

都市の特徴を表現した建物3Dモデル自動生成

Seki, Hayato / 関, 隼利

(出版者 / Publisher)

法政大学大学院情報科学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 情報科学研究科編

(巻 / Volume)

18

(開始ページ / Start Page)

1

(終了ページ / End Page)

6

(発行年 / Year)

2023-03-24

(URL)

<https://doi.org/10.15002/00026288>

都市の特徴を表現した建物 3D モデル自動生成

Automatic Generation of 3D Models of Buildings Representing a City Feature

関 隼利*

Hayato Seki

法政大学大学院 情報科学研究科 情報科学専攻

Email: hayato.seki.2s@stu.hosei.ac.jp

Abstract—Designers design and create 3D background models for movies or TV games. There, many 3D models with similar features are needed to make variation, but this process needs lots user cost. Therefore, there is a need for user assistance tools that automatically generate 3D models with similar features. The author approaches this problem in two ways: for the user to design the algorithm that generates the 3D model, and to use an adversarial generative network. For the former, we implemented a procedural modeling method that generates a 3D models according to algorithms and parameters. For the latter, we assume that there is a need for a tool that can output similar 3D models based on a single reference, and we investigate the functionality of the method of Wu et al.

1. はじめに

映画やゲームの舞台を 3DCG で表現する際、表現したい都市の特徴を、アーティストが調査しデザインする。3DCG で都市を表現するためには、その特徴をもつ類似した建物 3D モデルを多数作成する必要がある。これを手作業で行うのは、手間がかかる。そこで筆者は、都市の特徴を再現した建物 3D モデル制作について、ユーザ補助を行うツールについて検討、制作している [1, 2]。なお本研究におけるユーザとは、プロシージャルモデリングによる 3D モデル生成の理解があるアーティストのことを指す。

本研究の目標は、都市の特徴を表現した建物 3D モデルを多数作成する際の、ユーザの手間を削減することである。そこで本研究では、都市の特徴を表現した建物 3D モデルを自動生成するユーザ補助ツール（以下、補助ツール）を提案する。筆者は 2 種類の方針でこの目標に取り組む。1 つは建物のプロシージャルモデリング手法である。もう 1 つは、参考となる 3D モデルをユーザが作成し、その 3D モデルを基に敵対的生成ネットワーク (GAN) が類似 3D モデルを生成する手法である。1 つ目の方針について、筆者は江戸の身分に着目した、民家の 3D モデルプロシージャルモデリング手法 [2] を提案した。これについては 3 節にて説明する。2 つ目の方針では、1 つ目の方針での課題を改善するため、Wu らの手法 [3] の実験を行い、その利用方法について検討した。これについては 4 節にて説明する。以上より本研究では、江戸民家の 3D モデルを生成する際、身分により 3D モデルを作り分けるプロシージャルモデリング手法を提案する。またこの手法の課題点解決のため、GAN を用いた手法を利用するための実験を行う。

2. 関連研究

2.1. 都市の特徴の表現

都市の建物は都市ごとに、ある共通の特徴を持つことがある。この特徴を探るため建築学や民俗学では、現地での調査や、現存する資料を基に研究が行われる。また行政が、適切な町並みを定義している地域もある。川越市の例 [4] では、屋根や窓など、建物の概形や造りや材などに適切なものの基準が設けられている。都市の特徴を探る手法として、情報科学の側面では、Doersch らの手法がある [5]。Doersch は、町並み画像のパッチの特徴量をクラスタリングし、その町並みを構成する要素を発見する手法を提案している。Doersch らの実験では、入力した画像内の街灯や窓枠、バルコニーから、画像がどの地域や都市で撮影されたかを推定していた。また建物の造りだけでなく、道路ネットワークやスカイラインも都市の特徴の要素の 1 つである [6, 7]。さらに一部の都市では、社会構造が建物の造りや材に影響することがある。例えば江戸の民家は、その家に住んでいる人の身分により、用いられる材や造りが異なる [8]。

前項の都市の特徴の要素を踏まえると、都市の 3D モデル生成は、大きく分けて 3 つの段階に分けることができる。それぞれ道路ネットワーク、建物の造り、建物の配置である。本研究の前半部では、このうち建物の造りに着目する。

Wonka らは、プロシージャルモデリングで建物 3D モデルを生成する手法を提案している [9]。この手法では面を分割することで、建物の正面（以下、ファサード）を表現する。Nishida らは CNN を利用し、単一画像から建物 3D モデルを生成する手法を提案している [10]。この手法では、建物全体が写った単一画像とユーザ入力の建物輪郭線からパラメータを推定し、建物 3D モデルを自動生成する。CNN によるパラメータ推定を可能にするため、建物形状やファサード、窓などの種類を事前定義する。本研究では、建物を生成するアルゴリズムにて、これらの手法から着想を得た方法で実装を行う。勝村らは、家屋の部品を並べることで江戸時代家屋の 3D モデルを生成する手法を提案している [11]。これは江戸時代の建物の大きさの規格が、3 尺ごとであることに着目したものである。この手法では身分によって、建物の特徴が異なることを考慮していない。

