道路俯瞰画像を用いた車エージェントの運転行動学習

米元聡[1]　國武佑哉[1]

概要：車エージェントの運転行動の学習手法について述べる．本研究では，車エージェントは道路俯瞰画像をもとに運転行動を決定する．運転行動を制御するための車モデルとしてキネマティックモデルを用い，ステアリング操作を離散的化して表現する．運転行動の学習には，深層強化学習の 1 つである DQN 法を用い，走行シミュレータで生成される道路俯瞰画像を知覚に用いる．走行シミュレーションにより，最高速度や，ステアリング角操作量などの車モデルのパラメータの学習への影響や，変動を加えた道路コースの学習への影響などについて検証を行った．

キーワード：自動運転シミュレーション，深層強化学習，道路俯瞰画像

A Road Overhead Image based Deep Reinforcement Learning Method for Self-Driving Car Agents

SATOSHI YONEMOTO[†1]　YUYA KUNITAKE[†1]

Abstract: This paper presents a road overhead image based deep reinforcement learning method for self-driving car agents. Kinematic vehicle model is used as the car model to control driving actions, and the steering operations are represented by discretizing the action spaces. Deep reinforcement learning is realized by DQN(Deep Q-Network) method which is able to learn from the road overhead images generated by our virtual driving simulator. We further demonstrate the learning models that are acquired by changing model parameters such as maximum velocity and steering angle, or acquired under different road surface conditions.

Keywords: Self-Driving Simulator, Deep Reinforcement Learning and Road Overhead Image

1. はじめに

現在，自動車メーカーにより自動運転技術の本格導入が進められている．実際の走行データを用いたシミュレータで性能を検証し，公道での走行実験を行う段階にある．実用化が進む一方，人工知能技術をベースにした自動運転スキル獲得の手法の開発が依然として活発に行われている．

特に，車視点の道路画像を入力とした深層強化学習による自動運転技術に注目が集まっている．「End-to-End Learning」という，自動で特徴抽出までを行うアプローチが提案されており，自動運転の場合は，道路画像を入力，車の行動を出力とする形のニューラルネットワーク学習に相当する．自動運転の場合は，道路や交通参加者，他車などを抽出した結果，すなわち抽象度を少し上げた道路情報のマップに変換した上で学習させる方が有効である．実際に自動車メーカーの自動運転技術においても，ダイナミックマップ技術，占有グリッドマップなどといった抽象化した道路画像を用いる手法が広く用いられている[1][2][3].

自動運転に関連する研究には，Waymo 社の模倣学習（Google 社の Self-Driving Car Project）[4]，TORCS シミュレータを用いた自動運転学習[5]などがある．より先進的な自動運転技術としては，以下のような手法が開発されている．障害物，他車の速度などの交差点での不確実性を組み込んだモデルを用いて，これらの不確実性を考慮しつつ潜在的なリスクの推定値を割り出し経路算出に反映する技術[6]や，CVaR を利用した低リスク運転行動方策の学習[7]などである．

本研究の最終的な目標は，実際の道路での走行を想定した，すなわち交差点や他車との関係を考慮した安全運転行動を獲得することである．そのためには，手法に交通ルールや目的地までの経路情報を組み入れる必要が生じることになる．

安全運転行動獲得のための第一ステップは，走行車線内を安定に走行すること，交通事故の潜在リスクを減らすために走行車線内の障害物を回避することである．本報告では，走行シミュレーション環境において，交通ルールの埋め込みや経路計画を伴わないレベルで，車エージェントが与えられた走行車線内を運転するための行動，および道路

1 九州産業大学
Kyusyu Sangyo Univ.

