2010年度　卒　業　論　文

ダンジョン形状の自動生成に関する研究

指導教員：渡辺 大地 講師

メディア学部 ゲームサイエンスプロジェクト
学籍番号　M0107352
成田 晃

2010 年度　　卒　業　論　文　概　要

論文題目

ダンジョン形状の自動生成に関する研究

メディア学部 **学籍番号：M0107352**	**氏名**	成田 晃	**指導教員**	渡辺 大地 講師

キーワード	自動生成, 3DCG, モデリング, ゲーム, ダンジョン, 洞窟, L-system

コンピュータゲームのなかには、ローグライクゲームのように、ダンジョン形状をランダムに生成するゲームがある。このダンジョン生成の手法は、領域を規則的に分割しているため、規則的なダンジョンの生成には適しているが自然な形状のダンジョンには適さない。

本研究では人工的で規則的なダンジョンと自然的で不規則な形状のダンジョンに対応した 3D ダンジョン形状の自動生成をおこなうことを目的とした。解決手段として、人工的なダンジョンの形状と自然的なダンジョンの形状の違いをパラメータにより制御することで作り分けることにした。これにより 1 つのアルゴリズムによって人工的なダンジョンの形状と自然的なダンジョンの形状の両方を扱うことができる。2 つの形状を 1 つのアルゴリズムで扱えるようになることで、人工的な形状と自然的な形状が混じった形状を表現できるようにした。

最後に次の 2 つの方法によって本研究の有用性を証明した。ひとつめは、人工的なダンジョンの形状と自然的なダンジョンの形状の 2 種類をパラメータを変えることにより生成した。これにより、2 つの形状を 1 つのアルゴリズムで扱えることを証明した。ふたつめに、人工的なダンジョンの形状と自然的なダンジョンの形状の 2 つの特徴を持ったダンジョン形状を生成した。これにより、人工的な形状と自然的な形状が混じった形状を表現出来ることを証明した。

目 次

図 目 次

第 1 章

はじめに

1.1 研究背景と研究目的

コンピュータゲームでの城や塔などの建造物内部や洞窟などのダンジョンの形状は、常に同じである。コンピュータゲームのなかには、ローグライクゲームのように、ダンジョン形状をランダムに生成するゲームがある。ダンジョン形状をランダムに生成することで、ダンジョン形状が同じゲームに比べ、リプレイ性が向上した。ローグライクゲームの例として “チョコボの不思議なダンジョン 2”[1] などがある。

Adams[2] は、マップのトポロジー記述の自動生成と生成したマップの難易度の評価方法を提案した。この研究はトポロジー記述による、トポロジーマップの生成が目的である。そのため実際にマップの形状生成は行っていない。

ローグライクゲームのダンジョン生成は、領域を規則的に分割し、分割した領域内に部屋を作り、部屋同士を通路でつなぐ手法を用いている。領域を規則的に分割しているため、規則的なダンジョンの生成には適しているが自然な形状のダンジョンには適さない。

本研究では人工的で規則的なダンジョンと自然的で不規則な形状のダンジョンに対応した 3D ダンジョン形状の自動生成をおこなう。人工的なダンジョンの形状と自然的なダンジョンの形状の違いは、パラメータにより制御をする。これによ

り 1 つのアルゴリズムによって人工的なダンジョンの形状と自然的なダンジョンの形状の両方を扱うことができる。2 つの形状を 1 つのアルゴリズムで扱えるようになることで、人工的な形状と自然的な形状が混じった形状を表現できる。

本研究では L-system を用いて 3D ダンジョン形状の生成をおこなう。L-system は初期条件で記号や形状を与え、それに対して繰り返し生成規則に基づき記号や形状を変化させることで、様々な自然物の構造を表現できる手法である [3][4][5]。L-system は、自然物の構造表現以外に自己相似図形やフラクタル図形などの形状表現に用いる [6]。本手法では規則的な形状と不規則な形状をパラメータ制御により生成できることから、加藤ら [7] の用いた手法を参考に L-system を用いて 3D ダンジョン形状の自動生成をおこなう。

1.2 本論文の構成

本論文の構成は次のとおりである。第 2 章ではダンジョンについて述べる。同時にどのようなダンジョン形状を研究対象とするのかについて述べる。第 3 章では本手法の 3D ダンジョン形状の自動生成手法を説明する。同時に目標とするダンジョン形状について述べる。第 4 章では本研究で扱った手法の評価を行い、考察や今後の課題と展望について述べる。第 5 章では本研究のまとめを述べる。

第 2 章

ダンジョンの概要

2.1　ダンジョンについて

コンピュータゲームにおけるダンジョンとは、コンピュータゲームの主人公であるプレイヤーが探索や冒険ができる迷路状の空間を指す。英語の dungeon とは本来、地下牢のことを指すが、一般的にはコンピュータゲームにおけるダンジョンを指すことが多い。ダンジョンには洞窟や山道などの地形や城や遺跡などの建造物など様々な種類があり、通路と部屋といった領域を持つ。本研究ではコンピュータゲームに登場する洞窟をダンジョンと呼ぶ。

洞窟は広義的に、人間が入ることができる地中にある空間のことである [8]。洞窟は狭義的に、洞窟入口の長径が奥行きまたは深さよりも短い洞窟のことを洞窟と呼び、洞窟の断面が洞窟の入口で最大になる洞窟のことを岩陰と呼ぶ。本研究で扱う洞窟は、狭義的な意味での洞窟である。

本研究では、洞窟を石灰洞や溶岩洞などの自然にできた洞窟と坑道などの人の手により人工的に作った洞窟の 2 種類で分類をおこなう。石灰岩が水などによって侵食されてできた洞窟のことを鍾乳洞または石灰洞と呼び、火山の噴火により流出した溶岩の中に形成された洞窟のことを溶岩洞と呼ぶ。自然にできた洞窟形状は、海食や風食などの侵食作用や水による溶食作用などにより形成した。自然にできた洞窟の特徴は、形状が不規則になることである。一方、鉱山などで探鉱、採鉱、運搬などをするために人工的に地下に作られた洞窟のことを坑道と呼ぶ。坑

道などの人工的に作った洞窟は、鉱山などで採鉱や採掘などをおこなう目的から計画的に掘るため、規則的な形状になる。図 2.1、図 2.2 では、それぞれ自然な洞窟と人工的な洞窟の形状の違いを示す。黒い領域は、道または部屋といった領域である。図 2.1 は、通路が伸びる方向も不規則である。それに対して図 2.2 では、規則的に通路が伸びている。図中の矢印は、洞窟の入口を示したものである。

