

ペイントツールのインタフェースを用いた 街並みの自動生成

黒土英太郎^{†1}, 床井浩平^{†2}

本研究では、ゲーム開発における効率性の向上のために、ペインティングツールのインタフェースを用いて生成した曲線を道路として見立て、周囲に建物を生成することで町並みを容易にモデリングできるシステムの開発を行った。提案手法は、ユーザーが対話的にモデリングを行いつつ、生成した建物を実際の町並みの特徴に合った位置に配置する。そのためにマウスをドラッグして得た点列を連結したものにメッシュを割り当てることにより道路として見立て、それにより区切られた領域内に乱数を用いて建物を初期配置する。そののち個々の建物を最も近い交差点まで拡大することにより、その領域を充てんする。

1. はじめに

近年のゲームデザインにおいて、背景となる街並みは、そのゲームの世界観や遊びのデザインについて重要な役割を持つ[1]。しかしそれらの形状はコンピュータの性能が上がるにつれて非常に複雑で有機的になり、意味を持たせるモデリングを行うコストが高くなっている[2]。

少ないパラメータを用いて3Dモデリングを行う技術に、プロシージャル技術がある。これは手続き的な手法を用いることで、様々なコンテンツをパラメータの変更だけで作成しようとするものである。図1は、L-System[3]と呼ばれるアルゴリズムを用いて線分を決められた文法に沿って枝分かれるることにより、樹木を生成した例である。

しかし、L-systemを含め、プロシージャル技術は、ある程度の制御をパラメータの変更によって行うことは可能だが、必ずしもユーザーの意図に合致した結果が得られるとは限らない。

そこで、デザイナーの意図を反映しつつ、容易に町並みをモデリングすることができるシステムを開発することは有用であると考えられる。



図1 L-systemによって自動生成された樹木

2. 研究目的

本研究では前述したプロシージャル技術における「必ず

しもユーザーの意図に合致した結果が得られるとは限らない」という問題に対処することを目的とする。

本稿では、ユーザーが道路網をペインティングツールのインタフェースを用いて描き、それをもとにユーザーの意図したものに近い町並みの形状を生成する手法を提案する。また、後述する生成試行回数に対する生成時間と得られた街並みの形状に主観評価を行い、そこで得た結果から考察して提案手法の利点や改善点を見出すとともに、今後の展望や課題を考察する。

3. 提案手法

本研究は、マウスをドラッグして得た座標をもとに曲線を描画したのちに、乱数から得た座標と線分の交点から建物を生成する手法を提案する。

3.1 道路の生成

ペインティングツールのインタフェースの作成するために、マウスをドラッグして得た点列を連結したものにメッシュを割り当てる(図2)。

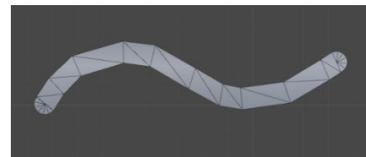


図2 メッシュを割り当てた曲線

3.2 建物の生成

描画した曲線に対して建物を生成するために、画面内のランダムな点に建物と見立てた立方体を生成し(図3)、その建物に一番近い交点まで拡張することで(図4)領域内に建物を充填させる手法を用いた。また、拡張を行ったときに建物が道路に衝突している間は建物を縮小し続け、建物が重なった場合は建物同士の体積を比較し、体積の小さい方を消去する。拡張を行った後は底面積に対する建物の高さを決定する[4](表1)。この手順を3000回行って(以下生成試行回数)、生成された街並みの例を以下に示す(図5)。

^{†1} 和歌山大学大学院システム工学研究科
Graduate School of Systems Engineering, Wakayama University
^{†2} 和歌山大学システム工学部
Faculty of Systems Engineering, Wakayama University

