures. Une trajectoire est ensuite estimée à travers les différents patches projetés dans un espace latent.

Les méthodes citées précédemment, bien que donnant de bons résultats, sont majoritairement paramétriques et proposent peu de contrôle sur le résultat de la part de l’utilisateur. De plus, les textures sont considérées dans leur entièreté, et aucune séparation des éléments n’est proposée.

2.2.2 Textures animées non paramétriques

Dans [11], les auteurs proposent de synthétiser des textures animées en 3D, qu’ils nomment *Dynamic element textures* à partir d’exemples en cherchant à créer des résultats réalistes. Les éléments présents dans l’exemple d’entrée sont divisés en échantillons s , pour lesquels une représentation $u(s) = (p(s), t(s))$ est faite, avec $p(s)$ la position spatiale de l’échantillon et $t(s)$ le temps associé. Le voisinage d’un échantillon s' est alors défini comme l’ensemble des différences $\hat{u}(s', s) = u(s') - u(s)$. Un élément pouvant être composé de plusieurs échantillons, les auteurs proposent pour respecter la cohérence lors de la synthèse du mouvement d’imposer des contraintes topologiques sur les éléments suivant leurs dimensions (particules, fils, surfaces). Afin d’apporter plus de variabilité dans les résultats produits, le mouvement est spécifié sur plusieurs échelles et est séparé par des filtres lors de l’étape d’analyse en mouvement global ou local. Lors de l’étape de synthèse, le voisinage est généré en

copiant des patches de l'exemple d'entrée comme dans [10], en tenant compte en plus des contraintes topologiques. La synthèse du mouvement tient finalement compte de la disposition des éléments afin de gérer les collisions entre particules.

Draco est un outil d'augmentation d'illustrations visant à ajouter de la vie sur des images fixes avec des interactions par le dessin [8]. Pour ce faire, les auteurs proposent une série d'outils basés sur la notion de texture cinétique (*kinetic textures*). Ces outils se divisent en deux catégories : les textures d'émission et les textures oscillantes. Les textures d'émission permettent à l'artiste de spécifier un chemin à partir duquel seront émises des primitives également spécifiées par l'utilisateur. Les textures oscillantes proposent quant à elles de donner à un objet dessiné un mouvement d'oscillation. Les deux types de textures proposent comme dans [11] une spécification multi-échelle du mouvement. Un premier mouvement global donnera une animation de large amplitude, tandis qu'un mouvement plus fin permettra de rendre l'animation créée moins inerte. Une seconde itération de Draco, Kitty, permet d'ajouter en plus des textures cinétiques un système d'interactions et de timeline aux illustrations [7]. Les travaux que nous proposons se situent à mi-chemin entre Draco, et les *Dynamic element textures*. L'idée de base en termes de spécification des contraintes sur la topologie des éléments de la texture et de la recherche de voisinage est proche de [11]. Néanmoins, nous nous concentrerons comme dans Draco exclusivement sur des textures en deux dimensions et ne chercherons pas particulièrement à obtenir de réalisme, mais plutôt à conserver le sens artistique. L'étape de synthèse du mouvement diffère par les méthodes de modification de celui-ci pour respecter la disposition des éléments déjà présents et le sens perceptuel.

CHAPITRE 3 SPÉCIFICATION DES GROUPES DE MOUVEMENT

Dans la texture d'entrée, une partie des éléments sont animés et servent d'exemples pour synthétiser de nouvelles animations à appliquer aux éléments statiques dans la texture d'entrée. Pour atteindre notre objectif d'animer toute la texture à l'aide d'un exemple, notre méthode doit être en mesure de s'appuyer sur la structure de cette animation d'exemple afin de savoir comment l'adapter dans le reste de la texture statique. Dans ce chapitre nous présentons en premier lieu les groupes de mouvement. Des contraintes liées à la hiérarchie des mouvements sont ensuite ajoutées dans la seconde partie. Ces deux sections serviront de motivation à la formalisation du concept de groupes de mouvement, une manière pour l'utilisateur de spécifier dans l'animation de référence des ensembles d'éléments à conserver groupés lors de la recherche de similarité dans la texture.

3.1 Groupes de mouvements

Dans un ensemble d'éléments animés, le mouvement perçu pour chacun des éléments dépend en partie du mouvement des éléments alentours. L'interprétation du mouvement par le cerveau donne lieu à de nombreuses inférences guidées par tout l'environnement du mouvement [4]. Dans une série de vidéos, le Dr. Johansson¹ met en évidence ces phénomènes avec notamment l'exemple de trois points sur un plan 2D dont deux sont animés d'une translation horizontale de part et d'autre du troisième réalisant une ellipse : si les positions horizontales des trois éléments sont synchronisées, le spectateur a l'impression de voir un mouvement diagonal plutôt qu'elliptique. Ces artifices perceptuels sont souvent utilisés dans le milieu de l'animation. Nous proposons d'introduire la notion de groupe de mouvements afin de permettre à l'utilisateur d'explicitier le fait que les mouvements des éléments d'un sous-ensemble doivent fonctionner ensemble. L'utilisateur peut réaliser des listes d'éléments, chacun des éléments ne doit pouvoir appartenir qu'à une seule de ces listes qui formeront chacune un groupe de mouvements.

1. <https://www.youtube.com/watch?v=1F5ICP9SYLU>

3.2 Hiérarchie de mouvement dans les groupes.

3.2.1 Mouvements et transformations

Le mouvement d'un élément est défini comme une fonction qui à chaque image de l'animation associe une position à l'élément (Figure 3.1). Le déplacement de l'élément d'une image à l'autre peut donc être exprimé par une transformation. Pour la suite des travaux, nous nous intéresserons plus particulièrement aux transformations dites projectives, c'est-à-dire des transformations bijectives qui envoient des droites, plans et espaces vers des droites, plans et espaces. L'ensemble de ces transformations contient la rotation (R), la translation (T) et la mise à l'échelle (S).

Ainsi pour réaliser une rotation d'un angle θ autour d'un point arbitraire $\mathbf{p}(x_p, y_p, 1)$ il faudra réaliser la transformation suivante $T_p R_\theta T_{-p}$. La possibilité de combiner les transformations offre un moyen d'expression simple pour des mouvements complexes : la composition des mouvements.

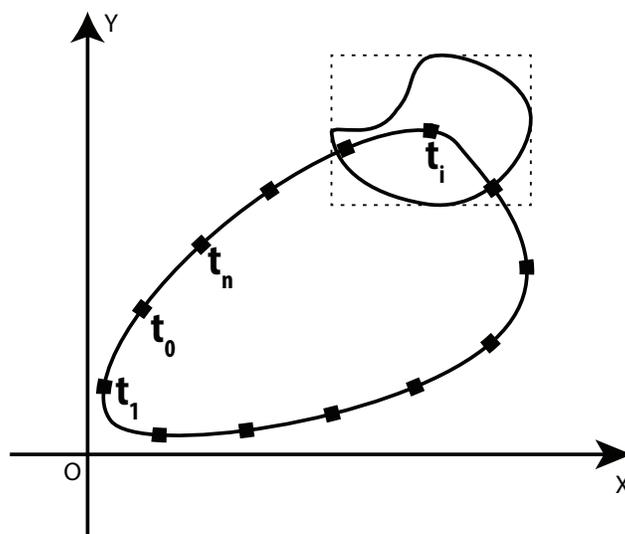


Figure 3.1 Élément animé d'une translation. Chacun des points représente la position de l'élément à une image t_i . La courbe formée par ces points est appelée la trajectoire.

3.2.2 Hiérarchie des mouvements

La spécification d'animations complexes est souvent réalisée par de la composition de mouvements. C'est-à-dire une séparation du mouvement en une combinaison hiérarchisée de mouvements plus simples à décrire. Généralement, les logiciels d'animation proposent de spécifier manuellement des relations parent-enfant entre les éléments afin de créer des mouvements

composés. Lorsqu'un élément possède un parent, les caractéristiques spatiales de cet élément sont exprimées par rapport à son parent et non plus par rapport à l'origine de l'image. Par exemple, un satellite en rotation autour d'une planète, elle-même en rotation autour d'une étoile, possèdera une trajectoire hélicoïdale enroulée autour de la trajectoire de sa planète. Ce mouvement peut être simplement exprimé par la composition de deux rotations : la planète autour du soleil, puis le satellite autour de la planète. Dans ce cas de figure, le soleil sera parent de la planète qui sera parente du satellite (Figure 3.2). Des liens de parenté sont établis entre les éléments pour définir la hiérarchie du mouvement. Lorsque c'est le cas dans l'animation de référence, il est nécessaire que nous conservions ces propriétés lorsque l'animation sera adaptée à des éléments statiques de la texture d'entrée.

Du fonctionnement de la composition des mouvements, nous tirons une exigence pour la spécification des groupes de mouvements. Lors de la recherche de similarités dans la texture statique, si un élément de référence possédant un parent est trouvé il faut que son parent soit également présent. Le non-respect de cette condition implique que les mouvements ne peuvent être analysés correctement lors de l'étape d'adaptation. Il est donc nécessaire que dans un groupe de mouvement formé par l'utilisateur, toute la hiérarchie du mouvement des éléments soit contenue. De plus, si une hiérarchie de mouvement est présente dans l'animation de référence, il faut que l'utilisateur en fasse un groupe de mouvements.

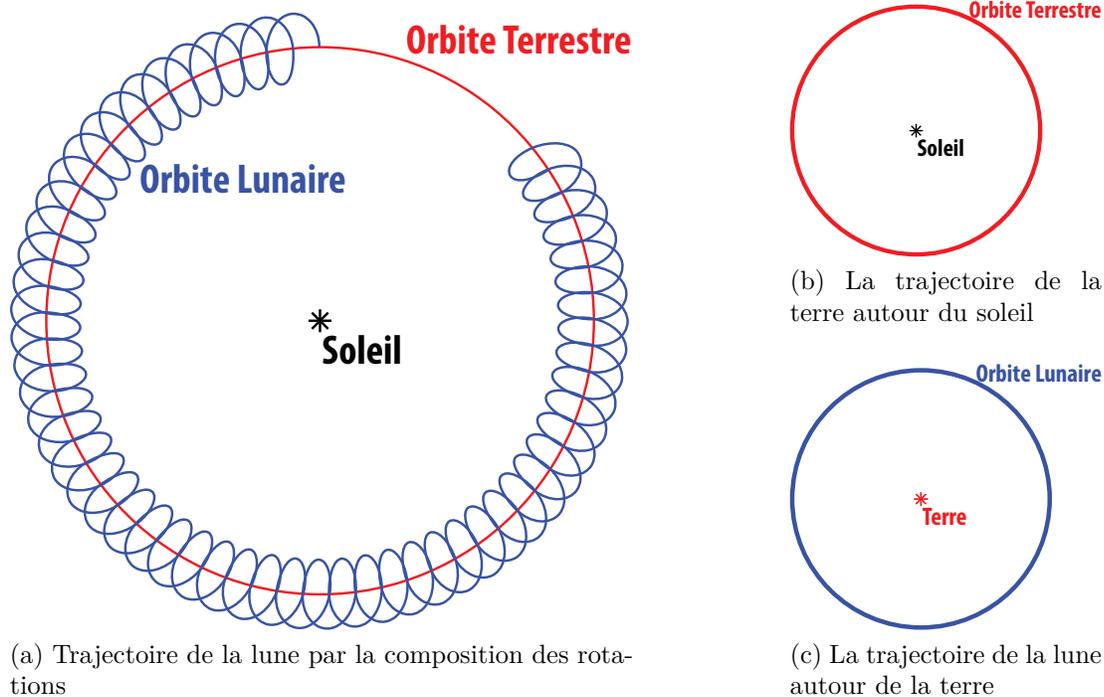


Figure 3.2 Composition de mouvements.

CHAPITRE 4 RECHERCHE DE SIMILARITÉS DANS LA TEXTURE

Dans ce chapitre, nous présentons la méthodologie proposée pour rechercher dans la texture statique des groupes similaires, selon leurs couleurs et disposition spatiale, à ce qui a été spécifié lors du chapitre précédent. La stationnarité inhérente aux textures implique que la texture statique d'entrée est composée d'une distribution des groupes spécifiés dans le motif par l'utilisateur. Chacun de ces groupes peut varier en matière d'organisation spatiale, mais la stationnarité assure que cette variabilité est faible. La méthode proposée a pour objectif de retrouver les occurrences des groupes dans la texture statique pour y appliquer, et adapter, les mouvements définis par l'utilisateur. Cette méthode doit donc être robuste à ces légères variations possibles d'organisation.

4.1 État de l'art

La recherche de groupes similaires d'éléments dans la texture présente des défis proches de ceux rencontrés lors de la synthèse de texture par l'exemple. Dans [1], le voisinage dans la texture d'entrée est décrit par une triangulation de Delaunay, et la synthèse de la texture est faite par dissémination de graines dont la croissance est contrôlée par le voisinage décrit plus tôt. Dans [6], les auteurs utilisent un procédé similaire en introduisant des priorités dans l'ordre de synthèse des voisinages de référence, et finit par remplir l'espace restant avec les éléments moins importants. Nous nous inspirons de cette méthode pour définir l'ordre d'affectation des mouvements dans la texture statique donnée en entrée par l'utilisateur. Les groupes de mouvements comprenant plusieurs éléments sont traités en premier, puis les éléments animés ne faisant pas partie d'un groupe de mouvement sont attribués aux éléments restant de la texture statique.

Les auteurs de [10] cherchent également à synthétiser à partir d'un exemple des textures dans un domaine spécifié par l'utilisateur. La texture d'entrée est composée d'éléments distincts qui sont eux-mêmes divisés en échantillons. À l'initialisation du processus, des patches de la texture d'entrée sont copiés dans le domaine de sortie. Une fois ces patches copiés, à chacun des échantillons sur le domaine de sortie, les auteurs font correspondre un échantillon de la texture d'entrée en comparant leurs voisinages. Contrairement aux approches présentées précédemment, la description du voisinage est réalisée en exprimant la disposition spatiale des éléments présents dans un certain rayon autour d'un élément de la texture de sortie. Les correspondances entre les éléments d'entrée et de sortie sont établies en minimisant une fonction d'énergie dépendant de la position et d'un autre nombre quelconque de paramètres

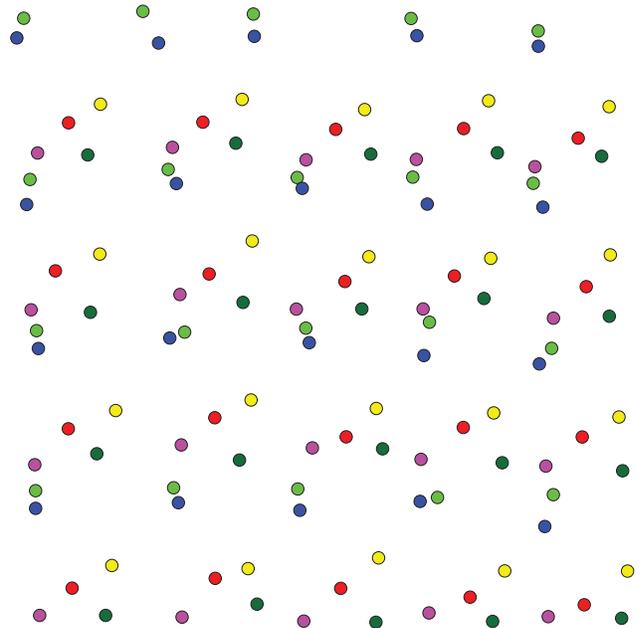
(couleur, forme, etc.). Nous proposons une méthode de description du voisinage de référence inspirée de ces derniers travaux couplée à une méthode gloutonne pour traiter la recherche des voisinages similaires aux groupes de référence dans la texture de sortie.

