

# 歩行空間における自律走行のための セマンティック・セグメンテーションの教師画像自動生成手法

Automatic Image Labeling Method for Semantic Segmentation  
toward Autonomous Driving in Pedestrian Environment

小野塚 友也<sup>†</sup>  
Yuya Onozuka

松實 良祐<sup>†</sup>  
Ryosuke Matsumi

小竹 元基<sup>†</sup>  
Motoki Shino

## 1 はじめに

近年、公共交通機関が減少する地域における移動支援のひとつとして、パーソナルモビリティの自律移動が注目されている [1]. 歩道や生活道路のような歩行空間を対象とした自律移動システムは多数提案されており、LRF 等のセンサを用いて構築された地図情報に基づく厳密な自己位置推定を必要とする手法 [2] が多い。しかし、すべての道路において事前に詳細な地図を整備するには、多くの時間とコストが必要であり、導入障壁が大きい。それに対し、未知の屋外環境での自律走行を目指し、詳細な事前地図を必要としない手法も提案されている [3].

詳細な事前地図を必要としないシステムでは、走行可能領域を認識するため、LRF 等により走行環境の幾何形状から障害物を検出し、走行不可領域として分類する手法が一般的である [3]. しかしながら、車道と歩道が混在する環境のように、物理的に明確な走路境界がない環境も存在し、そのような環境においては物理的特徴を検出する手法のみでは不十分である。財前ら [4] により提案された走行領域認識システムでは、画像上の対象をピクセル単位で分類する Semantic Segmentation を用い、走行領域を走行推奨領域、走行可能領域、走行不可領域の 3 属性に分類することを可能とした。

本研究では、歩行空間における詳細な事前地図を必要としないシステムを目指し、Semantic Segmentation により検出した走行推奨領域に基づく自律移動システムを対象とする。このシステムでは図 1 に示すように、既存のデジタル地図情報に基づく大域経路計画と大まかな自己位置推定、および、検出した走行推奨領域に基づき局所経路計画を行い、運動計画の段階で適切に安全を補償する操作量を決定し、それを目標に車体を制御することで自律移動を実現する。

## 2 本研究の位置づけ

財前らの研究 [4] では、走行推奨領域を分類するため、Semantic Segmentation の手法として Convolutional Neural Network (CNN) を用いたアーキテクチャである SegNet [5] を用いた。教師画像として、対象環境で撮影した画像 300 枚に対して作業員が 3 つの属性にラベリングしたデータセットを使用した。この教師画像を用いて学習モデルを生成した結果、対象環境においてはもっともらしい結果が得られることが確認された。しかし、対象場面の拡張や性能向上のため、追加の教師画像を生成するには、多くの時間と作業者の負担を要する。この課題は既に類似研究でも指摘されており、人手に頼らず教師画像を生成する手法も提案されてきた。具体的には、時系列の画

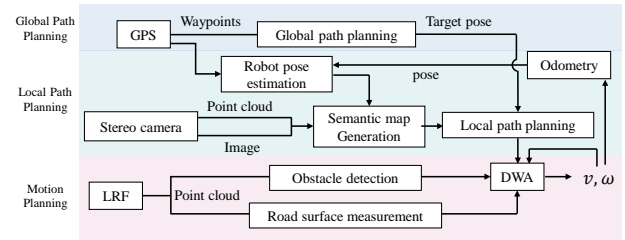


図 1 自律走行のためのシステムアーキテクチャ

像データと自己位置情報を用い、経験に基づき走行可能な経路を分類する手法 [6] や、深度センサにより平面を検出し、走行可能領域として分類する手法 [7] があるが、歩行空間に適用した例は少ない。

