

自律駐車のためのニューラルモーションプランニング

Dongchan Kim and Kunsoo Huh

概要- 本論文では、深層生成ネットワークと従来のモーションプランニング手法を組み合わせたハイブリッドモーションプランニング戦略を紹介する。A*やハイブリッドA*のような既存の計画手法は、複雑な環境下でも実現可能な経路を決定する能力があるため、経路計画タスクで広く使用されているが、効率性の点で限界がある。これらの制限を克服するために、ニューラルネットワークに基づく経路計画アルゴリズム、すなわちニューラル・ハイブリッドA*を導入する。本論文では、条件付き変分オートエンコーダ(CVAE)を用いて、駐車環境の情報が与えられた場合の計画空間に関する情報を学習するCVAEの能力を利用し、探索アルゴリズムを導くことを提案する。デモで学習された実現可能な軌道の分布に基づいて、効率的な拡張戦略が利用される。提案手法は、与えられた状態の表現を効果的に学習し、アルゴリズムの性能の面で改善を示す。

I. INTRODUCTION

近年、自律走行技術が普及している。駐車は自律走行に不可欠なタスクである。自律走行車は、障害物や空きスペースを含む周辺環境を認識し、衝突を避けながら所望のゴール状態に到達する必要がある。現在、自律型駐車システムが市販されている。さらに、自律走行車が車を駐車することを可能にする自律走行車駐車場も利用可能である[1]。

現在までに、人工ポテンシャル場(APF)[2]、高速探索ランダムツリー(RRT)アルゴリズム[3]、RRT*アルゴリズム[4]、部分運動計画(PMP)アルゴリズム[5]、A*アルゴリズム[6]、ハイブリッドA*アルゴリズム[7]など、多くの経路計画アルゴリズムが開発されている。ポテンシャル場法では、まずAPFベースの無衝突ホロノミック経路を使用し、次に生成された経路を使用してさらに最適化を行う。

そして、残りのアルゴリズムは、サンプリングに基づく方法と探索に基づく方法の2つのアプローチに分けられる[8]。サンプリングに基づく手法には、ランダムサンプリングに基づく手法と決定論的サンプリングに基づく手法の2つのアプローチがある。まず、ランダムサンプリングに基づく手法では、RRTとRRT*アルゴリズムが存在する。RRTに基づく手法は、与えられたサンプリング分布から得られたノードを接続するツリーを構築する。ノードを通り抜けることで、実現可能な経路が得られる。RRT*アルゴリズムは、再配線手順を利用することで計算時間を短縮するために使用される。そして、PMP法は決定論的サンプリングに基づく方法[5]を表す。この方法は、計算負荷を軽減するために、空間全体を検索するのではなく、局所的な検索を行う。

Dongchan KimとKunsoo Huhは、韓国ソウル市漢陽大学自動車工学科に所属している jookker, khuh2@hanyang.ac.kr

探索に基づく方法には、発見的計画法と状態格子に基づく計画法がある。まず、A*とハイブリッドA*アルゴリズムがヒューリスティックに基づく手法に含まれる。A*ベースの手法は、A*アルゴリズムを用いて駐車場を効率的に探索し、大規模な駐車場誘導システムとして機能する。最適経路を求めるためにヒューリスティック関数を採用する。ハイブリッドA*ベースの方法は、連続的な状態を探索するために使用され、それによって、グリッドの中心のみを訪問することができる古典的なA*の制限を緩和する。格子ベースの計画法は、構成空間の状態集合への離散化を利用する[9], [10]。格子は運動計画問題におけるグラフ探索としての解を提供する。

前述のアルゴリズムの中で、自律走行タスクに広く使われているA*アルゴリズムとハイブリッドA*アルゴリズムの応用に焦点を当てる。特に、ハイブリッドA*は、簡略化された車両モデル[7]を用いて、クラスタ化された環境において運動力学的な経路を生成することができ、設計された経路は人間の運転スタイルに近い。したがって、本研究ではハイブリッドA*アルゴリズムを適用する。しかし、ハイブリッドA*アルゴリズムは、構成空間のサイズが大きくなるにつれて、非常に時間とメモリを要する傾向がある。

ハイブリッドA*アルゴリズムの限界を克服するためには、状態拡張手順である探索の戦略を改善する必要がある。計画問題におけるディープニューラルネットワークの適用に関する既存の研究[11], [12]に触発され、本研究では、実現可能な軌道の学習分布であるガイダンスマップの助けを借りた経路計画アルゴリズムを利用する。