4.2 Méthodologie

Nous disposons en entrée de groupes de mouvement spécifiés par l'utilisateur comprenant un ensemble d'éléments décrits par leur disposition spatiale. Un élément est noté $e_{[0...N]}^{\{r,c\}}$, r désignant l'appartenance à un groupe de référence comprenant N éléments et une racine, c l'appartenance à un voisinage cible. Étant donné que la forme des éléments n'est pas un critère discriminant dans la méthode proposée, les éléments sont représentés par le centroïde de leur boîte englobante (Figure 4.1) dont la position est notée $\mathbf{p}(e)$ ainsi que par leur couleur moyenne $\mathbf{c}(e)$. Trouver dans la texture statique des dispositions similaires à celles des groupes d'entrée est donc équivalent à comparer des nuages de point. Nous parcourons tous les éléments de la texture statique afin de comparer pour chacun si son voisinage est similaire à l'un des groupes d'entrée spécifiés par l'utilisateur (Figure 4.2).



(a) Texture d'entrée



(b) Représentation centroïdale des éléments de la texture

Figure 4.1 Représentation centroïdale

La comparaison du voisinage d'un élément avec un groupe est réalisée en définissant pour chaque groupe un élément central qui servira de référentiel pour le calcul du voisinage. Cet

élément, appelé racine, est choisi de manière à minimiser les distances avec tous les autres éléments du groupe et ainsi permettre de réduire le nombre de candidats potentiels pour un voisinage lors de la recherche dans la texture statique. L'élément le plus proche du centre de la boîte englobante du groupe est sélectionné comme racine et est noté e_0^r , avec r désignant l'appartenance à un groupe de référence.

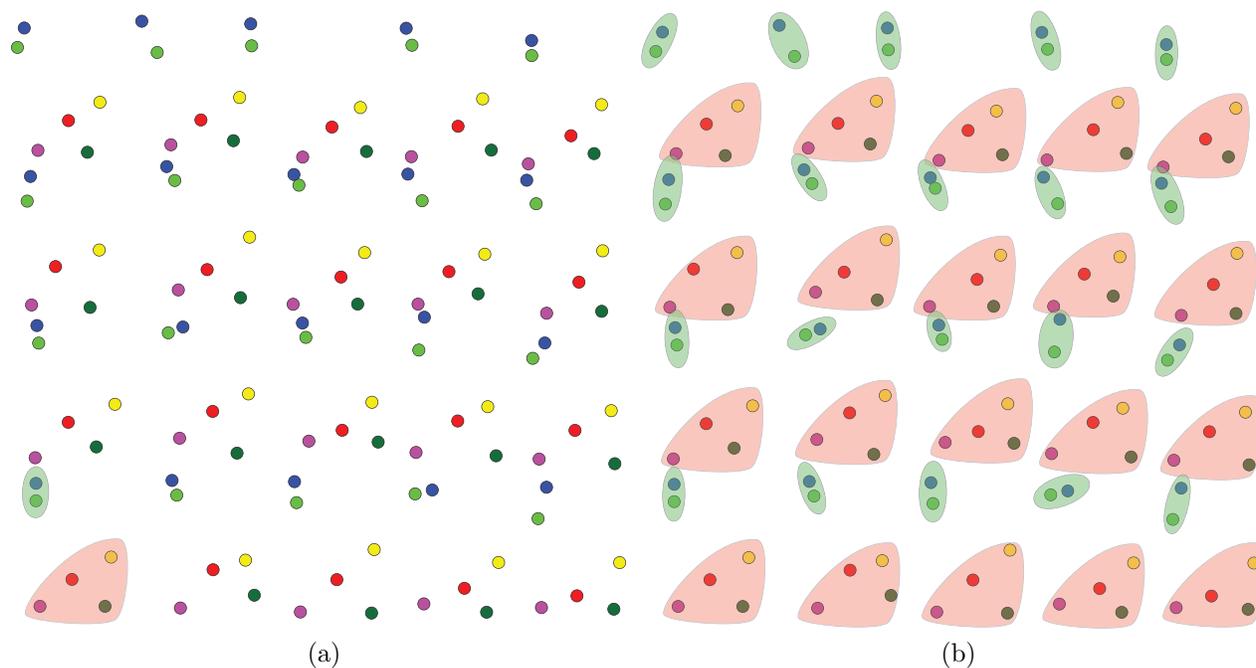


Figure 4.2 Résultat attendu de la recherche de similarités. Dans la texture (a), deux groupes de mouvement sont précisés par l'utilisateur. (b) présente le résultat que nous attendons en sortie de l'étape de recherche de similarité des groupes.

4.2.1 Nombre d'occurrences d'un groupe

Le nombre total d'éléments attribués à un groupe donné est déterminé en fonction de la proportion de l'exemple que ce groupe représente. Dans la Figure 4.2, 6 éléments font partie de l'exemple spécifié par l'utilisateur dont 2 appartiennent au groupe vert et 4 au groupe rouge. Le groupe vert représentant 33% de l'exemple, nous nous attendons à ce qu'environ 33% des éléments de la texture statique soient attribués à des groupes verts dans la texture animée.

4.2.2 Complexité d'un groupe

Lors du parcours des éléments de la texture statique, l'ordre dans lequel les groupes sont comparés est important. Dans la Figure 4.2, le groupe de deux points entouré en vert présente une disposition des éléments beaucoup plus simple que le groupe entouré en rouge. Il est préférable de traiter en premier lieu les groupes présentant les dispositions les plus discriminantes. Pour établir un ordre de comparaison des groupes, nous proposons une notion de complexité d'un groupe dépendant du nombre d'éléments qu'il comporte et de la dispersion des éléments autour de sa racine :

$$C = \frac{N}{N-1} \sum_{i=1}^N d(e_0^r, e_i^r) \quad (4.1)$$

où $d(.)$ est la distance euclidienne entre les centroïdes des deux éléments e_0^r et e_i^r . Le groupe présentant la mesure de complexité la plus élevée est comparé en premier au voisinage de tous les éléments de la texture statique, puis le second et ainsi de suite. Le voisinage de chacun des éléments e de la texture statique est comparé au voisinage de la racine des groupes de référence.

4.2.3 Éléments isolés

Les éléments n'appartenant pas à un groupe et pour lesquels un mouvement est spécifié sont considérés comme des groupes isolés dont la complexité est nulle. Il n'est pas possible pour ces éléments de déterminer à quels éléments de la texture statique les attribuer, puisqu'ils n'ont pas de voisinage. Une fois que tous les groupes ont été attribués, les éléments solitaires sont répartis aléatoirement dans les éléments restants de la texture statique. Le respect de la proportion d'éléments attribués à chaque groupe par rapport à la proportion d'éléments de ce groupe dans l'exemple permet d'assurer d'avoir un nombre suffisant d'éléments solitaires à attribuer aux éléments de référence isolés.

4.2.4 Détermination des combinaisons dans un voisinage

Afin de déterminer la similarité du voisinage d'un élément e_0^c à un groupe de référence comportant N éléments, nous établissons une liste des combinaisons de N éléments réalisables dans ce voisinage. Pour chaque élément e_i^c du groupe de référence, l'ensemble des éléments candidats dans le voisinage de e_0^c est noté $n_i = \{e_i^c\}$ et comporte tous les éléments satisfaisants

l'équation suivante :

$$(1 - \sigma_n)d(e_0^r, e_i^r) \leq d(e_0^c, e) \leq (1 + \sigma_n)d(e_0^r, e_i^r). \quad (4.2)$$

Le facteur σ_n permet à l'utilisateur de définir la variabilité de la texture d'entrée. Une valeur élevée de ce paramètre permet à l'utilisateur de préciser que la disposition spatiale des éléments des motifs varie fortement par rapport à la disposition des groupes de référence (Figure 4.3).

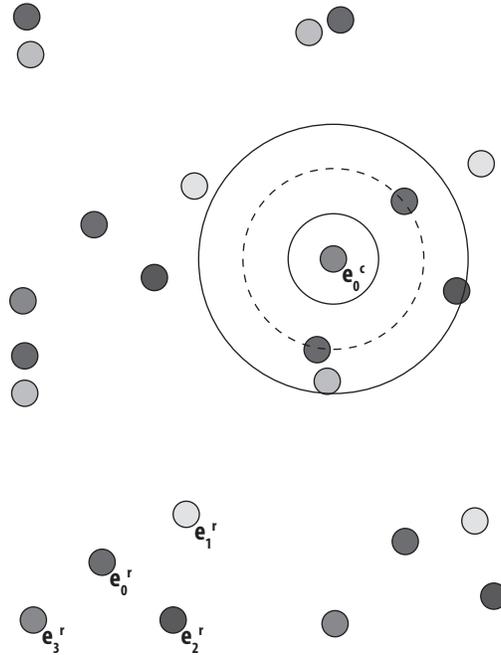


Figure 4.3 Voisinage d'un élément. Avec e_0^c comme racine, tous les éléments entre les deux cercles sont considérés comme candidats au voisinage n_1 avec $\sigma_n = 0.5$.

Pour chaque combinaison de N éléments réalisable à partir des ensembles n_i , nous établissons une mesure de similarité avec le groupe de référence en calculant trois dissimilarités notées D_{o1} , D_{o2} , D_c .

4.2.5 Mesures de similarité d'un voisinage à un groupe

Dissimilarité de premier ordre. Ce que nous appelons la dissimilarité de premier ordre représente la différence de distance moyenne des éléments cibles à leur racine et des éléments de référence à leur racine. Cette dissimilarité, rapide à calculer, permet de donner une première idée de la similarité des deux voisinages (Figure 4.4).

$$D_{o1} = \frac{1}{N} \sum_{i=1}^N \left(d(e_0^r, e_i^r)^2 - d(e_0^c, e_i^c)^2 \right). \quad (4.3)$$

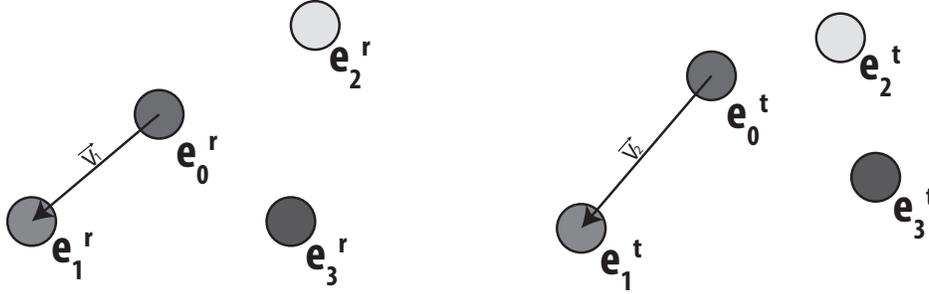


Figure 4.4 Dissimilarité de premier ordre. La différence de distance à la racine est donnée par la différence entre les distances $d(e_0^c, e_i^c)$ et $d(e_0^r, e_i^r)$ dans l'image.

Dissimilarité de second ordre. Afin de déterminer la similarité de la disposition spatiale des éléments les uns par rapport aux autres, nous proposons de calculer une dissimilarité de second ordre. Cette dissimilarité sera déterminée en alignant successivement les éléments du voisinage de référence sur le premier élément du voisinage cible. Pour ce faire, nous exprimons les coordonnées de tous les éléments de la combinaison courante et du groupe de référence par rapport à leur racine. Par la suite, pour chaque élément e_j^r du groupe de référence, nous calculons la matrice de rotation R_j qui sera appliquée à toute la combinaison courante. La dissimilarité calculée est la distance moyenne entre chacun des e_i^r et e_i^c correspondants. Nous tiendrons compte de l'alignement présentant la dissimilarité la plus faible (Figure 4.5) :

$$D_{o2} = \min_j \left(\frac{1}{N} \sum_{i=1}^N d(e_i^r, R_j e_i^c)^2 \right) \quad (4.4)$$

avec R_j la matrice de rotation permettant d'aligner le point e_1^c avec le point e_1^r .

Couleur. La dissimilarité de couleur représente la différence de couleur entre les éléments de la combinaison courante et du groupe de référence. La Commission International de l'Éclairage (CIE) propose un espace colorimétrique dit psychométrique adapté à la mesure de similarité perceptuelle entre deux couleurs. L'espace Lab composé de trois composantes : L représente la luminance, tandis qu'a et b sont les axes de chrominance. la dissimilarité de

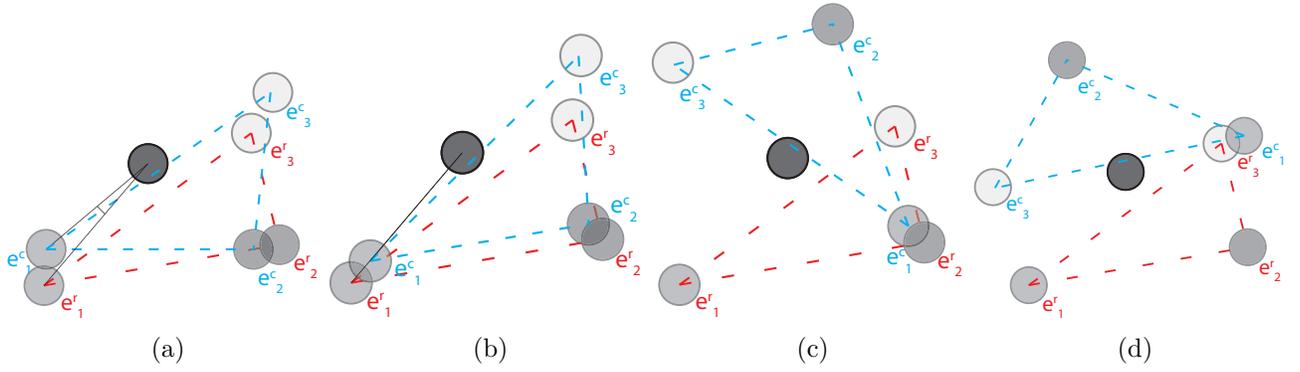


Figure 4.5 La racine du groupe de référence est alignée sur e_0^c (a). e_1^c est aligné sur chacun des points du groupe de référence (b) (c) (d)

couleur est la moyenne des différences de distance euclidienne entre les $\mathbf{c}(e_i^r)$ et $\mathbf{c}(e_i^c)$.

$$D_c = \frac{1}{N} \sum_{i=0}^N \|\mathbf{c}(e_i^r) - \mathbf{c}(e_i^c)\|^2. \quad (4.5)$$

4.2.6 Fonction de coût

L'expression de la similarité d'un voisinage dépend des trois dissimilarités calculées précédemment. Aux trois dissimilarités est appliqué un lissage gaussien afin de créer une fonction de coût associée à chaque combinaison. La résolution du problème posé passe donc par la minimisation de la fonction suivante :

$$\mathcal{D} = e^{-\frac{D_{o1}}{\sigma_1}} + e^{-\frac{D_{o2}}{\sigma_2}} + e^{-\frac{D_c}{\sigma_c}}. \quad (4.6)$$

Les paramètres σ_1 et σ_2 dépendent de la disposition spatiale du groupe et de la variabilité de la texture statique. Ils sont donc fixés comme suit :

$$\sigma_1 = \sigma_2 = \sigma_n \left[\frac{1}{N} \sum_{i=1}^N d(e_0^r, e_i^r) \right]. \quad (4.7)$$

σ_c a été fixé de manière empirique à 30 mais peut être contrôlé par l'utilisateur afin de donner plus ou moins d'importance à la différence de couleur entre les éléments.

