



HAL
open science

”Application des IFS à la composition de tissus tridimensionnels virtuels”

Xavier Marsault

► **To cite this version:**

Xavier Marsault. ”Application des IFS à la composition de tissus tridimensionnels virtuels”. Colloque ”Autosimilarité et applications”, CEMAGREF, May 2002, France. pp.1-12. halshs-00265241

HAL Id: halshs-00265241

<https://shs.hal.science/halshs-00265241>

Submitted on 1 Apr 2008

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

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

Application des Iterated Function Systems (IFS) à la composition de tissus urbains tridimensionnels virtuels

Xavier.Marsault@aria.archi.fr

UMR CNRS MAP, "Modèles et simulations pour l'Architecture, l'urbanisme et le Paysage"
Laboratoire ARIA, Ecole d'Architecture de Lyon
3, rue Maurice Audin, 69512 Vaulx en Velin

Résumé

On présente dans cet article les premiers résultats d'une étude initiée dans le cadre du projet de recherche régional sur la réalité virtuelle (DEREVE [DER 2000], en cours), sur la génération automatique de villes à l'aide d'une technique fractale. L'objectif de cette étude est de tenter de capter la cohérence interne de modèles de villes par le biais de la détection de leurs similitudes internes, en cherchant un modèle mathématique approximatif.

On utilise une méthode de compression fractale d'images pour coder par IFS le couple (empreinte au sol, hauteur) du bâti du plan d'une ville, et l'on régénère un modèle 2D½ auto-similaire approximant au mieux cette ville. Des opérateurs d'interpolation, de filtrage, ou génétiques (mutation, fusion) sont ensuite introduits pour produire automatiquement de nouvelles morphologies pseudo-urbaines et architecturales ressemblant à la réalité, s'appuyant sur un ou plusieurs plans ainsi codés, et géométriquement allégées pour permettre la navigation en temps réel.

Mots clés : Ville fractale. IFS. Image. Modèle 3D. Niveau de détail. Génétique. Interpolation. Fusion. VRML.

1. Introduction

L'investigation fractale des tissus urbains, à des fins d'analyse ou de composition, est finalement assez récente. Des études ont montré qu'une large majorité de structures urbaines connues présentent des caractéristiques spatiales complexes, de nature fractale ou multifractale, à certaines échelles d'observation [Batty M., Longley P.A., 1994], [Frankhauser P., 1994]. Différents travaux se sont articulés autour de trois aspects : méthodes d'analyse de l'organisation spatiale des villes [Frankhauser P., 1997], approches génératrices de structures géométriques, et outils de simulation de données physiques et de comportements humains (flots de piétons ou de véhicules, émissions sonores et thermiques...). Une voie de recherche tentante est d'essayer d'appliquer des techniques fractales à la modélisation et à la génération de tissus urbains virtuels.

On notera que, de manière générale, l'obtention de modèles géométriques de motifs urbains ou de villes entières, se fait soit par le biais de simulateurs de croissance spatiale, soit par des simulateurs temporels basés sur une histoire (de type Sim City), soit enfin par des générateurs statiques où le modèle urbain est figé dans le temps [Parish Y., Muller P., 2001]. Dans le premier cas, on rencontre de nombreux travaux, dont les plus importants portent sur les automates cellulaires [Torrens P., 2000], les modèles DLA (diffusion limitée par agrégation) ou divers modèles organiques inspirés des lois de la physique en milieu réplusif [Makse H.A., 1996], [Thèse de Eric Bailly.....].

2. Les IFS : Iterated Function Systems

2.1. Généralités

La théorie des IFS est entièrement basée sur la propriété d'invariance par changement d'échelle. Elle permet de générer des objets fractals à l'aide d'un ensemble de fonctions contractantes traduisant cette propriété, formant un « système de fonctions itérées » ou IFS. Elle fut étudiée par Hutchinson dans le cadre mathématique de l'auto-similarité [Hutchinson J., 1981], puis par Barnsley dans le cadre de la géométrie fractale [Barnsley 88], avec des applications en compression d'images [Barnsley M., 1992-1993], [Jacquin A.E., 1992].

On rappelle qu'un objet mathématique est dit « auto-similaire » s'il peut être partitionné en sous-objets qui se déduisent du tout par une « similitude ». Il conserve ainsi sa géométrie à plusieurs échelles d'observation. En fait, on observe assez peu pour des objets du monde réel, et généralement pas pour les structures urbaines, une telle propriété d'ensemble. Par contre, localement, cela peut être plus ou moins mis en évidence, et l'application d'un modèle fractal IFS à un tissu urbain peut s'envisager.

2.2. Compression d'images

La compression fractale des images se heurte, dans un premier temps, au fait que la plupart des images ne sont pas des objets fractals, et qu'il est peu probable de trouver un générateur fractal de l'image entière. D'où l'utilisation des *lifs* (local IFS) comme solution permettant de partitionner l'image en blocs qui vont être codés par des transformations fractales (invariantes par changement d'échelle) d'autres blocs plus grands de cette même image. On associe donc à une image une série de transformations itératives contractantes locales de parties d'elle-même.

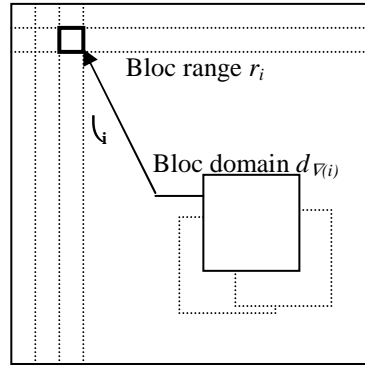


Figure 1 : transformation spatiale d'un domain bloc en un range bloc

L'algorithme de compression de base est celui de Jacquin [Jacquin A.E., 1992]. On rappelle qu'étant donné un partitionnement carré uniforme de l'image en N « range blocs » r_i , et un espace de recherche de « domain blocs » d_i (Fig. 1), la méthode de Jacquin consiste à trouver une application $i \rightarrow \alpha(i)$ et N *lifs* ω_i tels que : $r_i \rightarrow \hat{r}_i = \omega_i(d_{\alpha(i)})$

avec $\sum_i \|\omega_i(d_{\alpha(i)}) - r_i\|^2$ minimum pour chaque i . Chaque *lifs* ω_i est composé d'une transformation projetant un

domain bloc d_i à la place et à l'échelle du bloc r_i (une décimation des pixels suivi d'une isométrie) et d'une transformation massique affine sur les niveaux de gris. La transformation isométrique iso_i est choisie parmi les 8 transformations possibles d'un bloc carré (identité, rotations -90 , 90 ou 180 degrés, symétries en x , en y , ou par rapport aux diagonales). Si un range bloc r_i est uniforme, aucune recherche de domain bloc n'est entreprise, et l'on code dans le *lifs* ω_i uniquement le niveau de gris du bloc r_i . Dans le cas contraire, on introduit une transformation massique avec facteur de contraste σ_i et décalage β_i telle que $\hat{r}_i = iso_i(\sigma_i \cdot r_i + \beta_i)$.

