

Title	ルールベースの図書館シーンの自動生成
Author(s)	北, 直樹; 宮田, 一乗
Citation	第9回NICOGRAPH春季大会 論文&アート部門コンテスト 予稿集: SII-2
Issue Date	2010-3-26
Type	Conference Paper
Text version	author
URL	http://hdl.handle.net/10119/11425
Rights	Copyright (C) 2010 芸術科学会. 北直樹, 宮田一乗, 第9回NICOGRAPH春季大会 論文&アート部門コンテスト 予稿集, 2010, SII-2.
Description	

ルールベースの図書館シーンの自動生成

北 直樹¹⁾ 宮田 一乗²⁾

1) 北陸先端科学技術大学院大学 知識科学研究科

2) 北陸先端科学技術大学院大学 知識科学教育センター

1. はじめに

近年、映画やコンピュータゲームにおけるコンテンツ制作において、コンピュータグラフィクス(以降、CG を記す)が多く用いられている。ハードウェアの性能向上に伴いCG の表現力が増大した一方で、CG で制作するオブジェクトの規模やディテールも膨大かつ詳細なものが求められ、それらを手作業で制作すると開発コストが大きくなる。そこで作業の効率化のため、コンピュータを用いたコンテンツの手続き的な生成や自動生成などの技術が重要になる。

本論文では、ルールベースの文法を用いた本棚／大規模図書館シーンのモデリング手法を提案する。ルールを追加していくことで従来の手続き的な技法以上に、様々なバリエーションのモデルの生成が容易になる。また、図書館をはじめとする人工の建造物によく見られる対称性や反復構造、加えて表現対象物の階層構造を考慮した手法を採用することで、さらに多様なバリエーションのモデルが生成可能であることを示す。このような生成手法は特に、大規模なモデルの生成に有効であると考えられる。

2. 関連研究

本章では、本研究に関連する研究について紹介し、本研究の位置づけを明らかにする。

2.1 文法ベースのモデリング

文法を用いたモデリングは自然形状の生成から建造物の生成まで様々な対象に用いられている。L-System[1]は植物の生成手法として広く用いられている。L-Systemでは初期状態に置換規則を再帰的に適用していくことで自己相似的な図形を生成することが出来る。City Engine[2]ではこのL-Systemを拡張することで道路ネットワークの生成に応用している[3]。また、ビルモデリングにはSplit Grammars[4]やCGA shape[5]が用いられる。Split Grammarsは与えられた図形を再帰的に分割していき最終的に3Dモデルを生成するデザイン文法であり、ビルの詳細な形状の生成に用いられる。

これら、文法を用いたアプローチでは、パラメータを変更したり、生成ルールに重み付けをして確率的にルールを採択することで生成されるモデルにバリエーションをつけることが可能となる。

2.2 自動生成

自動生成では文法ベースのモデリングと同じく、各種制御パラメータを変更することで様々なバリエーションのモデルが生成可能である。野村らは大規模工場の景観の自動生成手法を提案した[6]。この手法では建造物全体のシルエットを手書きで入力することで工場の概形を、そしてパラメータにより工場を構成する各部位のレイアウトを制御することができ、生成される工場にバリエーション付けすることが出来る。

2.3 自由文脈デザイン文法

本研究では実装にあたり、Structure Synth[7]を用いた。Structure Synthでは一般的なプログラミング言語とは異なり、自由文脈デザイン文法(Context Free Design Grammar, 以降CFDGと記す)をベースとしたスクリプト言語EisenScriptにより3DCG作成を行う。CFDGはChris Coyneによって形式言語の文法を拡張しCGに応用出来るようにしたものである。CFDGの実装としてはContext Free Art[8]がある。Context Free Artは形式言語を2次元に拡張したものであるが、Structure SynthはContext Free Artを3次元に拡張したものである。