4.2.7 Seuils d'abandon d'une combinaison

Afin d'accélérer les calculs, nous proposons d'instaurer des seuils à partir desquels une combinaison est considérée comme présentant une dissimilarité trop élevée pour être une candidate potentielle à l'accueil du groupe de mouvement courant. Les seuils utilisés sont égaux à $\sigma_1, \sigma_c, \sigma_2$ et appliqués dans cet ordre. Ainsi, lorsque la dissimilarité de distance moyenne entre les éléments de la combinaison et la racine e_0^c dépasse σ_1 , l'étape d'alignement des combinaisons au groupe de mouvement de référence n'aura pas lieu accélérant considérablement le calcul. Cette méthode présente le désavantage de risquer une assignation incomplète des groupes de mouvements aux éléments dans la texture statique. Pour assurer une diffusion maximale du mouvement dans la texture, une seconde itération est réalisée sans tenir compte des seuils si il reste des points de la texture statique qui n'ont pas été assignés à un groupe.

4.3 Expérimentations

Les expérimentations ont été réalisées sur quatre textures possédant des dispositions différentes afin de tester la robustesse de la méthode proposée. Chacune des textures présente plusieurs versions dans lesquelles les éléments ont été déplacés aléatoirement dans l'espace sur des distances dépendant de la taille minimale de la boîte englobante d'une instance du motif de base de la texture (Figure 4.6) Les textures d'exemple ont été réalisées avec Illustrator, et les perturbations dans la disposition des éléments appliquées avec les outils de transformation de position aléatoire de chacun des éléments proposés par le logiciel. Pour ce qui est de la spécification de l'animation et de l'implémentation, les tests sont réalisés avec After Effects 2014 afin de tirer parti de la large palette d'outils proposés pour la spécification des mouvements par images clés, par le dessin ou par des méthodes procédurales. Une interface basique a été réalisée permettant de grouper les calques et définir les paramètres de la recherche de similarités dans la texture statique.

```

1: Calculer  $\sigma_1, \sigma_2$  avec l'Équation 4.7
2:  $iteration := 0$ 
3: tant que  $iteration \neq MAX\_ITERATIONS$  faire
4:   pour chaque groupe de mouvement faire
5:     pour chaque élément  $e$  faire
6:        $e_0^c = e$ 
7:       pour chaque  $e_i^r$  dans le groupe de mouvement courant faire
8:         pour chaque élément  $e \neq e_0^c$  faire
9:           si  $e$  vérifie l'Équation 4.2 for  $e_i^r$  OU  $iteration = MAX\_ITERATIONS - 1$ 
           alors
10:            Ajouter  $e$  à l'ensemble  $n_i$ 
11:          fin si
12:        fin pour
13:      fin pour
14:      Déterminer toutes les combinaisons de  $N$  éléments réalisables avec les ensembles
       $n_i$ 
15:      pour chaque combinaison faire
16:        Calculer  $D_{o1}$  associée avec l'Équation 4.3
17:        si  $D_{o1} > \sigma_1$  ET  $iteration \neq MAX\_ITERATIONS - 1$  alors
18:          Abandonner la combinaison
19:        fin si
20:        Calculer  $D_c$  associée avec l'Équation 4.5
21:        si  $D_c > \sigma_c$  ET  $iteration \neq MAX\_ITERATIONS - 1$  alors
22:          Abandonner la combinaison
23:        fin si
24:        Calculer  $D_{o2}$  associée avec l'Équation 4.4
25:        si  $D_{o2} > \sigma_2$  ET  $iteration \neq MAX\_ITERATIONS - 1$  alors
26:          Abandonner la combinaison
27:        fin si
28:        Calculer error  $\mathcal{D}$  avec l'Équation 4.6
29:      fin pour
30:      Ordonner combinaisons par ordre croissant de dissimilarité  $D$ 
31:      tant que Il reste des éléments à assigner ET des combinaisons valides faire
32:        Assigner le groupe à la combinaison avec le plus petit  $D$ 
33:        Retirer les combinaisons dont les éléments ne sont plus tous disponibles
34:      fin tant que
35:    fin pour
36:  fin pour
37:   $iteration = iteration + 1$ 
38: fin tant que

```

Les résultats présentés donnent le pourcentage de précision de la recherche de similarité dans la texture par rapport au résultat auquel nous nous attendons. Le résultat attendu est celui

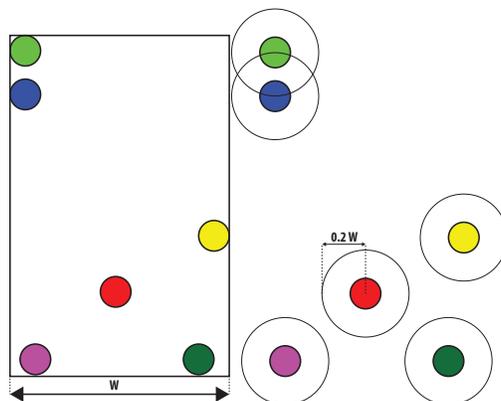


Figure 4.6 Perturbation d'une texture. Pour une perturbation de 20%, les cercles représentent la zone dans laquelle se trouvera chacun des éléments après la perturbation

Exemple 1						
Perturbation	10%		20%		30%	
Couleur	Oui	Non	Oui	Non	Oui	Non
$\sigma_n = 0.1$	100%	80.7%	100%	84.2%	91.2%	78.9%
$\sigma_n = 0.2$	100%	96.5%	100%	84.2%	86.8%	71.9%
$\sigma_n = 0.3$	100%	100%	100%	93.0%	91.2%	81.6%
$\sigma_n = 0.4$	100%	100%	100%	96.5%	100%	89.5%
$\sigma_n = 0.5$	100%	100%	100%	100%	100%	96.5%
$\sigma_n = 0.6$	100%	100%	100%	100%	100%	96.5%
$\sigma_n = 0.7$	100%	100%	100%	100%	100%	96.5%

Tableau 4.1 Variation de σ_n pour Exemple 1

Exemple 2						
Perturbation	10%		20%		30%	
Couleur	Oui	Non	Oui	Non	Oui	Non
$\sigma_n = 0.1$	77.7%	88.7%	61.0%	38.0%	42.3%	12.3%
$\sigma_n = 0.2$	77.7%	88.7%	61.0%	38.0%	42.3%	12.3%
$\sigma_n = 0.3$	92.7%	96.0%	83.7%	43.3%	42.3%	12.3%
$\sigma_n = 0.4$	92.7%	96.0%	94.7%	45.7%	59.3%	14.3%
$\sigma_n = 0.5$	92.7%	96.0%	95.3%	45.7%	71.7%	17.1%
$\sigma_n = 0.6$	92.7%	96.0%	95.3%	45.7%	83.7%	20.4%
$\sigma_n = 0.7$	92.7%	96.0%	95.3%	46.3%	89.7%	20.4%

Tableau 4.2 Variation des σ_n pour Exemple 2

calculé avec 0% de perturbation dans la texture que nous appelons résultat de référence. Une fois la recherche de similarité dans la texture perturbée terminée, nous comparons les