本論文では、物理的に明確な走路境界が存在しない環境を含む歩行空間における、走行推奨度に基づく自律走行の実現を目指し、Semantic Segmentation のための教師画像を自動で生成する手法を提案する。また、本研究における自律移動システムは、走行推奨度に基づき経路計画をした上で、操作量決定の段階で安全性を担保する構成になっている。そのため、自律走行を実現するための走行推奨領域検出では、電動車いすが走行しうる領域を広く検出可能な性能が求められる。そこで本論文では、より広い領域を走行推奨領域として認識することを目的とし、それを達成するための教師画像を自動で生成する際の機能検証を行う。また、自動で生成した教師画像を用いて学習を行い、教師画像の工夫次第でより広い領域を走行推奨領域として認識できることを確認する。

## 3 教師画像自動生成手法

本章では、教師画像を自動で生成する手法と、その機能検証を行った結果について述べる。なお、本システムでは財前ら [4] の研究のように、3 つの属性にラベリングされた教師画像を自動的に生成することを最終目標としているが、本論文ではその一環として、走行推奨領域とそれ以外の 2 分類の教師画像を自動で生成するシステムを構築した。ここで、Barnes ら [6] の手法を参考に、実際に安全に走行できた領域を走行推奨領域としてラベリングする。

### 3.1 走行推奨領域の自動ラベリング手法

時系列の画像データと車両の位置情報を用い、実際に走行できた領域をラベリングする手法について述べる。ある時刻  $t$  の画像  $I_t$  に対し、時刻  $t+k$  の車輪の接地点の 3 次元位置を投影することを考える。ここで、接地点の 3 次元位置は時刻  $t$  のカメラ座標系  $C_t$  における位置とし、 ${}^{C_t} \mathbf{X}_{t+k} = [X, Y, Z]^T$  と表

<sup>†</sup> 東京大学 The University of Tokyo

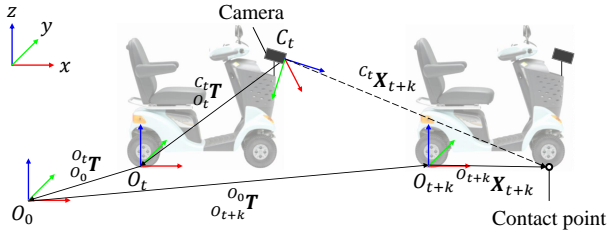


図 2  $C_t \mathbf{X}_{t+k}$  の導出に必要なオドメトリと車両幾何の関係



図 3 実験車両とセンサの取り付け位置

す。また、 $C_t \mathbf{X}_{t+k}$  の同次座標を  $C_t \tilde{\mathbf{X}}_{t+k} = [X, Y, Z, 1]^T$  と表す。このとき、カメラの透視投影行列  $\mathbf{P} \in \mathbb{R}^{3 \times 4}$  を用いると、画像上の投影点  $u, v$  は、同次座標  $\tilde{\mathbf{m}} = [u, v, 1]^T$  を用いた式 (1) より求められる。ただし、 $s$  はスケールファクタであり、左辺の第 3 要素が 1 となるように定める。

$$s\tilde{\mathbf{m}} = \mathbf{P} C_t \tilde{\mathbf{X}}_{t+k} \quad (1)$$

また、 $C_t \mathbf{X}_{t+k}$  は図 2 より式 (2) のように表される。ただし、 $O_0, O_t, O_{t+k}$  はそれぞれオドメトリより求められる車両の初期座標、 $t$  秒後の座標、 $t+k$  秒後の座標を表し、 $\mathbf{T} \in \mathbb{R}^{4 \times 4}$  は座標変換行列を表す。 ${}^{C_t} \mathbf{T}, {}^{O_{t+k}} \mathbf{T}$  は車両幾何、 ${}^{O_t} \mathbf{T}, {}^{O_0} \mathbf{T}$  はオドメトリよりそれぞれ求められる。

$$C_t \mathbf{X}_{t+k} = {}^{C_t} \mathbf{T} {}^{O_t} \mathbf{T} {}^{O_0} \mathbf{T} {}^{O_{t+k}} \mathbf{T} {}^{O_{t+k}} \mathbf{X}_{t+k} \quad (2)$$