上のリスクである障害物を安全に回避する行動を獲得する手法について述べる．このような手法が必要となるロボットカーの制御の問題では，如何に早く走行できるかも重要となるが，本研究では安定な走行の実現に重点を置いている．

本研究は，ステアリング，加減速の制御のみを扱う簡易な車モデルの導入，カーブなどを変更可能な道路コースのランダム生成，車上方からの道路俯瞰画像の生成，DQN 法を用いた道路俯瞰画像からの運転行動学習(行動の離散化，報酬の設定)，という 4 つの要素技術から構成される．

2. 提案手法

本研究で提案する道路俯瞰画像を用いた運転行動の学習手法についての概要を述べる．

2.1 走行環境

図 1 に本研究で扱う自動運転車エージェントの走行環境（走行シミュレータ）で利用する道路コースの生成例を示す．走行シミュレータは，道路コースのランダム生成，車上方からの道路俯瞰画像の生成，車エージェントの操作・描画，DQN 法による運転行動の学習機能から構成される．

2.2 道路コースのランダム生成

本研究では道路を，ベジエ曲線セグメントの連結により構成する．ループ状の道路コースの生成も可能である．また，道路幅を変更可能である．道路はすべて同じ高さにあるものとする．連結する曲線セグメントの端点を好きな位置に設定することにより基準となる道路コースを構成しておく．車エージェントの学習時，道路コースをランダム生成する場合は，曲線セグメントの端点に位置の変動を加える（図 1 左）．その変動の加え方により，道路の難易度を変えることができる．なお，道路コースのランダム生成による走行環境の変更は，学習のエピソード終了時に 10%の確率で起こるように設定する．学習したモデルを用いて試行する場合は，毎回変更される．

2.3 道路俯瞰画像の生成

走行環境における車エージェントの位置・方向より，道路俯瞰画像を生成する.走行できる領域を黒ピクセルとし，走行車線の境界を示すエッジ（白線に相当）および障害物を黒以外のピクセルとして生成する．また，車載カメラ画像のような車視点から道路をとらえた画像を学習には利用せず，本研究では，車上方から俯瞰した車両前方画像とする．詳細は 3.3 節で説明する．

2.4 DQN 法による運転行動の学習

本研究では，離散的な行動を扱える DQN(Deep Q-Network)法を運転行動の学習に利用する．車を制御するためのモデルとしては，主に車のゲーム開発に利用されるキネマティックモデルを採用し，ステアリング制御，アクセル／ブレーキの制御のみで行動できるものとした．詳細は 3.2 節で説明する．DQN 法では，状態観測（知覚）に画像を利用する．その画像として，2.3 節で述べた道路俯瞰画像を用いる．報酬は，走行したステップ数に比例させ，走行車線を外れるか，一定時間走行した場合にエピソード終了とする．

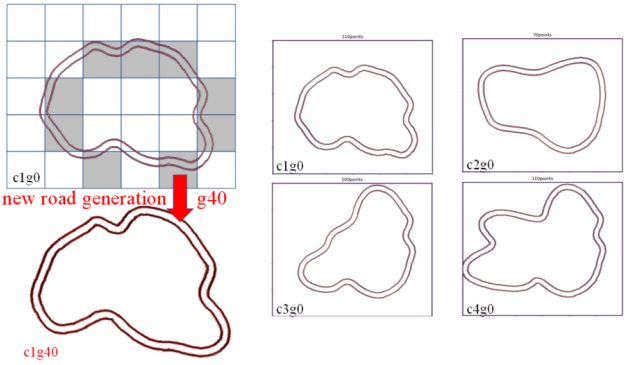

図 1　自動運転車エージェント用の道路コース生成
Figure 1　Driving Course Generation for Self-driving Cars.