2.4 本手法の位置づけ

本手法ではルールをベースとしたモデリングを行う。これまでのルールベースのモデリングが対象物に注目してモデリングを行うアプローチを採っているのに対し、本手法では対象物、特に人工物によく見られる反復構造や、表現対象物の階層構造を考慮したモデリング手法を提案する。また、これまでのルールベースのアプローチでは、パラメータ変更や生成ルールの重み付けにより生成されるモデルのバリエーション付けを行っていた。一方、本手法では反復構造の生成ロジックを制御し、適用されるルール自体を別ルールで代替することによるバリエーション付けについても提案する。

本研究では、本手法を用いたモデリング対象として図書館シーンの生成を行う。図書館において、反復構造は様々なところに見てとれる(図1)。局所的に見れば本棚における本の配列、また大局的に見ればフロアにおける本棚の配置などである。これは配架の際、分類ごとに配列や区画をまとめる必要があるためと考えられる。ただし本研究では見た目にもっともらしいCGの生成を目的としており、実際の配架に則しているかどうかは

考慮の対象外とする。



図1. 図書館における反復構造.

3. 図書館シーンのモデリング

本章では図書館シーンのルールベースのモデリング手法について述べる。

3.1 アルゴリズムの概要

図書館シーンを生成するアルゴリズムは、以下の手順で構成される:

1. 本棚の作成.
2. 本の作成.
3. 本棚に本を配列する.
4. 本棚をフロアに配置する.

図書館シーンの生成を行うため、まず本棚のモデリングを行う。本研究では大規模な図書館シーンの生成を目標としているため、3において本棚に本を配列する際には本同士、そして本棚と本の衝突検出は行わない。4においても本棚とフロアとの衝突検出は行わない。本手法では、衝突検出を行わずに見た目にもっともらしいモデルの生成を行うために、複数冊の本をルールとしてまとめることでこれを解決する。

3.2 本棚の作成

まず、本棚の作成を行う。本棚は外枠と棚板から構成されており、例えば7段組みの本棚の構造をルール化すると次のようになる(図2):

```
rule bookshelf {  
  shelf_frame  
  7 * { y height } shelf_board  
}  
  
rule shelf_frame { ... }  
rule shelf_board { ... }
```

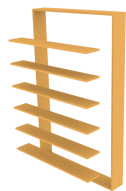


図2. ルールによる本棚の作成例.

ルールは最終的にプリミティブ(Box等)と並進移動, 回転移動等

の移動操作へ変換される。本棚の例では外枠と棚板のルールがBoxから成っており、図2の例では、棚板をy軸にheightだけ平行移動を7回繰り返すことで作成する。

3.3 本の作成

次に、本の作成を行う。本は中身とカバーから構成されており、本棚同様、以下のようにルール化出来る(図3):

```
rule book {  
  pages  
  cover  
}  
  
rule pages { ... }  
rule cover { ... }
```



図3. ルールによる本の作成例.

本の項は白色、カバーは予め用意したカラープール(カラーパレット)からランダムに選んだ色を用いる。本手法ではこのように、要素間の関係をルールとして記述することでモデリングを行う。

3.4 本の本棚への配列

次に、本棚に本を配置する。実際の本棚においては、本は必ずしも整理されて配列されているとは限らず、他の本に寄りかかる形で斜めになっている本があったり、横積みになっていたりすることが考えられる。本研究では最終的に大規模な図書館シーンの生成を目標としているため、計算の効率を考慮して本同士、そして本と本棚との衝突検出は行わない。しかし、上記のような配列を衝突検出を行わずに実現するためには、何らかの制約を加える必要がある。そこで本手法では複数冊の本をまとめて扱うことで衝突の問題を回避し、また様々な配置方法を実現する。

ここで、数冊の本をまとめたものをロットと呼ぶことにする。いくつかのバリエーションのロットを作成し、それらのロットの中から2組選んで組み合わせたものをカップリングとする。そして、そのようにカップリングしたロットを本棚に配列する。カップリングは、重み付けしたロットのルールを確率的に選択することで行う。以降、ロットの作成、ロットのカップリング、そしてカップリングしたロットの本棚への配置方法について述べる。また、そのレンダリング結果を示す。