オドメトリには、車両モデルに依存しない 3 次元オドメトリである visual odometry として、ステレオカメラによる ORB-SLAM[8] を採用した。また、visual odometry は画像上の特徴点を十分に検出できない場合に自己位置をロストしてしまうため、その際は wheel odometry により補間する。

時刻  $t$  における画像  $I_t$  に対し、式 (1), (2) を用いてある時刻の左右の前輪の投影点と、その単位時間前の左右輪の投影点の 4 点で構成される四角形内を連続的に塗りつぶすことで、実際に走行できた領域をラベリングする。投影する車輪の位置がカメラの視野角外、もしくはある閾値以上の地点に到達した時点で投影を終了する。本研究では、使用するステレオカメラ (Stereolabs 製, ZED stereo camera) の最大深度より閾値を 20 m とした。

### 3.2 走行軌跡の投影精度評価実験

対象とする走行環境には、直線だけでなく、右左折が可能な場面が存在する。そのような場面では、複数の方向を走行推奨領域として広く予測できることが望ましい。そのため、使用する教師画像には、右左折のような旋回時の走行領域を投影した教師画像が必要であると考えられる。このことから、教師画像自動生成システムの要求機能を、直進時、および右左折時のような走行時の走行領域を精度よく投影できることと定める。その要求機能を検証するため、走行軌跡の形状に応じた投影精度を評価する実験について述べる。まず、図 3 に示す実験車両を用い、図 4 に示す入力画像のようにあらかじめ路面にマーキングしたコースを手動で走行し、時系列の画像と車両の速度や操舵角などのデータを取得する。ここで得られたデータを用いて自動でラベリングした領域と、路面のマーキングをもとに手動でラベリングした領域を比較することで走行軌跡の投影精度

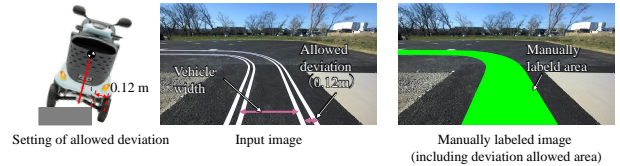


図 4 許容逸脱量も含めマーキングしたコースの手動ラベリング

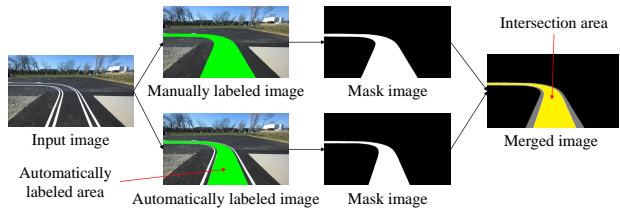


図 5 投影精度評価のためのデータ処理フロー

を評価する。このとき、比較対象となる手動でラベリングした領域は、投影精度の十分性の観点から設定したコース幅に、許容逸脱量を左右に加えた領域とした。許容逸脱量は、コース外が車輪で接地できない高さの路面と仮定した際に、横転しない限界逸脱量と定義し、図 4 の左図のように今回の車両を想定した際の横転しない限界逸脱量 0.12 m に設定した。投影精度評価では、図 5 に示すように手動でラベリングした領域と自動でラベリングした領域の共有領域 (Intersection area) の面積を求め、図 5 に示す Automatically labeled area に対する Intersection area の面積比を投影精度とした。

#### 3.2.1 実験条件

直進、旋回時の投影精度評価のため、曲率半径と速度を 3 条件ずつ設定し、表 1 に示す計 9 条件での実験を行った。曲率半径の最も小さい条件は、電動車いすでの走行時に壁に対して空けておきたい範囲 [9] をもとに、0.8 m と設定した。また、速度はセニアカーの仕様から最小速度と最大速度を設定した。各種演算はオフラインで実行するため、走行時は画像や速度等の車両情報を Robot Operating System[10] の bag ファイル形式で保存した。