2.2. 敵対的生成ネットワークを用いた 3D モデル生成手法

筆者は、画像内の都市の特徴を変換し、その後画像から建物 3D モデル生成する手法を提案した [1]。この手法では、画像内の建物に見られる都市の特徴を、GAN を用いて変換する。その後変換した画像から、Nishida らの手法を用いて [10] 建物 3D モデルを生成する。

* Supervisor: Prof. Takafumi Koike

3Dモデルを扱うGANも存在する[12]。これらの方法では、学習データを多数用意する必要がある。そのため、ユーザの手間を削減することが目的である本研究では、目的にそぐわない。ここでWuらは、3Dモデルのボクセル化解像度を段階的に変化させて学習することで、単一の3Dモデルのみから3D形状の特徴を学習することができる手法を提案した[3]。そこで本研究の後半部分では、Wuらの手法を利用するための実験を行う。

3. 江戸民家プロシージャルモデリング手法

映画やゲームの背景として都市を3Dモデルで表現する際、その都市の特徴を、建物3Dモデル内で表現する必要がある。特徴をデザインする際に、デザイナーが調査を行う。また過去の都市をモチーフにする場合、時代考証が必要になる場合がある。筆者は、映像やゲームの舞台で取り扱われる江戸について、重要な考証要素の1つである、身分に着目した。身分をパラメータとし、その身分に合った民家の3Dモデルを生成するプロシージャルモデリング手法を検討した。以下、本節で説明する本提案手法を、身分民家PMと表記する。

身分民家PMでは江戸の身分と、建物の造りや材の組み合わせを、身分パラメータで制御し江戸民家の3Dモデル生成する。これにより、ユーザは江戸の身分ごとの建物の違いを作り分ける際に、その手間が削減できると考える。身分パラメータについては、3.2項にて説明する。

提案手法の処理プロセスを図1に示す。プロセスは、事前定義と3Dモデル生成に分けられる。事前定義では、建物の造りや材の設定をユーザが行う。また、それらと身分パラメータの組み合わせを設定する。建物の造りや材については、3.1項にて説明する。3Dモデル生成では、建物3Dモデルを生成する。ユーザは身分パラメータを入力する。入力された身分パラメータを基に、プログラムが3Dモデル生成のためのパラメータ(3Dモデル生成パラメータと表記)を取得する。プログラムはこのパラメータを利用し、建物3Dモデルを生成、出力する。

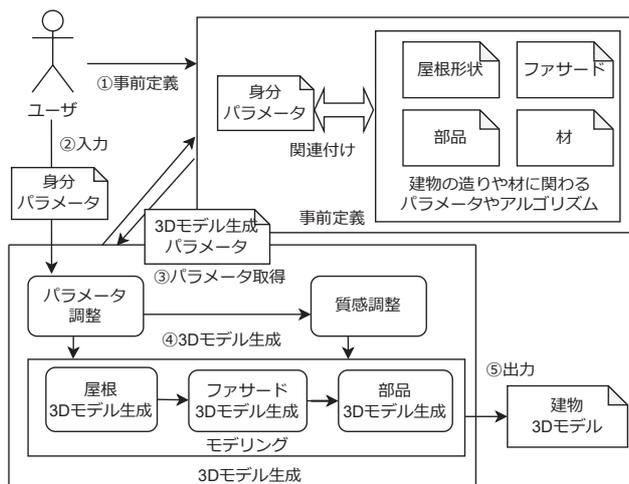


図1. 提案した、江戸民家プロシージャルモデリング手法の処理プロセス。ユーザが、建物の造りや材に関わるパラメータやアルゴリズム、身分パラメータを設計する。またそれらの関連付けを行う。次に身分パラメータをプログラムに入力する。入力を基にプログラムは、3Dモデル生成パラメータを取得する。取得したパラメータを調整し、建物3Dモデルを生成し出力する。[2]の図を一部改変。

3.1. 3Dモデルの設計と生成

3Dモデル生成の工程は2つに分けられる。1つはモデリングで、3Dモデルの形状を生成する。これにより、建物の造りを表現する。もう1つは質感表現で、建物の材を表現する。