3.4.1 ロットの作成

本の配列の問題を解決するために、本研究では複数冊の本をロットというかたちにまとめることを考える。ロットのパターンとして、次の(4a)から(4d)のルールを作成する(図4):

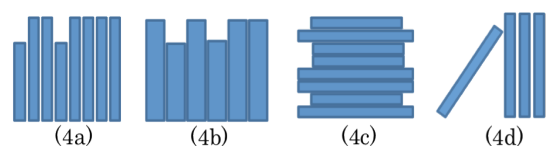


図4. ロットのバリエーション.

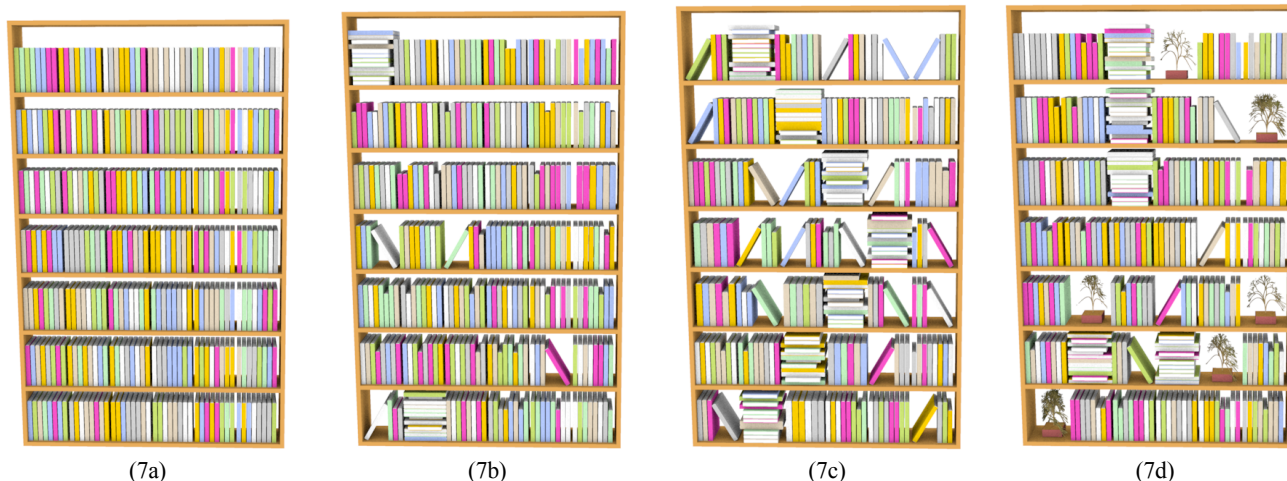


図7. 本棚のレンダリング結果.

(4a), (4b)は普通に配列されたロットであり, (4b)に関しては本の厚みが(4a)とは異なる. (4c)は横積みみのロットである. 本の背表紙が水平面から見て前後左右のどの方向を向くかでランダムに選択し積み上げていく. そして(4d)は斜めになっている本があるロットである. また, 本の高さに関しては2種類のルールを用意し, それを重みに応じて確率的に採択することでバリエーションを出す. ロットのルールは本を繰り返し平行移動したり, 回転移動することで構成される. また, 各ロットのルールには重み付けを行う. これにより生成される3Dモデルにバリエーションを持たせることが可能となる.

3.4.2 ロットのカップリング

ロットのカップリングは各ロットのルールから2つのルールを重み付けに従い確率的に選択し, それらをルールとしてまとめることで行う. 例えば(a)と(b)のカップリングを{(a), (b)}と書くことにする. 図5は{(4a), (4c)}, {(4d), (4a)}のようにカップリングした例である(図5):

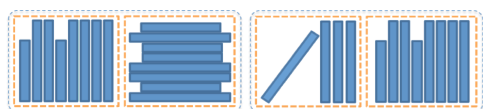


図5. ロットのカップリング例.