#### 3.2.2 実験結果

表 1 にそれぞれの条件における投影精度を示す。なお、表内の括弧内の数値は映像の再生速度を 0.5 倍にした際の結果を示している。表 1 から、直線の条件ではすべて 100% を達しているが、速度、曲率が大きくなると映像を等倍速で処理した場合には 100% を達成できなくなる。これは速度と曲率の増加に伴いヨーレートが増加し、フレームの水平方向の移動が大きくなった結果、visual odometry で特徴点を追跡できなくなったことが原因であると考えられる。使用した計算環境では、

表 1 投影精度 [%] (( ) 内は再生速度を 0.5 倍にした際の精度)

		速度 [km/h]		
		2	4	6
曲率	$\infty$ (直線)	100	100	100
半径	1.6	100	99.3(100)	98.4(100)
[m]	0.8	99.9(100)	98.8(100)	98.3(100)

visual odometry の計算周期が約 10 Hz であるのに対し、映像のフレームレートが 15 fps であったため、使用されないフレームが存在していた。そこで、映像データの全フレームを扱えるよう、データを保存したファイルの再生速度を 0.5 倍にして計算を行ったところ、すべての条件において visual odometry が安定に動作し、100% を達成できることを確認した。

#### 4 セマンティック・セグメンテーションによる 走行推奨領域推定

本章では、自動で生成した教師画像を用いた Semantic Segmentation の出力結果の特徴を把握する。まず学習に使用したデータセット、ネットワークアーキテクチャについて述べ、その学習結果を分析する。

##### 4.1 データセット

教師画像の生成および走行推奨領域の出力結果の分析のために、図 6 に示すような柏市内の歩行環境を図 3 に示す実験車両により手動で走行し、時系列の画像と車両の速度、操舵角等のデータを取得した。ここで得られたデータを使用して、前章で述べた手法を用い、走行推奨領域とそれ以外にラベリングされた画像を生成した。表 2 にラベリングした画像の枚数を示す。表 2 に示すように、走行軌跡の投影終了時の横方向の移動量と走行環境に応じて分類した。表中の training データは学習に使用する画像を示し、test データは学習したモデルの評価に使用する画像を示し、それぞれ別の走行環境で収集したデータを使用して生成した画像である。training データは全ての画像に対して左右反転の処理を加え、2 倍の枚数に増やした。また、横方向の移動量とは、走行軌跡を投影し始めた時刻の車両の進行方向に対して垂直方向に移動した距離である。得られた走行データには直進する場面が多く、右左折やカーブなどでの旋回場面が少なかったため、全データを使用して学習すると直進場面に対して過学習になる懸念があった。そこで、学習の際に直進時の画像と旋回時の画像の枚数を合わせて使用するために横方向の移動量に着目して画像を分類した。なお、本論文では投影終了時の車両左右の横方向移動量が 6 m 未満であるかに応じて自動で振り分けた。また、走行環境ごとの推定結果の特性を把握するため、以下のような基準で手動で画像を分類した。

- 車道：軌跡すべてが自動車が進入可能な領域に存在。
- 歩道：軌跡すべてが自動車が進入不可能な領域に存在。
- 車道+歩道：軌跡の一部が車道と歩道に存在。
- 横断歩道：軌跡の一部が横断歩道に存在。

ここで、横断歩道は車道+歩道の一部であるが、車道上に存在する歩行環境であるため、別の取り扱いとした。

得られた教師画像のうち、本論文の学習に使用した training データ数と学習中の検証に使用した validation データ数を表 3 に示す。前述の通り、過学習を避けるため、横方向の移動量が 6 m 未満のデータ数と 6 m 以上のデータ数が均一かつ走行環境



図 6 走行環境のスナップショット

表 2 ラベリングした画像の枚数 (左右反転画像含む)

	横方向 移動量	車道	歩道	車道 +歩道	横断 歩道
training データ	6 m 未満	10704	4636	682	336
	6 m 以上	776	184	362	422
test データ	6 m 未満	7466	1150	604	150
	6 m 以上	306	72	298	184

