

# 国土地理院データを用いた自動運転のための 仮想テストコース開発

速水郁海<sup>1</sup> 井上虎<sup>1</sup> 鈴木大暁<sup>1</sup> 別府瞭<sup>1</sup> 桑名祐弥<sup>1</sup> 瀬戸直也<sup>1</sup>  
巽竜雅<sup>1</sup> 川合康央<sup>1</sup>

**概要:** 自動運転車や電気自動車の制御アルゴリズムの開発にあたり、実車を用いたテストでは試行回数に制限がかかるため、仮想空間による試験環境の整備が試行されている。一方で、仮想空間で車両テストを行う場合、試行回数の制限問題は解消するが、コンピュータ上での仮想空間内に、現実環境を完全に模した環境を構築することは、大容量のモデルとなり、またその開発には多額のコストが必要となる。本研究では、国土地理院のオープンデータと、ゲームエンジンなどの安価な開発環境を組み合わせることによって、自動車の制御アルゴリズム開発のためのシミュレータに必要な大規模な都市空間モデルを、外部環境のアセットとして開発したものである。

## 1. はじめに

現在、自動運転車や電気自動車の制御アルゴリズム開発において、実車両を用いた試験は、正確な数値を取得できるものの、都市部での走行環境の確保や車両の準備などで多額のコストがかかり、また、試験には実時間が必要となるため、その試行回数は限られている。そこで現在では、コンピュータを用いた仮想空間によるシミュレーションを用いた開発環境が試行されている。シミュレータは、コンピュータによる計算で行われるため、走行環境や試験回数に制限がなくなる。また、特定の演算を高速に処理することが可能であるため、実時間に依存しない試験を行うことが可能である。

一方で、今日の自動運転車や電気自動車では、LIDAR (Laser Imaging Detection and Ranging) 等を用いて、現実空間上で実際に走行して情報を取得し、傾斜や路面状況などの道路データから走行効率を計算するなどの制御を行うため、実環境に即した環境での試験が必要となってきた。しかし、コンピュータ上の仮想環境において、現実環境を完全に模した環境を構築するのは、非常に手間と人手にかかる作業であるため、別途多額のコストが必要となっている。

大島ら[1]は、CG 画像と実画像を合成することによって、効率的に臨場感の高い仮想環境を作成できるとしている、しかし、照明条件の変更や車両の配置を変更できないなど、シミュレーションとしての汎用性に欠ける部分が多い。また、小野らの研究[2]によるモデルベース CG による描画の再現では、自由度は高いが現実感が薄く、画像ベース CG による再現では、高い現実感があるが画像データとして保持していない状況の再現が難しいとしている。ここでは、モデ

ルベース CG では現実感が薄いとされており、これは精密な CG 構築と視点からの距離に応じた CG の簡略化などの点から、現実感を持たせるためには高度なモデリング作業が必要とされている。これらの研究では、道路の再現性として、車両走行コースなどの自動化などが出来ていない点から、モデルベースで作った CG のみで開発するよりも、実画像を交えて開発した方がより現実感が再現できるとしている。

本研究では、自動運転車や電気自動車の制御アルゴリズムの開発テスト環境だけでなく、その開発環境から見直し、オープンデータやゲームエンジンを活用することによって、より安価に現実を模した仮想道路環境を構築可能なのではないかと考えた。そこで、国土地理院の基盤地図情報など大規模なオープンデータと、ゲームエンジンやフリー・オープンソースソフトウェアなどの低コストで実現できる開発環境を組み合わせることにより、シミュレータに必要な情報を持つ大規模な都市空間モデルを、外部環境のアセットとして開発することとした。

## 2. 開発手法

本開発における対象地区として、市街地モデルの低コスト再現を実現するため、神奈川県藤沢市に位置する湘南台駅周辺の道路を選定した。本地区は、さがみロボット産業特区に位置し、自動運転の実証実験なども行われているため、本システムを用いたシミュレーション結果を、実空間での走行結果と比較することが可能となる。

本システムでは、地理情報を取得するため、国土地理院の基盤地図情報ダウンロードサービスを使用した[3]。本サービ

<sup>1</sup> 文教大学  
Bunkyo University.