2.3. Décompression d'une image codée par un IFS

De manière classique, la reconstruction de l'image s'opère en calculant l'attracteur A de l'ensemble des *lifs* à partir d'une

image quelconque I_0 . En notant $W(I) = \prod_{i=1}^N \hat{r}_i = \prod_{i=1}^N \omega_i(d_{\alpha(i)})$, $W^{k+1}(I) = W(W^k(I))$, l'attracteur est défini par :

$A(I) = \lim_{k \rightarrow +\infty} W^k(I_0)$. Le processus converge et se stabilise généralement au bout de quelques itérations, mais les facteurs de contraste σ_i ne peuvent être choisis n'importe comment. Pour assurer la convergence uniforme, on maintient généralement la contractance en tout bloc en spécifiant $\forall i, |\sigma_i| < 1$, bien que l'on pourrait dépasser localement cette limite de contractance de 1, comme cela a été montré [Hutgen B., Hain T., 1994].

3. Génération de modèles de villes par IFS

3.1. Pourquoi choisir les IFS ?

On souhaite trouver une logique interne à des modèles urbains grâce à un outil mathématique d'analyse et de synthèse. Avec les IFS, on travaille sur un espace de formes, continu, autorisant des perturbations et des interpolations. De par leur principe fractal, les IFS ont l'avantage d'analyser et de générer des structures à la fois globales et locales, à la différence des DLA ou des automates cellulaires, où des règles locales induisent un comportement global. On peut aussi introduire des altérations locales avec certaines propriétés, n'influant pas ou peu sur la totalité de la forme produite (les IFS à condensation autorisent l'inclusion d'éléments extérieurs au codage). Les IFS semblent donc prometteurs pour qui souhaite analyser, modifier et recomposer des modèles 3D de tissus urbains.

3.2. Genèse du concept

L'utilisation des IFS pour générer des empreintes de quartiers, avec leur hiérarchie de pleins et de vides, correspondant respectivement au bâti et à la voirie, est décrite par exemple par [Woloszyn P., 1998]. Il montre comment une simple règle de substitution, itérée trois fois à partir d'un motif d'origine, conduit à une image ressemblant à une structure urbaine (Fig. 2).

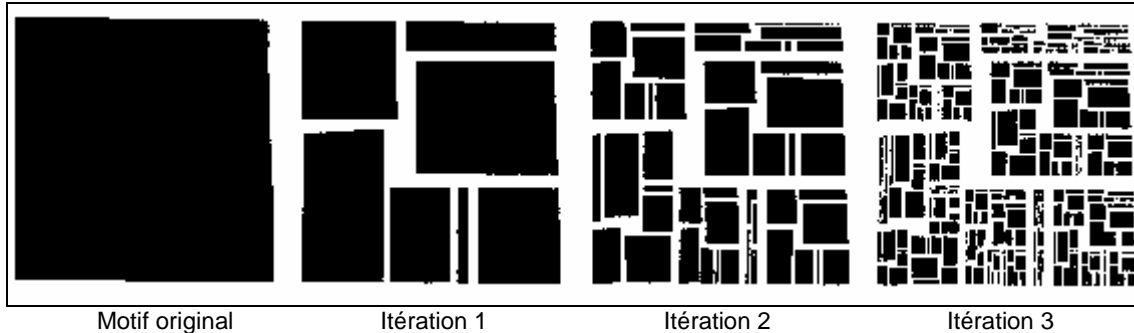


Figure 2 : exemple de génération de motifs pseudo-urbains par un IFS [Woloszyn P., 1998]

Cette simple technique permet d'engendrer progressivement des structures que l'on peut assimiler à la voirie principale, puis aux réseaux secondaires, et finalement au bâti de proximité. Mais, elle s'avère difficilement capable de reproduire ou d'approximer un tissu urbain existant, nettement moins régulier et plus complexe. Notre objectif étant de générer de la géométrie urbaine virtuelle ressemblant à la réalité, pourquoi ne pas partir finalement de modèles géométriques de villes réelles, et les analyser avec des IFS locaux, certainement plus appropriés pour obtenir de bonnes reconstructions ?

L'échelle urbaine concerne la distribution du bâti sur un certain territoire, et peut être décrite de manière certainement réductrice, mais suffisante (pour un rendu rapide entre autres), par un ensemble de volumes simplifiés et représentés dans une approche 2d ½ (vue de dessus + hauteur). Dans un premier temps, on néglige la surface non-bâtie, et l'on se concentre uniquement sur la distribution spatiale du bâti. Si l'on voulait travailler à l'échelle architecturale, il serait souhaitable de prendre en compte la dimension de surface et la distribution verticale des constituants des bâtiments. On laissera volontairement de côté cette échelle, afin de simplifier l'étude.

La technique de compression d'images par IFS décrite en 2.2 permet de reconstruire une image quelconque à partir de transformations locales de parties d'elle-même. Or, l'approche 2d ½ que l'on adopte permet de passer du modèle géométrique 3D au domaine des images (le niveau de gris codant la hauteur), et autorise finalement le codage de tissus urbains par des IFS. Il est important de noter qu'une telle démarche cherche à approximer un modèle géométrique urbain quelconque en un objet fractal auto-similaire, par le fameux théorème du collage [Barnsley M., 1993]. La précision va dépendre du nombre de générateurs (*lifs*) que l'on se donne pour épouser au mieux les diversités locales de l'image et capter la cohérence interne de la ville par le biais de la détection de ses similitudes internes.

3.3. Algorithme utilisé pour le calcul des LIFS

Pour valider la démarche, on est parti d'une solution simple : l'algorithme de compression de Jacquin présenté en 2.2. On a décidé d'étendre l'espace de recherche des « domain blocs » à toute l'image, leur taille D étant multiple de la taille B des range blocs, sans limite a priori, et pouvant changer pour chaque *lifs*. Pour être plus efficace, on scinde les range blocs en deux classes : les blocs uniformes (niveau de gris constant) et les autres (essentiellement des zones de contours). Lors de la recherche du meilleur appariement, on travaille aussi sur ces deux classes, afin de diminuer l'éventail des domain blocs à appairer.

VERS UN « MODELE » DE LA VILLE

Par ailleurs, la question du choix peut se poser lorsque plusieurs domain blocs sont candidats au meilleur appariement. Dans ce cas, on peut sélectionner celui dont le voisinage est le plus proche (au sens quadratique) du voisinage du range bloc traité. Cette option augmente nettement la durée des calculs, mais elle est la seule qui puisse rendre vraiment compte des liens topologiques entre les blocs B d'une part, et les blocs D d'autre part.

Remarque : les traitements effectués et présentés dans cet article ne tiennent pas encore compte de cette finesse de codage.

Il faudra d'ailleurs se donner des seuils, et éviter surtout de ne retenir que les blocs D qui atteignent strictement le maximum. Le rôle du paramètre sigma est aussi à étudier dans cette hypothèse.