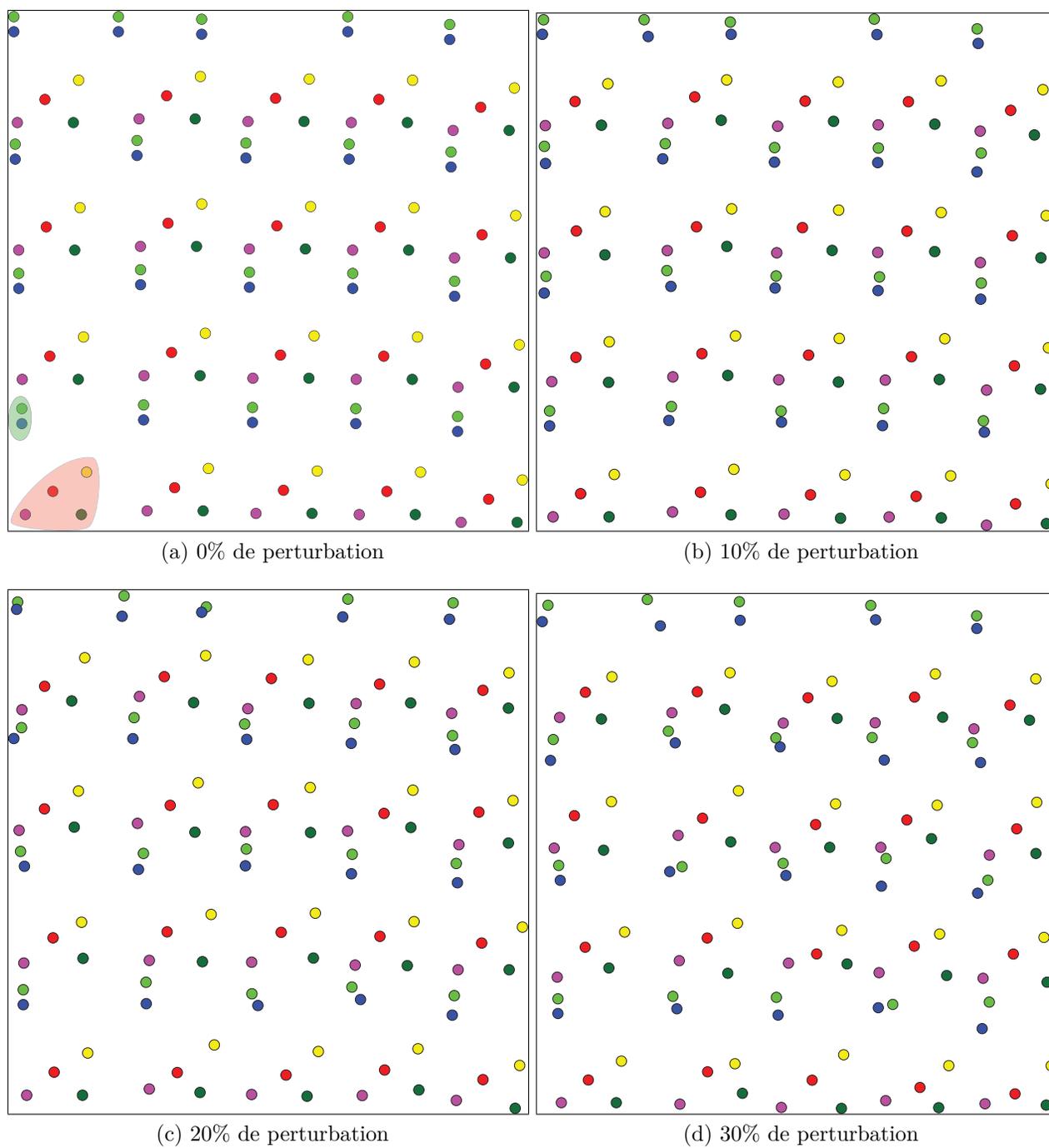


Figure 4.7 Texture Exemple 1

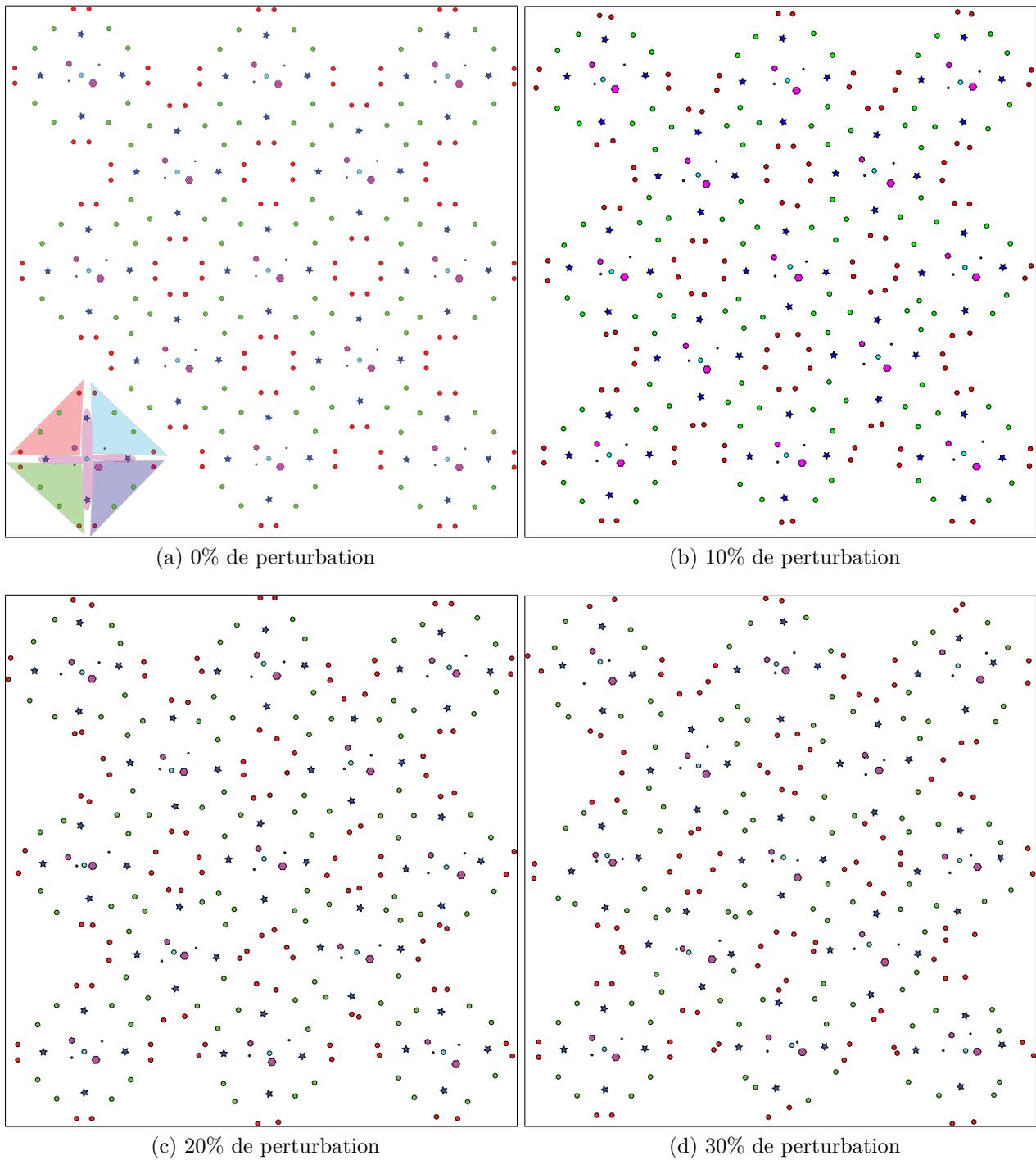


Figure 4.8 Texture Exemple 2

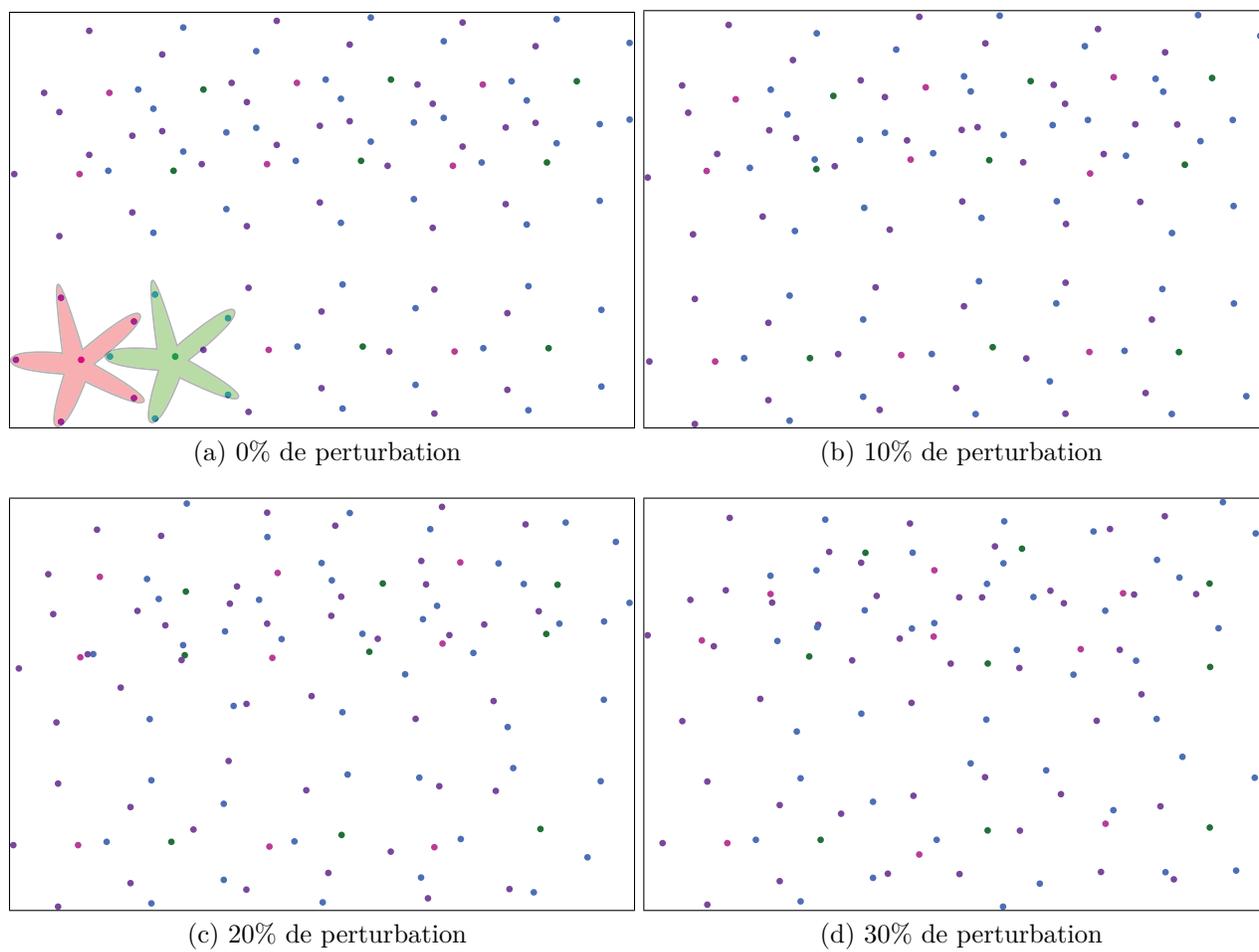


Figure 4.9 Texture Exemple 3

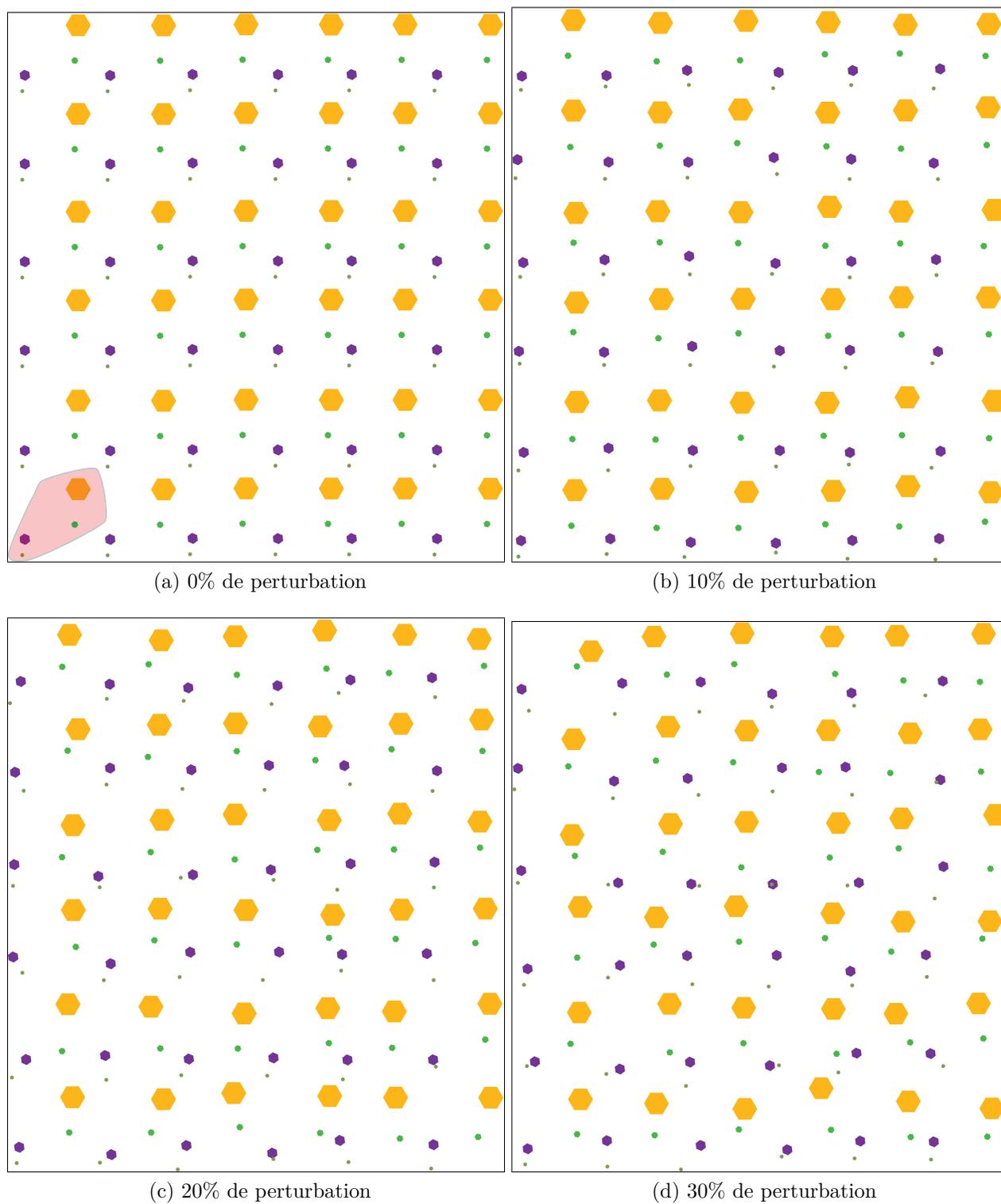


Figure 4.10 Texture Exemple 4

Exemple 3						
Perturbation	10%		20%		30%	
Couleur	Oui	Non	Oui	Non	Oui	Non
$\sigma_n = 0.1$	77.0%	33.3%	60%	30.2%	38.5%	30.2%
$\sigma_n = 0.2$	80.0%	43.8%	60.3%	32.1%	39.6%	31.3%
$\sigma_n = 0.3$	100%	43.8%	70.5%	18.8%	47.9%	18.8%
$\sigma_n = 0.4$	100%	43.8%	75.2%	27.3%	58.4%	15.6%
$\sigma_n = 0.5$	100%	43.7%	75.2%	49.5%	75.3%	26.0%
$\sigma_n = 0.6$	100%	43.8%	85.0%	50.8%	80.0%	30.1%
$\sigma_n = 0.7$	100%	43.8%	100%	55.1%	87.0%	31.5%

Tableau 4.3 Variation des σ_n pour Exemple 3

Exemple 4						
Perturbation	10%		20%		30%	
Couleur	Oui	Non	Oui	Non	Oui	Non
$\sigma_n = 0.1$	100%	100%	100%	100%	85.0%	68.5%
$\sigma_n = 0.2$	100%	100%	100%	100%	88.6%	68.5%
$\sigma_n = 0.3$	100%	100%	100%	100%	90.0%	78.9%
$\sigma_n = 0.4$	100%	100%	100%	100%	100%	89.5%
$\sigma_n = 0.5$	100%	100%	100%	100%	100%	94.3%
$\sigma_n = 0.6$	100%	100%	100%	100%	100%	94.3%
$\sigma_n = 0.7$	100%	100%	100%	100%	100%	94.3%

Tableau 4.4 Variation des σ_n pour Exemple 4

résultats obtenus dans la texture perturbée avec le résultat de référence :

$$score = \frac{\{\text{résultat de référence}\} \cap \{\text{résultat calculé}\}}{\text{résultat de référence}} \quad (4.8)$$

Le paramètre σ_n est prépondérant pour la qualité de l'affectation. Plus les motifs de la texture statique présentent de variations, plus σ_n devra être élevé pour y être robuste. Dans la texture Exemple1 (Figure 4.7) les éléments sont peu nombreux et leur densité relativement faible, la méthode donne un résultat parfait même en cas de fortes variations. On peut voir que même pour des textures dont les éléments présentent une plus grande densité comme pour l'Exemple 2 (Figure 4.8), la méthode atteint 100% d'éléments assignés au groupe attendu lorsque σ_n est choisi en proportion de la perturbation de la texture. La méthode proposée présente pour la texture Exemple 3 (Figure 4.9) une très faible efficacité. Les groupes de mouvement rouge, bleu, cyan et vert présentent des dispositions extrêmement similaires et des éléments de la même couleur (Figure 4.10a). Ici, l'invariance à la rotation rend la spécification en groupes

de mouvements moins intuitive. En considérant les 5 groupes de mouvement comme un seul, il est possible d'éviter ce problème.

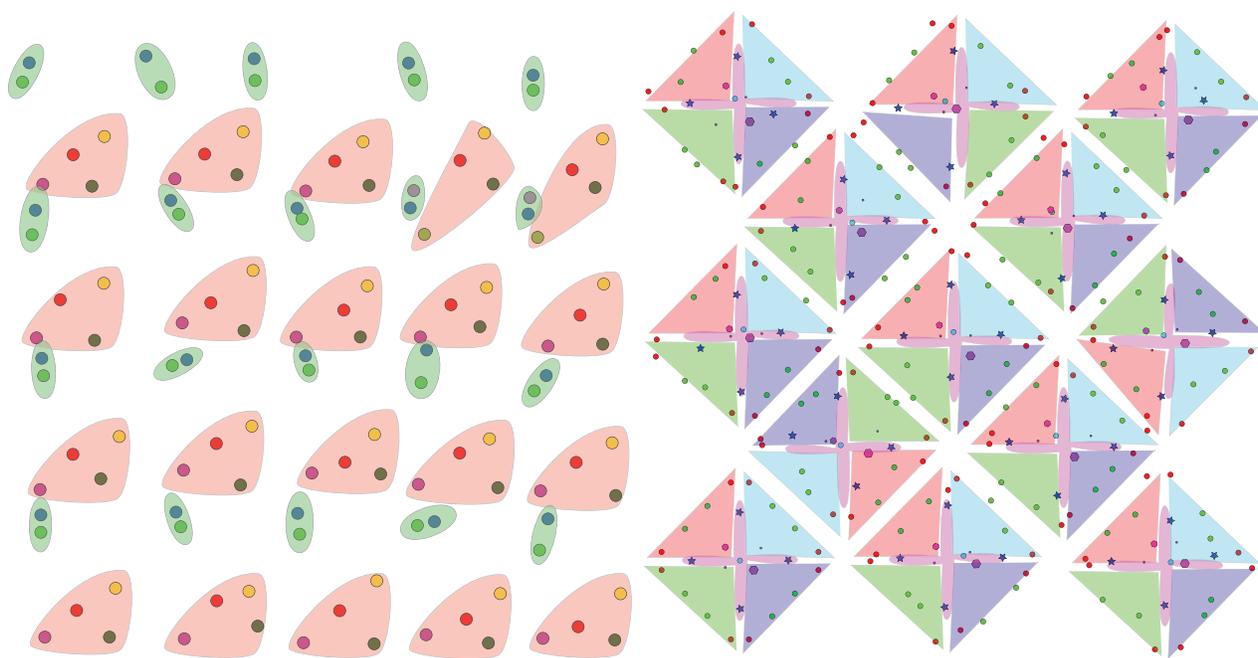
4.4 Discussion et limitations

Nous proposons dans ce chapitre une méthode permettant de retrouver dans la texture statique des ensembles d'éléments présentant une disposition similaire aux groupes de référence spécifiés par l'utilisateur en étant robustes à la faible variabilité des voisinages. Les attributions des groupes dans la texture sont dans la majorité des cas conformes aux attentes, néanmoins la principale faiblesse de la méthode proposée repose dans le nombre de calculs à réaliser lors de l'estimation de la dissimilarité de second ordre. La recherche gloutonne fait que le nombre de combinaisons à calculer croît rapidement avec la densité de la texture statique et la taille des groupes de référence. Ainsi, dans le voisinage d'un e_0^c nous aurons un nombre \mathcal{N} de combinaisons à évaluer :

$$\mathcal{N} = \prod_{i=1}^N \#n_i \quad (4.9)$$

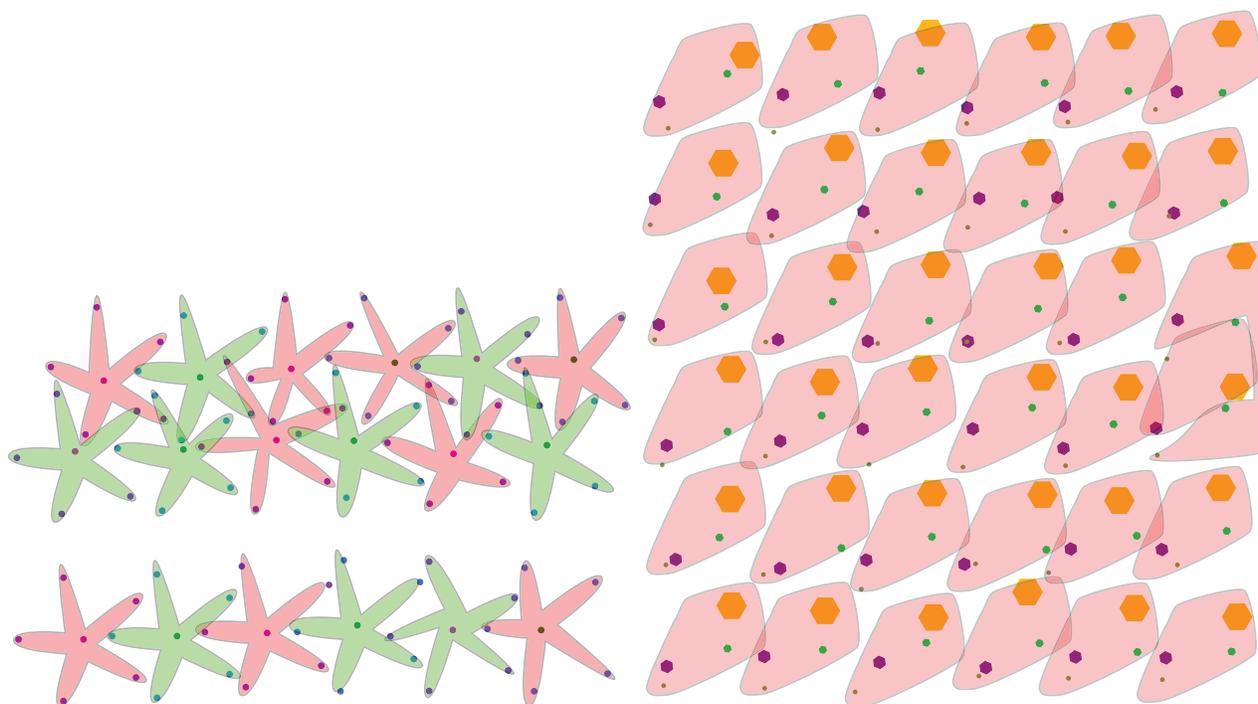
Néanmoins, les résultats obtenus nous permettent de valider la méthode sans la complexifier davantage. Le fait de considérer tous les éléments de référence comme faisant partie d'un super-groupe de mouvement dont les groupes de mouvement seraient des sous-parties permettrait après la recherche du premier groupe dans la texture d'inférer la position des autres groupes du motif, et d'attribuer les éléments solitaires non plus aléatoirement, mais par rapport aux groupes de mouvements attribués dans la texture. Nous pouvons remarquer que la qualité des résultats de la recherche de voisinages dans la texture est fortement dépendante de l'utilisateur. Le paramètre σ_n a un impact important sur la justesse de la recherche de groupes similaires dans la texture statique. La manière dont les groupes sont spécifiés est également importante, lorsque les groupes spécifiés ne sont pas assez différents les uns des autres (Figure 4.11) la méthode peine à les discriminer si ce n'est par la couleur. Il serait possible d'améliorer la performance de la méthode en proposant à l'utilisateur de déterminer par lui-même les paramètres σ_1 et σ_2 , l'impact de ces paramètres n'est néanmoins pas majeur et la spécification à main rend la méthode moins intuitive à utiliser. Le fait de proposer l'invariance à la rotation de la méthode la rend plus précise dans beaucoup de cas, mais peut avoir un effet négatif si les groupes de mouvement différents présentent des dispositions similaires (Tableau 4.2). Dans l'Exemple 2, la méthode peine à arriver à 100% de groupes retrouvés si la couleur n'est pas active. L'Exemple 3 (Figure 4.9) présente la particularité d'être composé de groupes de mouvement ne différant que par la couleur de ses éléments. Il