近年、深層生成ネットワークが盛んに研究されている。例えば、データセットの生成モデルを学習するための一般的な手法である変分オートエンコーダ(VAE)が利用された[13]。VAEは、低次元の潜在空間における高次元の動きの表現を可能にする。また、動きを再現するための最適動きも利用可能である。条件付き変分オートエンコーダ(CVAE)は、VAE法[14]に条件入力を追加で組み込む。CVAEは、ガウス潜在変数を用いた構造化出力予測のための深い条件付き生成モデルである。入力オブザベーションは、出力を生成するガウス潜在変数の事前分布を変調する。CVAEネットワークを用いた軌跡予測やマルチモーダル予測に関する研究はいくつかあり、追加条件により性能が向上する可能性がある[15], [16]。

本研究では、CVAEネットワークベースのハイブリッドA*アルゴリズムを提案する。CVAEモデルは、

駐車環境と初期状態およびゴール状態が与えられたときに、実現可能な軌道の予測分布を提供するように学習される。本研究の貢献は以下のようにまとめられる：

- ・ 深層生成ネットワークと従来の計画手法を組み合わせた自律駐車のためのニューラルハイブリッドA*アルゴリズムを提案する。
- ・ CVAEアーキテクチャは、初期状態、ゴール状態、障害物を含むマップ情報が与えられたときに、実現可能な軌跡分布を学習するために利用される。
- ・ 提案手法は、テストシナリオの計算時間とノード数を大幅に削減する。
- ・ 提案手法の実現可能性は、シミュレーションにおける様々な自律駐車シナリオで実証された。

本稿の残りの部分は以下のように進められる。セクション2では、従来のハイブリッドA*アルゴリズムについて詳しく説明する。セクション3では、ニューラル・ハイブリッドA*アルゴリズムについて説明する。CVAEアーキテクチャを紹介し、自律駐車のための提案ハイブリッドアルゴリズムを説明する。セクション4では、シミュレーションによる提案アルゴリズムの検証について詳述する。最後に、セクション5で研究を要約し、導き出された結論を示す。

II. PRELIMINARIES

A. ハイブリッドA*アルゴリズム

1) 遷移モデル：状態 $x_k = (x_k, y_k, \theta_k)^T$ は計画ステップ k の状態を表し、 x_k, y_k, θ_k はそれぞれ x 軸位置、 y 軸位置、方位角を表す。離散化された形の状態遷移モデルは以下のように表される：

$$\begin{aligned} x_{k+1} &= x_k + d \cos(\theta_k) \text{dir} \\ y_{k+1} &= y_k + d \sin(\theta_k) \text{dir} \\ \theta_{k+1} &= \theta_k + \frac{d}{L} \tan(\delta_k) \text{dir} \end{aligned} \quad (1)$$

ここで、 δ_k は離散化されたステアリング角集合Dに属するステアリング角候補であり、dirは車両の運動方向を表し、dは1回の探索ステップにおける膨張量である。2つの制御動作には、ステアリングの角度と方向が含まれる。Dは -40° から 40° の間で設定されたステアリング角を 10° の間隔で含み、dirは後方移動と前方移動にそれぞれ-1か1の値を持つ。

2) ハイブリッドA*：ハイブリッドA*アルゴリズムは、(1)で説明した遷移モデルを用いて運動力学的経路を生成する。ハイブリッドA*の実装では、X次元とY次元で2m、ヘディング角で 15° の解像度を使用する。この情報は、候補ノードがあるグリッドセルにあるかどうかを決定するために使用され、最もコストの低いノードだけが保持される。

アルゴリズム1は、ハイブリッドA*アルゴリズムが従う手順を示す。現在の状態とゼロコストがオープンリストに追加される。最もコストの低い状態(x_k)は、オープンリストから除外され、クローズドリストに追加される。ゴール状態に達したら探索は終了する。そうでない場合は、利用可能なアクション(ステアリング角度と方向)を使用して、遷移モデルを介して次の状態を取得する。