3.4.3 カップリングしたロットの本棚への配置

カップリングしたロットの本棚への配置は, 1段にカップリングを3つ並べ, 本棚の段数分積み上げることで行う(図6). このように本, ロット, カップリング, 本棚といったある程度のまとまりでルール化していくことにより, 様々な階層レベルでのバリエーション付けが可能となる. 例えば, 1つのロットを他のルールに置換することで, より細かな部分の変更を, そしてカップリングや配置方法を変更することで大きく印象の異なったモデルも生成可能であると考えられる.

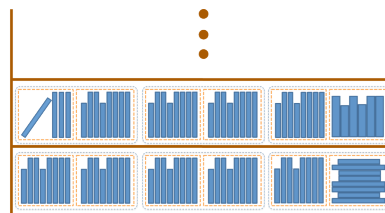


図6. カップリングの本棚への配置例.

3.4.4 本棚のレンダリング結果

本棚のレンダリング結果を図7に示す. ここで, (7a)は(4a)のロットで, かつ本の高さを1種類に限定したもののみを用いてレンダリングしたものである. (7b)は(4a)~(4d), そして(4d)のルールで斜めに寄りかかっている本の向きを反転させたもの(4d'と記す)をそれぞれ15:1:1:1の出現比率でルールに重み付けしてレンダリングしたものである. また, (7c)は(4a)から(4d')のルールに等確率の出現比率で重み付けをしたものである.

さらに, 図書館では見られないが例えば個人で所有する本棚などでは, 本以外のものが棚にディスプレイされていることがある. ここではそれを再現するために植物(ここでは盆栽ロットとした)を生成するルールを追加した. 幹の成長方向を確率的に選択することで, 1つのルールから様々なモデルを生成することが可能となる(図8):

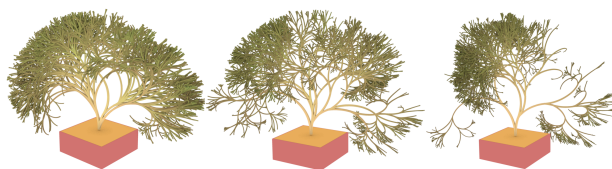


図8. 盆栽ロットとそのバリエーション.

盆栽ロットは他の本ロットと同様に扱われ, カップリングされて本棚へ配置される. (7d)はそのレンダリング結果である. このように, ルールを追加していくだけで容易に生成される3Dモデルにバリエーションをつけることが可能である.

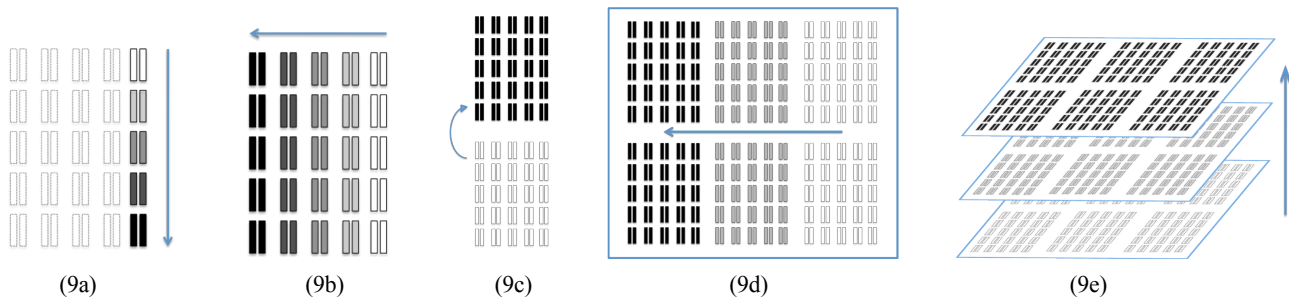


図9. 本棚のフロアへの配置アルゴリズム.