On a cherché à obtenir un codage le plus précis possible, plutôt qu'une compression efficace, moyennant une limite de temps raisonnable et donc une optimisation des traitements. Malgré cela, le codage en IFS des images de 2048 x 2048 pixels présentées en Figure 7 a pris trois jours sur un Pentium III 800 MHz.

3.4. Schéma des traitements proposés

Des opérateurs d'interpolation convexe, de mutation, de fusion et de filtrage (Fig. 3) ont ensuite été développés pour générer de nouveaux modèles de tissus pseudo-urbains s'appuyant sur un ou plusieurs IFS provenant d'images de tissus urbains réels ou synthétiques, ou les modifier. Pour cela, on a été amené à développer une analogie avec la génétique, objet du paragraphe 4.

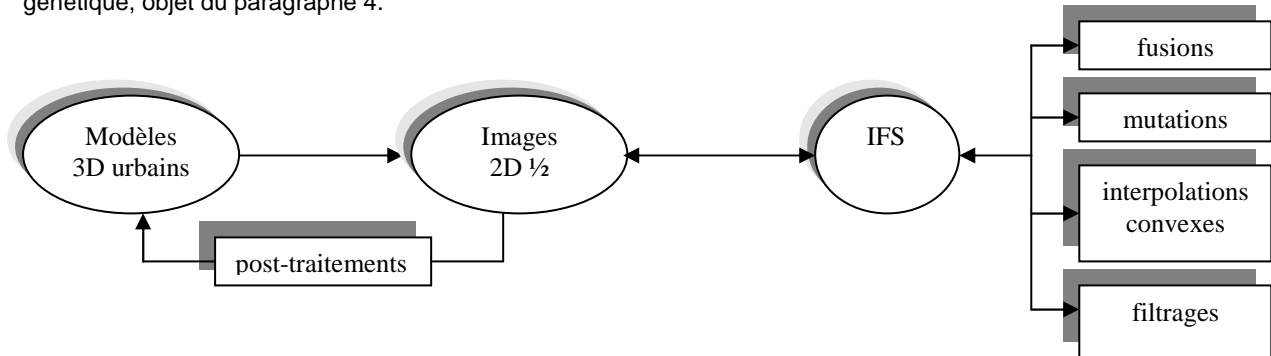


Figure 3 : schéma général des traitements

L'ensemble des programmes et l'interface graphique (Fig. 14) sont développés en C et C++ sous Windows, avec la librairie Qt (Université de Trolltech). Un programme de conversion image → modèle 3D par extrusion de pixels a été écrit pour visualiser les résultats. Il est en cours d'optimisation en fonction de caractéristiques propres au bâti généré, et de post-traitements liés au codage fractal. Les hauteurs sont quantifiables durant l'étape d'extrusion.

3.5. Interpolation pseudo-convexe de deux IFS de tissus urbains

On suppose que deux images distinctes I_1 et I_2 sont codées respectivement par les IFS $\{\omega_i^1\}$ et $\{\omega_i^2\}$. Une première idée à tester dans le cadre théorique des IFS, consiste à définir des IFS convexes paramétrés :

$$\omega_i(\lambda) = \lambda.\omega_i^1 + (1-\lambda).\omega_i^2$$

dont on sait que l'attracteur $I(\lambda) = A\{\omega_i(\lambda)\}$ est continu en fonction de λ [Gentil C., 1992], si le semi-groupe d'itération est convexe. Comme ça n'est pas le cas ici avec le semi-groupe des 8 isométries du carré, on parlera désormais d'interpolation pseudo-convexe. Ainsi, on passe de manière à la fois discrète et continue d'une image $I(\Leftrightarrow)$ à une autre, donc d'un modèle de ville à un autre, après régénération 3D. On observe les résultats que montre la Figure 8. Une solution plus générale consisterait à associer à chaque *lifs* un paramètre λ_i propre :

$$\omega_i(\lambda_i) = \lambda_i.\omega_i^1 + (1-\lambda_i).\omega_i^2 \text{ et } I(\lambda) = A\{\omega_i(\lambda_i)\}$$

Même si le nombre de λ_i peut être très élevé, cela permettrait d'explorer nettement plus de solutions qu'avec un seul paramètre. Comme cas particulier, on peut choisir λ_i dans $\{0;1\}$, ce qui correspond au choix exclusif de l'un des deux IFS locaux ; mais on perd la continuité du résultat en fonction des paramètres λ_i (cf. 4.2).

- Calcul pratique de cette interpolation

On écrit d'abord dans un cas général : $W_1(I_1) = \prod_{i=1}^{N_1} \hat{r}_i^1 = \prod_{i=1}^{N_1} \omega_i^1(d_{\alpha(i)})$ et $W_2(I_2) = \prod_{j=1}^{N_2} \hat{r}_j^2 = \prod_{j=1}^{N_2} \omega_j^2(d_{\gamma(i)})$. Les

partitionnements initiaux des images pouvant être différents, on n'a pas nécessairement $N_1=N_2$. Dans le cas d'un partitionnement carré de Jacquin commun aux deux images, à chaque bloc r_i sont associés deux blocs d_{i_1} et d_{i_2} (qui peuvent être de tailles distinctes), correspondant aux deux *lifs* de I_1 et I_2 : ω_{i_1} et ω_{i_2} . On définit le bloc interpolé :

$d_i(\lambda) = \lambda.d_{i_1} + (1-\lambda).d_{i_2}$, obtenu par combinaison convexe de points des domain blocs. L'interpolation des

transformations s'obtient de manière analogue. On obtient finalement pour chaque range bloc un *lifs* ω_i^λ , et l'on décode

l'IFS en calculant naturellement l'attracteur de $W_\lambda(I_1, I_2) = \prod_{i=1}^N \omega_i^\lambda(d_i(\lambda))$.

On a été amené à définir un autre opérateur d'interpolation, qui évite le calcul d'IFS intermédiaires entre $W(I_1)$ et $W(I_2)$, ce qui est pratique lorsqu'on ne sait pas le faire. En partant d'une image quelconque I_0 , on commence par calculer les deux images $A(W_1(I_0))$ et $A(W_2(I_0))$. Puis, à chaque itération k du processus de décodage, on calcule :

$$W^k(\lambda) = \lambda.W_1(A(W^{k-1}(\lambda))) + (1-\lambda).W_2(A(W^{k-1}(\lambda)))$$

Cela revient donc à fusionner les résultats partiels du décodage des IFS à chaque itération, ce qui induit nécessairement un biais dans la convergence propre de chacun des IFS. Mais en pratique, dans le cas précis du semi-groupe des isométries du carré, on a pu vérifier que cet opérateur converge après une dizaine d'itérations, et que les images obtenues ressemblent fortement à celles produites avec la méthode précédente. Pour autant, on ne fournit encore aucun élément pouvant expliquer ce résultat. Le point pratique qu'on retiendra est que cet opérateur permet de calculer une forme d'interpolation entre deux images utilisant chacune leur propre technique de codage par IFS et leur propre partitionnement.