表 3 学習に使用する教師画像の枚数

	横方向 移動量	車道	歩道	車道 +歩道	横断 歩道
training データ	6 m 未満	300	300	100	100
	6 m 以上	100	100	300	300
validation データ	6 m 未満	50	50	50	50
	6 m 以上	50	50	50	50

それぞれのデータ数が均一になるよう training データを 1600 枚抽出した。また、validation データとして test データからすべての条件のデータ数が均一になるよう 400 枚抽出した。

##### 4.2 ネットワークアーキテクチャ

CNN ベースの Semantic Segmentation のアーキテクチャは、多クラス分類における精度向上を目指し、多数提案されている [11]。しかし、今回は高精度を目指すことが目的ではなく、自動で生成した教師画像を用いた学習による出力の特徴を把握することが目的であるため、代表的なアーキテクチャとして先行研究 [4] と同様に SegNet を用いた。

##### 4.3 評価指標

本システムでは走行しうる領域を広く認識することを目的としているため、図 7 に示す Labeled area に対する走行推奨領域の未検出が少ないことが望ましい。そこで、未検出の少なさを評価可能な再現率 (Recall) を用いる。Recall は図 7 に示す Labeled area に対する Intersection area の面積比として求められる。また、自動でラベリングした 1 回の走行軌跡を示す領域より広い範囲が出力されることが望ましい。そこで、走行推奨領域として分類されている領域面積の大きさを示す評価指標として、図 7 に示す Labeled area に対する Prediction area の面積比 (以下、Prediction over Label (PoL) と呼ぶ) を用いる。

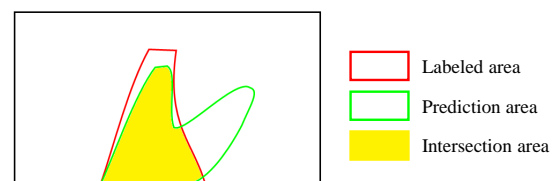


図 7 評価指標の計算に用いる領域

表4 PCの仕様

CPU	Intel Core i7-8750H (2.2GHz × 12)
RAM	32GB
GPU	NVIDIA Geforce RTX 2060 6GB

#### 4.4 学習条件

学習に使用したPCの仕様を表4に示す。画像サイズはメモリ制約から320×180、バッチサイズは4とした。SegNetのエンコーダを構成する特徴抽出器のみ、機械学習ライブラリのPyTorchで利用可能な学習済みモデルを使用した。最適化手法として確率的勾配降下法を用い、momentumを0.9、weight decayを0.0001と設定した。デコーダの初期学習率は0.01とし、エポックの繰り返し数に応じて学習率を減少させる手法を採用した。ただし、エンコーダの学習率はデコーダの1/10とし、損失関数は交差エントロピー誤差を用いた。

#### 4.5 走行推奨領域の推定結果

SegNetで学習を行い、走行環境と横方向移動量のそれぞれの条件について表2に示すtestデータで予測結果を出力した。表5, 6にRecallとPoLの平均値をそれぞれ示す。

まず、両評価指標について横断歩道に着目すると、Recallの大きさに対し、PoLが大きい傾向にある。図8の(a), (b)に示すように、横断歩道の手前で正解領域が示す方向以外の方向も走行推奨領域として出力していることが、予測領域の増加につながり、PoLが大きいと考えられる。他にも図8の(c)のように複数方向を走行推奨領域として予測している場面があり、trainingデータとして前方以外の方向への走行した際の画像が含まれることで、走行推奨領域として予測できる領域が増加すると考えられる。

また、表5の $ld \geq 6$ の部分に着目すると、歩道での精度が他の条件と比べて非常に小さいことがわかる。これは、図8の(d)より、正面が壁で1フレームでは進むべき方向が分かりにくい環境において、経路を予測することの困難さが要因だと考えられるが、教師画像の工夫次第で改善できると考えられる。