est donc tout à fait naturel que la méthode soit fortement limitée en l'absence de la prise en compte de la couleur dans la recherche de similarités.



(a) Texture d'exemple 1, résultat avec $\sigma_n = 0.2$ sans prise en compte de la dissimilarité de couleur

(b) Texture d'exemple 2, résultat avec $\sigma_n = 0.2$ sans prise en compte de la dissimilarité de couleur



(c) 20% de perturbation

(d) 30% de perturbation

Figure 4.11 Exemples de mauvais résultats de la recherche de groupes similaires

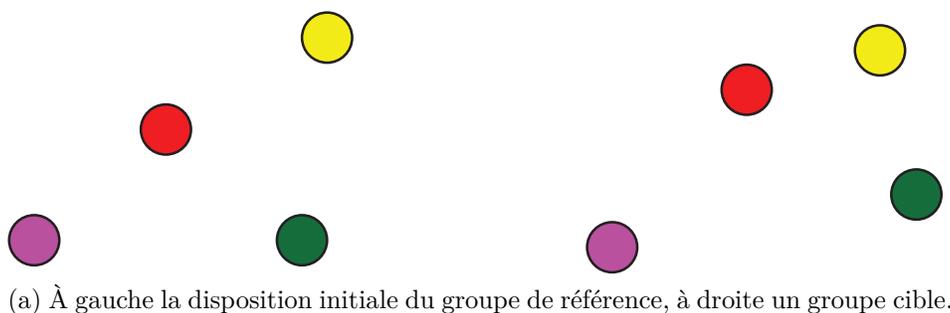
CHAPITRE 5 ADAPTATION DU MOUVEMENT

Ce chapitre présente la méthode visant à animer la texture statique d'entrée en affectant les mouvements depuis l'exemple vers les combinaisons trouvées lors de la recherche de similarité. Réaliser un simple copier/coller du mouvement de référence occasionne des pertes de cohérence dans l'animation dues à la variabilité des motifs de la texture (Figure 5.1). Nous proposons de se baser sur la hiérarchie établie par l'utilisateur lors de l'étape de spécification afin de réaliser une adaptation du mouvement à la différence de disposition spatiale entre les groupes de références et les groupes cibles. Le processus est réalisé en deux étapes. En premier lieu, les structures hiérarchiques et spatiales sont analysées afin de déterminer dans quelle proportion le mouvement doit être adapté. Le mouvement sera ensuite déformé afin de respecter les contraintes suivantes :

- La position initiale d'un élément dans la texture ne devra pas être modifiée.
- Si la première et la dernière position de l'animation sont les mêmes, l'animation est une boucle et cette continuité doit être conservée.
- Le point minimum de la trajectoire d'un élément par rapport à son parent devra être conservé.
- Les déformations appliquées à la trajectoire sont rigides et homogènes.

5.1 Hiérarchie des mouvements.

Lorsque des éléments d'un groupe possèdent un parent, cela signifie que l'utilisateur a précisé que les mouvements dépendent les uns des autres. Lors de l'adaptation du mouvement, nous devons reconstruire la structure hiérarchique des éléments. Le fait de préciser cette structure de parenté implique que l'utilisateur indique explicitement que cet ensemble de points possède des structures spatiales et hiérarchiques importantes pour le respect du mouvement. Lorsqu'un élément possède un parent, les positions successives définissant son mouvement sont exprimées dans le repère local de celui-ci. La différence de position d'un élément e_i^c par rapport à son élément de référence e_i^r dans le repère de leurs parents, à laquelle nous référerons par perturbation, devra être répercutée sur toutes les images lors de l'application du mouvement de référence.



(b) À un autre instant de l'animation, les éléments du groupe de référence sont tous situés à la même position, tandis que les éléments du groupe cible subissent un décalage dû à la variabilité des motifs.

Figure 5.1 Pertes de cohérence de l'animation.

5.2 Contraintes et perception.

La perception d'un mouvement n'est pas chose aisée à formaliser. Notre but est d'adapter les mouvements de référence à des éléments disposés différemment. Les caractéristiques explicites que nous avons en entrée sont peu nombreuses en comparaison de la richesse des détails pouvant caractériser le mouvement d'un ensemble de points. Ce que nous considérons comme sens artistique donné à un mouvement est une combinaison entre l'enchaînement des positions et les inférences et interprétations faites par notre système visuel. Cette partie perceptuelle n'est pas caractérisable de manière mathématique, nous proposons une série de contraintes extraites de l'observation de résultats préliminaires à respecter afin de conserver une partie des éléments visuels forts d'un mouvement.

Conservation de la position initiale des éléments. Nous considérons que lorsque la texture est reçue en entrée, tous ses éléments sont exprimés au même instant. C'est-à-dire que l'utilisateur spécifie explicitement qu'à un moment précis de son animation, la disposition de tous les éléments sera celle précisée en entrée. Lors de la modification du mouvement, nous conserverons cette position initiale notée $\mathbf{p}_0(e_i)$ pour un élément e_i . Ainsi, la texture

présentera uniquement une variabilité spatiale et non temporelle.

Conservation des boucles dans l’animation. Les textures dynamiques visent souvent à produire des animations cycliques comme nous l’avons vu dans le Chapitre 2 avec les textures vidéos. L’aspect cyclique d’une animation est assuré par deux conditions : le fait que la première et la dernière image soient les mêmes, et la consistance dynamique du mouvement. C’est-à-dire que la direction globale du mouvement est la même (Figure 5.2). Dans notre cas, le mouvement synthétisé l’est à partir d’un seul exemple de l’utilisateur et comprend la première et la dernière image. Nous n’avons donc pas à nous occuper du respect de la dynamique du mouvement, elle est explicitement présente. En revanche, le fait d’avoir des positions identiques à la première et la dernière image est important.

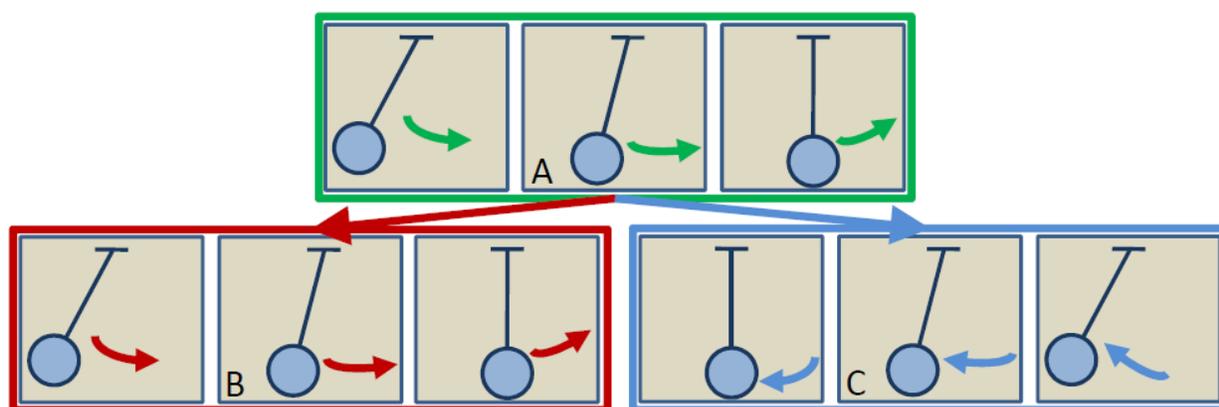


Figure 5.2 Dynamique du mouvement. Une transition de A vers B conserve la dynamique du mouvement, une transition de A vers C ne la conserve pas. Source : [17]

Conservation des contacts entre un point et son parent. Lors d’une animation, les contacts entre les éléments représentent une caractéristique visuelle forte. Lorsque l’on regarde une texture complète animée, la présence ou non d’un contact entre deux points peut nuire à la régularité perceptuelle de la texture. De la même manière que l’utilisateur a, à l’aide des groupes de mouvement, explicitement spécifié que la distribution spatiale d’un ensemble d’éléments est importante, en établissant une hiérarchie entre les éléments il spécifie que les mouvements représentent un ensemble avec une forte cohérence. Lorsqu’un élément e_j est défini comme enfant d’un élément e_i , ses transformations seront spécifiées dans le repère local de e_i et il sera soumis à toutes les transformations que subiront ce repère local. Le fait que l’élément e_j entre en contact avec son parent (ou soit proche d’un contact) sera considéré

comme une caractéristique visuelle forte et nous chercherons à la conserver. La position ayant la plus petite distance par rapport au parent sera notée $\mathbf{p}_{min}(e_j)$ et servira de point fixe pour calculer une interpolation du mouvement.

Déformation du mouvement. La vitesse du mouvement dépend de la répartition des images sur la trajectoire, on considère des transformations homogènes afin de conserver au maximum le pas. Nous disposons donc de deux points pour guider la déformation du mouvement de référence d'un élément e_i : la position minimum $\mathbf{p}_{min}(e_i^c) = \mathbf{p}_{min}(e_i^r)$ ainsi que la position initiale $\mathbf{p}_0(e_i^c)$. Nous souhaitons réaliser une transformation homogène de la trajectoire, c'est-à-dire composée d'une translation, rotation, et d'un facteur d'échelle qui est le même sur les composantes horizontales et verticales.

5.3 Méthode d'adaptation du mouvement

La perturbation est séparée en deux composantes : un angle et la distance à la racine. Lors de la synthèse du mouvement, nous considérerons que toutes les positions ont subi une rotation de l'angle de perturbation θ_i . La trajectoire entière subit donc une rotation autour du parent afin d'aligner e_0^r et e_0^c . Afin de respecter la seconde contrainte, nous ne pouvons pas réaliser une simple translation du mouvement. Dans le cas d'une translation, les contacts entre un point et son parent ne pourront pas être respectés. Nous choisissons donc de considérer qu'une perturbation de distance sera représentée comme étant un changement d'échelle du mouvement par rapport au parent. Nous pouvons représenter ce processus de transformation de la trajectoire à l'aide d'une série de matrices de transformation :

$$p(e_i^c) = T^{-1}SRTp(e_i^r). \quad (5.1)$$

La matrice T représente une translation ayant pour valeur les coordonnées de $\mathbf{p}_{min}(e_i^r)$. Ce point étant commun à la trajectoire de référence et à la trajectoire adaptée, il sera utilisé comme centre pour la rotation et la mise à l'échelle qui suivra. La matrice R est une matrice de rotation de l'angle $\angle(\mathbf{p}_0(e_i^r), \mathbf{p}_{min}(e_i^r), \mathbf{p}_0(e_i^c))$. La matrice S contient la mise à l'échelle requise pour faire correspondre les positions $\mathbf{p}_0(e_i^r)$ et $\mathbf{p}_0(e_i^c)$. Le processus entier est illustré dans la Figure 5.3.

La méthode proposée pour la transformation permet dans le cas général de respecter toutes les contraintes que nous avons énoncées précédemment. L'amplitude des déformations appliquées au mouvement dépend entièrement des perturbations présentes dans la texture, nous respectons donc la régularité de la texture. Dans certains cas particuliers, nous sommes

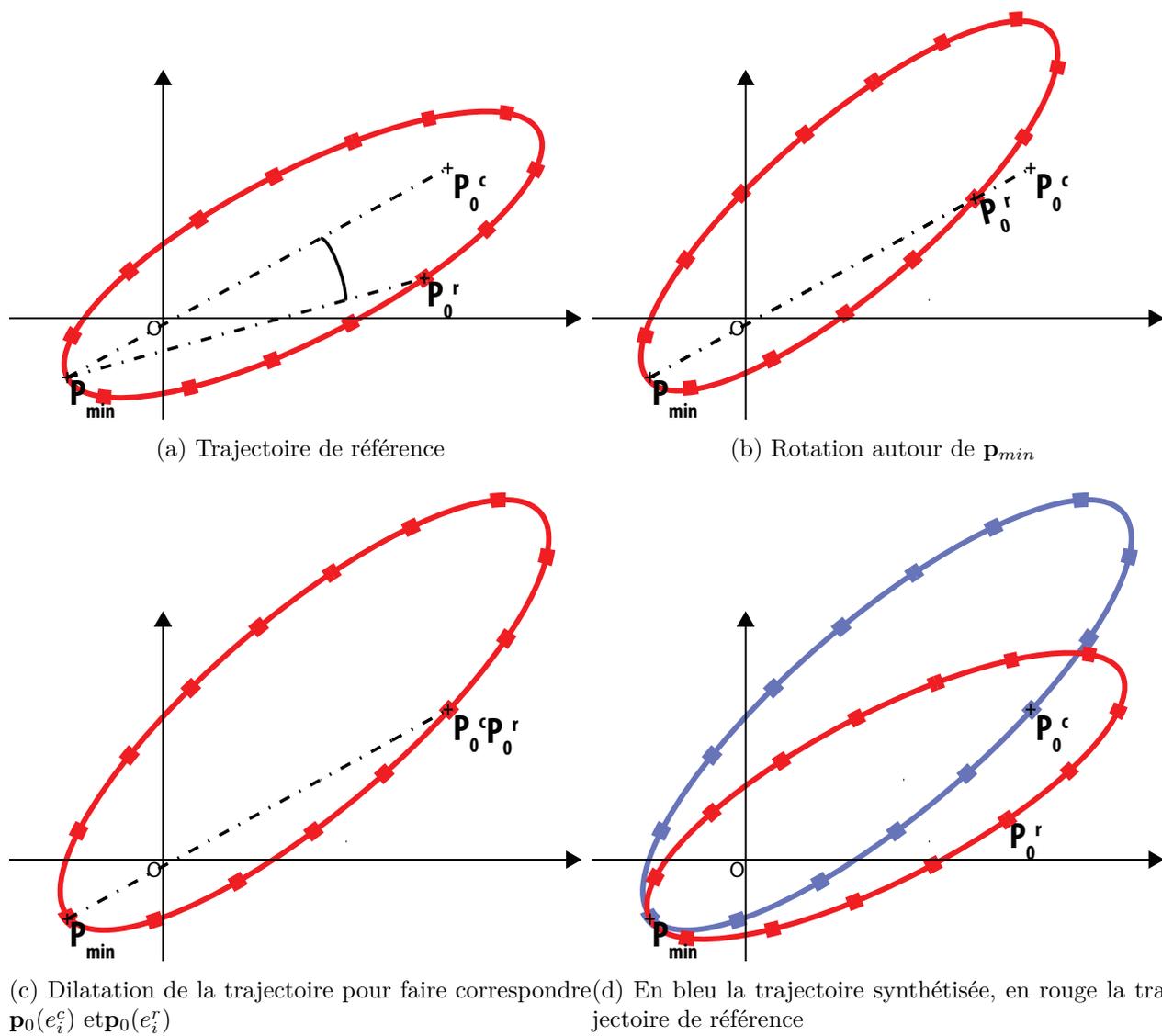


Figure 5.3 Processus d'adaptation de la trajectoire

confrontés à des situations de sur-contrainte ou sous-contrainte du système nous demandant d'adapter les transformations.