4. Formalisation de type génétique

4.1 Description

L'analogie génétique a été introduite par [Vences L., Rudomin L., 1997]. Elle consiste à considérer l'IFS d'une image I comme un chromosome dont les gènes sont les *lifs* ω_i : $IFS = (\omega_1, \omega_2, \dots, \omega_N)$, pour ensuite exploiter des traitements issus de la génétique. Cette approche peut surprendre, mais elle peut se justifier. En effet, outre le passage de l'IFS (phénotype) à l'image (génotype), l'information qui sert à décoder un fragment de l'image est répartie dans un grand nombre de *lifs*. Des modifications locales d'un *lifs* peuvent se répercuter (ou pas) sur de nombreuses autres zones de l'image, l'ensemble des *lifs* formant un système complexe et hautement non linéaire, comme en génétique. Dans le même prolongement d'idées, on rencontre des domaines blocs qui sont très utilisés, et qui subissent plusieurs transformations isométriques et massiques différentes, d'autres qui ne servent qu'une seule fois, et d'autres enfin qui sont simplement décodés et non codants (cf. 5.5). Un dernier aspect de cette analogie réside dans la concision des codes génétiques, que l'on retrouve ici, puisqu'on sait élaborer des méthodes performantes de compression par IFS.

En suivant ce schéma, on a commencé à appliquer deux mécanismes propres aux algorithmes évolutionnistes [Renders J.M., 1995] : la mutation et la fusion (ou croisement), pour engendrer de la nouveauté s'appuyant sur de l'existant. Généralement, le croisement consiste à créer une population d'individus partageant des gènes entre les deux parents, et la mutation consiste à altérer certains gènes lors des croisements, ou échanger un ou plusieurs gènes sur un même chromosome. Dès lors qu'on cherche à « croiser » deux images I_1 et I_2 , on peut définir des opérations sur les chromosomes : mutations et fusions des *lifs*. Les mutations peuvent s'opérer a priori sur n'importe quelle composante d'un *lifs*, c'est-à-dire sur l'un au moins des paramètres $(\beta, \sigma, x, y, x_D, y_D, iso)$ que l'on pourrait appeler « nucléotides ».

4.2. Fusion directe de deux IFS de tissus urbains

Si elle se pratique de manière aléatoire et non supervisée, l'approche de fusion directe *lifs* à *lifs*, même hautement combinatoire, donne de très mauvais résultats. De même, le fait d'exprimer trop souvent le même *lifs* à de multiples endroits (copie de gènes) conduit vite à des dégradations et à une uniformité dues à la non conservation de la répartition spatiale initiale (topologie). Une autre voie testée consistait à permuter des pavés de même taille dans l'image (une autre forme de mutation), pour modifier la topologie du modèle urbain en échangeant le contenu des pavés. Elle a donné de médiocres résultats.

Dans toute la suite, on va donc conserver cette topologie initiale et tenter de grouper les *lifs*. On note cependant que la fusion directe se prête très bien à l'application d'algorithmes génétiques (les λ_i étant les paramètres, comme expliqué en 3.5). Encore faut-il définir des fonctions de coûts et des objectifs, c'est à dire injecter dans ces algorithmes de l'intelligence relative à la connaissance des villes, ce qui est l'objet de travaux en cours.

4.3. Fusions asymétrique et symétrique de deux IFS

La fusion directe considère que, pour un bloc range donné, les *lifs* des deux images sont interchangeable, alors qu'ils codent a priori des zones dont le contenu est différent. Si l'on poursuit l'analogie avec la génétique : la fusion des

gamètes est très certainement un processus combinatoire, mais elle ne se fait pas entre n'importe quels gènes. Certains sont mis en relation - on parle d'appariement - selon leur fonction et des règles qui ne nous sont pas encore toutes connues.

Un premier pas que l'on propose pour tenir compte de ce mécanisme consiste à calculer des appariements de range blocs entre les deux images, basés sur le contenu des images. **DECRIRE LA METHODE D'APPARIEMENT.** L'opération d'appariement n'étant pas du tout symétrique, on associe à chaque image la liste des blocs appariés dans l'autre image. Le principe est décrit en Figure 4. Il s'agit de la fusion asymétrique : un bloc r_1 de l'image 1 est apparié au bloc r_2 de l'image 2, lequel est codé par un *lifs* de base d_2 . On associe finalement à r_1 le *lifs* du bloc d_2 . En faisant ainsi pour tous les range blocs, on peut générer deux images : $1\text{mod}2$ et $2\text{mod}1$. La première est l'image 1 modulée par l'appariement avec l'image 2 (Fig. 10).

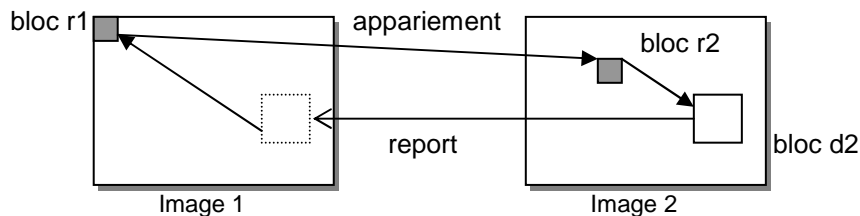


Figure 4 : principe de la fusion asymétrique

Une autre façon d'utiliser ces appariements consiste, pour chaque bloc r de l'image que l'on souhaite obtenir, à sélectionner les blocs appariés r_{12} et r_{21} de r dans les images 1 et 2. Ces deux blocs sont calculés respectivement par $lifs_1$ et $lifs_2$. Comme ils correspondent à une information spatiale appariée, on peut espérer de meilleurs résultats, à condition, toutefois, de ne pas briser la topologie des éléments du bâti (ce qu'on ne sait pas encore bien réaliser), car on a vu en 4.2 que cela conduit au phénomène d'émiettement.

Toutes les opérations de fusion directe et d'interpolation peuvent aussi être reprises sur cette paire ($lifs_1$, $lifs_2$) et sont cette fois symétriques par rapport aux deux images. Par exemple, l'interpolation consiste à calculer dans « l'espace des domain blocs d_1 et d_2 » le bloc interpolé $d(\lambda) = \lambda.d_1 + (1-\lambda).d_2$, et à lui appliquer les transformations isométriques et massiques interpolées. Cependant, on ne présentera pas de résultats dans cet article, l'approche étant à peine ébauchée pour l'instant.