図 2.1: 自然な洞窟形状

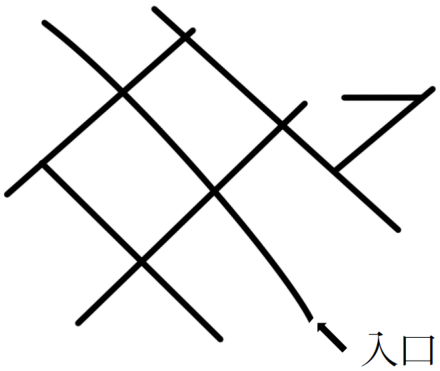


図 2.2: 人工的な洞窟形状

2.2 関連研究

ダンジョンの自動生成はいくつか種類があるが、ここでは 2 つの手法を示す [9] [10]。

ひとつめの手法について述べる。はじめにダンジョンを作る領域を決め、その領域を何度か 2 つに分ける。終了条件はランダムで終わらせる場合や分割した最小領域の縦または横の大きさが一定の値以下になった場合である。最後に、分割した領域に収まるように部屋を作り、各部屋を通路でつなぎ、ダンジョンが完成する。図 2.3 は、その過程を示す。図中の赤い線は領域を分割している線であり、水色はダンジョン形状である。

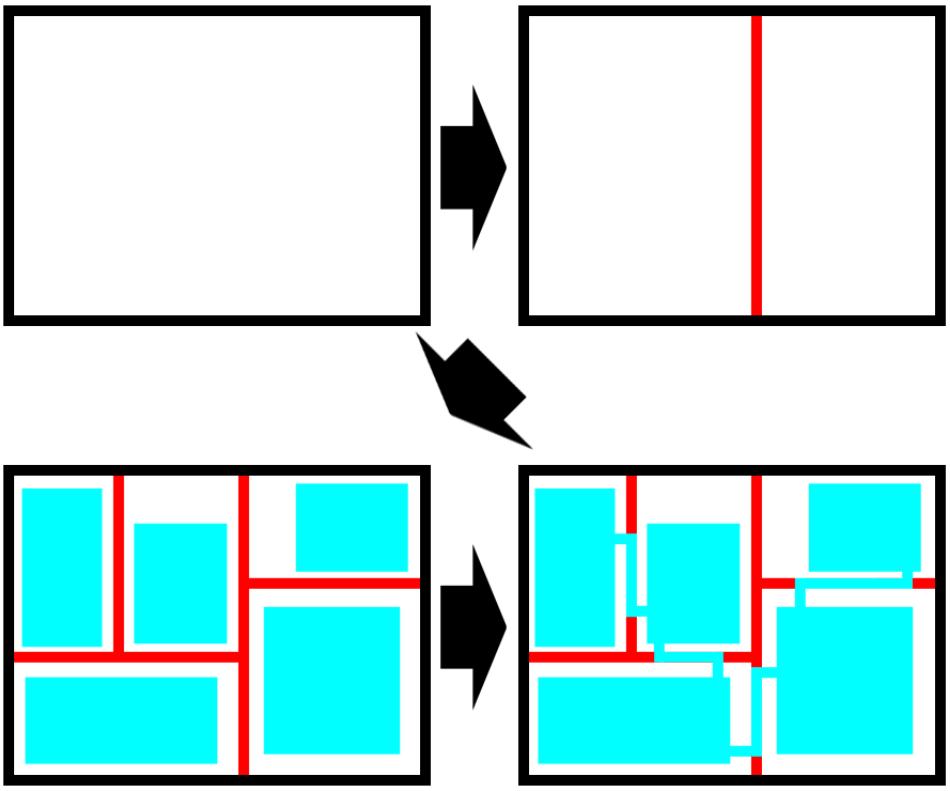


図 2.3: ダンジョンの生成手順

ふたつめの手法について述べる。はじめにダンジョンの部屋の最大サイズを決定する。次にダンジョンの大きさを決定する。m は横方向、n は縦方向の部屋の数とし、ダンジョンの大きさは、部屋を $m \times n$ 個、配置できる領域とする。区切られた 1 つの領域に収まるように部屋を 1 つ作るが、すべての領域に部屋はつくらない。その理由は、ダンジョン形状にメリハリを付けるためである。次に部屋と部屋をつなぐ通路を 1 本作り、つがなている部屋を 1 つのグループとし、グループ分けをおこなう。すべての部屋がつながるまで、各グループから、ランダムで部屋を選び、その部屋同士をつなぐ。すべての部屋同士がつながると、ダンジョンが完成する。図 2.4 は、その過程を示す。図中の水色の領域はダンジョン形状である。

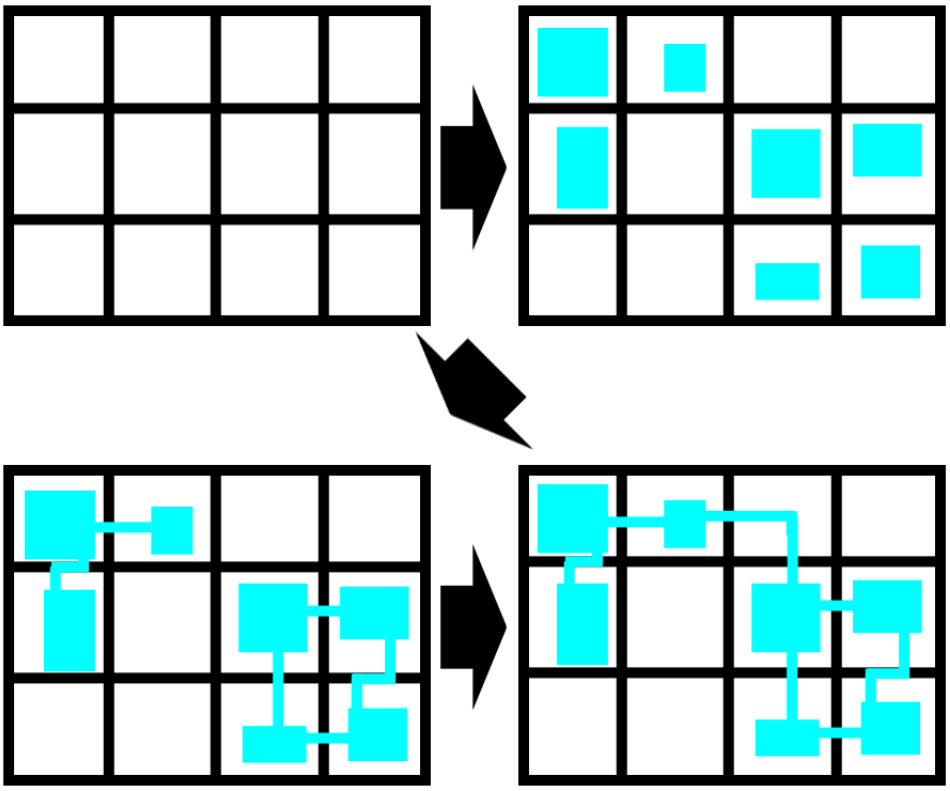


図 2.4: ダンジョンの生成手順

従来のダンジョン自動生成手法では、領域を規則的に分けてしまっているために、ダンジョン形状が規則的になってしまう。そのため、自然な形状のダンジョンをつくることに適さない。