5.3.1 Cas particuliers

$$\mathbf{p}_0(e_i^r) = \mathbf{p}_{min}$$

Lorsque la position initiale de l'élément de référence correspond au point minimal de la trajectoire, les contraintes ne sont plus cohérentes avec les éléments dont nous disposons. La trajectoire adaptée devrait passer par \mathbf{p}_{min} et $\mathbf{p}_0(e_i^c)$ au même moment. Nous considérons donc le centre du repère (le parent, donc) comme centre de rotation. Cette manipulation nous permet d'avoir un impact minime sur la forme globale de la trajectoire et de conserver la position initiale des éléments (Figure 5.4). Du point de vue de l'Équation 5.1 cela revient à retirer la matrice T , l'équation devient alors :

$$p(e_i^c) = SRp(e_i^r). \quad (5.2)$$

$$\mathbf{p}_0(e_i^c) = \mathbf{p}_{min}$$

Lorsque la position initiale de l'élément cible correspond au point minimum de la trajectoire de référence, le même type de contradiction entre les contraintes que dans le cas précédent apparaît. Nous souhaitons que l'élément possède la même position à deux temps différents. Imposer une immobilité de l'élément pour le nombre d'images présentes entre \mathbf{p}_0 et \mathbf{p}_{min} permettrait de respecter les contraintes de respect du point \mathbf{p}_{min} et $\mathbf{p}_0(e_i^c)$. Cette méthode implique de ne plus avoir un mouvement continu en créant un saut entre les deux positions. Nous procédons donc de la même manière que lorsque $\mathbf{p}_0(e_i^r) = \mathbf{p}_{min}$. Nous procédons à une rotation autour du parent, puis une mise à l'échelle afin de faire correspondre $\mathbf{p}_0(e_i^c)$ et $\mathbf{p}_0(e_i^r)$.

$$\mathbf{p}_0(e_i^r) = \mathbf{p}_{min} = O$$

Lorsque la position initiale de l'élément de référence est située à l'origine du repère, c'est-à-dire sur son parent, nous ne pouvons pas estimer l'angle de rotation de la trajectoire pour correspondre à l'angle de perturbation. Pour remédier à cette situation, nous cherchons le premier $p(e_i^r) \neq O$ et utilisons l'angle correspondant.

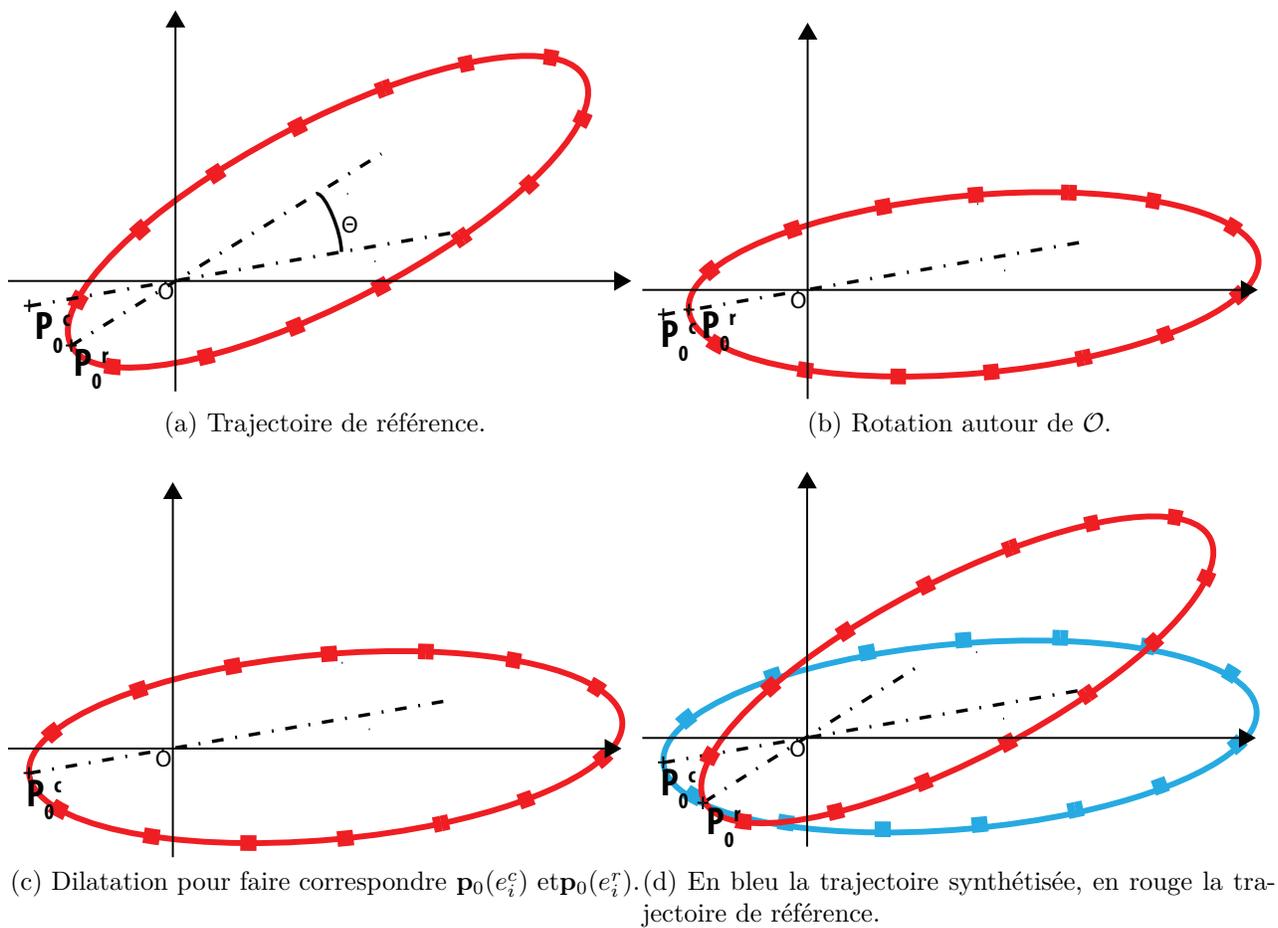


Figure 5.4 Adaptation de la trajectoire de référence lorsque $\mathbf{p}_0(e_i^r) = \mathbf{p}_{min}$.

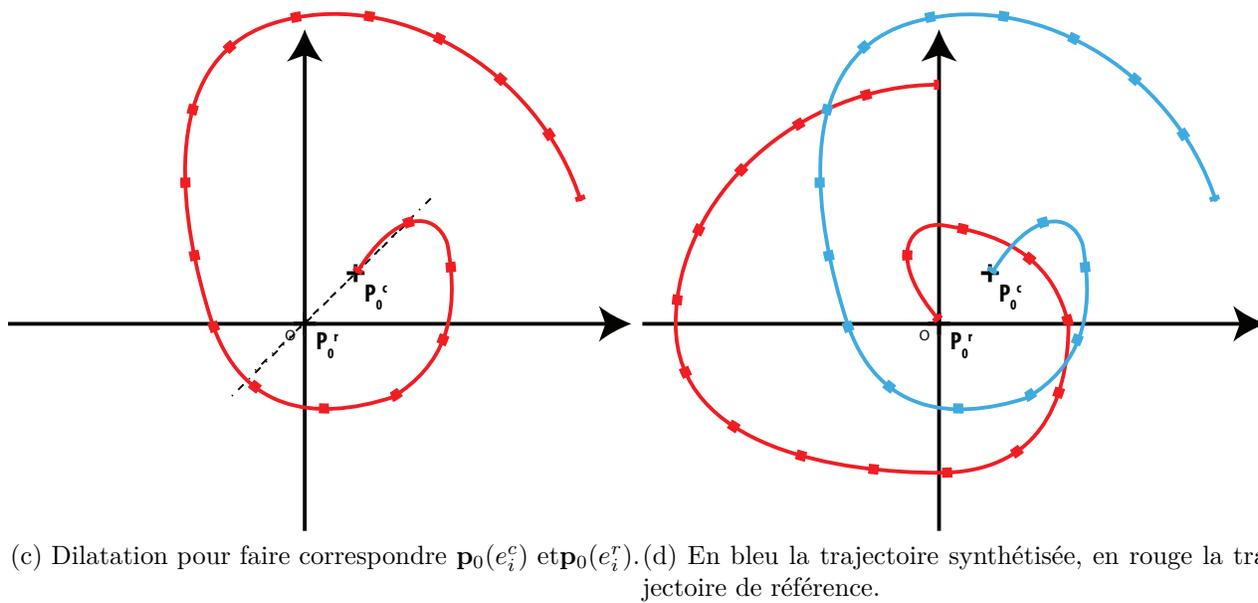
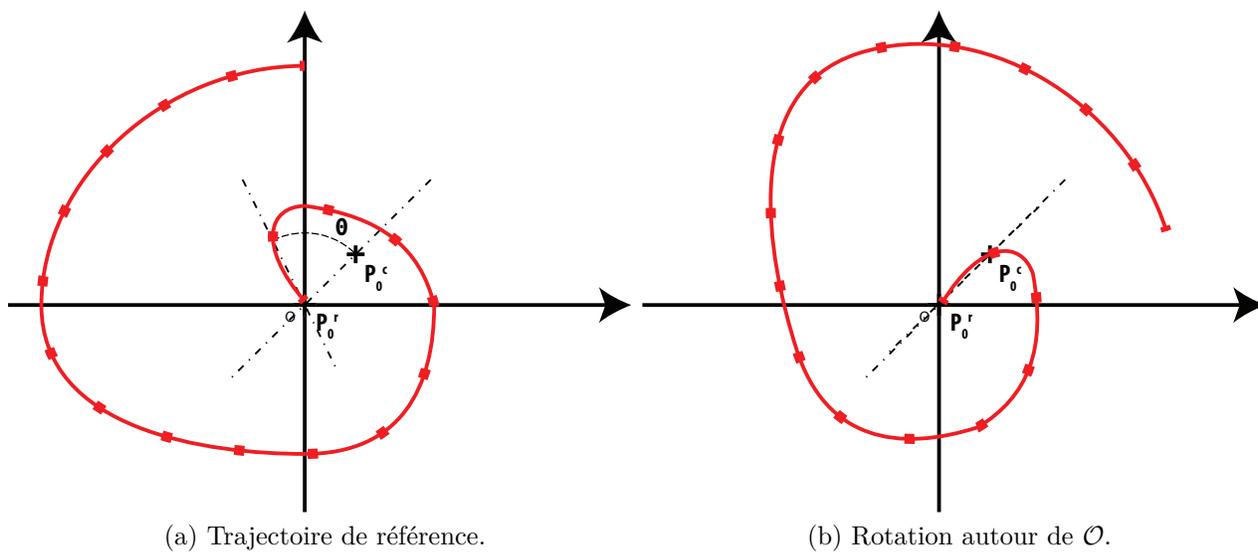


Figure 5.5 Adaptation de la trajectoire de référence lorsque $\mathbf{p}_0(e_i^r) = \mathbf{p}_{min} = \mathcal{O}$.

5.4 Expérimentations

5.4.1 Données

Les expérimentations sont réalisées sur trois exemples contenant à chaque fois peu d'éléments afin que nous puissions nous concentrer sur la forme des trajectoires synthétisées. Pour chacun des tests, la texture est composée de 4 éléments. Deux servent à la spécification du mouvement de référence (un parent, un enfant) et deux autres servent de cible.

5.4.2 Méthodes appliquées

Les résultats de la méthode proposée dans le chapitre sont comparés avec trois itérations précédentes de celle-ci, ainsi qu'à une copie directe du mouvement de l'élément de référence vers l'élément cible. Pour les trois méthodes, la manière dont l'angle de perturbation est appliqué est la même. Les coordonnées du centroïde de l'élément sont ici exprimées en coordonnées polaires, on a $\mathbf{p}(e) = (\rho, \theta)$.

Méthode 0. Cette méthode est la plus naïve que l'on puisse proposer puisqu'il s'agit simplement d'une translation du mouvement.

$$\rho(e_i^c) = \rho(e_i^c) + [\rho_0(e_i^c) - \rho_0(e_i^r)]. \quad (5.3)$$

Méthode 1. La première méthode proposée applique à la trajectoire de référence un facteur d'échelle égal au rapport entre les distances de l'élément de référence et de l'élément cible à leurs racines respectives comme suit :

$$\rho(e_i^c) = \frac{\rho_0(e_i^c)}{\rho_0(e_i^r)} \rho(e_i^c). \quad (5.4)$$

Méthode 2. La deuxième méthode proposée prend en considération la plage de variation de la distance élément-parent afin de réaliser une mise à l'échelle de la trajectoire de référence en conservant les contacts avec le parent :

$$\rho(e_i^c) = \rho(e_i^r) \left[1 + \frac{\rho_0(e_i^c) - \rho_0(e_i^r)}{\rho_{max}(e_i^r) - \rho_{min}(e_i^r)} \right]. \quad (5.5)$$

Méthode 3. La méthode proposée dans ce chapitre.

5.4.3 Résultats

Les résultats fournis par la méthode 1 sont satisfaisants la plupart du temps. Néanmoins, on peut remarquer que la rotation de la trajectoire n'est pas toujours satisfaisante (Figure 5.6d) puisque la rotation est faite par rapport à l'origine du repère contrairement à la méthode 3 pour laquelle dans le cas général (Figure 5.6) la rotation et la mise à l'échelle sont réalisées par rapport au point minimum de la trajectoire. Lorsque $\mathbf{p}_0(e_i^r) = \mathbf{p}_{min} = O$ et de manière plus générale lorsque l'élément de référence se rapproche trop près de son parent ($\rho_0(e_i^r) \leq 0$) la division par $\rho_0(e_i^r)$ fait diverger la trajectoire synthétisée (Figure 5.10). La méthode 2 permet d'éviter la division de la trajectoire par 0, mais ne respecte pas la position initiale du point cible. La méthode 2 donne des résultats constants dans tous les cas de figure testés, la forme globale de la trajectoire et la mise à l'échelle sont conservées. Néanmoins, la position initiale de l'élément est systématiquement modifiée, et le point minimum conservé que lorsqu'il est situé sur le parent (Figure 5.7). La méthode 3 valide toutes les conditions dans tous les cas de figure testés. Les positions $\mathbf{p}_{min}, \mathbf{p}_0$ sont toujours respectées et guident la mise à l'échelle de la trajectoire synthétisée et permettent d'obtenir une mise à l'échelle respectant plus la forme globale (Figure 5.6d).