スでは、任意の地域における様々な地図情報を、汎用的な XML データで取得することが可能である。ダウンロードした XML データを、基本項目と数値標高モデルの表示ソフトウェアである基盤地図情報ビューア[4] に読み込み、これを Shape 形式のデータに変換して書き出すこととした。次に、地理情報システムのオープンソースソフトウェアである QGIS[5]に、変換した Shape ファイルを、新規ベクタレイヤとして読み込んで表示させることとした。QGIS 上では、マップキャンパスにタイルマップを追加するためのプラグインである TileLayerPlugin[6], や JavaScript ライブラリを使用した 3D 視覚化プラグイン Qgis2threejs[7], three.js, WebGL などを使用し、高さ情報を持った三次元の地理情報データを作成した。QGIS 上で加工された三次元データは、地形、道路縁、建築物の外周線の 3 つのレイヤとして、STL 形式のデータとして書き出した (図 1)。

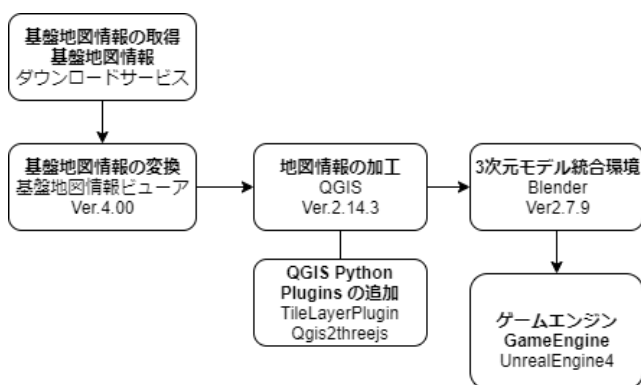


図 1 本システムの開発フロー

これらの加工された三次元地理情報データを、オープンソースソフトウェアの 3DCG 統合環境である Blender[8]上に読み込んだ。取り込んだ道路縁データは、道路縁に不要なポリゴンが生成されている状態で読み込まれるため、事前にこれらのポリゴンを消去して、縁データを整備する必要がある。その後、道路面に相当する面にポリゴンを生成し、高低差のある市街地の道路面を詳細に作成した (図 2)。本手法を用いることにより、広範囲の街並みを一度に作成することが可能となる。しかし、道路縁データを読み込む際に生成されているポリゴンは、道路線が繋がっていない、ポリゴンが不自然に重なっている等の問題が発生するため、手動で整理する必要がある (図 3) (図 4)。この作業を施さずに面張り作業を行った場合には、何重にも重なったポリゴンが発生する、面が裏面にも生成してしまう、不自然な空洞が開いてしまう等の問題が発生する。

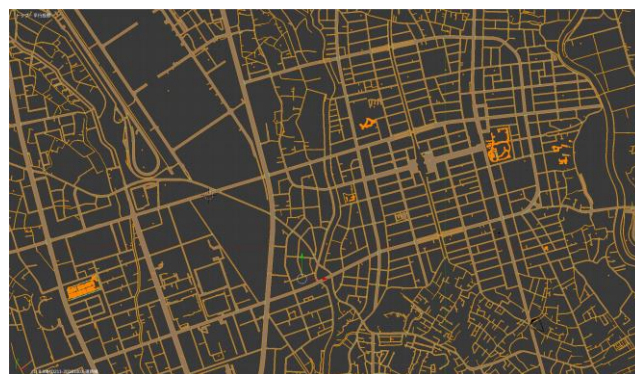


図 2 道路縁データから道路面を生成したモデル

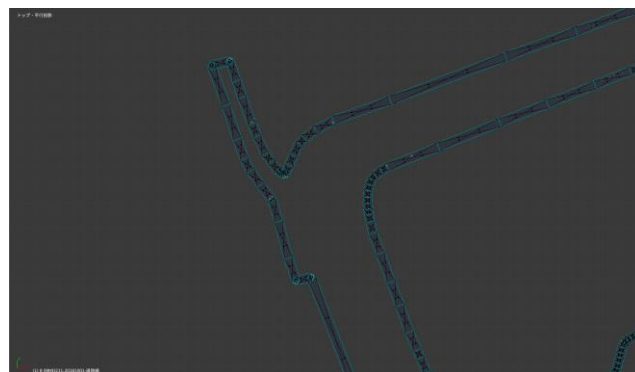


図 3 道路縁データから生成されたポリゴン

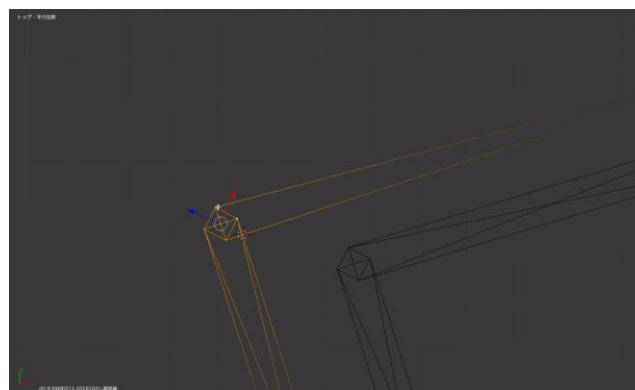


図 4 不自然に重なった不要なポリゴン

Blender 上で生成された三次元モデルデータは、FBX 形式のデータとして書き出し、これをゲームエンジン Unreal Engine4[9]に読み込み、仮想車両走行のためのシステムを用意することとした (図 5)。操作の方法については、キーボード操作のほかに、入力インターフェースとして、ハンドルコントローラを用いることとし、ハンドルとペダルによる操作を、仮想空間内の車両の動きと同期させた (図 6, 7)。また、比較のために簡易な自動運転アルゴリズムを搭載した車両を用意し、同じ目的地に対して走行する NPC (Non Player Character) を用意することとした (図 8)。