2.3 本研究で扱うダンジョン形状について

本研究では、人工的にできた規則的な洞窟と自然にできた不規則な洞窟の 2 つに注目した。自然にできた洞窟の形状は不規則で、人工的に作った洞窟は規則的な形状になるといった特徴を持つ。本研究ではそれぞれの特徴を持ったダンジョン形状をそれぞれ、人工ダンジョン、自然ダンジョンと呼ぶ。また、洞窟の母岩などによる種類の違いや岩陰は考慮しない。

本研究では人工ダンジョンと自然ダンジョンに対応した3Dダンジョン形状の自動生成をおこなう。人工的なダンジョンの形状と自然的なダンジョンの形状の違いは、パラメータにより制御をする。これにより1つのアルゴリズムによって人工的なダンジョンの形状と自然的なダンジョンの形状の両方を扱うことができる。2つの形状を1つのアルゴリズムで扱えるようになることで、人工的な形状と自然的な形状が混じった形状を表現できる。図2.5では、本研究で目標とするダンジョン形状を示す。この図は右側が人工ダンジョンであり、左側が自然ダンジョンである。中央にある領域は人工ダンジョンと自然ダンジョンの両方の特徴を持つ部屋である。

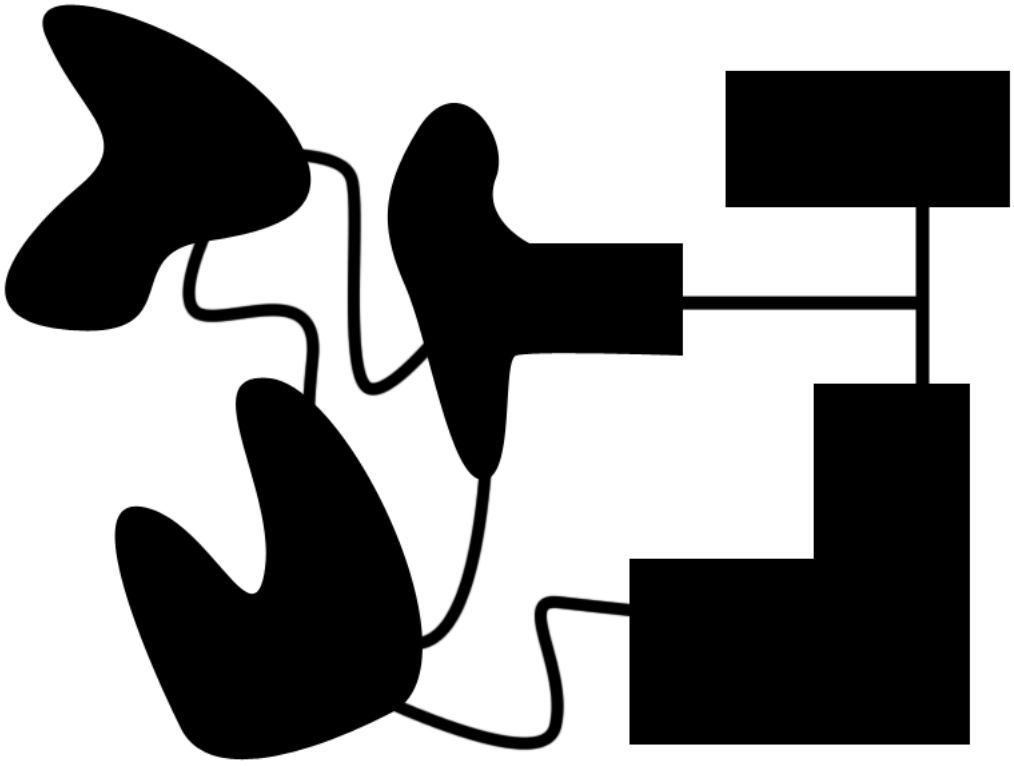


図2.5: 人工的な形状と自然的な形状が混じった形状の例

第 3 章

ダンジョン形状の生成手法

本研究で目標とするダンジョン形状は、ダンジョンのすべての部屋がつながっていて、ダンジョンの部屋同士が重ならずに整合性が取れているものとする。図 3.1 では、これらの条件を満たすものを示し、図 3.2 では、これらの条件を満たさないものを示す。図 3.1 は、ダンジョンのすべての部屋がつながっていて、ダンジョンの部屋同士が重ならずに整合性が取れている。図 3.2 は、緑の部屋がどの部屋ともつながっておらず、赤い部屋が重なっている。この場合、階層を分けなければダンジョンとしての整合性が取れない。

