

ペイントツールのインタフェースを用いた街並みのモデリングツールの開発

○黒土 英太郎、床井 浩平
和歌山大学大学院

Development of A Tool to Design A Cityscape with The User Interface of A Painting Tool

Kurotsuchi Eitaro, Tokoi Kohei

1. はじめに

近年のゲームデザインにおいて、背景となる町並みはそのゲームの世界観や遊びのデザインにお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、デザイナーの意図を反映したモデリングを行うにはコストが高くなっている [2]。

2. 研究目的

本研究では、プロシージャル技術における「必ずしもデザイナ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 [3] に対処することを目的とする。提案手法はデザイナーが自分の考えた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、自分の意図した結果に近い町並みの形状を生成させることができる。また、得られた街並み形状に対してアンケート形式による主観評価を行い、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

3.1 交差判定および交点座標の取得

描いた線分 AB および CD において、点 A および B の座標をそれぞれ (x_a, y_a) , (x_b, y_b) 、点 C および D の座標をそれぞれ (x_c, y_c) , (x_d, y_d) とする。式 (1) 及び (2) において、 $t_{ab}(x, y) = 0$, $t_{cd}(x, y) = 0$ なら、点 (x, y) はそれぞれの直線上に位置する。

$$t_{ab}(x, y) = (x_a - y_b)(y - y_a) + (y_a - y_b)(x_a - x) \quad (1)$$

$$t_{cd}(x, y) = (x_c - y_d)(y - y_c) + (y_c - y_d)(x_c - x) \quad (2)$$

このとき、 $t_{ab}(x_c, y_c)$ 及び $t_{ab}(x_d, y_d)$ の符号が異なれば、直線 AB と線分 CD は交差している。これは式 (3) で判定できる。

$$t_{ab}(x_c, y_c)t_{ab}(x_d, y_d) < 0 \quad (3)$$

同様に $t_{cd}(x_a, y_a)$ と $t_{cd}(x_b, y_b)$ を求め、式 (4) によって直線 CD と線分 AB との交差判定を行う。

$$t_{ab}(x_c, y_c)t_{ab}(x_d, y_d) < 0 \quad (4)$$

式 (3) と式 (4) がともに真なら、線分 AB と線分 CD は交差している。

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、図 1 のように画面内のランダムな位置に建物を生成し、生成した建物に一番近い交点まで拡張することにより、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 (図 2)。拡張を行った後は底面積に対する建物の高さを表 1 にもとづいて決定する [4]。この手順を 3000 回行って生成された街並みの例を図 3 に示す。

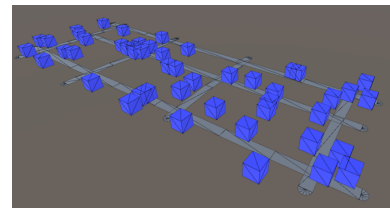


図 1: ランダムな位置に生成した建物

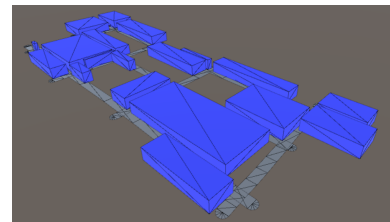


図 2: 拡張を行い、重なった建物を消去した町並み

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

表 1: 底面積の平均と高さの平均の関係 [3]

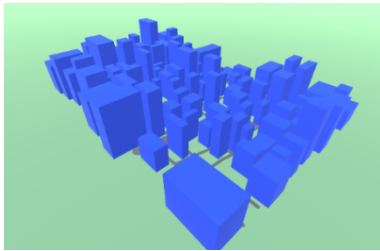


図 3: 生成した仮想都市

4. 評価と今後の検討

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、デザイナー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、建物の大きさの統一性が向上した。

しかし、建物の生成が乱数に依存している点から、デザイナーの意図に合わせた建物の生成ができないという大きな課題がある。そこで、本研究の今後の課題を、デザイナーがある程度調節可能なパラメータを持つ建物生成システムの実装とし、ウェイトペイントを用いた建物の形状制御の検討を行う。

4.1 右手法による閉領域検出

今後、ユーザが閉領域ごとに異なるパラメータの値を設定することを想定し、右手法による閉領域の検出システムを実装した (図 4)。処理の流れは以下のとおりである。

1. 任意の点 P を決定する
2. 点 P から右 (+X 軸) 方向に進行する
3. 交点に到達した場合、右折する
4. 始点に到達したら閉領域を検出する

なお、2 の処理では、候補となる各方向ベクトルを正規化したものとベクトル (1, 0, 0) の内積が最大となる方向を選択し、3 の処理では侵入したベクトルと候補ベクトルの外積が最小となる方向をとる。

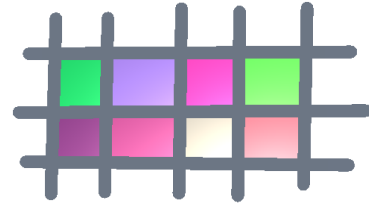


図 4: 閉領域検出結果

4.2 ウェイトペイントを用いた重みの設定

ウェイトペイントとは、3DCG のアニメーションを制作する際に多く用いられるインタフェースであり、「色を塗る」動作によってボーンが与える影響度をメッシュに与えるものである。Autodesk 社の Maya や、オープンソースと知られる Blender など、多くの DCC ツールで採用されている点から、多くのデザイナーにとって馴染みのあるインタフェースであると言える。

4.3 ボロノイ分割による建物の底面形状の決定

ボロノイ分割とは、閉領域内の隣接した母点間を結ぶ垂直二等分線によって各母点の最近隣領域を分割する手法である。本研究では母点の分布を先述したウェイトペイントによって制御し、結果として出力されたボロノイ領域を生成する建物の底面形状として扱うことを検討する。

参考文献

- [1] 「デザイナーの仕事 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://wccfttech.com/gta-v-most-expensive-videogame-history/>
- [3] 宮田 一乗, “ゲームとエンタテインメント技術×第 2 回”, 映像情報メディア学会誌 Vol.63, No.8, pp.1107-1112(2009)
- [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] 宮田 一乗, 伊藤 貴之, 嶋田 憲司, “正方形粒子の最密充填手法を用いた石畳テクスチャの生成”, 情報処理学会論文誌, Vol.42, No.11, pp.2743-2751(2001)