アルゴリズム1 関数ハイブリッドA*。

Require: x_s : start state, g : goal state, $h(x)$: heuristic, O : obstacle

```

1:  $\mathbb{C} \leftarrow \emptyset$ 
2:  $N = \{x_s, 0, 0, 0, \emptyset\}$ 
3:  $\mathbb{O} \leftarrow N$ 
4:  $KeepSearching \leftarrow 1$ 
5:  $KeepSearching \neq 1$ である間
6:   if  $\mathbb{O}$  is not empty then
7:      $N = \text{Extract node with minimum } c + h \text{ from } \mathbb{O}$ 
8:      $Add N \text{ to } \mathbb{C} \text{ and Delete from } \mathbb{O}$ 
9:     if  $N$  is  $g$  then
10:        $trajectory = \text{Backtracking}(N)$ 
11:        $KeepSearching \leftarrow 0$ 
12:     else
13:       foreach  $a \in \text{Available action}(N)$  do
14:          $\{x_n, c_n\} \leftarrow \text{transition}(N.x, a, N.c)$ 
15:         if  $IsCollide(x_n, O)$  is 0 then
16:            $N_n = \{x_n, c_n, h(x_n), a, N\}$ 
17:           if  $N_n \in \mathbb{C}$  then
18:             continue
19:           else if  $N_n$  has the same state with
20:              $n \in \mathbb{O}$  with smaller cost then
21:                $Replace n \text{ with } N_n$ 
22:           else
23:              $Add N_n \text{ to } \mathbb{O}$ 
24:
25: return  $trajectory$ 

```

検索された状態(x_{k+1})が衝突がなく、オープンリストにあるが、ステップコストとヒューリスティックの合計がリスト内の状態より低い場合、対応する状態が置き換えられ、状態(x_{k+1})がオープンリストにない場合、オープンリストに追加される。このプロセスをオープンリストの状態に対して繰り返し、選択された状態がゴールであれば、探索は終了し、クローズドリストを介してバックトラックにより最適な軌道が返される。全体の疑似コードをアルゴリズム1に示す。Nは{状態、行動、経路コスト、ヒューリスティック、親ノード}からなるノードを表す。 o と c はそれぞれオープンリストとクローズドリストを示す。さらに、IsCollide手続きは、車両が衝突していないときに0を返し、遷移手続きは、与えられた制御アクションの次の状態を得るために(1)を使用する。

III. ニューラル・ハイブリッドA*アルゴリズム

A. CVAE Architecture

自律駐車のための実現可能な軌道の分布を学習するために、条件付き入力を持つエンコーダ・デコーダネットワークを含むCVAEアーキテクチャが使用される。本研究で利用する条件付き入力は、初期点とゴール点の位置と方位、障害物、自由空間を含むマップ情報を表現している。全体のアーキテクチャを図1に示す。マップ情報 c を表現するために2次元(2-D)画像が使用される。

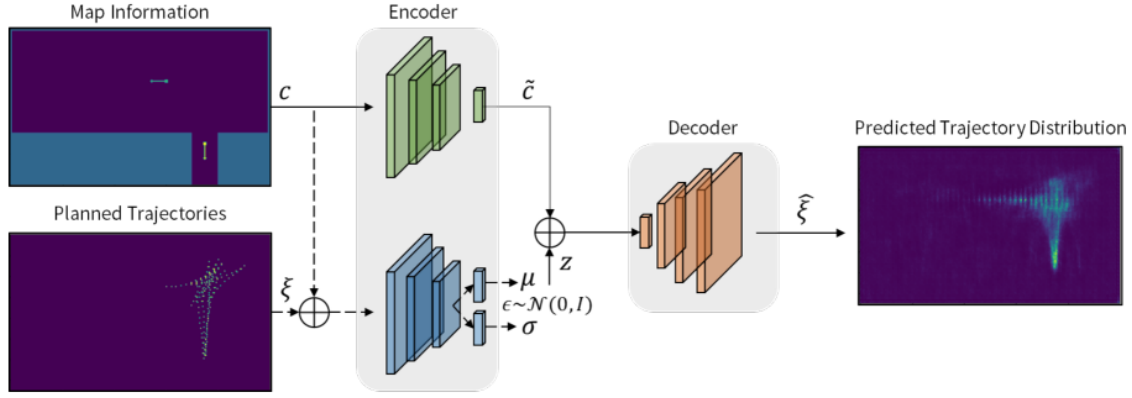


図1. ニューラル運動計画のためのCVAEアーキテクチャの提案。

幅は250m、高さは150mで、分解能は0.1mである。画像の各ピクセルは、自由空間が0、障害物が1、開始位置が2で方位情報を示す矢印が付いた状態が3、ゴール位置が3で方位情報を示す矢印が付いた状態が0という特定の値を持つ。真の軌道 ξ は、ハイブリッドA*アルゴリズムを用いて得られた計画軌道である。合計5つの軌跡が生成され、2次元画像中の対応する画素に1の値が割り当てられる。点線で描かれた真の軌道 ξ は、学習段階でのみ使用される。