3.5 本棚のフロアへの配置

次に、作成した本棚を用いて図書館シーンを生成する方法について述べる。本研究では、大規模な図書館シーンを生成するために、次のアルゴリズムに従って本棚をフロアに配置していく(図9):

1. 背中合わせにした1組の本棚をN回繰り返し配置し、区画の列を作成する(9a).
2. (9a)の列をM回繰り返し配置し、区画を作成する(9b).
3. (9b)を複製し、(9c)のように対称的な配置になるように回転移動する.
4. フロア全体を覆うように(9c)を繰り返し平行移動しながら配置する(9d).
5. (9d)を積み上げることで図書館シーンとする(9e).

ここで、本報告では $N=5$, $M=5$ とした。(9a)から(9e)の手順をルール化することで大規模図書館シーンを生成する。このシーンの生成結果は、中央の通路に対して線対称に本棚の区画が並んでいるフロアを各階に持つ図書館シーンとなる。このように、モデリング対象をルールとして作成するだけでなく、繰り返しや移動などの構成要素の配置の制御構造もルールにする。そして、そのルールを制御することで、制御構造を制御することが容易になり、生成される3Dモデルに多彩なバリエーションを与えることが可能となる。このことについては次章で詳述する。

4. ルール制御によるバリエーション付け

本章では、対象物のルール、あるいは制御構造のルールを変更したり制御することで、生成される3Dモデルにバリエーションを与える方法について述べる。

本研究では対象物をルール化したものとして、本棚ルールや盆栽ルールを作成した。本棚に配列される本はロット、カップリングというように、いくつかの階層レベルでルールとしてまとめることで棚に配置される本にバリエーションを与えることが可能となった(7a-7c)。また、ルールを追加することでも容易にバリエーションが与えられることを示した(7d)。そして、(9a)から(9e)では、制御構造をルール化した。本章では、制御構造をルール化することで多様なモデルが生成可能であることを、(9a)、(9c)、(9d)の各ルールの制御による具体例で示す。

4.1 区画の列単位のルール制御

(9a)は区画を構成する列を、本棚を反復的に繰り返し配置することで生成するルールである。(9a)における最初の本棚のみの対象物に変更することを考える(図10左)。変更の結果、フロアには中央の通路にそった本棚のみが別の対象物(ルール)に変更されたモデルを作成することが可能となる(図10右)。

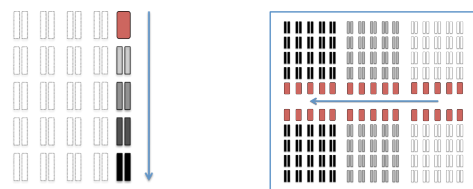


図10. (9a)のルール制御によるバリエーション1.

同様に、(9a)の反復の末尾の2つを別のルールに置き換えることで生成されるフロアに変化を与えることも可能である(図11)。

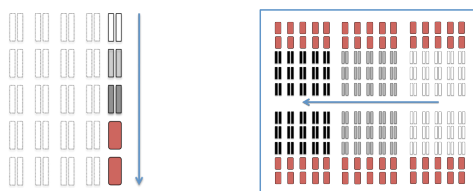


図11. (9a)のルール制御によるバリエーション2.

4.2 区画単位のルール制御

(9c)は区画を複製したものを通りに対して線対称に配置するルールである。ここでは、(9b)で作成した区画を複製する代わりに、別途作成したルールを通路に対して線対称に配置する(図12左)。その結果、生成されるフロアは4.1に比べてより大局的に変更されたモデルとなる(図12右)。

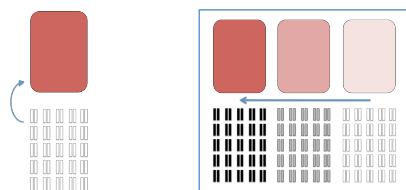


図12. (9c)のルール制御によるバリエーション.