Nishidaらの手法[10]に倣い、作成する建物3Dモデルの形状を事前定義する。事前定義では建物形状を、屋根、ファサード、部品の3部分に分類する(以下これらをまとめて、建物の3部分と表記する)。これらそれぞれについて、江戸民家の造りを考慮し、形状と、生成するためのアルゴリズムとパラメータを設計する。3Dモデル生成時は、建物の3部分を順に生成する。

質感表現では事前定義した質感を、3Dモデルに割り当てる。この際、江戸民家の材を考慮する。

3.2. 身分パラメータ

江戸の民家は、その家に住んでいる人(以下、家主)の身分によって、用いられる材や造りが異なる[8]。例えば商人や農民が、武士よりも豪華な住居に住むことは禁じられていた。そのため住居を見れば、その家主の身分が推測できる。

そこで身分ごとの特徴を制御し、適切な建物3Dモデルを生成する身分パラメータを提案する。プログラムは、身分パラメータと関連づいたモデリングのアルゴリズムと、割り当てる質感を、事前定義に基づき選択する。この関連付けは、知識ベースにて設定する。これによりユーザは、身分パラメータを選択するだけで、家主の身分に合った建物3Dモデルを得ることができる。

3.3. 身分民家PMの実装

ユーザはモデリングソフトで建物3Dモデルを作成、利用することを想定する。そのためオープンソースモデリングソフトBlenderと、Blender Python APIを用いて実装を行った。Blenderのバージョンは3.2.0である。

建物の3部分のそれぞれについて、設計を行った。この際、[4, 8, 13]を参考にした。また、江戸東京たてもの園、日本民家集落博物館の展示物を参考にした。配置する3Dモデルの生成方法は2種類ある。1つはプロシージャルモデリングのアルゴリズムを事前設計し、生成する方法である。もう1つは、筆者が事前にモデリングを行い、プログラムが配置する方法である。設計した建物の3部分について、順に説明する。

屋根は、上辺が短い台形2つと、3角形2つで斜面が構成される屋根とした。この屋根は寄棟屋根と呼ばれる。この寄棟屋根について、2種類の形状を設計した。図2内屋根がその形状である。左から順にそれぞれ茅葺き、瓦葺きの屋根として設計した。

ファサードは、面を再帰的に分割することで表現する。これはWonkaらの手法[9]から着想を得たものである。ファサードの分割方法については、図2内ファサードに示す分割方法を設計した。図内の白い面は壁か柱、水色の面は壁や柱以外の部品を示す。実線は常に分割される境界線を示し、破線は分割されない場合がある境界線を示す。ファサードの1F-農、2F-商では、横方向への分割数は固定でない。建物の1階は、図内の1Fとつくファサード分割のいずれかである。2階以上の階は、図内A-共か2F-商のいずれかである。

部品は、図2内部品に示す形状を設計した。窓、扉、^{ひし}庇、壁に分類し、それぞれ形状を設計した。

質感は、Blender シューダノードのマテリアルを利用し、表現した。この際、模様やノイズを表現するテクスチャを組み合わせ、質感を表現した。バイクしたテクスチャ画像を利用して、繰り返し感が生まれてしまう。そこで模様やノイズを組み合わせることで、繰り返し感を避けた質感表現が出来る。実装したマテリアルは、漆喰、土壁、石、竹、鏝、瓦、茅、苔、和紙、木とした。

身分パラメータは、士農工商のうち、農民と商人の2種類を実装した。身分パラメータが農民の場合は、図2内の-農、-共とつくものが生成される。また2階以上の建物は生成されない。商人の場合は、図2内の-商、共とつくものが生成される。また各3Dモデルに割り当てる質感を、表1とした。実装したマテリアルのうち、この表に対応するものを各3Dモデルに割り当てる。

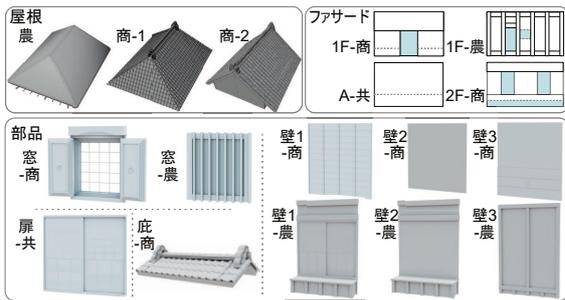


図2. 実装した屋根、ファサードの分割、部品の一覧。ファサード内、白い面は壁か柱、水色の面は壁と柱以外の部品を示す。実線は常に分割される部分を示す。破線部分は、分割されないことがある。[2]より引用。