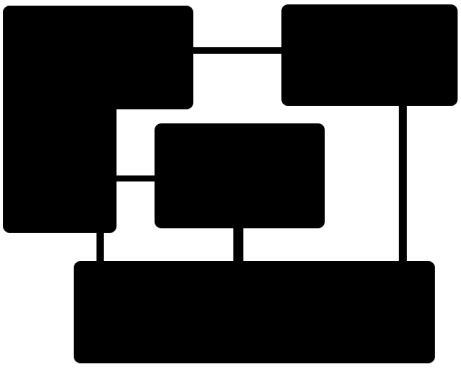


図 3.1: 条件を満たすダンジョン形状

図 3.2: 条件を満たさないダンジョン形状

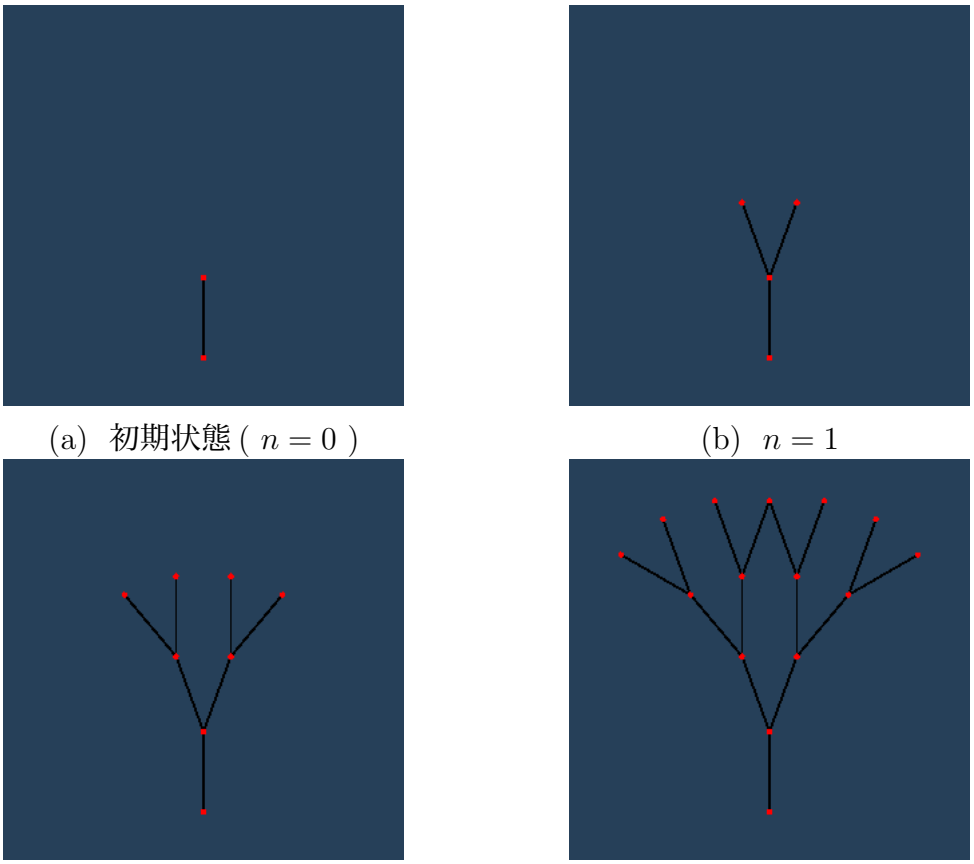
本研究では、L-system を用いて、ダンジョンの外形であるダンジョン形状の線形状を生成する。その後 2 通りの手法を用いて、ダンジョン形状を生成する。ひと

つめは、L-system を用いて生成した線形状からできた領域の取捨選択を行い、ダンジョン形状を生成するふたつめは、L-system を用いて生成した線形状からできた領域を縮小し、ダンジョン形状を生成する。次に本手法の生成手順を次に示す。

1. ダンジョン形状に影響するパラメータを設定
2. パラメータをもとに確率論的 L-system を用いてダンジョンの概形であるダンジョン形状の線形状を生成
3. 線形状から部屋になるだけの面積を持った領域を検出し部屋を生成
4. ダンジョンの部屋と部屋の間に通路を生成
5. 線形状をもとにダンジョン形状を生成
6. 生成した形状に人工的な加工を加える

3.1 L-system

本研究では、自然物の表現と人工物の両方の表現を用いる。そのため自然物の表現と人工物の両方の表現に適した L-system を用いる。L-system は様々な自然物の構造や自己相似図形やフラクタル図形などを表現できる手法である [3][6]。L-system は初期条件で記号や形状を与え、それに対して繰り返し生成規則に基づき記号や形状を変化させることで形状を生成する。図 3.3 は、その過程を示す。図 3.3 は、生成規則を適応するたびに、樹木の生成が行われている様子である。図 3.3a は初期状態であり、図 3.3b、図 3.3c、図 3.3d はそれぞれ n 回の生成規則を適応した。n は生成規則を適応した回数である。



(a) 初期状態 ($n = 0$)　(b) $n = 1$

(c) $n = 2$　(d) $n = 3$

図 3.3: L-system による樹木形状の生成過程

図 3.3 のように、記号や形状に対して毎回 1 つの生成規則を適応する L-system は、決定論的 L-system と呼ぶ。それに対して記号や形状に対して毎回、特定の条件や確率によって生成規則を適応する L-system は、確率論的 L-system と呼ぶ。加藤らの [7] 用いた L-system は確率論的 L-system である。

本研究では確率論的 L-system の生成規則が特定の条件や確率によって生成規則を変えることができ、制御が容易で不規則さを表現できることから確率論的 L-system を用いる。また加藤らの [7] のように、分岐角度に揺らぎを与えることで自然な形状と人工的な形状を再現する。

3.2　ダンジョン生成に影響するパラメータ

本研究で用いる確率論的 L-system の生成規則は、直進する場合、2 つに分岐する場合、3 つに分岐する場合の 3 つとした。どの生成規則を適応するかは、ランダムに決定する。ダンジョンの部屋内部にある鋭角の数が多い場合、ダンジョン形状が歪なってしまうからである。そのため生成規則を 3 分岐までにした。

生成規則を適応する際、線分が伸びる長さと成長方向を決めるための揺らぎの角度が必要になる。線分の長さは同じだが、揺らぎの角度は、線分を生成するたびにランダムで値を決定する。

生成をおこなおうとした線分がそれまでに生成した線分と交差した場合、次の 2 つからランダムで処理を決定する。

- 交点まで生成を行おうとする線分を伸ばし、そこでその線分の生成を止める
- 交点を気にせずに線分を伸ばし、今後もその線分は成長を続ける

すべての領域を部屋とする場合、極端に小さい部屋や極端に大きな部屋ができてしまうためである。そのためダンジョンの部屋の最大の大きさと最小の面積を決める必要がある。また部屋同士をつなぐために必要な通路の幅の長さを決める必要がある。この通路の幅は定数である。

これらの本研究ではこれらのパラメータは、人の手によって設定する。しかし、これらのパラメータは、パラメータセットを用いるかこれらのパラメータをすべてランダムで決定することで、人の手を介さずにダンジョン形状の生成を行える。

ダンジョン形状の生成に必要なパラメータを表 3.1 に示す。

表 3.1: 生成時に必要なパラメータ

直進する確率
2 つに分岐する確率
3 つに分岐する確率
線分の伸びる長さ
線分が伸びる際の揺らぎの角度
線分が交差した際交点で止まる確率
線分が交差しても進む確率
ダンジョンの部屋の最大の面積と最小の面積
通路の幅

3.3 ダンジョンの線形状の生成

本研究でダンジョン形状を生成する際に、ユーザが 3.2 節の表 3.1 で示したパラメータを設定する。これにより人工ダンジョンと自然ダンジョンの違いを出し、ダンジョンの生成をおこなう。