4.3 フロアへの配置に対するルール制御

(9d)は(9c)のルールを反復的に繰り返すことでフロア全体を覆うルールである。ここでは、(9a)から(9c)の手順で別のルールを作成する(9a'-9c')。そのようにして作成した別ルール(図13左)に、(9d)において、繰り返しの末尾のみ(図13左)を適用することを考える。すると、結果としてフロアの3分の1のみ別の対象物に置き換えられたモデルを作成することが可能となる(図12右)。

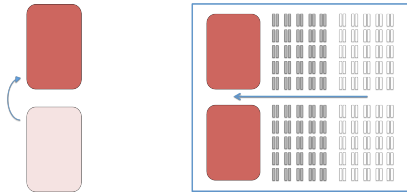


図13. (9d)のルール制御によるバリエーション。

4.4 実装方法

ここでは、上記のルール制御をどのようにスクリプト上で実装するかについて述べる。図9の(9a)から(9d)のアルゴリズムをEisenScriptの文法の書き方で疑似コード化すると次のように書ける：

```
rule RuLe9d maxdepth 3 {  
  Rule9c  
  { x方向に平行移動 } Rule9d  
}  
rule RuLe9c { 2 * { y軸に対して180度回転して平行移動 } Rule9b }  
rule RuLe9b maxdepth 5 {  
  Rule9a  
  { x方向に平行移動 } Rule9b  
}  
rule RuLe9a maxdepth 5 {  
  pair_shelf  
  { z方向に平行移動 } Rule9a  
}
```

ここで、Rule9aからRule9dは(9a)から(9d)をルール化したものである。また、パラメータのmaxdepthは再帰の回数を指定するものであり、Rule9a中のpair_shelfは(9a)で背中合わせにした本棚のルールである。ルール制御を行うには、EisenScriptのRule retirementを用いる。これは、あるルールがmaxdepthにより再帰の末尾に到達した際、別ルールを代替として使うことを可能とするものである。これにより例えば、図10で示した区画の列単位のルール制御は次のように書くことができる：

```
rule RuLe9a maxdepth 4 > yet_another_bookshelf { ... }
```

Rule9aのmaxdepthを4回にして、ルールを代替するための指定記号>を使用し、yet_another_bookshelfで5回目を代替する。これによりルールを別のルールで代替することができ、ルール制御が可能となる。すなわち、この最大再帰回数の制御、ルールの代替により、手間をかけずに生成モデルのバリエーション付けを行うことができる。

5. 結果と考察

本章では、大規模図書館シーンと前章で示したルール制御によるバリエーション付けのレンダリング結果を示す(図14、図15)。図14は図9で示した配置アルゴリズムにより作成したシーンのレンダリング結果である。シーンの生成にあたり、簡単な外装と家具も作成した。大規模なシーンであっても、本棚に並ぶ本にバリエーションが確認出来る。単純な反復による配列に見られるアーティファクト(人為的作用による配置の周期性)が除去された、よりもっともらしいシーンが生成されたと言える。

また、図15は前章で示した図10から図13のレンダリング結果である。(15a)では、区画を構成する本棚の列の反復の最初を家具ルールで代替することで中央の通路側の本棚がひとつ減り、代わりに家具が通りに沿って配置されているシーンを生成した。(15b)では逆に、反復の末尾の2回を別の本棚のルールで代替することで異なったシーンを生成した。(15c)では通りに対して線対称を成すように本棚の区画を回転複製する際に、複製ではなく別ルールにより構成した区画を配置することで、フロアの半分に読書機のコーナーを作成した。ここでは、別途読書機コーナーのルールを作成し、それを利用した。(15d)はフロアを覆うように区画を反復的に配置する際、反復の最後を読書機コーナーのルールで代替することで、フロアの向かって奥に読書機のコーナーを作成した。