表1. 身分パラメータと質感の関係。[2]より引用。

身分	質感			
	屋根/庇	壁	窓	扉
商人	瓦	漆喰、木	漆喰	木
農民	茅	土壁	木	木

3.4. 身分民家PMによる建物3Dモデル生成結果

身分民家PMを実装した際のBlender画面が、図3上段である。また上段右の赤丸内が、パラメータ入力画面である。この入力画面は、Blender上のサイドバーに表示される。

生成した建物3Dモデルの例を、図3内下段に示す。図3下段左側が、身分パラメータが農民のとき、右側が商人のときに生成された建物3Dモデルの例である。またこれらの3Dモデルを利用して、シーンを制作した。図4がそのシーン例である。身分に合った建物3Dモデルを、事前定義通り生成できた。

この手法の課題点は2つある。まず、配置する部品をユーザが事前に設計や定義をする必要があり、この作業はユーザ負担がある。また、都市全体を表現するためには、生成した建物3Dモデルを道路ネットワークに配置する必要がある。

4. 身分民家PMの改良

前節の身分民家PMは、ユーザがアルゴリズムやパラメータの設計の全てを定義する必要がある。表現したい特徴を維持したまま異なる形状が出力されるよう、

これらすべてを設計するのは、ユーザ負担が大きい。そのためユーザが用意した単一のリファレンスから、類似した3Dモデルを出力できるツールが求められる。

そこで本節では、単一リファレンスから類似の3Dモデルを出力するユーザ支援ツール制作に向け、単一3Dモデルから特徴を学習するWuらの手法[3]を調査する。Wuらのプロジェクト名Single Shape Gen[14]の頭文字を取り、以降Wuらの手法をSSGと表記する。

4.1. SSG

3DCGで立体を表現する方法のうち、本節では2種類を取り扱う。1つはある点群について、点同士をつなぎ辺や面を定義する、サーフェスモデルである。もう1つは、3D空間を格子状に分割し、それぞれの領域を2値で表現するボクセルである。ここでは前者を単に3Dモデル、後者をボクセルと表記する。また3Dモデルをボクセルに変換することを、ボクセル化と表記する。この際、格子の分割数をボクセル解像度と表記する。

SSGは、単一の3Dモデルからその形状の特徴を学習し、類似した3Dモデルを出力できる。SSGの学習の際、学習用3Dモデルはボクセル化される。このボクセルの解像度を段階的に変化させ、順に学習することで、単一の学習データから特徴を学習できる。

ある3Dモデルを x とする。0以上の任意の整数 N と i （ただし $i \leq N$ ）として、 x をボクセル化したものを x_i とする。 x をボクセル化したボクセル群を $\{x_0, \dots, x_i, \dots, x_N\}$ とする。ユーザ指定の自然数 S_N 、1未満の係数 r としたとき、 x_i は x をボクセル解像度 $S_N \cdot r^{N-i}$ でボクセル化したものである。各 x_i について、ネットワークが形状の特徴を学習する。この際各 x_i での学習は独立している。

4.2. 実験準備

補助ツールへのSSGの利用について考察するため、身分民家PMで利用、生成した3Dモデルを、SSGに学習させ、学習させたモデルの出力結果を観察する。実際にネットワークを学習する前に、学習用3Dモデルの事前準備のための実験と考察を行う。この実験では、3Dモデルの状態が閉じているか否かで、3Dモデルをボクセル化した際のボクセルの状態について比較を行う。その後、各3Dモデルでネットワークを学習し、それぞれ表現できるか実験する。なお実験には、公開されているSSGのスク립ト[14]を用いた。

SSGの学習では、3Dモデルを事前にボクセルに変換する必要がある。そこでまず、3Dモデルをボクセル化する際の条件を確認した。3Dモデルはまず大きさが正規化され、その後ボクセル化される。3Dモデルのボクセル化には、[15]を基にしたボクセル化ソフトbinvox[16]を用いる。ボクセル化された3Dモデルはその後、内部を埋める処理がされる。そこで閉じていない3Dモデルと閉じた3Dモデルをそれぞれボクセル化し、その様子を観察する。図5は、開いた3Dモデルと閉じた3Dモデルをそれぞれボクセル化した状態を比較した図である。図5-a左は開いた3Dモデルであり、半球の底面が閉じていない。一方、図5-b左は閉じた3Dモデルであり、半球の底面が閉じている。図内各項目の右は、3Dモデルをボクセル化した状態であり、右下は右をスライスした状態を示す。図5-b右下のように、閉じた3Dモデルをボクセル化した場合、内部が埋まっている。一方開いた3Dモデルをボクセル化した場合、3Dモデルの表面のみがボクセル化される。そこでファサードや部