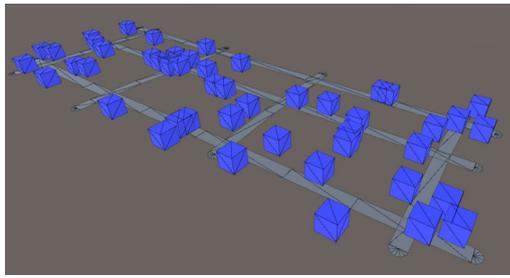


図 3 乱数を用いた建物の生成

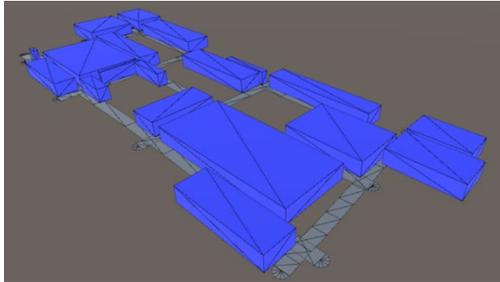


図 4 拡縮を行い、重なった建物を消去した街並み

底面積	高さ
1~6	1~5
6~10	1~11
10~15	7~11
15~20	7~20

図 5 底面積の平均と高さの平均の関係[3]

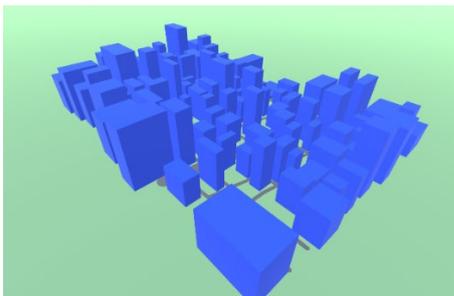


図 6 生成された街並み

4. 実験

本研究のシステムで作成された街並みを評価するため、生成試行回数に対する生成時間と景観に対する主観評価を行った。本研究のインターフェースを用いて作成した道路網（図 7）に対して生成試行回数を 500, 1000, 2000 に設定し、それらの生成時間と景観を比較する（図 8, 9, 10）。各生成試行回数に対する生成時間は以下のとおりである。（表 2）

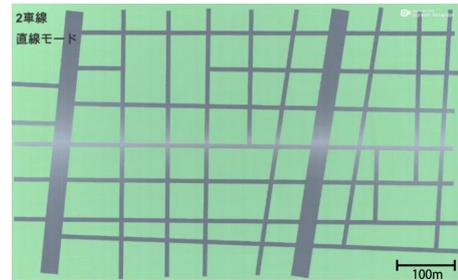


図 7 作成した道路網

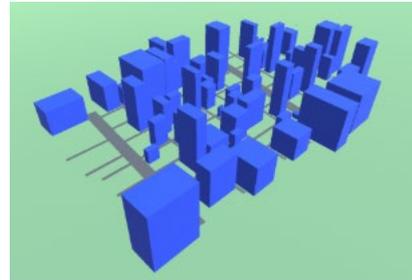


図 8 生成試行回数 500 の街並み

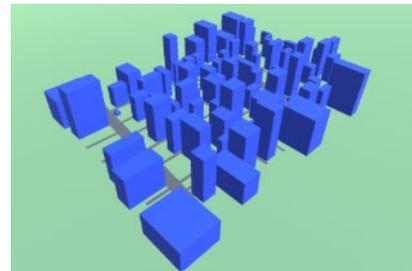


図 9 生成試行回数 1000 の街並み

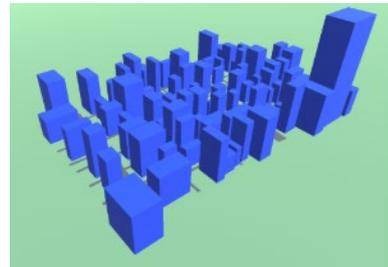


図 10 生成試行回数 2000 の街並み

生成試行回数	生成時間 (秒)
500	0.72
1000	0.96
2000	2.89

表 1 生成試行回数に対する生成時間

4.1 評価

本研究で採用した、ペインティングツールのインタフェースによって、ユーザの意図した道路網を生成させることができた。また、建物の生成試行回数を増やすことで、建物の総数が増加した点や、道路に囲まれた区画内に建物が充てんされやすくなった点、建物の大きさの統一性が向上した点も評価できる。

しかし、生成試行回数を増やすことにより生成時間も増加してしまう点や、建物のオブジェクトを立方体の拡張によって生成しているために、建物の多様性が乏しいという点が課題として挙げられる。