はじめにダンジョンの概形を決めるためにダンジョンの線形状を、確率論的 L-system を用いて作成する。本研究で用いる確率論的 L-system の生成規則は 3 つある。それぞれ、直進する場合の確率を P_1 、2 つに分岐する場合の確率を P_2、3 つに分岐する場合の確率を P_3 とする。これらの生成規則は同時に発生することはない。これを式 (3.1) に示す。

$$P_1 + P_2 + P_3 = 1 \tag{3.1}$$

線分を生成するとき、線分の長さ l と角度の揺らぎ β が必要である。β は、線分を生成するたびにランダムで値が決定する。揺らぎの角度 β の最大値を $\beta_{\max}$ とし、線分が伸びるときの揺らぎの角度 β の値のとりうる範囲を、式 (3.2) に示す。図 3.4 は、線分を生成する様子である。

$$-\beta_{\max} < \beta < \beta_{\max} \tag{3.2}$$

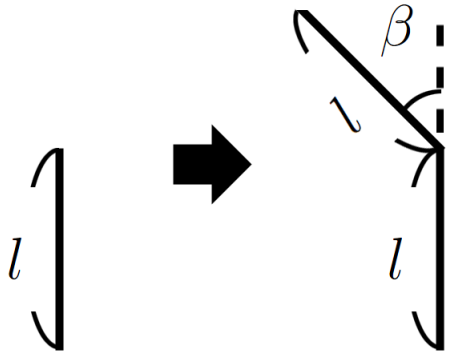


図 3.4: 線分を生成する様子

新たに生成をおこなおうとする線分がそれまでに生成した線分と交差した場合、交点まで生成するかそのまま線分を伸ばすかを決める。この 2 つはランダムで決定する。それぞれの確率は、その交点まで線分を伸ばす確率を P_s、そのまま線分を伸ばす確率を P_c とする。P_s、P_c は同時に発生することはない。これを式 (3.3) に示す。図 3.5 は、線分生成の様子を示す。図左の点線は生成をおこなおうとする線分である。図右上は交差した場合の形状であり、図右下は交差しない場合を示す。

$$P_s + P_c = 1 \tag{3.3}$$

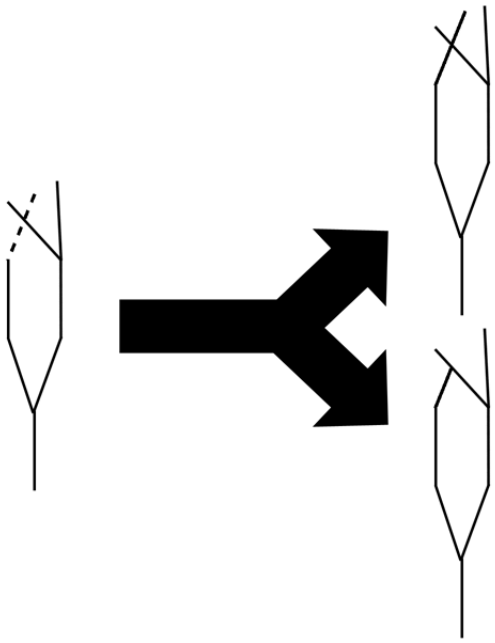


図 3.5: ダンジョンの概形生成の例

これらの値を用いてダンジョンの概形であるダンジョン形状の線形状を生成する手順を示す。はじめに基準となる 2 つの線分を原点からランダムな方向に生成する。これを初期状態として生成規則を適応させ、線分を生成する。生成規則を適応させていくに従い、新たに生成を行おうとする線分とそれまでに生成した線分と交差する場合がある。この場合、線分を交点まで伸ばし生成を止めるか、そのまま線分を伸ばし線分を生成するかを決める。これは P_s 、 P_c の確率によって決まる。生成規則を適応させ、生成規則が適応できなくなるまでこれらを繰り返す。図 3.6 は、ダンジョンの概形生成の例である。

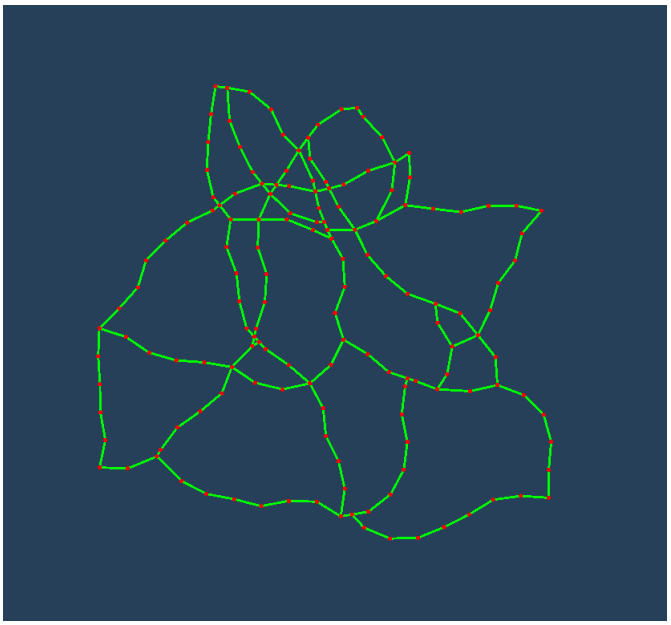


図 3.6: ダンジョンの概形生成の例

3.4　ダンジョン形状の生成

生成したダンジョンの線形状をもとに、2 通りの方法を用いてダンジョン形状を生成する。L-system を用いて生成した線形状からできた領域の取捨選択を行い、生成するダンジョンを取捨選択型ダンジョン形状と呼ぶ。L-system を用いて生成した線形状からできた領域を縮小し、生成するダンジョンを領域縮小型ダンジョン形状と呼ぶ。取捨選択型ダンジョン形状は、部屋の数が少なく通路が長いといった特徴がある。領域縮小型ダンジョン形状は、部屋の数が多く通路が短いといった特徴がある。表 3.2 に、取捨選択型ダンジョン形状と領域縮小型ダンジョン形状の特徴を示す。

表 3.2: 各ダンジョン形状の特徴

	取捨選択型ダンジョン形状	領域縮小型ダンジョン形状
部屋の数	少ない	多い
通路の長さ	長い	短い

ダンジョン形状の線形状から線で囲われた領域を見つけ部屋となりうる領域とする。部屋となりうる領域からベクトルの外積を用いて、部屋となりうる領域の面積 S を計算する。部屋となりうる領域の面積 S の最大値を $S_{\max}$ とし、最小値を $S_{\min}$ とし、S の値のとりうる範囲を、式 (3.4) に示す。

$$S_{\min} < S < S_{\max} \tag{3.4}$$

式 (3.4) に示した範囲に S が含まれない場合、その領域は部屋にはならない。すべての領域を調べ、部屋となりうる領域を決定する。

3.4.1 取捨選択型ダンジョン形状の生成

取捨選択型ダンジョン形状の生成では、部屋となりうる領域を決定した後、部屋の取捨選択をおこなう。ダンジョンの部屋同士が近すぎると部屋同士の間に通路を作ることができない。そのため部屋同士は隣接することができないとする。つまり部屋となりうる領域同士は、同一の点を共有できない。もし、部屋同士が隣接していた場合、どれかひとつを残して、他の部屋はすべて部屋として扱わない。これらの判断はランダムでおこなう。