6. まとめ

本稿ではルールベースのモデリングによる本棚、大規模な図書館シーンの生成手法を提案した。ルールの追加や重み付けを制御することで、多様なパターンのモデルが生成可能となる。また、対象物をルール化するだけでなく制御構造もルール化することで、反復構造や対称性といった人工物によく見られるパターンを利用した生成モデルのバリエーション付け手法も提案した。

本論文では、直線的な反復構造の制御について示したが、曲線や、その他のパターンを利用したバリエーション付け手法を取り入れることでさらに多彩なバリエーションが実現可能である。また、本研究ではEisenScriptで実装を行ったが、EisenScriptは汎用プログラミング言語が備えているべきフロー制御構造や条件分岐を持たないなど、その言語構造からして、思い描く最終形を意図したモデリングに向いているとは言い難い。EisenScriptやCFDGが主として用いられるのは、意図しない結果が描かれることを期待したアート制作の場である。より高度な制御を行う場合には独自の実装が必要である。今回モデリング対象にした図書館においては、実際には本の配置は配架に基づいており、フロアレイアウトも配架に基づいていると考えられる。そのため、必ずしも対称性を持っている訳ではない。そのような対象物をモデリングする場合、本手法では柔軟性に欠ける。以上の点は、今後の課題として改良する必要がある。

参考文献

- [1] Prusinkiewicz, P. and Lindenmayer, A., “The Algorithmic Beauty of Plants”, Springer Verlag, 1991.
- [2] CityEngine, Procedural Inc.
- [3] Parish, Y. I. H., and Muller, P., “Procedural modeling of cities”, Proceedings of ACM SIGGRAPH 2001, ACM Press, E. Fiume, Ed., pp.301-308, 2001.
- [4] Wonka, P., Wimmer, M., Sillion, F., and Ribarsky, W., “Instant Architecture”, ACM Transactions on Graphics 22.3, 669-677, 2003.
- [5] Muller, P. Wonka, P., Haegler, S., Ulmer, A., and Van Gool, L., “Procedural modeling of Buildings”, Proceedings of ACM

SIGGRAPH 2006 / ACM Transactions on Graphics, ACM Press, New York, NY, USA, vol.25 pp.614-623, 2006.

- [6] 野村健太郎, 宮田一乗, “大規模工場の景観の自動生成”, 第8回NICOGRAPH春季大会 論文 & アート部門コンテスト予稿集 SI-2, pp.1-6, 2009.

- [7] Christensen M. H., “Structural Synthesis using a Context Free Design Grammar Approach”, Generative Art Conference, Milano, 2009.

- [8] C. Coyne, M. Lentczner, J. Horigan, Context Free Art, <http://www.contextfreeart.org/>