## 5 まとめ

本論文では、物理的に明確な走路境界が存在しない環境を含む歩行空間において、走行推奨領域に基づく自律走行を目指し、まずSemantic Segmentationのための教師画像を自動で生成する手法の開発とその機能検証を行った。visual odometryを活用した走行領域の自動ラベリング手法については、速度、曲率が大きい条件においても、十分な精度でラベリング可能であることを確認した。ただし、処理速度と映像の更新周期の関係によっては性能が低下することも確認されており、十分な性能を得るには処理速度を向上させるか映像の更新周期を遅くするなどの工夫が必要であることもわかった。

次に、自動で生成した教師画像を用いてSegNetで学習を行い、予測結果の特徴を分析した。その結果、複数の方向を走行した際の教師画像を学習に用いることで、より広い領域を走行推奨領域と予測できることが確認された。

今後適用すべき教師画像を検討し、経路計画まで含め、自律移動システムとして検証していくことを考えている。

謝辞：本研究は、戦略的イノベーション創造プログラム(SIP)第2期/フィジカル空間デジタルデータ処理基盤「移動空間デジタルデータのエッジ処理とクラウド連携による安心・安全・安価な複数台自動走行パーソナルモビリティの社会実装」の一環として行ったものであり、関係各位に感謝します。

表5 Recall [%] ( $ld$ : 横方向移動量)

$ld$	車道	歩道	車道+歩道	横断歩道
$ld < 6$	89.8	87.2	83.5	70.9
$ld \geq 6$	75.8	48.5	80.9	82.1

表6 PoL [%] ( $ld$ : 横方向移動量)

$ld$	車道	歩道	車道+歩道	横断歩道
$ld < 6$	98.9	100.7	102.5	101.0
$ld \geq 6$	90.4	84.8	96.9	105.0

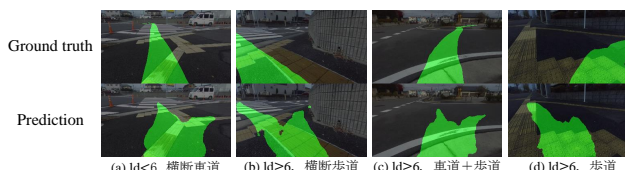


図8 Semantic Segmentationの出力結果

## 参考文献

- [1] 内閣府地方創生推進事務局, “近未来技術事例集,” 近未来技術等社会実装事業事例集 (平成31年3月), p. 15, 2019.
- [2] M. Hirai, T. Tomizawa, S. Muramatsu, M. Sato, S. Kudoh, and T. Suehiro, “Development of an intelligent mobility scooter,” in 2012 IEEE International Conference on Mechatronics and Automation, pp. 46–52, 2012.
- [3] 江口純司, 尾崎功一, “DGPSを用いて設定したウェイポイント経路追従による実環境自律移動法の開発,” 日本ロボット学会誌, vol. 30, no. 3, pp. 314–323, 2012.
- [4] 財前通平, 松實良祐, 林隆三, 小竹元基, “セマンティック・セグメンテーションによるシニアカーの走行可能領域検出手法の開発,” 交通・物流部門大会講演論文集, vol. 2017.26, p. 3007, 2017.
- [5] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 12, pp. 2481–2495, 2017.
- [6] D. Barnes, W. Maddern, and I. Posner, “Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 203–210, 2017.
- [7] J. Mayr, C. Unger, and F. Tombari, “Self-Supervised Learning of the Drivable Area for Autonomous Vehicles,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 362–369, 2018.
- [8] R. Mur-Artal and J. Tardos, “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras,” IEEE Trans. Robot., vol. PP, Oct. 2016.
- [9] Y. Morales, N. Kallakuri, K. Shinozawa, T. Miyashita, and N. Hagita, “Human-comfortable navigation for an autonomous robotic wheelchair,” in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2737–2743, 2013.
- [10] M. Quigley et al., “ROS: an open-source Robot Operating System,” in ICRA workshop on open source software, vol. 3, 2009.
- [11] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image Segmentation Using Deep Learning: A Survey,” arXiv:2001.05566v4 [cs.CV], 2020.