次に部屋の領域と部屋の領域との間の線分を通路とする。この線分のままでは、キャラクターが通ることができないので、通路となる線分から道幅 d だけ拡張し通路とする。図 3.7 は、線分の拡張し通路とする例である。実線はダンジョンの線形状とし、点線は道を拡張した線である。

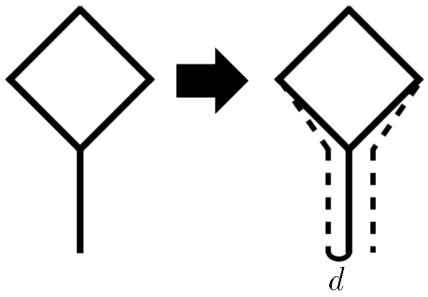


図 3.7: 線分の拡張し通路とする例

3.4.2　領域縮小型ダンジョン形状の生成

領域縮小型ダンジョン形状の生成では、部屋となりうる領域を決定した後、与えられた領域を縮小をおこなう。部屋となりうる領域を縮小し部屋を作る。各部屋の間に通路を作れるだけの空間を作る。その後、取捨選択型ダンジョン形状の生成と同様に、通路となる線分から道幅 d だけ拡張し通路とする。図 3.7 は、線分の拡張し通路とする例である。実線はダンジョンの線形状とし、灰色の領域は部屋の領域である。

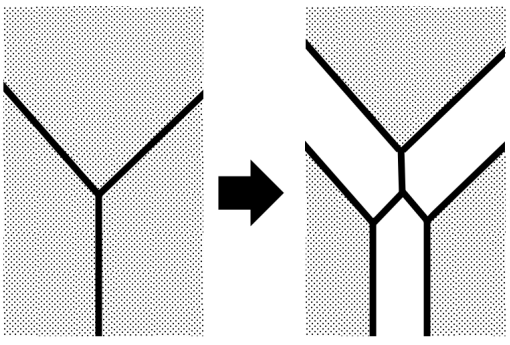


図 3.8: 領域縮小し部屋を作る例

3.5　人工的な形状の追加

自然な形状の中に人工的な形状を追加するために、ダンジョン形状を生成した後、生成した部屋の領域に人工的な形状を加える。ひとつの部屋の連続したいくつかの頂点のうち、ひとつを除き、すべての頂点を削除する。その後、削除されなかった頂点とその連続したいくつかの頂点の始点と終点をつなげる。どの部屋のどの部分に処理を施すかは、ランダムで決定する。図 3.7 は、人工的な形状を追加する例である。実線は処理を施す前の部屋の境界線とし、点線は実線は処理を施した後の部屋の境界線である。

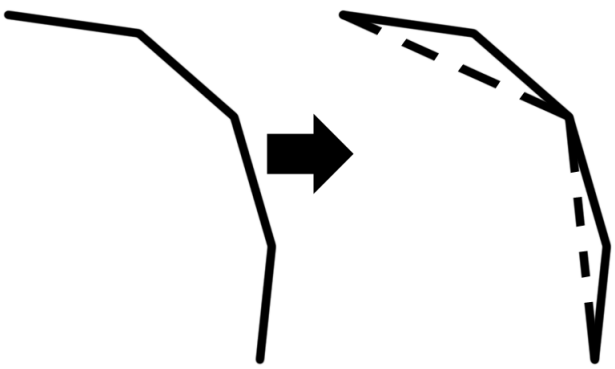


図 3.9: 人工的な形状を追加する例

第 4 章

検証と考察

4.1 実装と結果

3 章で述べたダンジョン形状の生成手法を Fine Kernel ToolKit System[11] を用いて実装を行った。データ構造は、ハーフエッジ構造 [12][13] を用いた。ハーフエッジ構造とは、次のようなデータを持つ。

- 頂点
- 稜線
- 半稜線
- ループ (面)

頂点は、形状の頂点を指し、稜線の端点となる。稜線とは、面の境界線を指す。稜線は必ず両端点に頂点があり、その頂点を始点とする半稜線がひとつづつある。半稜線とは、稜線に必ず 2 つある要素であり、稜線の片側の端点を始点とするものである。ループとは、半稜線によって囲まれた面のことである。ハーフエッジ構造は半稜線をたどることで簡単にループの検出ができる。そのため本研究ではハーフエッジ構造を用いた。

図 4.1 の a、b は、表 4.1 に示した同一のパラメータを用いて本手法により生成したダンジョン形状である。同一パラメータで、それぞれ違った形状ができた。

表 4.1: 生成時のパラメータ

直進する確率 (P_1)	0.9
2 つに分岐する確率 (P_2)	0.05
3 つに分岐する確率 (P_3)	0.05
線分の伸びる長さ (l)	30.0
線分が伸びる際の揺らぎの最大角度 ($\beta_{\max}$)	30.0
線分が交差した際交点で止まる確率 (P_t)	0.5
線分が交差しても進む確率 (P_c)	0.5
ダンジョンの部屋の最小面積 ($S_{\min}$)	2000.0
ダンジョンの部屋の最大面積 ($S_{\max}$)	40000.0
通路の幅 (d)	5.0

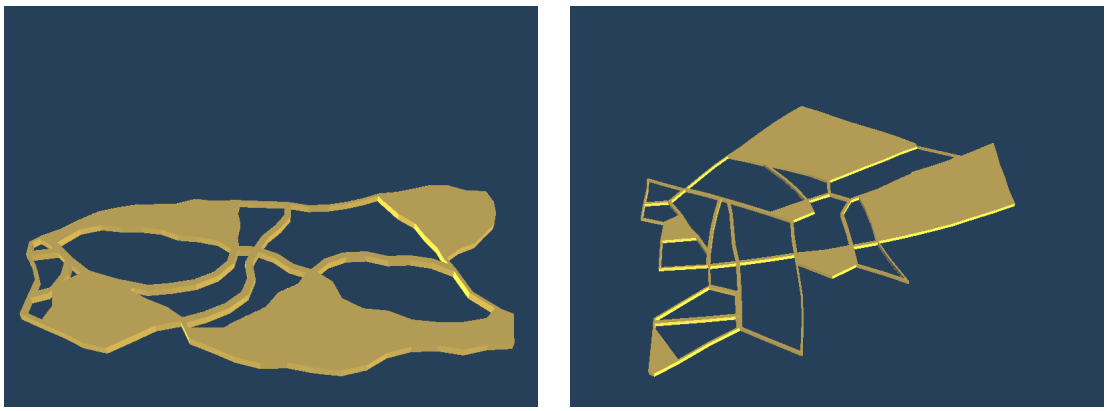
(a) (b)