CVAEは生成モデル $p_\phi(\xi, z)$ と推論モデル $q_\phi(z, \xi)$ からなり、潜在変数 z は再パラメータ化トリック[17]を用いて以下のように表現される：

$$z = \mu_\phi(c, \xi) + \sigma_\phi(c, \xi) \times \epsilon \quad (2)$$

ここで、 ϕ と ρ はそれぞれエンコーダーとデコーダーのネットワークのパラメータである。正規分布 $N(0, 1)$ を利用して標準化する。予測された軌跡分布 $\xi \sim$ と計画された軌跡集合 ξ との誤差を最小化するために、再構成損失はL2損失として定義される。CVAEアーキテクチャは損失関数を最小化することで学習され、以下のように定義される：

$$\mathcal{L} = \mathcal{L}_{REC} + \mathcal{L}_{KL} = \|\xi - \hat{\xi}\|^2 + \beta D_{KL}(q_\phi(z|c, \xi) || p(z)) \quad (3)$$

ここで、ハイパーパラメータ β は2つの損失のバランスをとる。前者の損失は再構成損失を表し、後者は多変量正規分布とエンコーダからの出力分布の間のKLダイバージェンス損失である。

真の軌道セット ξ はテストフェーズでは利用できない。したがって、 z は $N(0, 1)$ から直接サンプリングされ、デコーダ部分のみが利用される。条件 c はエンコーダに通され、推論用のデコーダで 사용되는潜在ベクトル c_{302} を生成する。

マップ情報などの異なる条件を与えることで、様々な予測軌跡分布を生成することができる。例えば、開始位置とゴール位置を方位値とともに変化させることで、異なる軌跡分布が生成される。図2は、スタート位置と目標駐車スペースが同じであるにもかかわらず、

アルゴリズム2 関数ニューラル・ハイブリッドA*。

Require: x_s : 開始状態, g : 目標状態, $h(x)$: ヒューリスティック, 0 : 障害物, D_{map} : 予測軌道分布図

```

1:  $\mathbb{C} \leftarrow \emptyset$ 
2:  $N = \{x_s, 0, 0, 0, \emptyset\}$ 
3:  $\mathbb{O} \leftarrow N$ 
4:  $KeepSearching \leftarrow 1$ 
5:  $KeepSearching$ が1である間
6:   if  $\mathbb{O}$  is not empty then
7:      $N = Extract\ node\ with\ minimum\ c + h\ from\ \mathbb{O}$ 
8:     Add  $N$  to  $\mathbb{C}$  and Delete from  $\mathbb{O}$ 
9:     if  $N$  is  $g$  then
10:      trajectory = Backtracking( $N$ )
11:      KeepSearching  $\leftarrow 0$ 
12:   else
13:     foreach  $a \in Available\ action(N)$  do
14:        $\{x_n, c_n\} \leftarrow transition(N.x, a, N.c)$ 
15:       if Rand() > 0.2 then
16:         if CheckDistMap( $x_n, D_{map}$ ) then
17:           continue
18:         if IsCollide( $x_n, O$ ) is 0 then
19:            $N_n = \{x_n, c_n, h(x_n), a, N\}$ 
20:           if  $N_n \in \mathbb{C}$  then
21:             continue
22:           else if  $N_n$  has the same state with
23:              $n \in \mathbb{O}$  with smaller cost then
24:             Replace  $n$  with  $N_n$ 
25:           else
26:             Add  $N_n$  to  $\mathbb{O}$ 
27:
28: return trajectory

```

ゴール位置の方位角が逆である場合に予測される軌跡分布を示している。この結果から、提案するCVAEネットワークは学習に成功し、地図情報の違いを区別できることがわかる。