5.5 Discussion et limitations

Nous proposons dans ce chapitre une méthode permettant d'adapter le mouvement d'un groupe d'éléments fourni en exemple par l'utilisateur à un groupe d'éléments présentant une disposition spatiale similaire. Les contraintes que nous avons exprimées en début de chapitre nous ont permis d'arriver après plusieurs itérations à une méthode fournissant un résultat satisfaisant dans tous les cas auxquels nous avons été confrontés même lorsque les contraintes deviennent incohérentes les unes avec les autres. La principale limitation de la méthode proposée est qu'elle ne prend en compte que les contacts entre les éléments et leur parent, ce qui peut mener à une perte d'information dans les textures possédant des groupes de mouvements complexes. L'analyse de la trajectoire de tous les éléments du groupe les uns par rapport aux autres est une solution envisageable pour remédier à ce problème qui demande de revoir la liste de contrainte et d'en ajouter de nombreuses. Vis-à-vis des résultats présentés, la solution proposée pour gérer l'angle lorsque $\mathbf{p}_0(e_i^r) = \mathbf{p}_{min} = O$ donne des résultats peu convaincants lorsque $\mathbf{p}_0(e_i^c)$ est proche de $\mathbf{p}_0(e_i^t)$. Plus la différence de distance entre les deux éléments est faible, plus l'estimation de l'angle de décalage de la trajectoire est imprécise.

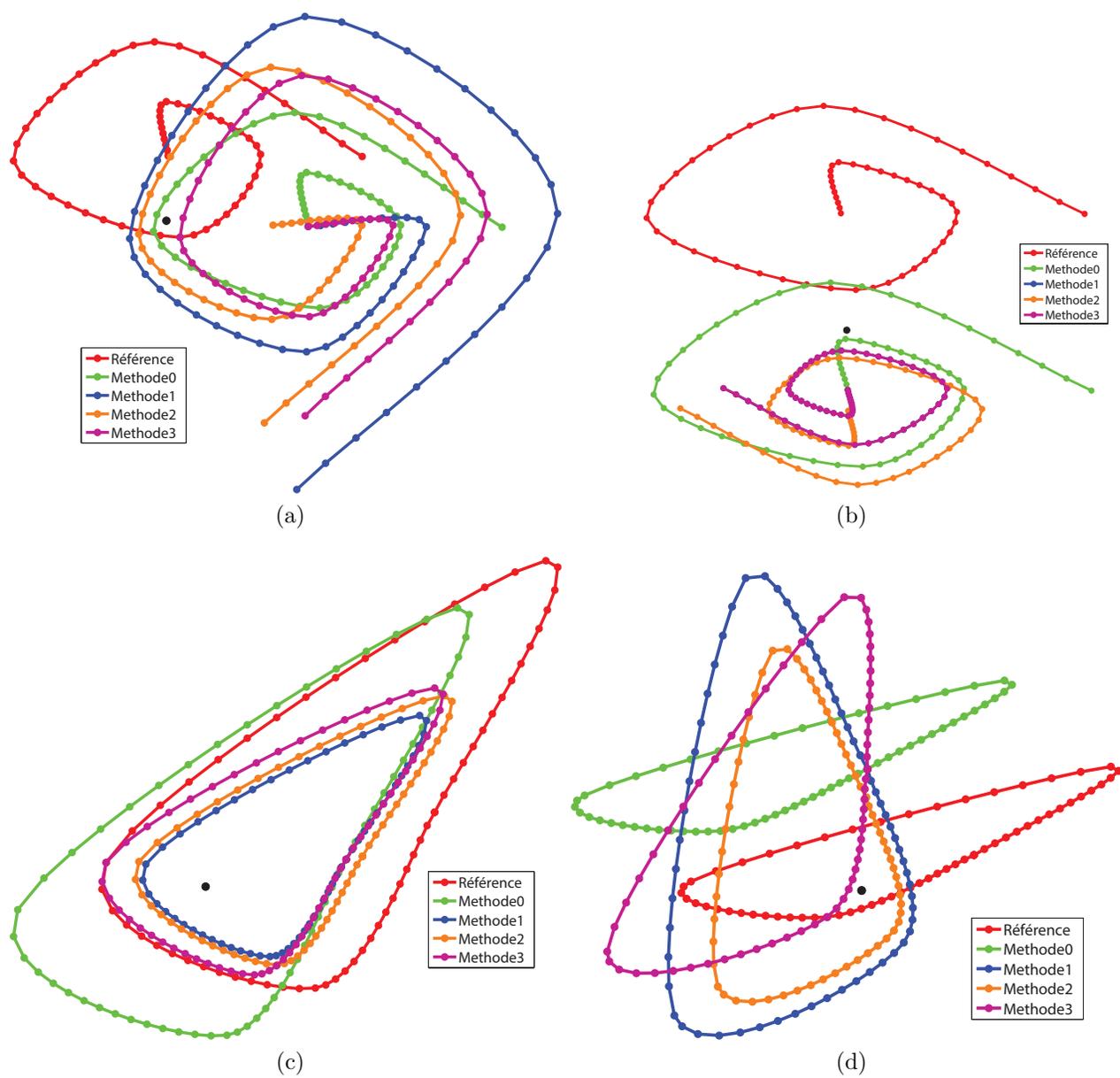
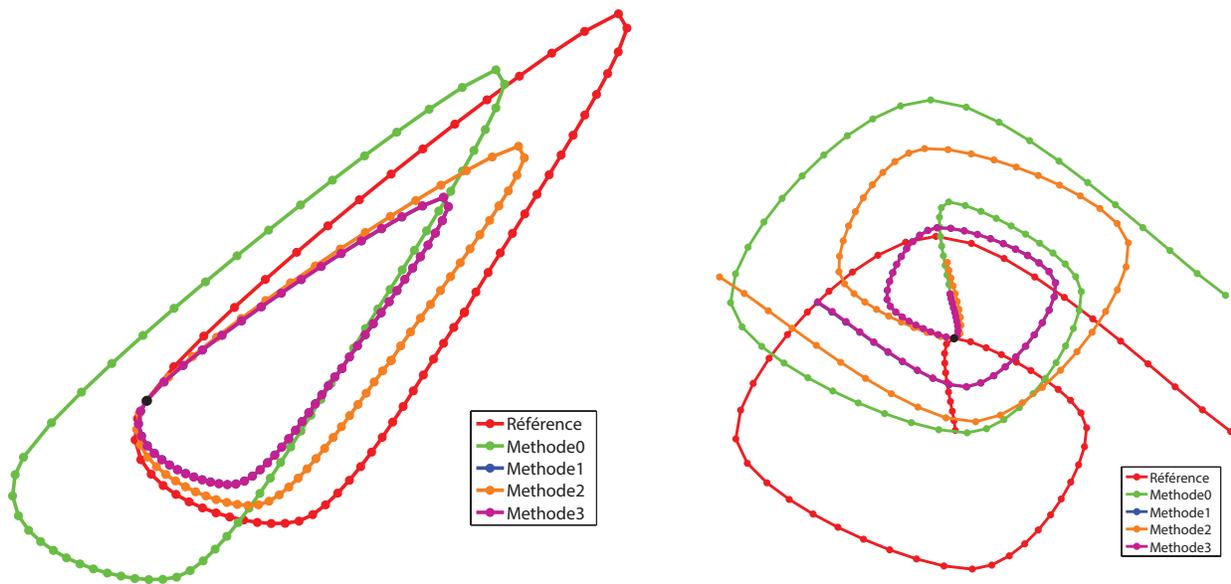
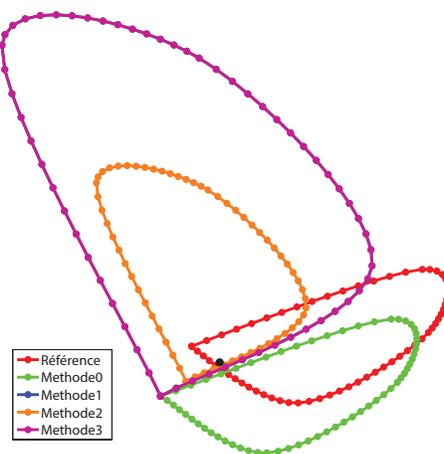
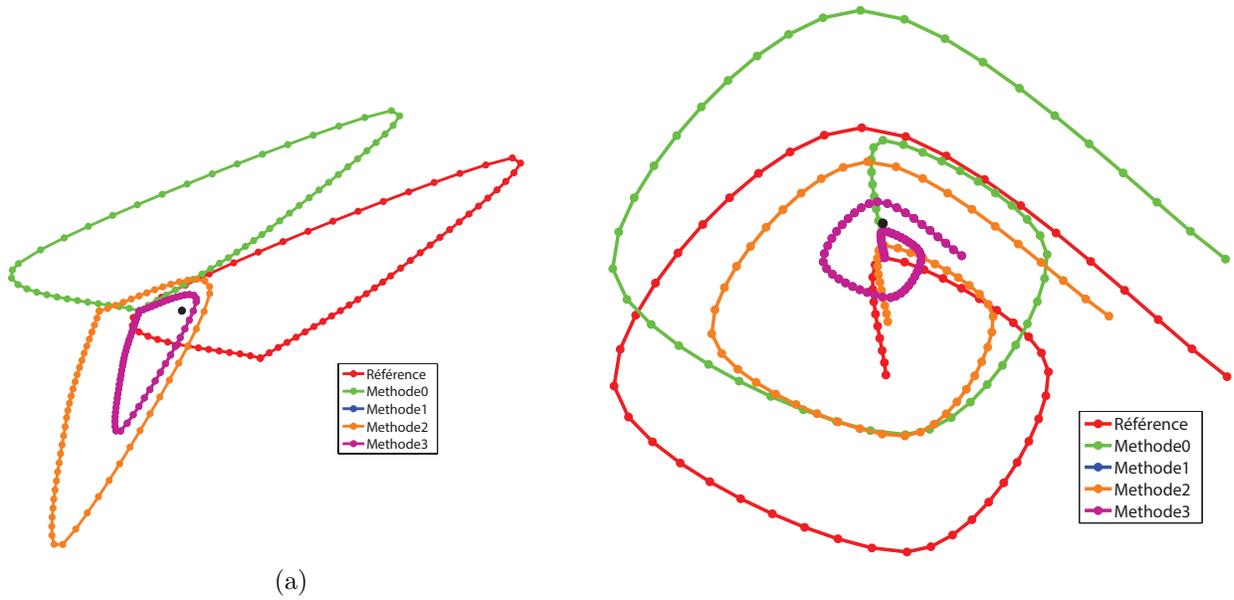
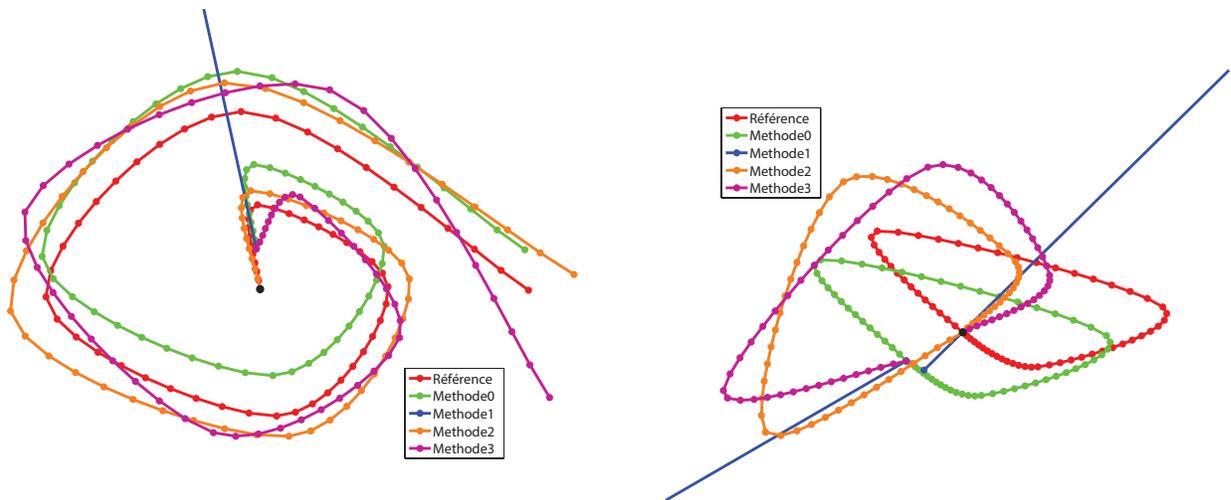


Figure 5.6 cas général.

Figure 5.7 $\mathbf{p}_{min} = \mathcal{O}$.Figure 5.8 $\mathbf{p}_0(e_i^r) = \mathbf{p}_{min}$.

Figure 5.9 $\mathbf{p}_0(e_i^c) = \mathbf{p}_{min}$.Figure 5.10 $\mathbf{p}_0(e_i^r) = \mathbf{p}_{min} = O$.

CHAPITRE 6 CONCLUSION

6.1 Synthèse des travaux

Au cours de ce mémoire, nous avons présenté un système permettant d’animer une texture vectorielle à partir d’un exemple de mouvement qui est décrit par l’utilisateur, puis analysé et adapté à tout le reste de la texture automatiquement.

La première partie de nos recherches a consisté à mettre à disposition de l’utilisateur une méthode lui permettant de spécifier des groupes de mouvement dans un exemple composé d’un ensemble d’éléments qu’il aura préalablement animé. Ces groupes permettent notamment d’explicitier les configurations hiérarchiques et perceptuelles du mouvement d’exemple afin de les conserver lors des étapes ultérieures de la méthode.

La recherche dans la texture statique de dispositions spatiales similaires aux groupes de mouvement spécifiés est réalisée par une approche gloutonne. À chacun des éléments de la texture statique, trois mesures (couleur, disparité des éléments voisins, distance inter-éléments du voisinage) sont associées. La recherche de similarités est équivalente à une minimisation de ces coûts. Afin de rendre le système le plus intuitif possible, nous avons réduit le nombre de paramètres que l’utilisateur doit fixer.

La dernière étape du système est l’adaptation des mouvements des éléments de référence aux dispositions similaires trouvées lors de l’étape précédente. Nous établissons une série de contraintes afin de guider la détermination d’une transformation rigide de la trajectoire du mouvement de référence afin de synthétiser la trajectoire cible. Ces contraintes ont été déduites de l’observation de résultats préliminaires lors des itérations successives de la méthode proposée. Ainsi, nous cherchons à respecter dans cet ordre : la position initiale explicitement précisée par l’utilisateur, la position du point minimum de la trajectoire par rapport au parent (si parent il y a) afin de conserver les contacts entre les éléments et leurs parents, une mise à l’échelle de la trajectoire formalisant la différence de position initiale des éléments de référence et cibles associées, la conservation des mouvements bouclés, et enfin la conservation de la forme globale de la trajectoire. Les trajectoires obtenues par l’adaptation du mouvement sur le résultat donné par la recherche de similarité dans la Figure 6.1, la Figure 6.2 montre quant à elle le résultat de ces trajectoires sur les éléments de la texture de base utilisée au début du Chapitre 4.