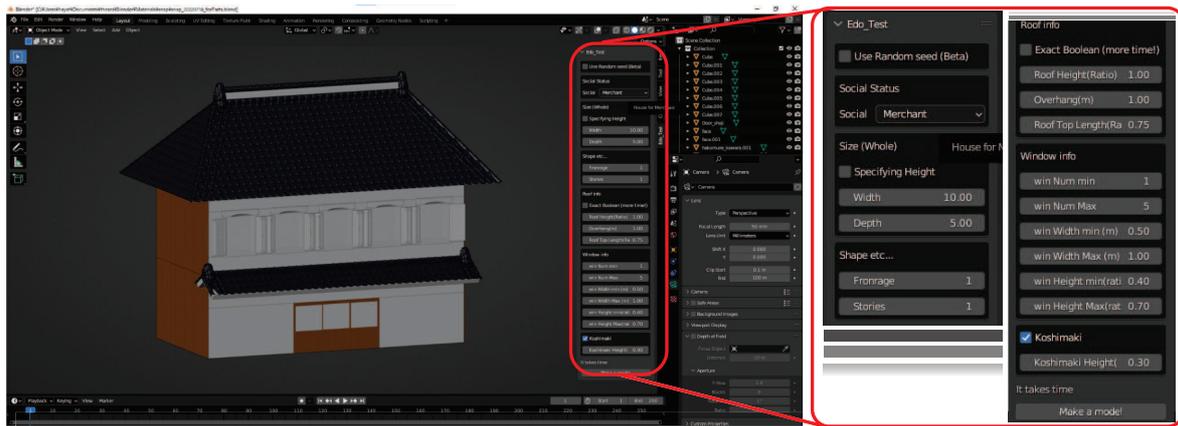


図 3. 本手法のプログラムを実装した Blender 画面 (上段) と, 生成した建物 3D モデル例 (下段). 上段-右) パラメータ入力部分の拡大図. [2] より引用.



図 4. 生成した建物 3D モデルを利用したシーン作成例. [2] より引用.

品などの一部の 3D モデルは, 筆者が手動で 3D モデルを閉じてからボクセル化することとする. なお手動での 3D モデル編集には, 3D モデルソフト Blender を用いる.

前項の通り, SSG では 3D モデルをボクセルとして扱う. そのため学習に用いる 3D モデルについて, 表現できる細部の大きさを考える必要がある. 3D モデルある軸方向の長さ $W_{LearnData}$, ボクセル解像度 R , 0 より大きい任意の値 M とすると, 表現できる幅 $W_{Represent}$ は下記の式 1 で表現できると考える.

$$W_{Represent} = \frac{W_{LearnData} \cdot M}{R} \quad (1)$$

例えば $M = 1$ のとき, 幅 $\frac{W_{LearnData}}{R}$ の範囲で 3D モデルの有無を表現できる. しかしこれでは, 幅が変化する模様が表現出来ない恐れがある. そこで M を掛けることで, その幅の変化に対応することができる. なお以降, ボクセル解像度 $R = 256$ とする.

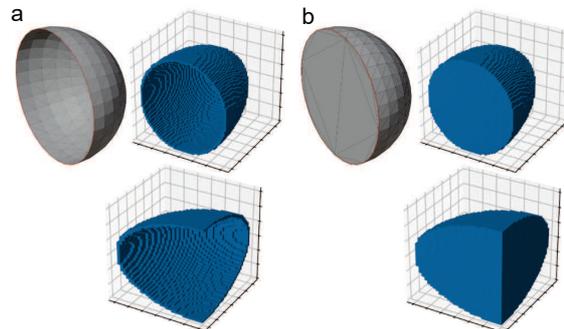


図 5. ボクセル化のテスト. a) 半球の底面を閉じていない状態, b) 半球の底面を閉じた状態. それぞれの左は 3D モデルを示し, 右はボクセル化した状態, 右下は右をスライスした状態を示す. b 右下では内部が埋まっている.

4.3. 身分民家 PM 改良に向けた SSG の実験

まず学習に用いる 3D モデルの種類と大きさについて考察を行う. 学習, 生成する 3D モデルの単位は, 複数の構成要素を含むものと, 単一要素のもの 2 通りを考える. 複数の構成要素を含むものは, 建物全体, 建物の 3 部分のそれぞれとする. 単一要素のものは, 部品とする. これは建物の 3 部分は, それぞれ部品が組み合わさり構成されており, 建物全体は建物の 3 部分で構成されているためである. また大きさは, 建物全体の 3D モデルの幅を 7.0m とする. $M = 2$ のとき, $W_{Represent} \approx 5.5 \times 10^{-2}m$ であり, 今回の目的には適している. なお学習に用いる 3D モデルは, 身分民家 PM で生成し一部編集したものや, 事前に用意した 3D モデルを用いる.