3. DQN 法による運転行動の学習

3.1 DQN(Deep Q-Network)法

深層強化学習は，強化学習に深層学習を組み合わせた手法であり，DQN 法はその手法の 1 つである．DQN 法は，ゲーム画面をもとに AI コントローラがよりスコアを稼ぐ行動を学習していくというゲーム AI の問題のために提案された方法[8]で，改良手法が，2 人ゲーム(AlphaGo[9])，ロボットの制御，自動車の自動運転[10]などに応用されている．

行動の連続値の学習を行える DDPG 法[11]などの拡張手法もあるが，本研究では離散的なステアリング角操作量を行動により制御できればよいことから DQN 法を用いる．また本研究では，標準的な DQN 法を用いており，特に学習アルゴリズム上の改良は行っていない.そのため DQN 法の詳細については省略する．

3.2 車のモデル

本研究では，車の制御を行うためのモデルとして，車の動きをおよそ再現でき，実装が容易なキネマティックモデルを採用する．キネマティックモデル(kinematic vehicle model)は，質量や力を計算上考慮せず，速度，加速度，空間内の位置のみを考慮するモデルであり，主に車のゲーム開発に用いられている簡易なモデルである．図 2 にキネマティックモデルの概要を示す．

以下，キネマティックモデルにおける車の位置，車体角度の計算方法について述べる．

加速度の大きさを a $[m/s^2]$, ステアリング角を s $[rad]$ とする．車の全長を$L[m]$ とする． sはエージェントの行動として決定する．加速は$a \leftarrow a + dt$ で実現し，加速度の大きさaにマイナス値を指定することで減速できる．

計算手順は以下の通りである．

(a) 速度ベクトル$v[m/s]$を車の進行方向の成分のみ更新する（$v \leftarrow v + (adt, 0)$）．ただし，車の進行方向$x$の速度成分$vx$は予め定めた最大速度$vmax$で制限する．
(b) 旋回半径(turning radius) $t_r[m]$を計算する（$t_r = L/\tan(s)\ [m]$, sは微小な角度とする）．
(c) 角速度(angular velocity) $a_v[rad/s]$を計算する（$a_v = vx/t_r[rad/s]$）．

(a)～(c)より，車の位置$\boldsymbol{p}$，車体角度θを$\boldsymbol{p} \leftarrow \boldsymbol{p} + v_r{}^{-\theta}dt$，$\theta \leftarrow \theta + a_v dt$と更新する．ここで，$v_r{}^{-\theta}$は，速度$v$を$-\theta$だけ回転したベクトルを表す．

以上まとめると，車エージェントが決定する行動（ステアリング角操作量s）と，現在維持している加速度aをもとに車エージェントの動きが決定される．ただし，最高速度を制限しているため，最高速度に達した後はその速度での運転となる．通常，減速の判断は走行車線内に障害物が存在するか，停止線，前方の他車の存在に応じて決まるものであるため，本報告の段階での行動学習においては加速度の加減は学習の対象とせず，一定の加速度まで増加させるルールとしている．

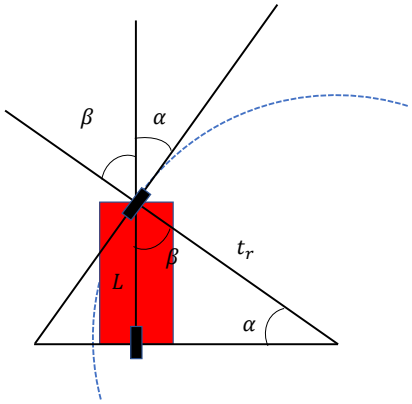

図 2　キネマティックモデル
Figure 2　Kinematic Vehicle Model.

3.3 道路俯瞰画像

車エージェントの状態観測に用いる道路画像としては，車載カメラ画像（車の運転席から見た道路の透視投影像）や距離画像，車上方から俯瞰した車両前方画像，交通参加者，信号などを細かく記述したセマンティックセグメンテーション画像など様々なものが用いられる．例えば，Waymo[4]では交差点全体をとらえる大きさの俯瞰画像をエキスパート運転の模倣学習に用いている．本研究も道路俯瞰画像を採用するが，本報告の段階では他車の動きや交差点については考慮しないため，車両前方の狭い範囲に限定して用いる．図 3 に道路俯瞰画像および対応する車視点画像の例を示す．安定走行と障害物回避のみ扱うことを想定しているため，走行車線の境界，障害物を foreground ピクセルとした 64×64 サイズの道路画像を採用する．