B. Neural Hybrid A*

Neural Hybrid A*は、CVAEネットワークから得られた軌跡分布図情報を使用する。

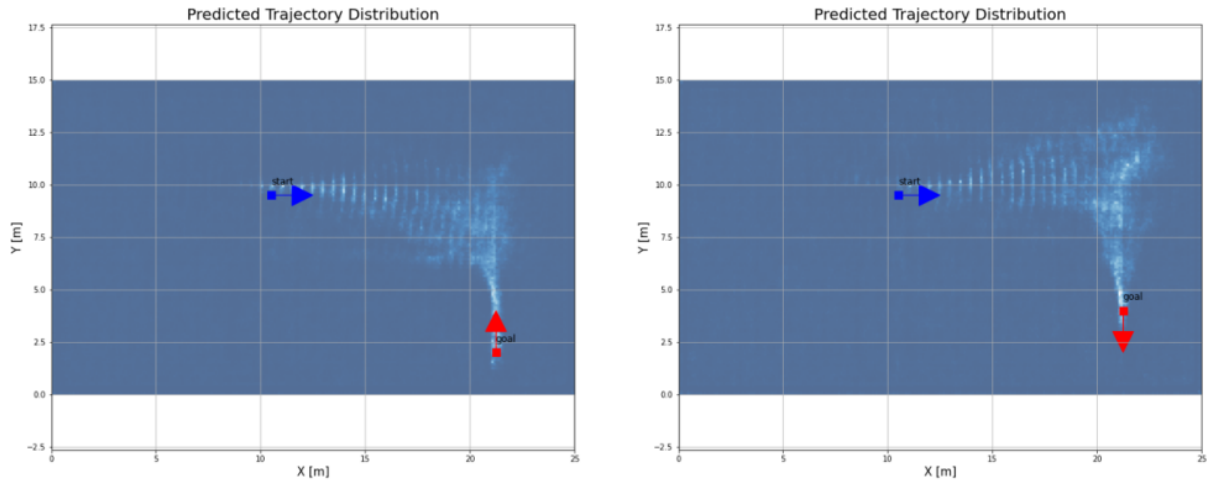


図2. 予測された軌跡分布の例: (a)ゴール位置での方位角が 90° の場合、(b)ゴール位置での方位角が -90° の場合の結果。 at

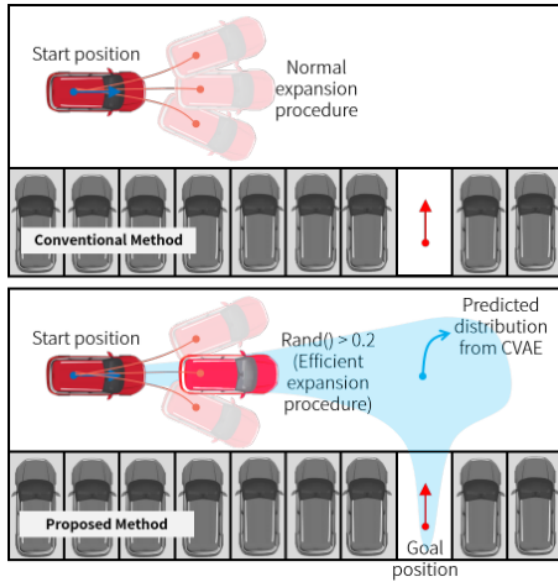


図3. 予測された分布に基づく提案された計画戦略の例。

詳細はアルゴリズム2に示されており、要件には新しい入力が含まれ、15行目から17行目は新たに追加されたものである。Dmapは予測された軌跡の分布図を表す。15行目、乱数 $\text{Rand}() \in (0, 1)$ をチェックし、Dmapを利用するかどうかを判断する。本研究では、ニューラルネットワークモデルの助けを借りて、ノード拡張の80%を進める。CheckDistM ap手順は、Dmapの現在の状態がある閾値より大きい値を持つ場合、0を返す。16行目と17行目でわかるように、Dmapが状態 x_n の位置で閾値以下の値を持つ場合、13行目で選ばれたアクションは不要と判断され、そうでない場合は次のステップに進む。

このように、この状態拡張戦略では、学習された軌跡分布に基づいて主に拡張を行うことで、制御動作を効率的に選択することができる。図3は、前述の計画戦略の画像例である。従来の方法を用いた上図では、あらかじめ定義されたアクションセットによる展開が行われる。下の画像では、CVAEから予測された分布に基づいて、効率的な展開法が実行される。

IV. SIMULATION

本節では、データセットの取得、実装の詳細、CVAEモデルの評価、および結果について述べる。ニューラルネットワークモデルは、オープンソースのPythonライブラリであるPyTorch [18]を用いて設計・学習した。運動計画アルゴリズムであるハイブリッドA*はPythonで実装されている。

A. Dataset

モデルを学習するために、Hybrid A*アルゴリズムを用いてエキスパート軌道セットを収集する。各シーンは5つの軌跡の組み合わせを持つ。ハイブリッドA*アルゴリズムは決定論的手法であり、同じ解が得られる。したがって、提案するネットワーク学習では、行動集合Dの順序をランダムに混合し、異なる軌道を得ることができる。データ収集のために、合計1,000のシナリオをシミュレートした。開始位置は自由空間でランダムにサンプリングし、方位角は 0° または 180° からサンプリングする。駐車場はランダムに選択され、それに応じてゴール位置が選択される。ゴール位置の方位角は -90° または 90° からランダムにサンプリングされる。収集したデータセットは、提案するCVAEネットワークを学習するための条件付き入力とラベルとして利用される。