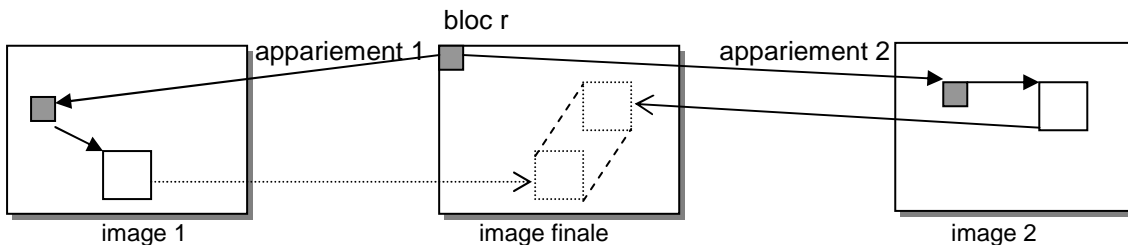


Figure 5 : principe de la fusion symétrique

4.4. Fusion par pavés de deux IFS de tissus urbains

Une fusion par pavés consiste à croiser des séquences de *lifs* groupés spatialement plutôt que des *lifs* isolés (analogie avec des groupes de gènes qui s'expriment conjointement), notamment pour préserver la topologie de certains quartiers. On considère un même partitionnement carré de Jacquin pour les deux images, dont les blocs range sont de $B \times B$ pixels. On définit un pavage carré de taille supérieure à B ($kB \times kB$, k entier) afin de regrouper plusieurs range blocs connexes. La fusion s'opère, pour un pavé donné, en gardant sélectivement soit les *lifs* de la première image, soit ceux de la seconde. Il y a donc continuité à l'intérieur de chaque pavé, et de possibles discontinuités aux frontières entre les pavés.

Il existe de multiples configurations à base de pavages carrés dans une image, mais on va en limiter le plus possible les discontinuités. Il est aussi intéressant de pouvoir varier la taille des pavés, ce qui conduit à pouvoir varier des échelles de croisement. Deux cas se présentent : les pavés sont recouvrants, ou ne le sont pas. On n'envisagera pas un pavage de type quadtree, car il présente des discontinuités trop visibles dues aux pavés non recouvrants.

C'est pourquoi on s'est tourné vers la technique de recouvrement alternative suivante : l'algorithme va « peindre la toile urbaine » avec des coups de pinceaux de taille décroissante, en changeant de pinceau alternativement (un pinceau pour chaque modèle urbain). On va répartir en premier lieu les pavés de taille importante, et finir par les pavages de petite taille (un urbaniste planifie à petite échelle avant de s'intéresser au contenu des quartiers).

Les deux images sources sont représentées par IFS_1 et IFS_2 . Les positions des pavés ainsi que le taux de remplissage (rapport du nombre de blocs de l' IFS_1 par rapport à l' IFS_2) sont fournis par l'utilisateur ou produits par un générateur

pseudo-aléatoire, l'utilisateur fournissant toujours les bornes maximales et minimales de la taille des pavés. L'algorithme pré-remplit ensuite l'image avec l'un des IFS. Puis il passe à l'autre, et il fera cela alternativement jusqu'à ce qu'un test d'arrêt soit vérifié. A chaque changement d'IFS, la taille des pavés est décrétementée d'une unité. A chaque étape, l'algorithme injecte des pavés jusqu'à ce que le taux de remplissage de l'IFS sélectionné soit atteint. Il se donne pour cela une marge, cette marge s'atténuant au fur et à mesure des étapes.

On a aussi défini un critère de non-intersection permettant d'utiliser comme « bloc admissible » tout bloc dont le bord n'intersecte pas les bâtiments des images à fusionner de plus d'un seuil de tolérance, fixé par l'utilisateur. Ce critère est défini comme la somme des différences de niveaux sur les bords du pavé. De plus, on s'arrange pour préserver aussi la continuité du travail antérieur à chaque étape de la fusion, et pas seulement la topologie des images initiales. Attention : cette précaution ne garantit pas que tous les domaines blocs appartiennent à des zones non découpées dans l'image finale. Mais c'est une première limitation des intersections.

En jouant sur les bornes minimum et maximum et de la taille des pavés, on modifie la topologie du modèle généré en autorisant plus ou moins de discontinuités. On observe en Figure 9 les premiers résultats de cette technique qui assure un bon brassage des gènes des modèles croisés.

On constate le rôle déterminant des bornes (pav_{min} , pav_{max}) et du seuil de tolérance S sur le tissu final engendré et ses cassures topologiques. Lorsque S tend vers 255 avec pav_{min} assez petit, on obtient un mélange assez uniforme des deux tissus, respectant assez la topologie. Quand S décroît, pav_{min} étant toujours petit, on perd peu à peu le respect de la topologie. Et lorsque S est bas et pav_{max} élevé, il y a de nombreuses ruptures topologiques, mais en contrepartie l'apparition de bâtiments nouveaux.

AMELIORATIONS A ENVISAGER :

- injecter des pavés de blocs vides ;
- augmenter le nombre de tissus en input.

5. Quelques résultats et applications

5.1. Présentation des modèles pour les tests

Un jeu de deux villes réelles, St Genis et Vénissieux de la banlieue de Lyon (Figure 7), ainsi qu'un jeu d'images de synthèse (Figure 6) ont été utilisés pour tester les algorithmes. Bien que les IFS s'affranchissent des dimensions, les proportions ont été respectées entre les deux jeux de tests, quant au rapport hauteur / dimensions planes. Leurs tailles sont respectivement de 2048 x 2048 pixels et 256 x 256 pixels.