本研究において，車上方から俯瞰した道路画像を採用したのは，車載カメラ画像からの白線抽出結果を俯瞰画像として容易に変換できること，また，Waymo[4]のように，その他のセンサ情報や道路情報の統合によって上空から俯瞰した交差点を含む道路画像を入力に利用するアプローチが今後主流となると予想されることが理由である．

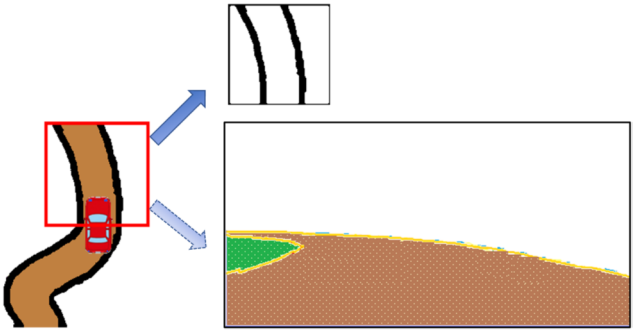

図 3　道路俯瞰画像と対応する車視点画像
Figure 3　Road Overhead Image and the Road Scene Image.

3.4 報酬と終了条件

DQN 法の報酬として，1 ステップごとに正常な走行が可能であれば+1 を与える．車中心が走行車線からはみ出す，障害物に衝突する，既定の走行時間（最大ステップ数）に達する場合はエピソード終了とする．

4. 走行シミュレーション

4.1 OpenAI Gym による車エージェントの実装

本研究では，OpenAI Gym という強化学習向けのエージェント開発のフレームワーク[12]を利用して車エージェントの実装を行った．この車エージェントを含む環境に DQN 法を組み込む．DQN 法の実装には Keras[13]を用いた．DQN 法では，離散的な行動を対象とするため制御の問題を扱う場合には工夫が必要である．以下，DQN 法を用いた OpenAI Gym の実装方法について述べる．

エージェントの設定は以下の(1)～(3)の通りとする．

(1) 行動

ステアリング操作として，左回転・直進・右回転のいずれかの行動をとる（したがって行動数は 3）．エピソード開始時よりアクセルを踏み続け，その後は（制限された）最高速度で走行を続けることになる．このため加減速の行動については行動学習の対象としていない．

(2) 知覚

64×64 ピクセルの道路俯瞰画像を知覚とする．道路俯瞰画像は車の位置，向きに基づき生成する．DQN 法における window length は 4 とした．

(3) 報酬・終了条件

1 ステップ走行するたびに走行可能と判断できれば報酬を+1 とする．車中心が走行車線内をはみ出した場合，障害物に重なった場合，一定の走行時間（最大ステップ数）に達

した場合に終了とする．

以上のエージェント設定を，OpenAI Gym 用のエージェントの操作である step()，reset()，render()に実装する．計算機としては以下の構成の PC を用いた．

CPU: Core i7-8700K, メモリ: 32GB, GPU: GeForce RTX 2070, OS：Windows 10，開発環境：Anaconda (Python 3.7.3)

Python のライブラリには Tensorflow, Keras, OpenAI Gym を用いた．

4.2 車モデルのパラメータの影響

車エージェントの寸法は全長 4m，全幅 2m とした．走行シミュレータ環境で車エージェントを 40 ピクセルで描画するため車モデルの計算においては全長 $L = 40$とスケールの調整を行っている．車モデルには，最高速度，行動 1 回あたりのステアリング角操作量，微小時間などのいくつかのパラメータ設定が必要である．走行シミュレータで車エージェントを手動で動かす際に設定するパラメータを標準のパラメータとした．具体的には，最高速度$v = 10$，行動 1 回あたりのステアリング角操作量$d = 10[deg]$, 微小時間$dt = 0.3$とした．