B. 実装の詳細

ネットワーク全体はエンコーダとデコーダで構成される。デコーダで使用するマップ情報を符号化するためのエンコーダと、潜在ベクトルを生成するためのマップ情報とともに計画された軌道を符号化するためのエンコーダがある。

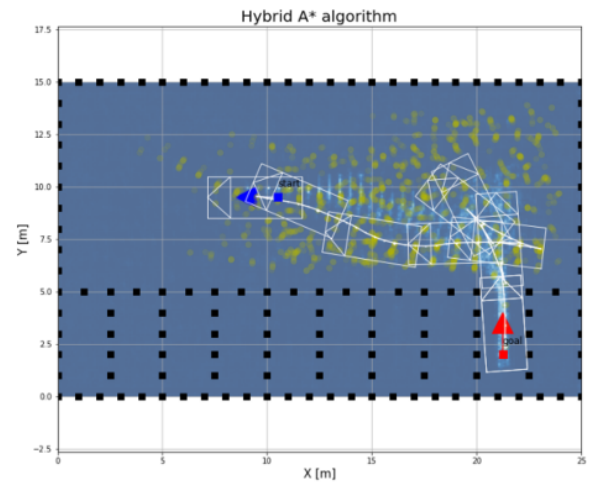
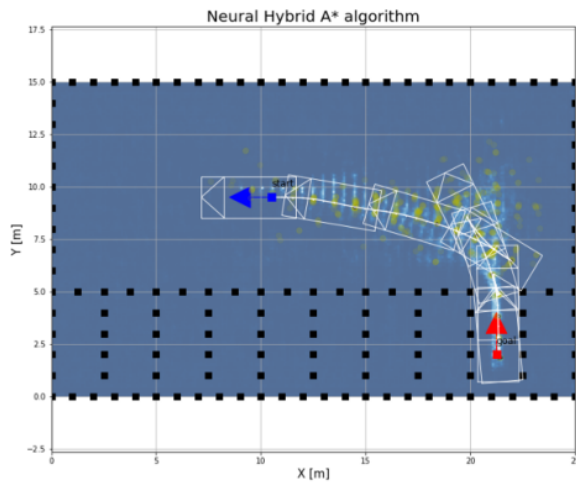


図4. シナリオ1における提案手法と従来手法の比較。青色の分布は、提案するCVAEネットワークから予測される実現可能な軌道の分布を表す。黄色の点は、プランニングプロセス中に拡張されたノードを示す。

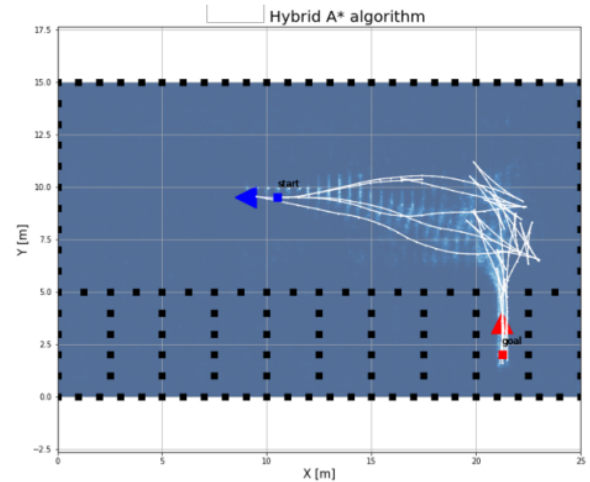
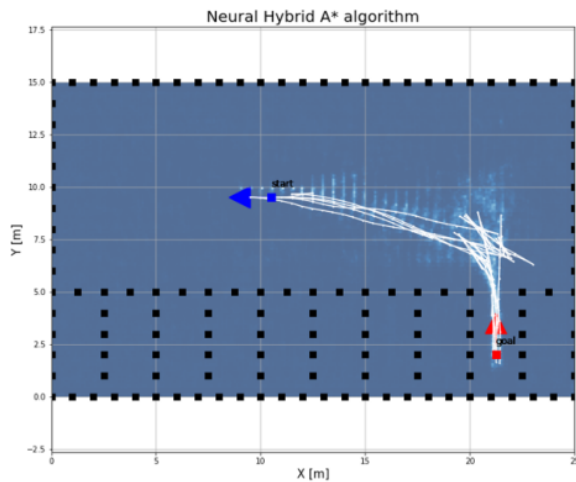