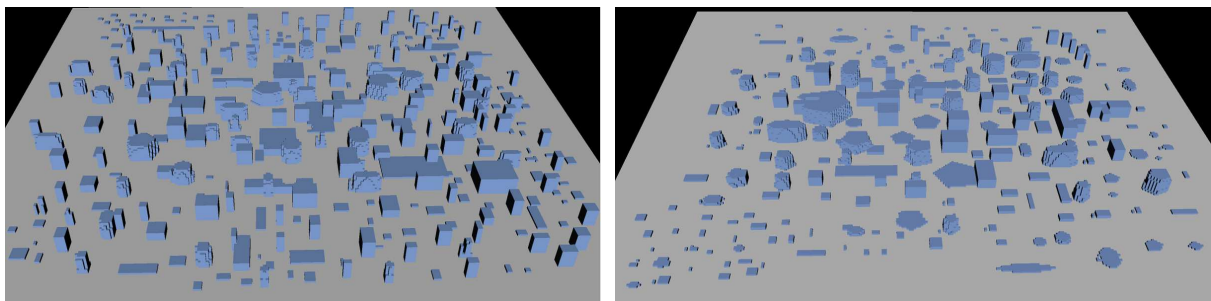


Figure 6 : jeu de tissus urbains tests synthétiques, en 3D

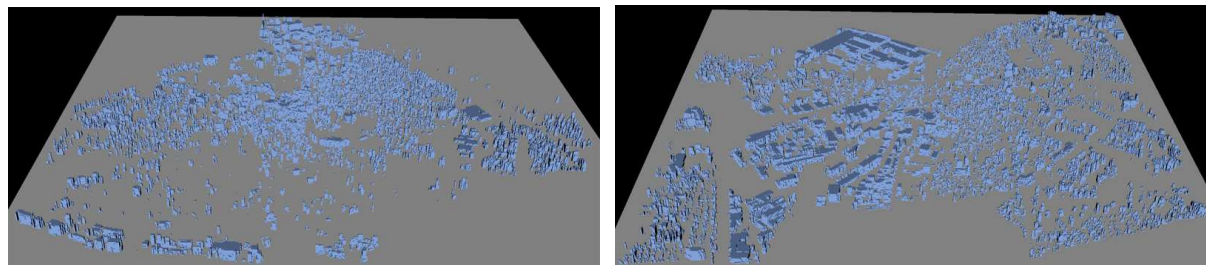


Figure 7 : jeu de tissus urbains tests réels, visualisés en 3D

Dans toute la suite, les résultats des calculs seront visualisés en 3D. On a choisi quelques exemples représentatifs parmi des centaines de modèles qui ont été calculés en faisant varier les paramètres de blocs, les configurations spatiales des pavés et les coefficients d'interpolation, et qui seront bientôt disponibles sur le site du laboratoire MAP-ARIA : www.aria.archi.fr. Les modèles ont été recadrés afin d'être plus lisibles.

5.2. Quelques exemples d'interpolations pseudo-convexes

La figure 8 montre l'évolution pseudo-continue des modèles 3D reconstitués après interpolation par IFS des deux tissus urbains choisis. La cas limite apparaît pour $\lambda=0.50$ où seulement quelques objets dominant, la plupart des autres ayant disparu, ou étant à l'état de traces. Les effets de cette interpolation par IFS seront étudiés plus en détail prochainement.

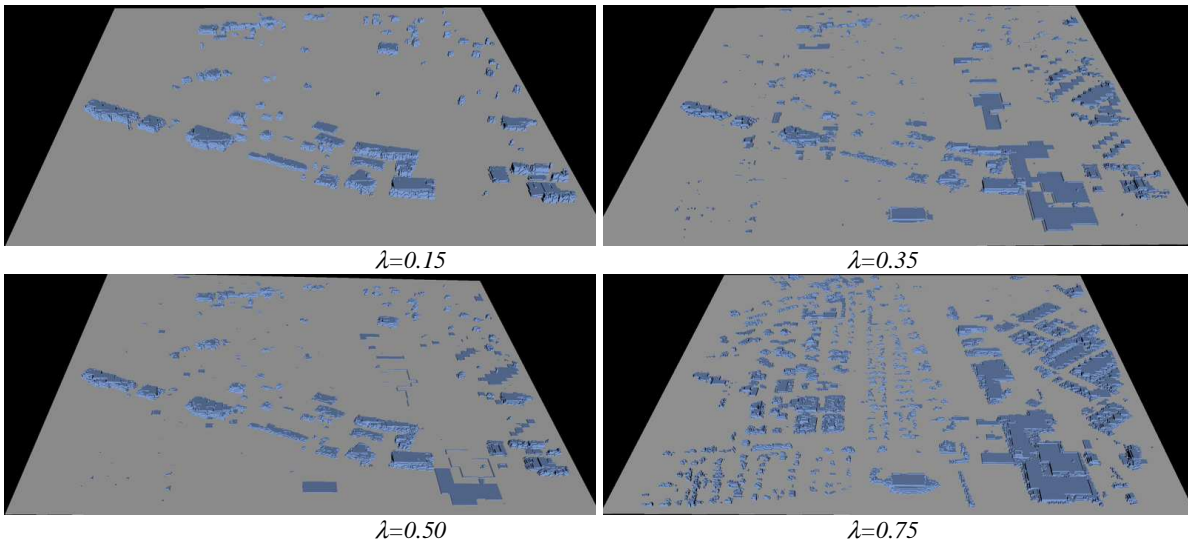


Figure 8 : jeu de 4 interpolations entre les modèles de villes réelles

5.3. Quelques exemples de fusions

La fusion par pavés décrite en 4.4 donne des résultats graphiques plutôt satisfaisants, et fournit des modèles d'objets (locaux) originaux, bien que s'appuyant sur des topologies urbaines (globales) existantes. Il n'y a qu'à observer l'image de gauche de la figure 9 et comparer avec les modèles originaux de la figure 6 pour s'en convaincre. On observe des résultats qualitativement identiques avec les images des villes réelles. Il est important de souligner ici que le nombre d'images intervenant dans ce type de fusion n'est pas limité a priori.

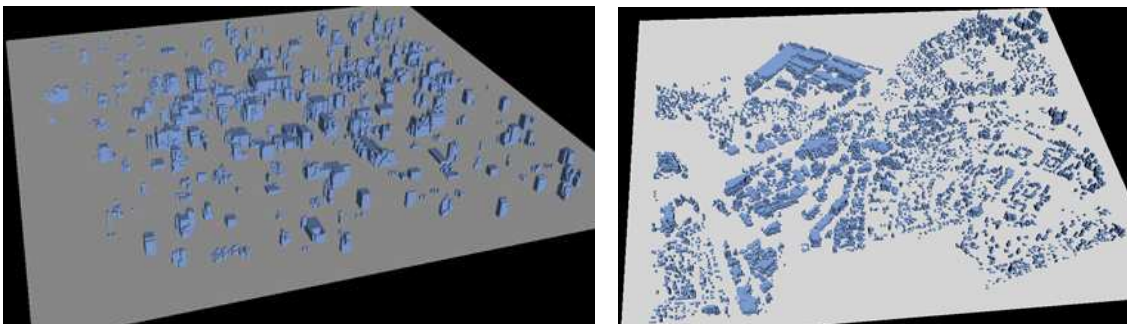


Figure 9 : essai de fusion par pavés sur villes synthétiques (gauche) et réelles (droite)

La figure 10 montre l'effet de la fusion asymétrique : on observe deux images réellement distinctes l'une de l'autre, mais on devine des traits de l'image 1 à droite et de l'image 2 à gauche. Les résultats obtenus sont meilleurs avec une taille $B=8$ qu'avec $B=4$, ce qui est normal, puisqu'on cherche des appariements, et que ceux-ci auront d'autant plus de sens que les motifs mis en correspondance représentent de l'information pertinente, donc des blocs de taille supérieure.