道路コースの道路幅を 2m の等間隔とした．まずこの標準のパラメータの設定および変動なしの道路コース(c1g0)を用いて行動学習を行った結果，5 万回程度で安定して運転行動を獲得することができた．

道路コースは，学習時にランダム生成され試行のたびに変化する．このため，学習時の報酬の推移にばらつきが出る．そこで車モデルパラメータおよび道路コースを変化させた場合の行動学習への影響を調査した．以降，道路コースのランダム生成時に与える変動の大きさを g0（変動なし）, g20（やや変動あり）, g40（変動大）と表記する．

(1) **最高速度による影響**

車モデルのパラメータの 1 つである最高速度vの設定による違いを調査した．最高速度vを低速から高速に 5,10,20,30 と変えて行動学習にどのような影響があるかを調べた．変動大でランダム生成した道路コース(c1g40)において 10 万ステップ学習を行い，学習時と同じ条件下で 100 エピソード試行した．図 4 にその結果を示す．例えば，学習したモデル V10D15 model は，最高速度$v = 10$，ステアリング角操作量$d = 15$のパラメータ設定で 10 万ステップ学習したモデルであることを表す．そのモデルを用いて，試行した際の成功率が 48%であることを示している．ここで成功率とは，100 エピソードのうち成功したエピソードの割合のことである．成功の判断基準は，指定の走行距離以上とした. 従ってこの計測では, 基準を速度の小さい順に 800, 400, 200, 133 ステップ以上とそれぞれ調整した．スコアは，成功した場合の走行距離の平均（報酬にもとづく平均）を表している．変動大でランダム生成された道路コースの難易度が高いためか，成功の基準を満たす割合が低くなっている．しかしながら，最高速度が 5～20 の場合の走行は比較的安定しており，この範囲の速度であれば適切であると判断した．高速の場合は小回りが利くようにステアリング角操作量をやや大きく設定した．最高速度vが高速の 30 の場合は，曲率の大きなカーブを曲がれずオーバーランすることが多かったため，成功率が低下している．

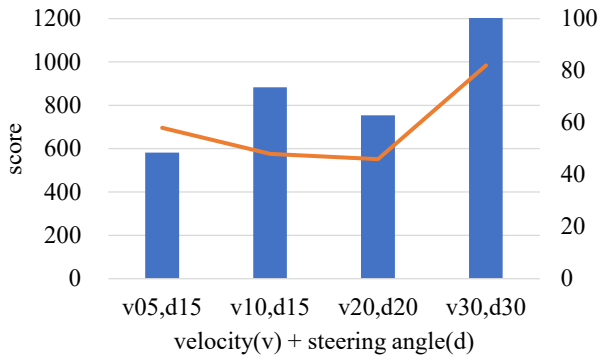

図 4　最高速度がスコア及び成功率へ与える影響

Figure 4　Influence of Maximum Velocity Parameter(v) on Score and Success Rate.

(2) **ステアリング角操作量による影響**

次に，左右にハンドルを切るためのステアリング角操作量dの違いについて調査した．操作量dを 5～30 と変えて運転行動学習にどのような影響があるかを調査した．変動大でランダム生成した道路コース(c1g40)において最高速度$v = 10$の設定で 10 万ステップ学習を行い，学習時と同じ条件下で 100 エピソード試行した．図 5 にその結果を示す．$d = 5$の場合は，車の向きをうまく変えることができず成功率 0%となった．最も操作量を大きくした$d = 30$で成功率が最大となった．しかし，$d = 30$の場合は，走行車線からはみ出しそうになると大きくハンドルを切るためふらついた走行となり，安定な走行とは言い難い．以上より，15～20 の範囲の操作量が適切であると判断した．

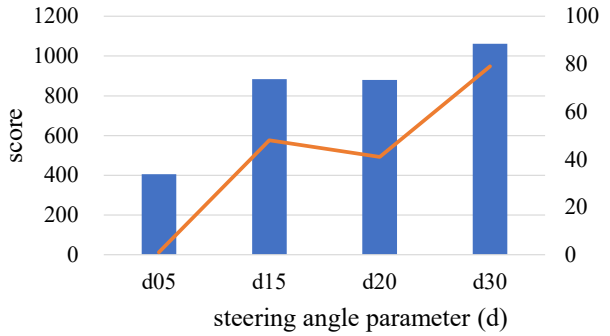

図 5　ステアリング角操作量による影響

Figure 5　Influence of Steering Angle Parameter(d) on Score and Success Rate .