SSG に建物 3D モデルを学習させ, SSG が生成した

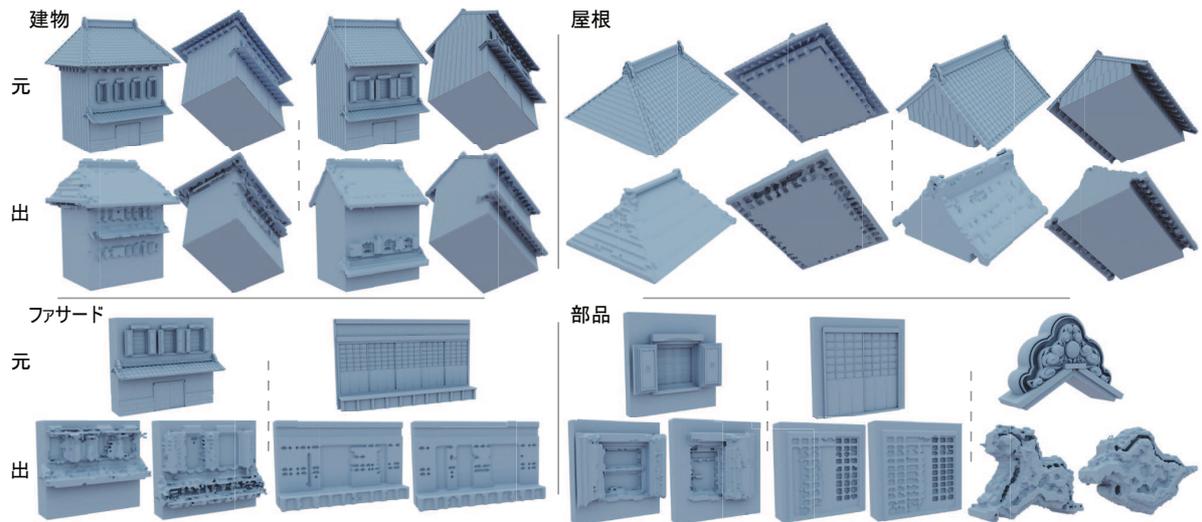


図 6. 建物, 屋根, ファサード, 部品の 3D モデルを SSG に学習させ, 出力させた 3D モデルの比較. それぞれの項目のうち, 上段は学習に用いた元の 3D モデル, 下段は SSG が生成した 3D モデルである.

3D モデルを図 6 左上に示す. 学習に用いた 3D モデルは, 身分民家 PM で生成し, 一部編集したものを用いた. 図 6 内, 破線より左は寄棟屋根の商人, 右は切妻屋根の商人の建物 3D モデルである. また各 3D モデルは 2 視点からレンダリングした結果を示す. 学習に用いた建物 3D モデルと形状を比較すると, いずれも建物の概形は類似している. 特に寄棟屋根や切り妻屋根は, その概形が表現されている. また, 庇や窓, 屋根裏の垂木も表現されている. しかし, 建物側面の凹凸や, 入口などの細部は表現されていない.

SSG に屋根 3D モデルを学習させ, SSG が生成した 3D モデルを図 6 右上に示す. 図 6 右上内, 破線より左は寄棟屋根, 右は切妻屋根の 3D モデルである. 学習に用いた部品 3D モデルと形状を比較すると, 寄棟屋根や切り妻屋根の概形が表現されている. また, 屋根裏の垂木や屋根頂上部両端の瓦の概形も表現されている. しかし建物 3D モデルでは表現されていた, 切妻屋根斜面と建物側面部の段差は表現されていない. また瓦や建物側面部の細部は表現されていない.

SSG にファサード 3D モデルを学習させ, SSG が生成した 3D モデルを図 6 左下に示す. 学習に用いた部品 3D モデルと形状を比較すると, 庇や窓, 腰掛の概形は表現されている. また, 図内右のドアの様子は表現されている. 図内左のドアは表現されていない.

SSG に部品 3D モデルを学習させ, SSG が生成した 3D モデルを図 6 右下に示す. 図内左下を破線で区切った 3 領域のうち, 左は観音開きの窓, 中央は障子のドア, 右は影盛の 3D モデルである. 学習に用いた部品 3D モデルと形状を比較すると, 観音開きの窓と障子のドアの概形は, 表現されている. 観音開きの窓は, 扉部分が片側のみ生成される例がある. 障子のドアは, 格子の様子が表現されている. 影盛は, 全体の概形にバリエーションがある. また輪郭より少し内側の溝は表現されている.