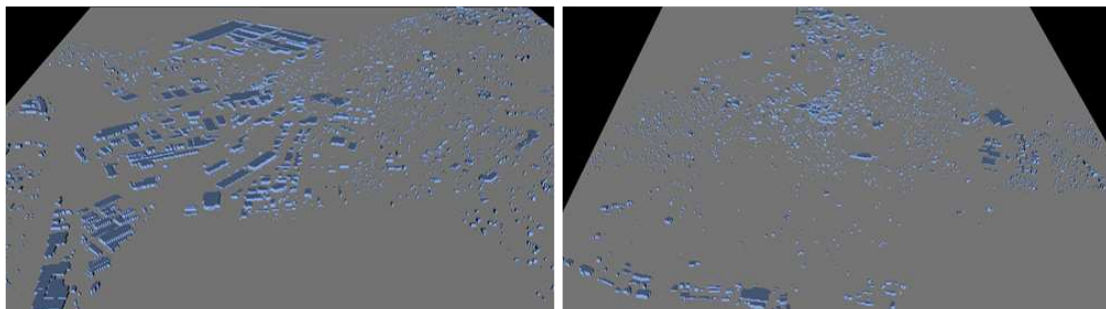


Figure 10 : essai de fusion asymétrique sur villes synthétiques

5.4. Zoom fractal et niveaux de détails

Comme les dimensions ne sont pas codés dans les IFS, on peut reconstruire mathématiquement le modèle à n'importe quelle échelle. Le zoom fractal en découle : il permet de générer par calcul à n'importe quelle échelle du détail géométrique en provenance du codage de l'échelle d'origine (Fig.11). Inversement, on peut aussi réduire un modèle. Dans le cas présent, une application intéressante qu'on propose est la génération des niveaux de détails (LOD) continus pour les objets fractals pseudo-urbains.

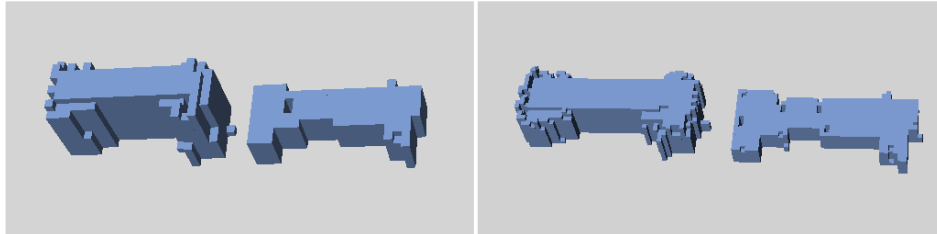


Figure 11 : effet du zoom fractal sur deux bâtiments

5.5. Filtrage fractal

Le codage des range blocs à partir de domain blocs étant surjectif, certains domain blocs sont plus utilisés que d'autres. On a calculé les images de la fréquence d'utilisation des domain blocs et du compteur d'appels des pixels des blocs domain. Bien sûr, la taille des range blocs influe sur la densité de remplissage de ces images, dans la mesure où plus elle est élevée, moins il est facile de trouver des domain blocs communs, et inversement. Elles montrent souvent qu'un petit nombre de domain blocs (et donc de pixels) de l'image sont utilisés par l'IFS lors du codage. D'où l'idée d'établir une classification des pixels de l'image en fonction de leur appartenance à ces blocs, selon leur fréquence d'utilisation dans le codage, puis de ne retenir que les plus représentés. On opère ainsi un filtrage de l'image dans l'espace des domain blocs. On préférera considérer les pixels et non les domain blocs, un pixel pouvant appartenir à plusieurs domain blocs. Le filtrage passe-bas s'opère ainsi : après avoir choisi une fréquence de coupure f_c , on met au niveau du sol (niveau 255) tout pixel dont la fréquence est supérieure à f_c . Une fréquence de coupure élevée élimine de petits détails de l'image. Lorsqu'elle diminue, on tend vers l'image des blocs non codés par *lifs*. Les résultats obtenus par filtrage passe-bas sont nettement plus intéressants qu'avec un filtrage passe-haut, et sont présentés en figure 12.

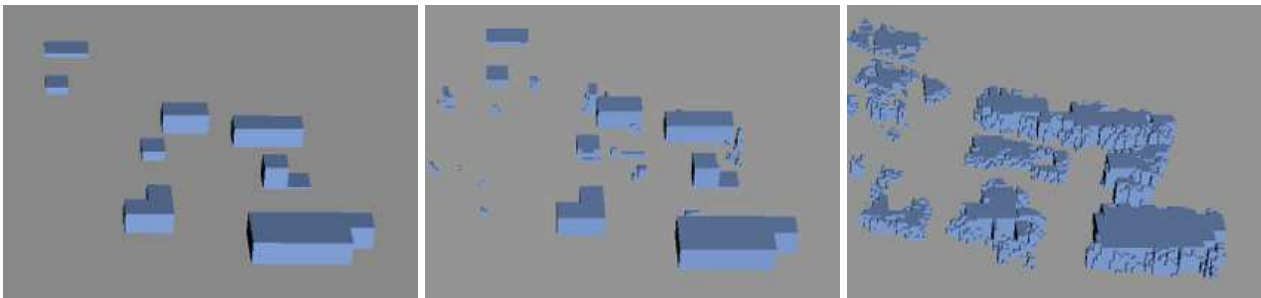


Figure 12 : Filtrage passe-bas par IFS ($f_c=1$; $f_c=10$; $f_c=100$)

5.6. Remarque générale

On obtient avec tous les traitements décrits des modèles orthomorphes dus à l'approche cubique de l'écriture en 3D de pixels ou de blocs de pixels connexes sur trame carrée. Les objets ressemblent à des empilements de blocs parallélépipédiques (voir figure 11). Des références actuelles en matière de modélisation de villes utilisent de tels objets, comme cela est décrit dans [Parish Y., Muller P., 2001], où les bâtiments sont composés par des L-systems. On pourra enfin remarquer que les célèbres « architectones » de Malevich (Figure 13) sont des modèles orthomorphes du type de ceux que l'on calcule ici, exceptée la symétrie.

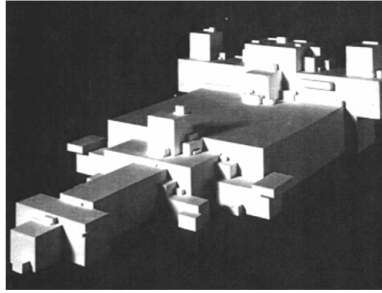


Figure 13 : maquette d'un des célèbres architectones de Kasimir Malevich (1926)

6. Conclusions et perspectives

6.1. Analyse critique des résultats obtenus

On a montré comment coder approximativement un modèle géométrique simplifié ($2D\frac{1}{2}$) d'une ville réelle ou synthétique par des IFS, en utilisant une méthode de compression d'images dérivée de celle de Jacquin. Le découpage d'un plan urbain par partitionnement présentait l'intérêt de pouvoir agir localement sur le contenu de petites zones. On s'est intéressé au partage d'informations provenant des IFS codant ces zones, dans le but d'obtenir de nouveaux modèles (par fusion), ou des modèles intermédiaires (par interpolation). Le fait de pouvoir opérer aussi sur des images synthétiques ouvre des possibilités de création immenses, et plusieurs idées ont été proposées pour cela. Notamment une approche de type génétique a été initiée. Au final, on a obtenu des modèles géométriques $2D\frac{1}{2}$ de tissus pseudo-urbains, avec des niveaux de détails fractals. Même une simple reconstruction du modèle original est capable de modifier les objets de la scène, en leur donnant un aspect fractal qu'ils n'avaient pas.