また、生成される結果の大部分が乱数に依存しているため、ユーザの意図した街並みが生成されない可能性が生じるという点が今後の大きな課題となる。

5. 今後の展望

今後の展望として先述した本研究の様々な課題を解決する手法に、以下の提案を行う。

5.1 Bentley-Ottman アルゴリズムによる交点列挙

本研究の課題の一つとして挙げられた、生成時間が長いという点の原因として、交点列挙がある。本研究で採用した交点列挙の手法は、全線分に対し総当たりで判定するので、線分が増えるにつれて計算量が増大してしまう。

そこで、Bentley-Ottman の平面走査法と呼ばれるアルゴリズム[5]を採用することで、計算量を削ることが期待される。

5.1.1 Bentley-Ottman アルゴリズム

「端点座標の対として与えられた n 個の線分に対して交差している線分対を報告する」という問題を扱う際、Bentley-Ottman アルゴリズムは、掃引直線を用いて平面を左から処理するアルゴリズムである (図 11)。ここで、線分の端点 p に対し p を端点とする線分を $s(p)$ とする。また y 軸に平行な掃引直線を考え、各端点で掃引直線と交わる線分を y 座標順にソートして保持するために 2-3 木 SWEEPLINE を用意する。また、掃引直線と交わる線分の交差順序が変化する場所 (線分の端点と線分対の交点) を管理するために、線分の端点と線分対の交点を x 座標を基準に保持するヒープ NEXTSTOP を用意する。アルゴリズムの概要は以下のとおりである。

1. 線分の端点を x 軸でソートし、 x 座標の小さいほうか

ら添え字を付け p_1, \dots, p_{2n} とする。

2. SWEEPLINE を空にする。 p_1 から p_{2n} を NEXTSTOP に追加する。

3. NEXTSTOP から点 v を取り出し、(ア) (イ) (ウ) を行う。 NEXTSTOP が空なら終了する。

(ア) v が線分 $s(p_i)$ の左端点 p_i ならば、以下を行う。

- ① p_i の x 座標における各線分の y 座標をキーとして $s(p_i)$ を SWEEPLINE に挿入する。
- ② SWEEPLINE 上で $s(p_i)$ と隣り合う線分と $s(p_i)$ との交差判定を行い、交差したらその線分対を報告し、交点を NEXTSTOP に追加する。

(イ) v が線分 $s(p_i)$ の右端点 p_i ならば、以下を行う。

- ① $s(p_i)$ を SWEEPLINE から削除する。
- ② SWEEPLINE 上で新規に隣り合う線分対の交差判定を行い、交差したらその線分対を報告し、交点を NEXTSTOP に追加する。

(ウ) v が線分 $s(p_i)$ と線分 $s(p_j)$ の交点ならば、以下を行う。

- ① $s(p_i)$ と $s(p_j)$ の SWEEPLINE における場所を入れ替える。
- ② SWEEPLINE 上で新規に隣り合う線分対の交差判定を行い、交差したらその線分対を報告し、交点を NEXTSTOP に追加する。

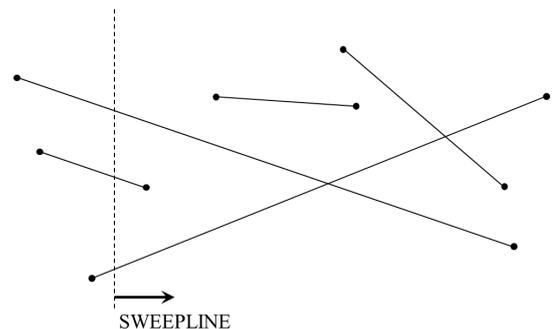


図 11 Bentley-Ottman の線分交差列挙

5.2 L-System による領域分割と建物のプロシージャルモデリング

建物の多様性が乏しいという問題点を解決するため、道路で囲まれた領域を L-System で分割し、分割された小領域を底面とする建物のプロシージャルモデリング手法を提案する。これにより底面形状が長方形だけでなく、多角形のものも生成されることが期待できる。

5.2.1 L-System

L-system は初期文字列に書き換え規則を繰り返し適用することで発生の様子を記述するもので[6]、第 1 章で述べた

ように植物の生長などを記述するのに使われている。以下に生成例を示す(図12)。以下の図では、初期文字列Fに書き換え規則 $F \rightarrow F[+F][-F]$ を繰り返し適応したものである。この書き換え規則の文字列から図形への変換は、タートル・グラフィックスの考え方[7]を用いて以下のように行う。なお、図12では回転する角度を 30° とした。