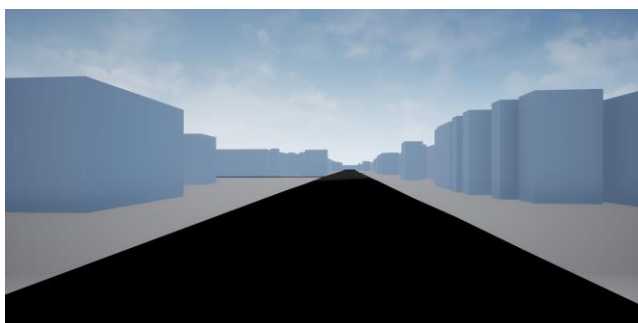
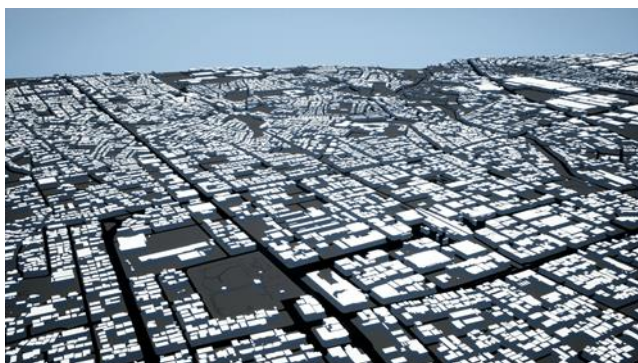


図5 ゲームエンジン上に読み込んだ3次元の地形・道路・建物データ



図6 一人称視点による車内から見た外界環境



図7 三人称視点による車外から見た外界環境

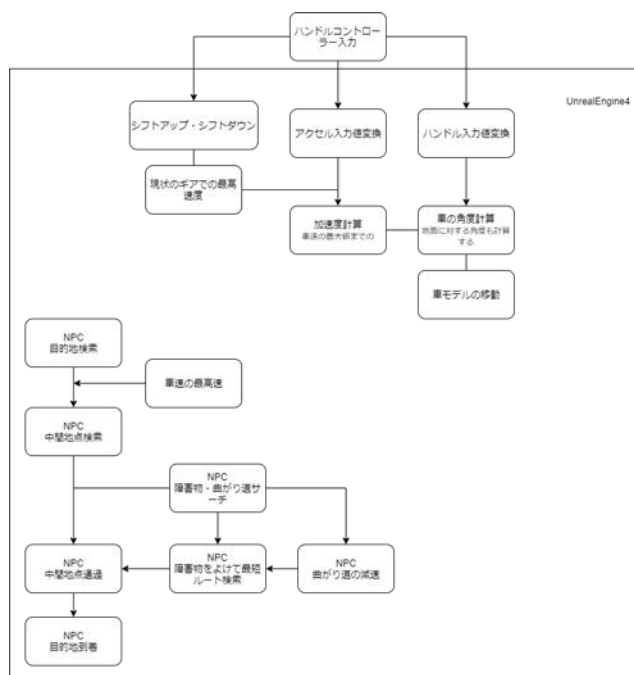


図8 システム構成図

道路の再現性としては、道路形状だけでなく、高低差のある市街地道路環境として、精度の高いオブジェクト配置にすることができた。一方で、道路面の作成において、道路面に相当する面のポリゴン生成と、その縁にある不要なポリゴンの詳細な編集等が、完全な自動化ができていないため、開発にいくつかの手作業による工程が残っている。そのため、道路条件の変更等、大規模なモデルの仕様変更が可能な要素が限られている。テクスチャ等の作成に関しても、手動作業によって開発が行われているため、それなりの作業コストが必要となる。また、本システムで用いたデータセットには、建物の高さのデータが含まれていないため、全て一定の高さを持った建物モデルとなっている。詳細な建物の形状を確認するためには、実際に現場でデータを取得しなければならないため、同じ形状の建物、景色を作るためには、より大容量の建物データセット、あるいは、写真や動画など、実データから計算して高さを割り出すシステム等を開発する必要がある。

一方で、ゲームエンジンを使うことによって、光源や日差し、天候などの環境条件に関しては、簡単に変更することが可能なものとなった。NPCに関しては、現在の開発環境には、処理速度から限られた台数のものとなったため、アルゴリズムの改良が必要である。