Parmi les difficultés rencontrées, on n'a pas considéré jusque-là de partitionnements adaptés aux objets morphologiques propres aux villes, ce qui fait que l'identité topologique des bâtiments est souvent rompue dans toutes les opérations proposées, exception faite (ou presque) de la technique de fusion par pavés. Dans ce cas, justement, on obtient théoriquement à partir de deux images initiales de tissus urbains un nombre immense de modèles qui, bien que ressemblant quant à leur distribution spatiale aux deux topologies initiales, sont en fait peuplés de bâtiments tout à fait originaux, dont les détails géométriques sont le fruit du croisement des *lifs*.

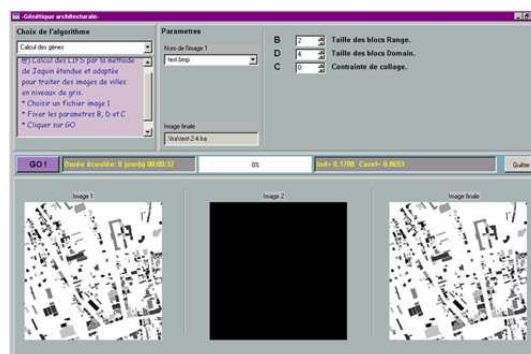


Figure 14 : interface graphique du programme de calcul

Une autre difficulté concerne l'ajustement de la taille B des « blocs range ». Elle doit être au moins de l'ordre de grandeur moyen des bâtiments, si l'on veut éviter les ruptures topologiques. Par ailleurs, il est normal d'adapter les blocs codants à la taille des objets qui ont du sens : habitations, quartiers... Si B est trop petit, on ne code que la frontière des objets. D'autre part, la précision du codage implique une résolution d'image suffisante pour décrire chaque bâtiment avec assez de détails. Ces deux contraintes vont dans le même sens : une très forte augmentation des temps de calcul des IFS. Pour l'éviter, on peut opérer sur des images de taille inférieure (donc sur des parties des images originales), ou bien diminuer la résolution pour conserver un paramètre B petit, adapté à la taille moyenne des objets. C'est ce que nous avons fait pour les deux villes réelles du jeu de test. Mais la meilleure solution consistera à poursuivre des recherches concernant les partitionnements adaptatifs au contenu des images, et les conséquences pour la mise en œuvre du codage et des croisements d'IFS.

6.2. Directions d'étude et perspectives envisagées

Plusieurs voies restent encore à explorer. D'un point de vue scientifique, il manque à cette étude :

- des expérimentations avec des algorithmes génétiques pour optimiser les fusions basées sur des critères liés aux villes réelles (par exemple, en se donnant des paramètres sur les alignements de bâtiments, sur des distributions connues d'un type de bâtiment ou d'habitation...).
- une approche par « IFS à condensation » : comment injecter de l'information extérieure dans les blocs, non codée par IFS, de manière à conserver des propriétés particulières des modèles générés ?
- une analyse de robustesse : une fois entamées les deux phases précédentes, il s'agira de tester les algorithmes dans des conditions particulières d'utilisation (images tests très simplifiées ou l'on s'attend à un résultat précis, ou au contraire images de natures totalement différentes), et de voir comment ils se comportent. Par exemple, avec deux images initiales choisies parmi : « Lenna » (le fameux modèle de test des chercheurs) + ville de synthèse, cercles concentriques + damier, ville concentrique + ville en damier.

Au niveau des formes produites et des modèles 3D, on envisage :

- d'effectuer une correction des images avant extrusion : traitements de lissage sur les objets générés, notamment pour éliminer l'effet de marche d'escalier de certains contours, et obtenir des formes géométriques plus régulières et plus faciles à texturer.
- de remplacer les pavés carrés dans le programme de fusion par des ensembles de régions connexes préservant la continuité des objets. On choisit ainsi comme région admissible un groupe de blocs range contenant totalement un ou plusieurs bâtiments.
- de proposer une méthode de mise en correspondance des objets générés avec diverses bibliothèques d'objets existants. Au lieu d'utiliser les objets tels qu'ils sont produits, avec les corrections éventuelles que cela suppose, on opère une substitution des édifices de notre choix, en respectant la contrainte des hauteurs et l'empreinte 2D.
- d'implémenter un moteur de calcul automatique de voiries basé sur les opérateurs de la « morphologie mathématique » (un algorithme déjà écrit pourra être fourni par le laboratoire iMAGIS, membre du projet DEREVE). On pourra se servir d'un calcul de densité du bâti, ou de la densité de population qui est fonction de la hauteur des bâtiments et des habitations.

7. Références

- [Barnsley M., 1993] Fractal image compression, AK Peters, Ltd, Wellesley.
- [Barnsley M., 1992] Image coding based on a fractal theory of iterated contractive image transformation, IEEE transactions on image processing, vol 1, n°1, pp18-30.
- [Batty M., Longley P.A., 1994], Fractal Cities: A Geometry of Form and Function, Academic Press, London and San Diego, CA.
- [DER 2000] Projet de recherche DEREVE de la région Rhône-Alpes, Développement d'un Environnement logiciel de REalité Virtuelle Elaboré, LIGIM, Université Lyon I, avril 2000.
- [Frankhauser P., 1994] La Fractalité des Structures Urbaines, Collection Villes, Anthropos, Paris, France.
- [Frankhauser P., 1997] L'approche fractale : un nouvel outil de réflexion dans l'analyse spatiale des agglomérations urbaines, Université de Franche-Comté, Besançon.
- [Gentil C., 1992] Les fractales en synthèse d'images : le modèle IFS, Thèse, LIGIM, Université Lyon I, LYON.
- [Hutgen B., Hain T., 1994] On the convergence of fractal transformes, Proceedings of ICASSP, p 561-564, 1994.
- [Hutchinson J., 1981] Fractals and self-similarity, Indiana University Journal of Mathematics, 30:713-747, 1981.
- [Jacquin A.E., 1992] Image coding based on a fractal theory of iterated contractive image transformations, IEEE transactions on image processing, 1(1):18-30, January 1992.
- [Makse H.A., 1996] Modeling fractal cities using the correlated percolation model, Session C18, *Fractal and granular media conference*, March 1996.
- [Parish Y., Muller P., 2001] Procedural modeling of cities, SIGGRAPH 2001.
- [Renders J.M., 1995] Algorithmes génétiques et réseaux de neurones, Editions Hermès, avril 1995.

[Torrens P., 2000] How cellular models of urban systems work, CASA, Angleterre.

[Vences L., Rudomin L., 1997] Genetic algorithms for fractal image and image sequence compression, Instituto Tecnológico de Estudios Superiores de Monterrey, Camus Estado de Mexico, Comptation Visual 1997.

[Woloszyn P., 1998] Caractérisation dimensionnelle de la diffusivité des formes architecturales et urbaines, Thèse, Laboratoire CERMA, NANTES.