図 4.1: 同じパラメータでの生成結果

図 4.2 の a、b は、表 4.2 に示した異なったパラメータを用いて本手法により生成したダンジョン形状である。図 4.2a では自然ダンジョン、図 4.2b では人工ダンジョンをパラメータの違いにより生成した。

表 4.2: 生成時のパラメータ

	a	b
直進する確率 (P_1)	0.9	0.8
2 つに分岐する確率 (P_2)	0.05	0.1
3 つに分岐する確率 (P_3)	0.05	0.1
線分の伸びる長さ (l)	30.0	40.0
線分が伸びる際の揺らぎの最大角度 ($\beta_{\max}$)	30.0	5.0
線分が交差した際交点で止まる確率 (P_t)	0.5	0.4
線分が交差しても進む確率 (P_c)	0.5	0.6
ダンジョンの部屋の最小面積 ($S_{\min}$)	2000.0	2000.0
ダンジョンの部屋の最大面積 ($S_{\max}$)	40000.0	40000.0
通路の幅 (d)	5.0	2.5



(a) (b)

図 4.2: 別のパラメータでの生成結果

図 4.3 は、表 4.3 に示したパラメータを用いて本手法により生成したダンジョン形状である。図 4.3 は形状の左半分が人工ダンジョンで、形状の右半分が自然ダンジョンの特徴をもったダンジョンである。この違いは線分が伸びる際の揺らぎの角度 β、左半分ではを小さくし、右側半分で大きくすることで実現した。$\beta_{r\,\max}$ は左半分での線分が伸びる際の揺らぎの角度とし、$\beta_{l\,\max}$ は右半分での線分が伸びる際の揺らぎの角度とする。

表 4.3: 生成時のパラメータ

直進する確率 (P_1)	0.9
2 つに分岐する確率 (P_2)	0.05
3 つに分岐する確率 (P_3)	0.05
線分の伸びる長さ (l)	30.0
左半分での線分が伸びる際の揺らぎの最大角度 ($\beta_{r\,\max}$)	6.0
右半分での線分が伸びる際の揺らぎの最大角度 ($\beta_{l\,\max}$)	30.0
線分が交差した際交点で止まる確率 (P_t)	0.5
線分が交差しても進む確率 (P_c)	0.5
ダンジョンの部屋の最小面積 ($S_{\min}$)	2000.0
ダンジョンの部屋の最大面積 ($S_{\max}$)	40000.0
通路の幅 (d)	5.0

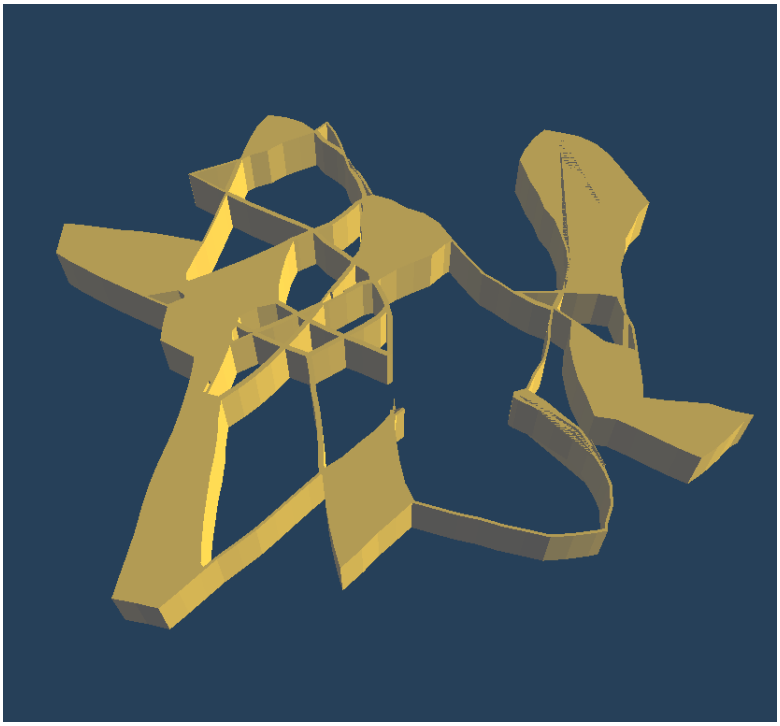


図 4.3: ダンジョンの生成手順

図 4.4 は、取捨選択型ダンジョン形状のダンジョン形状であり、図 4.5 は、領域縮小型ダンジョン形状のダンジョン形状である。ひとつの線形状から 2 通りのダ

ンジョンを生成することができた。

図 4.4: 取捨選択型ダンジョン形状

図 4.5: 領域縮小型ダンジョン形状

図 4.6 は、人工的な加工を加えていないダンジョンであり、図 4.7 は、人工的な加工を加えたダンジョンである。これにより自然な形状の中に人工的な形状が存在したダンジョン形状ができた。

図 4.6: 人工的な加工を加えていないダンジョン

図 4.7: 人工的な加工を加えたダンジョン

4.2　考察と問題点

本研究では L-system を用いて自然ダンジョンと人工ダンジョン、人工ダンジョンと自然ダンジョンの特徴をもったダンジョンを生成できた。自然ダンジョンは形状が不規則になり、人工ダンジョンは規則的な形状になった。人工ダンジョンと自然ダンジョンの特徴をもったダンジョンでは、それぞれの特徴を持った形状になった。これにより本研究の目的は達成できたといえる。