図5. シナリオ1で生成された5つの軌道：(a)提案手法の結果、(b)従来手法の結果。

エンコーダは、2次元畳み込み、バッチ正規化、整流線形ユニット(ReLU)活性化関数、完全連結層で構成される。畳み込みニューラルネットワーク(CNN)モデルは、出力チャンネルが16、32、64の3つの畳み込み層から構成される。最初のエンコーダでは、結果は32の隠れユニットを持つ完全連結層に渡され、これは符号化された条件の次元である。2番目のエンコーダでは、結果は潜在変数の次元である32の隠れユニットを持つ2つの完全連結層に渡される。

デコーダは、完全連結層、2次元デコンボリューション、バッチ正規化、ReLUで構成される。デコーダのCNNモデルは出力チャンネルが32、16、1の3つの畳み込み層から構成され、カーネルサイズとストライドはエンコーダと同じである。

学習過程では、学習率0.001の標準的なAdamオプティマイザを用いて最適化を行った。また、CVAE損失のパラメータ β は以下のように設定される。

0.1.

C. C. CVAEモデルの評価

提案手法の性能を評価するために、テストデータセットを生成する。初期位置や方位角が異なる地図情報と、トレーニングセットの地図情報を推論用に選択する。図5～図10は、提案するニューラル・ハイブリッドA*アルゴリズムと従来のハイブリッドA*アルゴリズムの結果を比較したものである。

図4では、最初のシナリオの結果を比較している。青い分布は、提案するCVAEネットワークから予測される実現可能な軌道の分布を表す。黄色の円は、このプロセスで拡張されたノードを表す。左図は提案手法の結果である。このように、拡張されたノードとともに、拡張は学習された分布に対して行われることがほとんどである。一方、右図は学習した分布の助けを借りない場合の結果である。ノードは分布に関係なく散在しており、計算量が多い。

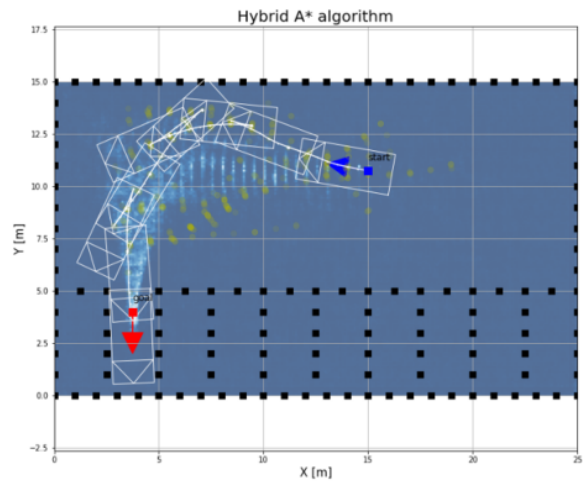
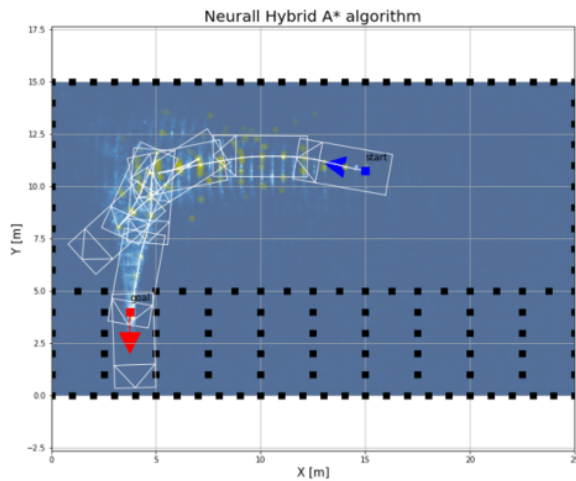


図6. シナリオ2における提案手法と従来手法の比較。

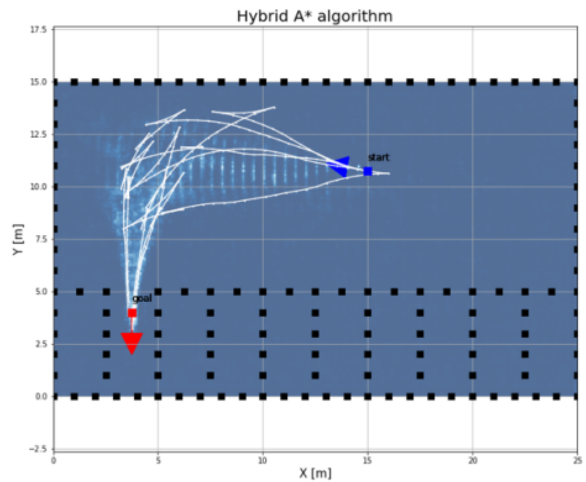
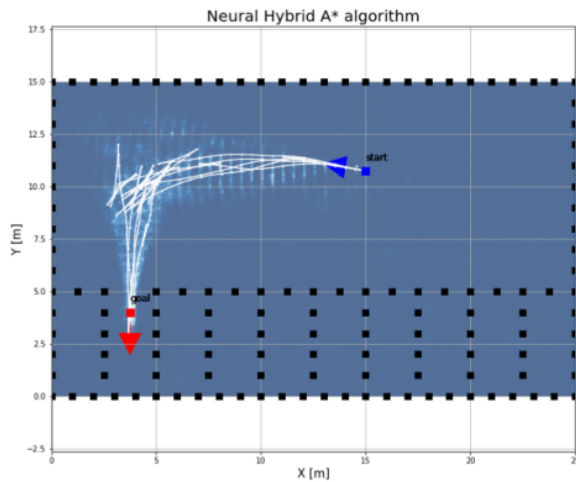