以上の実験より, 江戸民家の 3D モデルを SSG に学習させたところ, 概形や一部の大まかな形状が類似した 3D モデルが生成された. 建物やファサードなど, 複数の要素を含む 3D モデルでは, 生成された 3D モデル内の窓や庇などの個数や配置に, バリエーションがあった. 一方窓やドア, 影盛などの単一要素の 3D モデルでは, その概形にバリエーションが生まれることがあ

た. ただし, その窓やドアなどの細部が表現されることがあった. これは, SSG が 3D モデルをボクセルとして扱っていることが由来だと考える. 学習前に 3D モデルをボクセル化する場合, 解像度や段階を細かくするなど, パラメータ調整することで改善する可能性がある. ただし, SSG の論文内では, 現状のボクセル化の解像度は 256 までと記載があり, SSG のネットワークそのものを改善する必要もあると考える. 概形は表現されているため, 細部が表現されれば, 補助ツールへの利用が可能になると考える.

4.4. 都市生成に向けた SSG の追加実験

全項では, 江戸民家の建物や構成要素に着目し, SSG の利用を試みた. 本項では追加実験として, 都市の 3D モデルを学習データとして用い, 都市全体の 3D モデル生成が可能か実験を行った.

まず都市 3D モデルのうち, ある地点を中心にどれほどの範囲のモデルが利用可能か, 考察を行う. SSG から出力される都市 3D モデルに求めるのは, 建物や道路である. これは, 建物や道路の輪郭や境界が明確であれば, 道路網やスカイラインなどの都市の特徴抽出に繋がると考えるためである. そこで式 1 について, $W_{Represent}$ を細い道路の幅として, 扱える都市 3D モデルの幅を考える. 1 車線分の道路員を 3.5m のとき, 片側 1 車線道路では道路幅は約 7.0m とする. このとき $W_{Represent} = 7.0$ なので, $W_{LearnData} \approx \frac{1.8}{M} 10^3$ となる. ここで $M = 0.5, 1, 2$ とすると, $W_{LearnData} \approx 3.6, 1.8, 0.9 \times 10^3$ m となる. ここで底面の一辺がこれらの値となる正方形領域で都市 3D モデルを切り抜き, それぞれを学習データとする. ここで都市 3D モデルは, 国土交通省が主導し, 都市データを公開するプロジェクトである PLATEAU[17] 上で公開されている 3D モデルを利用する.

SSG に都市 3D モデルを学習させ, SSG が生成した 3D モデルを図 7 に示す. 学習に用いた都市は 2 つで, 大阪と東京である. それぞれ, 梅田駅と東京駅を中心に, 約 $3.6, 1.8, 0.9 \times 10^3$ m 四方の領域を抽出し, それぞれ学習に用いた. またそれぞれについて, 建物群と道路網, 建物群のみ, 道路網のみ, の 3 パターンを用意して, 学習に用いた. 学習に用いた都市 3D モデルと形状を, 順に比較する. 抽出した範囲が狭いほど, 建物や道

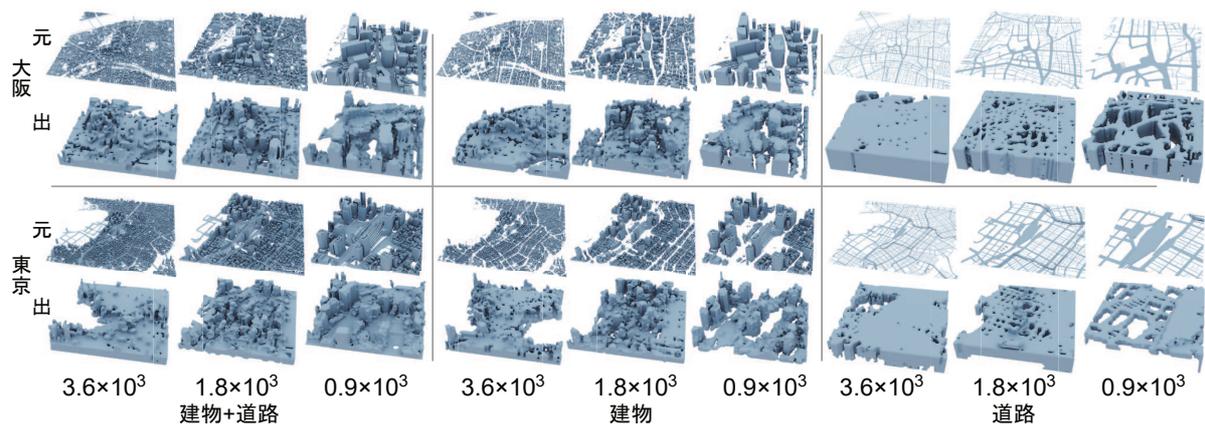


図7. 都市3DモデルをSSGに学習させ、出力させた3Dモデルの比較. 上半分は大阪, 下半分は東京の3Dモデルである. 縦線で3つに分けた領域のうち, 左が建物群と道路網, 中央が建物群のみ, 右が道路網のみ, の3Dモデルである. それぞれの項目の上段は学習に用いた元の3Dモデル, 下段はSSGが生成した3Dモデルである. また各項目の元3Dモデルは, 左から順に 3.6×10^3 , 1.8×10^3 , 0.9×10^3 m 四方の領域で, 都市3Dモデルを抽出したものである.