今後、データの加工、変換に必要な手順のうち、プログラムで実行可能な作業について検討し、自動化システムを作成していくこととする。また、車両の操作性に関しては、ゲームエンジンを使用している為、限定的な操作のみとなっている。これは、実車両の操作感をリアルに再現したのではなく、ゲームの操作に近い感覚のものとなっていることが課題である。今後、操作に応じた車両挙動を模した値をリ

リアルタイムにシミュレーションを行う HILS(Hardware-In-the-Loop-Simulation)を用いることで、現実の操作感と車両挙動を再現していくこととする。

本開発を行う際には、その対象範囲にもよるが、使用する都市空間データが大規模なものとなるため、相応の性能を持った開発環境を用意しなければならない。特に、CPU (Central Processing Unit) だけでなく、GPU (Graphics Processing Unit) に、高負荷に耐えうる性能が必要となる。しかし、本システムでは、専用のハードウェアではなく民生機での開発及び運用が可能であり、さらに開発環境もゲームエンジンやフリー・オープンソースソフトウェアを使用し、データもオープンデータを活用するなど、低コストでシステムを実現しており、従来の開発環境と比して、開発コストは大幅に低いものとなっているため、シミュレータ導入の際におけるコスト面での障壁を大幅に下げることが可能となった。

### 3. まとめ

本研究は、自動運転車や電気自動車の制御アルゴリズム開発において、実車両を用いた高コストな試験による試行回数の制限、それに伴うコンピュータでの仮想化における、現実環境を完全に模した環境の構築のコスト等の課題を解決するために、オープンデータやゲームエンジンなどの安価な開発環境を組み合わせることによって、シミュレータに必要な情報を持つ大規模な都市空間モデルを外部環境のアセットとして開発を行ったものである。本システムでは、特に、オープンデータの地図を用いて、市街地における高低差のある道路モデルを、低コストで実現したことが新規性である。

一方で、開発したシステムは、一定程度の外部環境の再現ができたとともに、開発工程や操作感にいくつかの課題が見られることとなった。今後、道路モデルデータ作成時におけるポリゴン面作成の自動化や、ガードレール等の道路構成要素の自動配置など、手作業で行われていた工程の手順をもとに、自動化可能な個所を支援するプログラムの開発を行う。また、操作と車両挙動の関係についても HILS などの外部データをリアルタイムで取得することによって、改善を図っていくこととする。本システムの、開発準備段階には、手動での道路面のデータ整理など、ある程度のコストが必要となる。しかし、その後の開発には、オープンデータやゲームエンジンなどの安価な開発環境を多用しているため、システム全体で見た際、従来のシミュレータに比して、大幅なコスト削減効果が出ていると言える。今後、HILS を用いた走行実験を行い、外界環境シミュレータとしての精度を確認するとともに、本システムの特徴を活かしたシミュレータとして改良を継続していく。

**謝辞** 本研究は JSPS 科研費 JP19K12665 の助成を受けたものです。

### 参考文献

- [1] 大島大輔, 山田康右, 竹之内篤, 山下浩行, 中野公彦, 鈴木高宏, 小野晋太郎, 平沢隆之, 洪性俊, 杉町敏之, 鄭仁成, 須田義大, 大口敬, 堀口良太, 白石智良. ドライビングシミュレータに対するニーズ及び先進的関連技術に関する調査研究. 生産研究, 2015, vol.67, no.2, pp. 87-92.
- [2] 小野晋太郎, 杉町敏之, 貝塚勉, 坂井康一, 和田健太郎, 平沢隆之, 大口敬, 須田義大, 大石岳史, 中野公彦. 仮想交通実験環境における異常時を含めた視覚環境等の再現性にかかる課題の研究. 生産研究, 2018, vol.70, no.2, pp. 63-68.
- [3] “基盤地図情報ダウンロードサービス - 国土地理院”. <https://fgd.gsi.go.jp/download/menu.php> (参照 2019-12-22) .
- [4] “基盤地図情報ビューア”. <https://fgd.gsi.go.jp/otherdata/tool/FGDV.zip> (参照 2019-12-22) .
- [5] “QGIS プロジェクトへようこそ!”. <https://www.qgis.org/ja/site/> (参照 2019-12-22) .
- [6] “TileLayerPlugin”. <https://github.com/minorua/TileLayerPlugin> (参照 2019-12-22) .
- [7] “Qgis2threejs”. <https://github.com/minorua/Qgis2threejs> (参照 2019-12-22) .
- [8] “Blender”. <https://blender.jp> (参照 2019-12-22) .
- [9] “Unreal Engine 4”. <https://www.unrealengine.com> (参照 2019-12-22) .