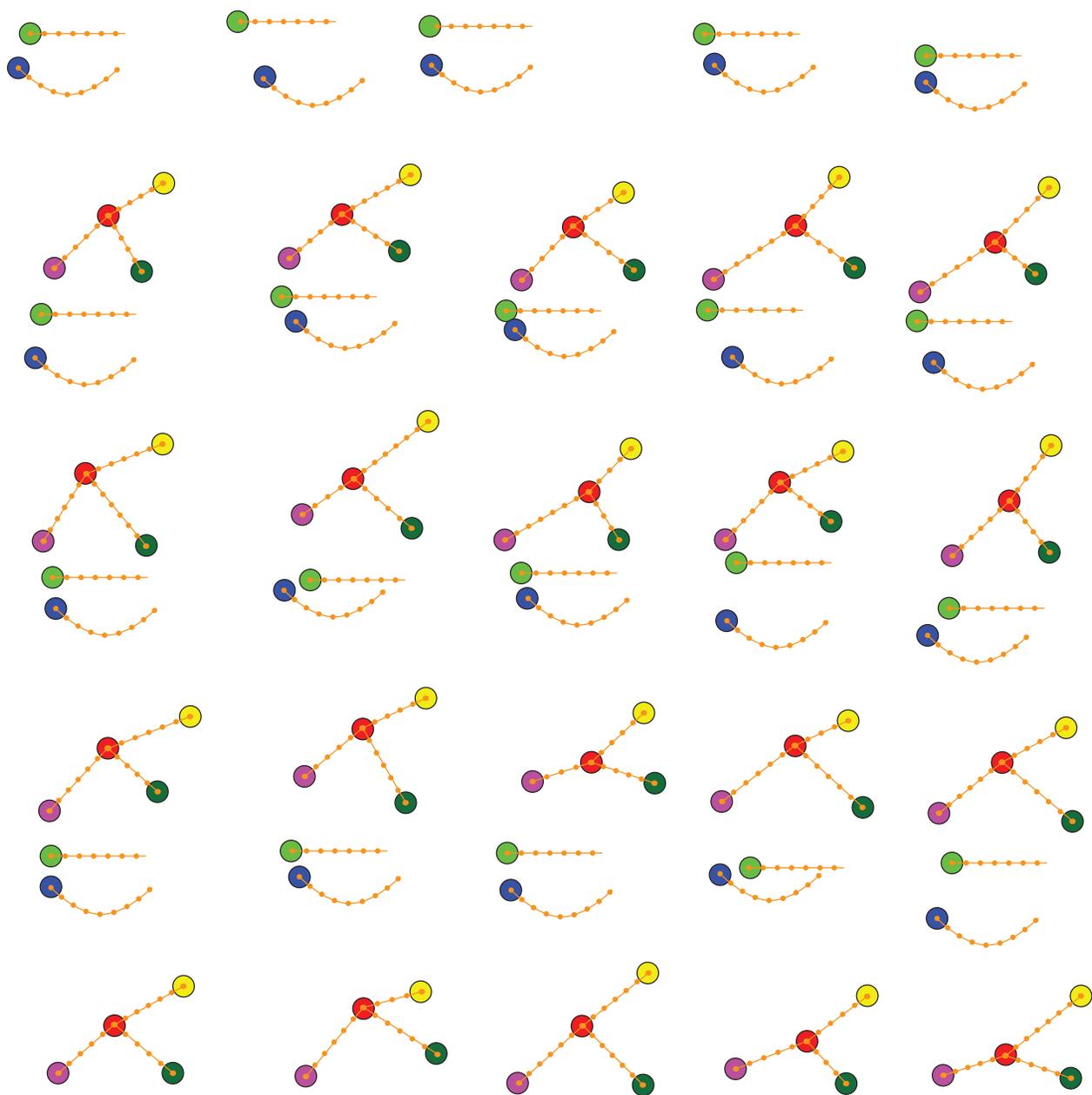


Figure 6.1 Toutes les trajectoires adaptées pour la texture d'exemple avec une perturbation de 30%.

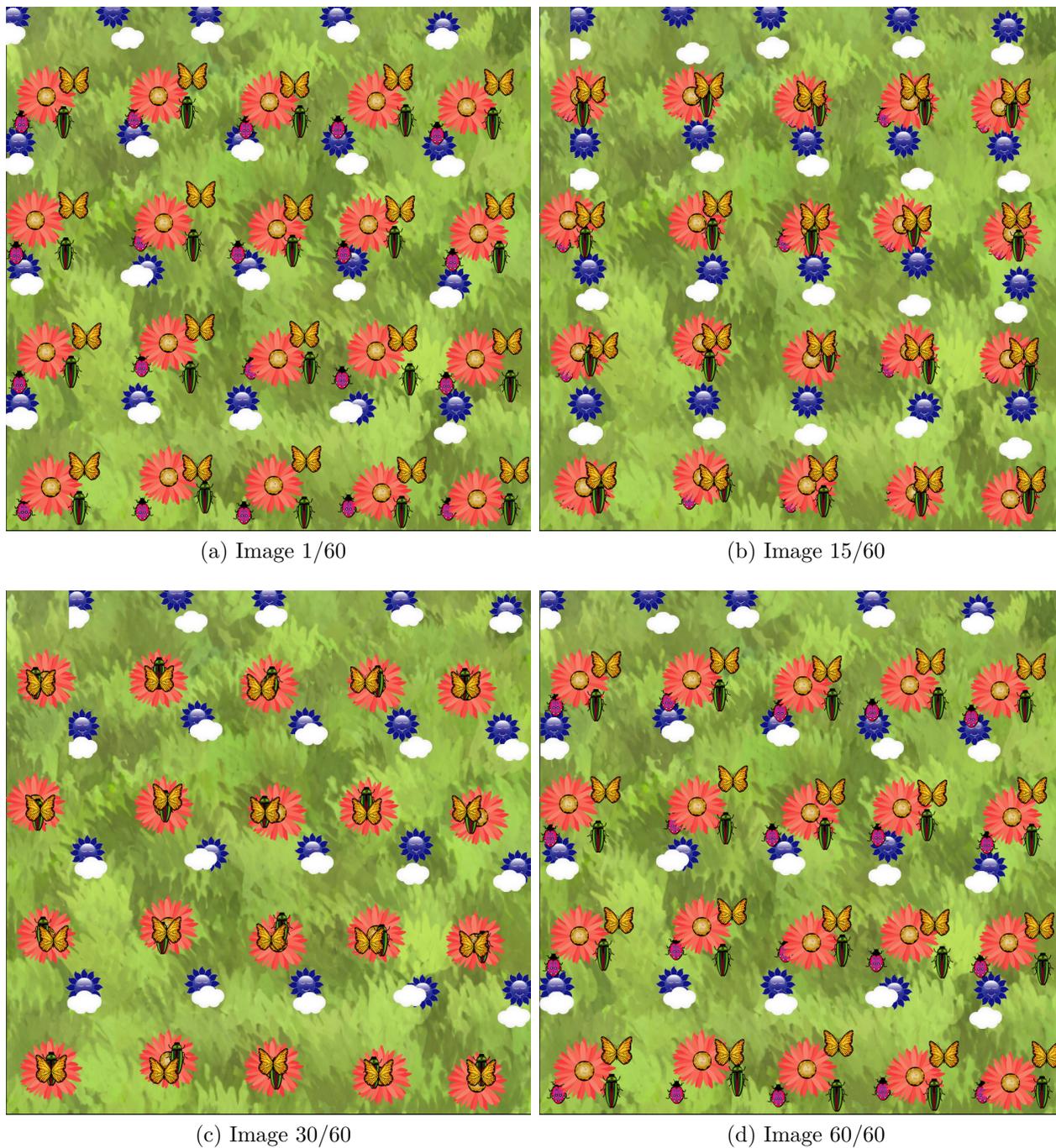


Figure 6.2 Extraits de l'animation synthétisée sur la texture utilisée au début du Chapitre 4.

6.2 Limitations de la solution proposée

Les méthodes proposées pour les différentes étapes du système donnent des résultats conformes aux attentes. Néanmoins, plusieurs limitations existent.

Le système proposé manque de validation, et bien que nous ayons cherché à le rendre intuitif il serait nécessaire de faire évaluer le travail entier par des artistes professionnels ou amateurs. Ce type d'évaluation demande une préparation importante et est longue à mettre en place. Le processus passe par la mise en place d'exemples guidés servant de tutoriaux pour permettre d'introduire les différentes caractéristiques de la méthode, suivis par une étape d'essais libres. La rédaction d'un questionnaire de satisfaction de l'utilisateur n'est pas aisée, puisque les éléments permettant de définir si le résultat est correct peuvent être subjectifs et varier d'une personne à l'autre.

Le système proposé repose sur le fait que l'utilisateur a spécifié des groupes suffisamment différents les uns des autres pour que la méthode de recherche de similarités les discrimine dans la texture statique comme nous avons pu le voir avec l'exemple 2 (Figure 4.8a). Dans des textures très denses avec des motifs entremêlés, le système risque de présenter des faiblesses.

6.3 Améliorations futures

Le fait de considérer que la texture statique d'entrée représente la disposition spatiale à un instant donné de l'animation implique que les animations des éléments sont synchronisées. Lever cette hypothèse permettrait d'envisager un résultat moins homogène dans le temps. Apporter cette modification à la méthodologie proposée demande de revoir entièrement l'étape de recherche de similarité. En ajoutant une dimension temporelle, l'approche gloutonne sera fort probablement très vite limitée par un nombre trop conséquent de minimums locaux de la fonction de coût. En revanche pour ce qui est de l'adaptation, le fait de ne pas considérer toute la texture statique comme étant exprimée au même moment permet d'écarter la plupart des cas particuliers en considérant \mathbf{p}_0 décalé de quelques images. De plus, obtenir dans la texture une variabilité temporelle augmentant avec la variabilité spatiale peut être pertinent. La gestion des contacts entre les éléments d'un groupe à un ordre plus élevé, c'est-à-dire les contacts entre deux éléments dont aucun n'est la racine, permettrait d'établir une meilleure conservation des caractéristiques perceptuelles du mouvement dans un groupe de mouvement. Cette modification demanderait soit d'ajouter une couche de spécification pour l'utilisateur lui permettant d'explicitement les interactions importantes entre les points, soit de réaliser une étape d'analyse plus poussée sur toutes les trajectoires des éléments des groupes de référence. Quelle que soit la solution, elle résultera par une augmentation des contraintes

sur la méthode d'adaptation. Afin d'affiner les résultats de la recherche de similarité dans les cas tels que la texture Exemple 4, plusieurs éléments sont envisageables. Il serait intéressant d'ajouter des caractéristiques de formes des éléments dans l'étape de recherche de similarité. Rendre l'invariance à la rotation facultative permettrait également de donner plus de contrôle à l'utilisateur sur le résultat obtenu sans trop alourdir son expérience.

Dans les motifs comprenant de nombreux groupes de mouvement, il est envisageable de proposer une description à plus haut niveau de l'exemple fourni par l'utilisateur. Une fois tous les groupes de mouvement fixés, une triangulation de Delaunay sur les racines de tous les groupes de mouvement permettrait, une fois les instances du groupe le plus complexe trouvées dans la texture statique, d'inférer la position potentielle des autres groupes, et ainsi d'alléger considérablement le coût calculatoire de l'étape de recherche de similarité.

RÉFÉRENCES

- [1] P. Barla, S. Breslav, J. Thollot, F. Sillion, et L. Markosian, “Stroke pattern analysis and synthesis”, dans Computer Graphics Forum, vol. 25, no. 3. Wiley Online Library, 2006, pp. 663–671.
- [2] R. C. Davis, B. Colwell, et J. A. Landay, “K-sketch : a kinetic sketch pad for novice animators”, dans Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2008, Conference Proceedings, pp. 413–422.
- [3] G. Doretto, A. Chiuso, Y. N. Wu, et S. Soatto, “Dynamic textures”, International Journal of Computer Vision, vol. 51, pp. 91–109, 2003. DOI : 10.1023/A:1021669406132. En ligne : <http://dx.doi.org/10.1023/A:1021669406132>
- [4] J. J. Gibson, The ecological approach to visual perception : classic edition. Psychology Press, 2014.
- [5] T. Hurtut, P.-E. Landes, J. Thollot, Y. Gousseau, R. Drouillhet, et J.-F. Coeurjolly, “Appearance-guided synthesis of element arrangements by example”, dans Proceedings of the 7th International Symposium on Non-photorealistic Animation and Rendering. ACM, 2009, pp. 51–60.
- [6] T. Ijiri, R. Mech, T. Igarashi, et G. Miller, “An example-based procedural system for element arrangement”, dans Computer Graphics Forum, vol. 27, no. 2. Wiley Online Library, 2008, pp. 429–436.
- [7] R. H. Kazi, F. Chevalier, T. Grossman, et G. Fitzmaurice, “Kitty”, pp. 395–405, 2014. DOI : 10.1145/2642918.2647375
- [8] R. H. Kazi, F. Chevalier, T. Grossman, S. Zhao, et G. Fitzmaurice, “Draco”, pp. 351–360, 2014. DOI : 10.1145/2556288.2556987
- [9] H. Loi, T. Hurtut, R. Vergne, et J. Thollot, “A Programmable Model for Designing Stationary 2D Arrangements”, Inria - Research Centre Grenoble – Rhône-Alpes ; INRIA, Research Report RR-8713, Avr. 2015. En ligne : <https://hal.inria.fr/hal-01141869>
- [10] C. Ma, L.-Y. Wei, et X. Tong, “Discrete element textures”, dans ACM Transactions on Graphics (TOG), vol. 30, no. 4. ACM, 2011, p. 62.

- [11] C. Ma, L.-Y. Wei, S. Lefebvre, et X. Tong, “Dynamic element textures”, ACM Transactions on Graphics (TOG), vol. 32, p. 90, 2013.
- [12] L. Olsen, F. F. Samavati, M. C. Sousa, et J. A. Jorge, “Sketch-based modeling : A survey”, Computers & Graphics, vol. 33, pp. 85–103, 2009.
- [13] C. W. Reynolds, “Flocks, herds and schools : A distributed behavioral model”, dans Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, série SIGGRAPH '87. New York, NY, USA : ACM, 1987, pp. 25–34. DOI : 10.1145/37401.37406. En ligne : <http://doi.acm.org/10.1145/37401.37406>
- [14] A. Schödl, R. Szeliski, D. H. Salesin, et I. Essa, “Video textures”, dans Proceedings of the 27th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 2000, Conference Proceedings, pp. 489–498.
- [15] R. M. Steinman, Z. Pizlo, et F. J. Pizlo, “Phi is not beta, and why wertheimer’s discovery launched the gestalt revolution”, Vision Research, vol. 40, pp. 2257–2264, 2000.
- [16] M. Thorne, D. Burke, et M. van de Panne, “Motion doodles : an interface for sketching character motion”, dans ACM SIGGRAPH 2007 courses. ACM, 2007, Conference Proceedings, p. 24.
- [17] L.-Y. Wei, S. Lefebvre, V. Kwatra, et G. Turk, “State of the art in example-based texture synthesis”, dans Eurographics 2009, State of the Art Report, EG-STAR. Eurographics Association, 2009, Conference Proceedings, pp. 93–117.
- [18] H. Xu, H. Zha, et M. A. Davenport, “Manifold based dynamic texture synthesis from extremely few samples”, pp. 3019–3026, 2014. DOI : 10.1109/cvpr.2014.386