4.3 道路コースの影響

次に道路コースの影響について調査した．具体的には，道路コースの難易度，道路幅の影響を調べた．

(1) 道路コースの難易度の影響

難易度の異なる道路コースをいくつか用意し学習を行った．道路コースの難易度は，基準とする道路コースの構造で決まる．また，曲線セグメントの連結時に与えるランダムな変動の大きさによっても道路コースの難易度は変化する．図 6 に各エピソード終了時に得られた報酬の推移を 3 つの学習モデルについて重ねたグラフを示す．これは同一のパラメータ設定(v10, d15)のもとで変動大でランダム生成した 3 つの道路コース(c1g40, c2g40, c3g40)についてそれぞれ 10 万ステップ学習した結果である．エピソードの終了条件である最大ステップ数を 1200 と設定した(赤色のライン)．これは道路コースの 3,4 周に相当する走行距離である．道路コース c1g40 を基準とすると，道路コース c4g40 はカーブが多め，道路コース c2g40 は緩やかなカーブの設定である．道路コース c4g40 では，報酬が 200 手前で伸び悩んでいることからここに難所があることがわかる．実際に，コースの難易度が高いほど報酬が少なくなっていることも確認できる．

なお，変動なしで学習したモデル C1G0 model, C2G0 model, C3G0 model の成功率は 100 エピソードの再現試行においてすべて 100%，C4G0 model は 0%であった．

次に変動の大きさによるコースの難易度の影響を調査した．変動なし(g0)，やや変動あり(g20)，変動大(g40)の 3 つの場合について，変動の大きさに比例して，スコアが伸び悩むことを確認した．

変動を加えるのはいわゆる「データの水増し」を狙った工夫と言える．そこで未知の道路に対して安定した運転行動が獲得できるか検証した．ランダムな変動を加えて学習したモデル(C1G40 model)，変動なしで学習したモデル(C1G0 model)の 2 つを用いてそれぞれ，変動大の未知の道路コースに対して 100 エピソード試行した．図 7 に 4 種類の未知の道路コース(c1g40, c2g40, c3g40, c4g40)に対する試行結果を示している（C1G40 model の場合，道路コース c1g40 は未知ではないが合わせて示している）．スコアは，成功した走行距離の平均を表している．予想に反し，変動なしで学習したモデルの方が，すべての未知の道路において優れた走行結果となった．考えられる要因は，変動ありの場合，多様な道路に対する運転行動を学習するには学習回数がより必要であること（実際，50 万回ほど学習させると，約 2 倍の成功率となることを確認した），設定した未知の道路に適応するための運転スキルは，道路コース(c1g0)内のカーブの攻略のみで十分である可能性などが考えられる．ある程度，学習が進んでから未知の道路コースに適応させるような工夫が必要と思われる．

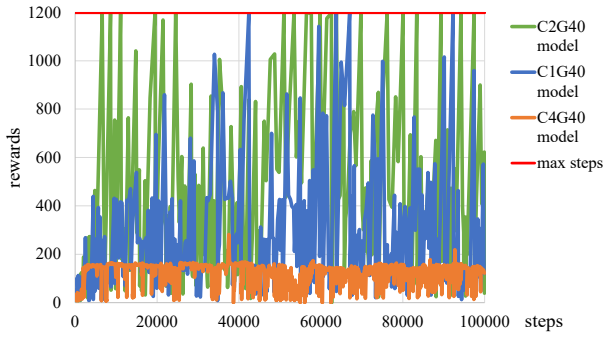

図 6　異なる道路コースに対する報酬の推移

Figure 6　Changes in Rewards on Different Roads.