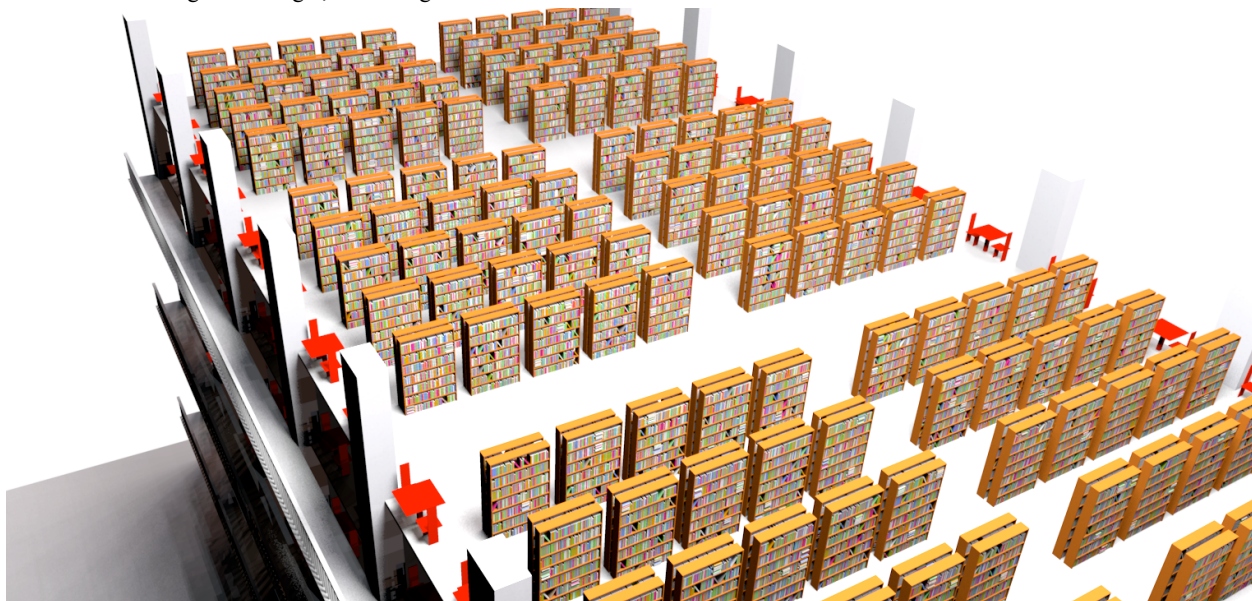
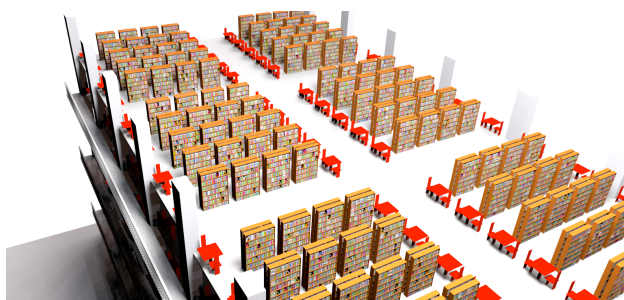
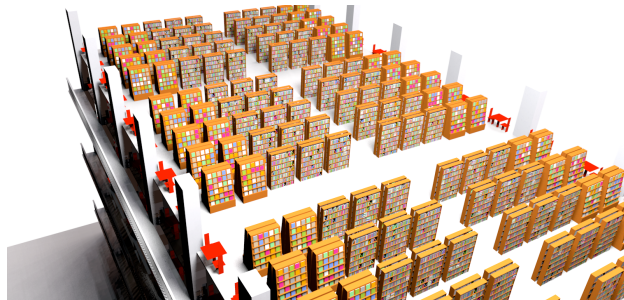


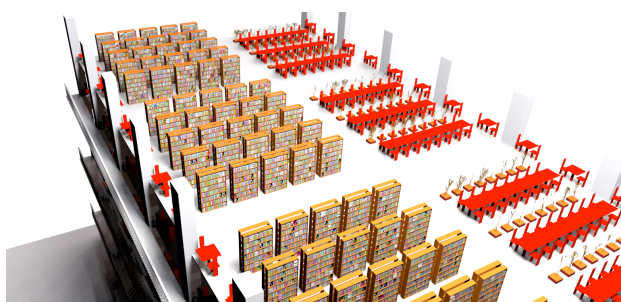
図14. 図9のレンダリング結果.



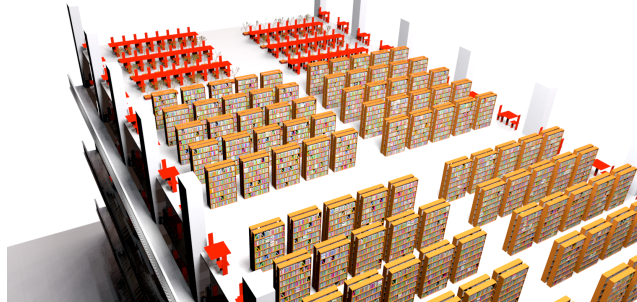
(15a) 図10のレンダリング結果.



(15b) 図11のレンダリング結果.



(15c) 図12のレンダリング結果.



(15d) 図13のレンダリング結果.

図15. ルール制御によるバリエーション.