- F: タートルが一定の長さで前進する。
- +, -: タートルが左右へ一定回数回転する。
- [,]: 分岐する。



図12 $F \rightarrow F[+F][-F]$ の書き換え規則による図形の生成

5.2.2 小領域を底面とする建物のプロシージャルモデリング

L-Systemにより分割された小領域を底面とした、建物の生成を行う。この際、小領域の頂点数、頂点位置に合わせた底面の生成と、その形状に合わせた適切な壁面を作成する必要がある。

そこで、Houdiniなどのモデラーを用いた建物自体のプロシージャルモデリングを行うことで、頂点数や位置に合わせた建物の生成が期待できる[8]。

5.3 環境ベクトル、スペースシンタックス理論による制御

先述したL-Systemにルールを設ける際や、建物の生成時の高さを決定する際に、乱数を用いた場合、「ユーザの意図した街並みが生成されない可能性がある」という課題を解決することはできない。

そこで、今後の予定の1つとして「町らしさ」に関する以下の項目を調査することがあげられる。

5.3.1 環境ベクトル $V=(v_1, v_2, v_3, d)$ [9]

都市は道路で囲まれたブロックで構成されており、各ブロックが相互に作用しあいながら発展衰退をおこす。そこで、都市およびブロックの状態を表す特徴として、以下のようなパラメータを定義する。

$V=(v_1, v_2, v_3, d)$: ブロックごとに設定される
環境ベクトル

- v_1 : 生活環境の度合い
- v_2 : 消費活動の度合い

v_3 : 生産活動の度合い

d : ブロック内の建物の延べ床面積 / ブロックの総面積

ただし、 $0.0 \leq v_1 \leq 1.0$, $0.0 \leq v_2 \leq 1.0$,

$0.0 \leq v_3 \leq 1.0$, $v_1 + v_2 + v_3 = 1.0$

5.3.2 スペースシンタックス理論[10]

スペースシンタックス理論とは、都市空間からConvexSpaceとAxialLineを生成することで、歩行者の量やほかの場所からのアクセスの容易さを指標として求めることのできる理論である。

6. おわりに

第3章で述べた通り、本研究ではペインティングツールのインタフェースを用いて描画した曲線を道路として見立て、そこに乱数を用いて建物を生成する手法を提案した。このシステムにより、ユーザのレベルに限らず、だれでも手軽に仮想都市を生成できることが期待できる。

参考文献

- 1) デザイナの仕事 ゲームCGデザイン
<https://www.nintendo.co.jp/jobs/introduction/design/work02.html>
- 2) GTA V Most Expensive Video Game in History - Budget More than High Budget Hollywood Films
<https://wccftech.com/gta-v-most-expensive-video-game-history/>
- 3) Przemyslaw Prusinkiewicz, Aristid Lindenmayer: The Algorithmic Beauty of Plants(1990)
- 4) 加藤 伸子, 岡野 紋, 狩野 均, 西原 精一: 遺伝的アルゴリズムを用いた仮想都市のための建物配置方式, 電子情報通信学会論文誌, Vol.J82-D-II, No.10, pp.1766-1774(1999)
- 5) Balaban: An optimal algorithm for finding segments intersections, Proc. 11th ACM Symp. Computational Geometry, pp. 211-219,(1995)
- 6) 加藤 伸子, 奥野 智江, 狩野 均, 西原 精一: Lシステム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, pp. 1104-1112(2000)
- 7) Abelson, H. and diSessa, A.A: The Turtle geometry, M.I.T. Press(1982)
- 8) ArtStation - Modular building, asset test with Houdini, Zheng Wang
<https://www.artstation.com/artwork/2mvyx>
- 9) 大谷 洋平, 山内 大介, 本田 真望, 水野 一徳, 福井 幸男, 西原 精一: 時間変化を伴う仮想都市の自動生成における建物配置, 情報処理学会 65 回全国大会, No4, pp.83-84(2004)
- 10) 荒屋 亮, 竹下 輝和, 池添 昌幸: スペースシンタックス理論に基づく市街地オープンスペースの特性評価日本建築学会計画系論文集, No589, pp.153-160(2005)