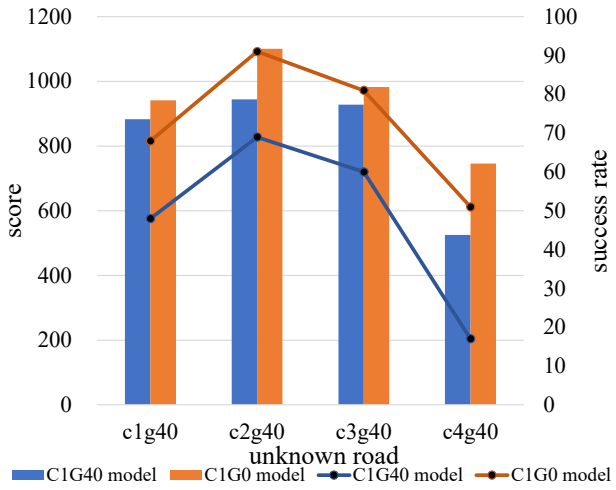

図 7　未知の道路コースに対するスコアと成功率

Figure 7　Score and Success Rates under Unknown Roads.

(2) 道路幅の影響

道路幅の違いが運転行動学習に影響するのか調査した．3m, 4m のやや広めの道路幅の場合と通常設定の 2m の道路幅の場合とを比較した．図 8 に異なる道路幅，変動大でランダム生成した道路コース（2 種類）に対してそれぞれ 10 万ステップ学習したモデルを用いて再現試行した結果を示している．道路コース c1g40 では，2m で学習したモデル(C1G40R2 model)は成功率が 48%であったが，3m で学習したモデル(C1G40R3 model)は 75%に向上した．これは道路幅が適度に広くなることにより，カーブにおいて運転を立て直す余裕があったためと思われる．4m で学習したモデルは，ほぼ変化なしであった．しかし，4m の場合はふらついた走行となった．これは，広い道路では手がかりとなる白線が道路俯瞰画像内に存在しなくなることを避け，より手がかりのある道路の端の方へと運転行動を起こすようになるからである．

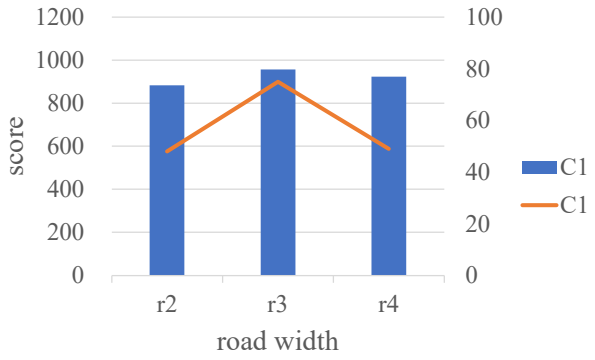

図 8　異なる道路幅に対するスコアと成功率
Figure 8　Score and Success Rates under Different Road Width.

4.4 障害物の影響

道路コース内に障害物がある場合の影響を調査した．まず，変動なしの道路コース(c1g0)を用いて 10 万ステップ学習した C1G0 model を用意する．このモデルの成功率は 100%である．次に，障害物がランダムな位置に一定区間出現する道路を生成し，このモデルを使ってさらに 10 万ステップ学習させ,障害物を避けるようになるか検証した(新たなモデルを C1G0B model とする)．図 7 に道路コースおよび道路俯瞰画像を示す．中央の画像が，車エージェントの知覚する道路俯瞰画像である．C1G0 model をそのまま適用すると，道路コースに障害物があるため成功率が 4%に下がるが，改善を試みた C1G0B model では 64%となり障害物にある程度対処できていることがわかる．しかし，他車，交通参加者などの様々な障害物に対応させるには，避けるだけではなくブレーキを用いて停止する機能が必須であるため，対策は今後の課題とする．