今後の課題として、次の 2 つの問題点がある。

1. ダンジョンの部屋同士が離れすぎてしまう

2. 分割した領域の数に比べ、生成する部屋数が少ない

図 4.8 には領域の削除前の画像を示し、図 4.9 と理想の削除結果の画像を示す。

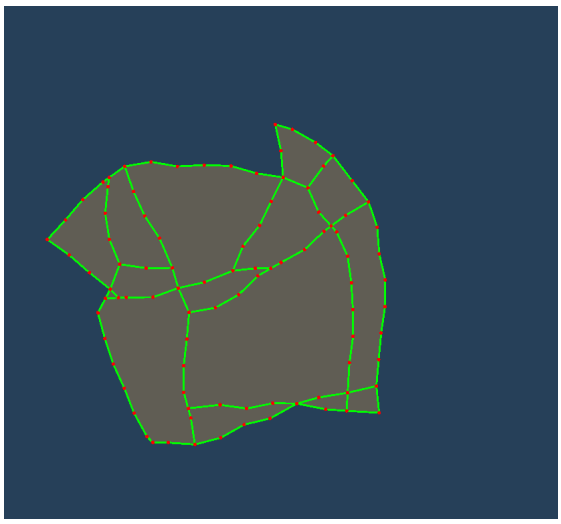


図 4.8: 領域の削除前

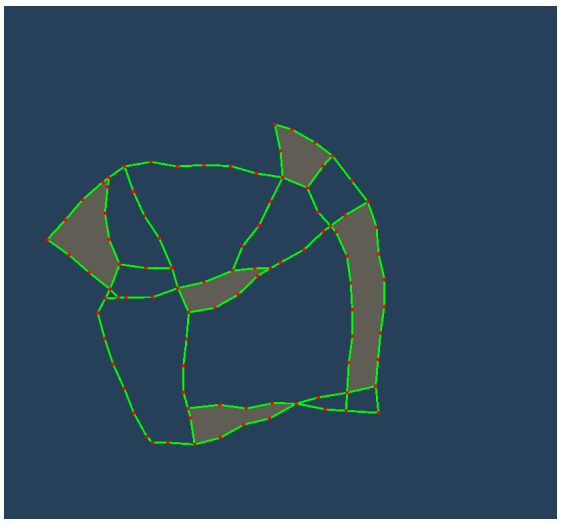


図 4.9: 理想の削除結果

次に、図 4.10 では、ダンジョンの部屋同士が離れすぎてしまう場合の画像を示す。図 4.10 では理想の削除結果である図 4.9 と比べ、部屋同士が離れている。

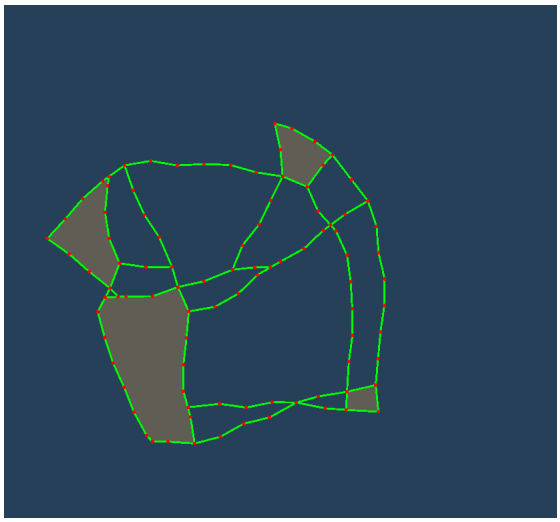


図 4.10: ダンジョンの部屋同士が離れすぎてしまう場合

そして図 4.11 には、分割した領域の数に比べ、生成する部屋数が少ない場合の画像を示す。理想の削除結果である図 4.9 では 5 つの部屋ができているが、図 4.11 では部屋の数が 2 つしかない。

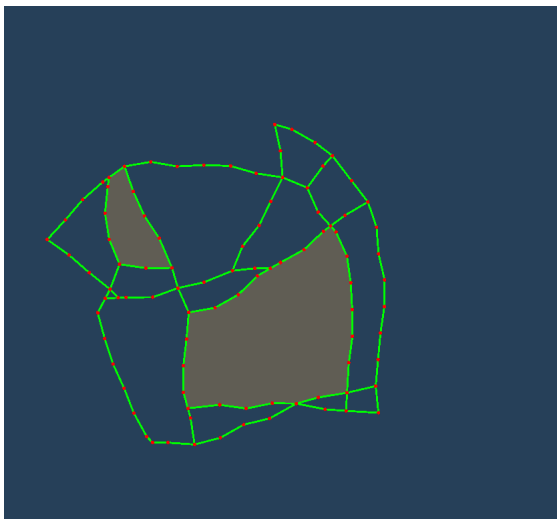


図 4.11: 分割した領域の数に比べ、生成する部屋数が少ない場合

これらの問題点の原因は、部屋となりうる領域同士は、同一の点を共有できな

いという部屋の生成規則の実装方法に問題があった。隣接した部屋同士を削除する方法として、ランダムに削除をおこなった。そのため部屋の離れすぎや過度な部屋の削除がおこった。

第 5 章

まとめ

本研究では、L-system を用いた 3D ダンジョン生成手法の提案した。またパラメータ制御により、人工ダンジョンや自然ダンジョンや人工ダンジョンと自然ダンジョンの特徴を持ったダンジョンの生成を目的とした。これは確率論的 L-system を用いることで実現した。

今後の課題として、領域の取捨選択方法の改善していく必要がある。これにより、より整合性のとれたダンジョン形状を作ることができるだろう。

謝辞

本研究を行うにあたり、研究方針や手法など数多くの助言や指導、ご教授頂いた渡辺大地講師、三上浩司講師、本校大学院の先輩方、ゲームサイエンス研究室の仲間たちに深く感謝いたします。

参考文献

[1] Square Enix Company, Limited. チョコボの不思議なダンジョン 2, 1998.

[2] D. Adams et al. Automatic Generation of Dungeons for Computer Games. *Bachelor thesis, University of Sheffield, UK. DOI= http://www. dcs. shef. ac. uk/intranet/teaching/projects/archive/ug2002/pdf/u9da. pdf*, 2002.

[3] Mech R, Prusinkiewicz P. Visual models of plants interacting with their environment. *SIGGRAPH 96*, pages 397–410, 1996.

[4] 桑原教彰, 鉄谷信二, 志和新一, 岸野文郎. フラクタルを用いた階層的な樹木形状表現による 3 次元樹木画像の高速生方法. **電子情報通信学会論文誌**, pages 1091–1104, 1995.

[5] 望月茂徳. 生物学・フラクタルモデルに基づいた紅葉のビジュアルシミュレーション. 2002.

[6] Przemyslaw Prusinkiewicz , Aristid Lindenmayer, James Hanan. Development models of herbaceous plants for computer imagery purposes. *SIGGRAPH*, 1988.

[7] 加藤 伸子, 奥野 智江, 狩野 均, 西原 清一. L-system を用いた仮想都市のための道路網生成手法. **情報処理学会論文誌**, Vol.41(No.4):110–01112, 2000.

[8] 上野 俊一，鹿島 愛彦. **洞窟学入門 暗黒の地下世界をさぐる**. 講談社, 1978.

[9] 山本 貴光. **デバッグではじめる *C* プログラミング**. 翔泳社, 2008.

[10] 日高 徹. ***Delphi* ゲームプログラミングのエッセンス**. ソフトバンククリエイティブ, 2002.

[11] 渡辺 大地. Fine Kernel Tool Kit System. `http://www.teu.ac.jp/media/~earth/FK/`.

[12] K. Weiler. Edge-based data structures for solid modeling in curved-surface environments. *IEEE Computer Graphics and applications*, 5(1):21–40, 1985.

[13] 齋藤満昭. 多重解像度 Marching-Cubes 法.