路網が繋がらずに生成された. 特に 3.6×10^3 m では, 建物や道路網の細部が一体化している.

都市の3DモデルをSSGに学習させたところ, 生成される3Dモデルは, 建物や道路が結合して出力された. これは4.3項と同様の要因であると考えられる.

5. まとめ

都市の特徴を表現した建物3Dモデルを自動生成する補助ツール制作を目的とし, 2つの方針に沿って実装や実験を行った. 1つは再現する特徴を踏まえたまま3Dモデルを生成するアルゴリズムを, ユーザーが定義する方法である. もう1つは, 参考となる3Dモデルをユーザーが作成し, その3Dモデルを基に深層学習モデルが類似3Dモデルを生成する手法である.

1つ目の方針について, 筆者は江戸の民家に着目したプロシージャルモデリング手法[2]を実装した. ユーザーが事前に設計した身分に対応する建物3Dモデルを, 多数生成することができた.

さらに2つ目の方針について, 敵対的生成ネットワーク(GAN)を用いたWuら[14]手法の実験を行った. 試した3Dモデルの種類は, 建物, 屋根, ファサード, 部品, 都市である. いずれも, 学習に用いた3Dモデルの概形は, 生成された3Dモデルの概形と類似していた. しかし細かな模様は表現されなかった. パラメータ調整やネットワークの改良によって, 細部が表現されれば, 補助ツールへの利用が可能になると考える.

参考文献

- [1] 関隼利, 小池崇文, “異なる都市間の建物スタイル変換の基礎検討”, 第83回全国大会講演論文集, 2021.
- [2] 関隼利, 小池崇文, “身分パラメータを導入した江戸民家のプロシージャルモデリングの検討”, 第27回バーチャルリアリティ学会大会論文集, 2022.
- [3] R. Wu and C. Zheng, “Learning to generate 3d shapes from a single example,” *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, 2022.
- [4] 川越市 都市計画部 都市景観課, “川越伝建地区まちづくりガイドライン(改定版)”, 2020.
- [5] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, “What makes paris look like paris?” *ACM*

Transactions on Graphics (SIGGRAPH), vol. 31, no. 4, pp. 101:1–101:9, 2012.

- [6] Z. Liu and S. Zhao, “Characteristics of road network forms in historic districts of japan,” *Frontiers of Architectural Research*, vol. 4, no. 4, pp. 296–307, 2015.
- [7] 梅津里香, 浅野純一郎, “都市のスカイラインのコントロール手法に関する研究”, 日本建築学会技術報告集, vol. 18, no. 38, pp. 319–324, 2012.
- [8] 太田博太郎, 新訂図説日本住宅史 彰国社, 1971.
- [9] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky, “Instant architecture,” *ACM Transactions on Graphics*, vol. 22, no. 3, 2003.
- [10] G. Nishida, A. Bousseau, and D. G. Aliaga, “Procedural modeling of a building from a single image,” *Computer Graphics Forum*, vol. 37, no. 2, 2018.
- [11] 勝村大, 谷村知洋, 青木香織, 阿部祥子, 田口博之, 森谷友昭, 高橋時市郎, “3d cgによる歴史的町並み復元のための家屋生成手法”, 画像電子学会誌, vol. 36, no. 4, pp. 382–389, 2007.
- [12] J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, and S. Fidler, “Get3d: A generative model of high quality 3d textured shapes learned from images,” in *Advances In Neural Information Processing Systems*, 2022.
- [13] 米山勇, “江戸東京たてもの園でわかる日本建築史 – 江戸の民家から昭和の木造モダニズム建築まで –”, 公益財団法人 東京都歴史文化財団 江戸東京たてもの園, 2020.
- [14] ChrisWu1997. Github - chriswu1997/singleshapegen. [Online]. Available: <https://github.com/ChrisWu1997/SingleShapeGen>
- [15] F. S. Nooruddin and G. Turk, “Simplification and repair of polygonal models using volumetric techniques,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 2, pp. 191–205, 2003.
- [16] P. Min. binvox. Patrick Min. [Online]. Available: <http://www.patrickmin.com/binvox>
- [17] 国土交通省. Plateau [プラトー]. 国土交通省. [Online]. Available: <https://www.mlit.go.jp/plateau/>