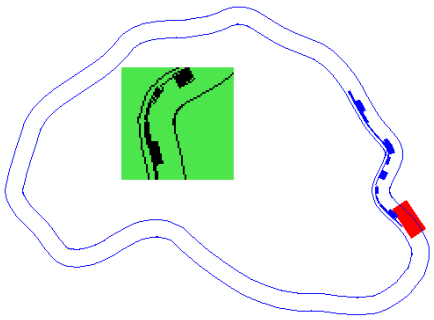

図 7　障害物のある道路コース (c1g0b) の例
Figure 7　An Example of Driving Course Having Obstacles.

5. おわりに

本報告では，道路俯瞰画像を用いた車エージェントの運転行動学習の方法について述べた．具体的には，深層強化学習の 1 つである DQN 法を用いて道路画像から運転行動を学習する手法について提案した．車モデルとしてキネマティックモデルを採用し，ステアリング操作を 3 つの離散的な行動で表現した．また，車エージェントの知覚として道路俯瞰画像を取り入れ，獲得報酬はステップ数に比例するものとして定義した．

基本となる道路コースについての学習性能を確認した後，最高速度や，ステアリング角操作量などの車モデルのパラメータの学習への影響や，変動を加えた道路コースの影響，道路幅，障害物の影響について検証を行った．

検証の結果，車モデルのパラメータを適切な範囲に設定すれば学習が安定することは確認できたが，ランダム変動を用いた道路画像の水増しによる学習性能の向上を確認することはできなかった．学習したモデルは，学習時とは異なる未知の道路コースにも適用できることが確認できた．しかし，変動大の道路に対する成功率がコースによっては半分以下と低いため，手法に改善の余地があるといえる．学習回数を増加させることである程度改善するが，その後伸び悩む傾向があることも確認した．あらゆる道路コースに対応できる学習モデルを獲得することを目指すが，考えられる道路コースを学習データとしてランダムに与えるだけでは学習が進まない可能性が高い．学習時に与える道路コースの難易度を学習レベルに応じて制御し，段階的に学習できる仕組みが必要である．

今後の課題としては，交差点での右左折や，他車を考慮した車線変更に対処できるよう経路計画を導入すること，安全運転行動の獲得手法へと拡張することなどが挙げられる．

参考文献

[1] 須田，青木，“自動運転技術の開発動向と技術課題”，情報管理, vol.57, no.11, 2015, p.809-817.
[2]青木，“自動運転車の開発動向と技術課題: 2020 年の自動化実現を目指して”，情報管理, vol.60, no.4, 2017, p.229-239.
[3] J. Chen, et al.. Model-free deep reinforcement learning for urban autonomous driving, In Proc. of IEEE Intelligent Transportation Systems Conference (ITSC), 2019, p.2765-2771.
[4] M. Bansal, et al.. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst, 2018, arXiv:1812.03079.
[5] C. Chen, et al.. DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving, Proc. of ICCV. 2015, p.2722-2730.
[6] S.G. McGill, et al.. Probabilistic Risk Metrics for Navigating Occluded Intersections, IEEE Robotics and Automation Letters, 2019, vol.4, no.4, p.4322-4329.
[7] Y.C. Tang, et al.. Worst Cases Policy Gradients, 2019, arXiv:1911.03618.
[8] V. Mnih, et al.. Playing atari with deep reinforcement learning, 2013, arXiv:1312.5602.
[9] D. Silver, et al.. Mastering the game of Go with deep neural networks and tree search, 2016, nature, 529(7587), 484.
[10] 勞世竑，陳謙，“自動運転システムにおける AI 技術”，計測と制御, 2018, vol.57, no.7, p.493-496.
[11] T.P. Lillicrap et al.. Continuous control with deep reinforcement learning, Proc. of ICLR2016, 2016.
[12] “OpenAI Gym”, https://github.com/openai/gym/
[13] “Keras: The Python Deep Learning library”, https://keras.io/