図7. シナリオ2で生成された5つの軌道：(a)提案手法の結果、(b)従来手法の結果。

図5は、生成された5つの軌道を用いた計画結果である。左の図に示すように、生成された軌道は、ほとんどが予測された分布上にある。右図は、CVAEネットワークの情報がないため、分布に関係なく計画結果がよりランダムであることを示している。

図6に第2シナリオの結果を示す。初期点の方位角は 170° とし、学習データセットには含まれない。生成された分布は、分布図上でほぼ展開が完了するため、ガイダンスマップとしてうまく利用されているようである。しかし、右図に見られるように、分布図を利用しない場合、より広い範囲をカバーすることで、ノード拡張のコストが高くなる。

図7は、生成された5つの軌跡の結果も示している。左図に示すように、生成された軌道は、ニューラル・ハイブリッドA*法を用いて予測された分布にほぼ沿ったものである。しかし、分布図がない場合、よりランダムな結果を示す。

図8～図11は、前述のシナリオと同様の結果を示している。図9に示す第3のシナリオでは、CVAEネットワークは、学習データセットと大きく異なる初期方位角 30° でも、実現可能な分布マップを提供する。

TABLE I

Cメトリクスの比較.

		Scen. 1	Scen. 2	Scen. 3	Scen. 4
Comparison (vs Hybrid A*)	Time	24.75%	30.33%	50.0%	10.86%
	Node	50.91%	53.30%	79.36%	35.35%

ニューラル・ハイブリッドA*アルゴリズムの性能を評価するために、計算時間とオープンリストのノード数からなる2つのメトリクスを、4つのテストシナリオの下で評価する。表 I は、性能比較をメトリクスの観点から示したものである。統計的な結果は、シナリオを5回評価し、2つの指標の平均値を使用することで得られる。

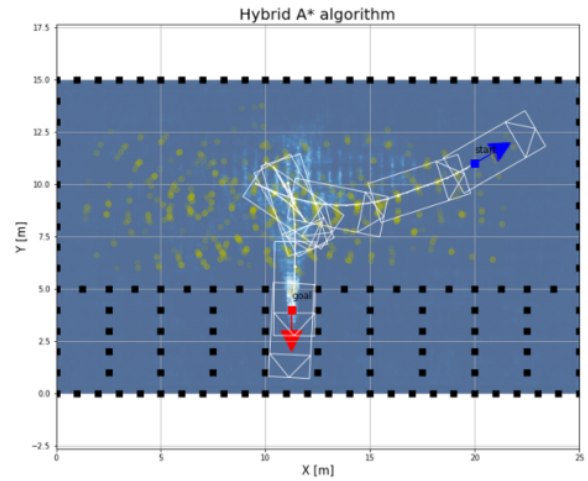
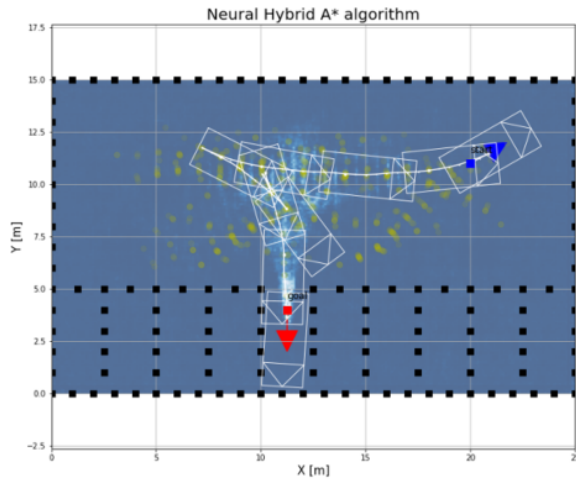


図8. シナリオ3における提案手法と従来手法の比較。

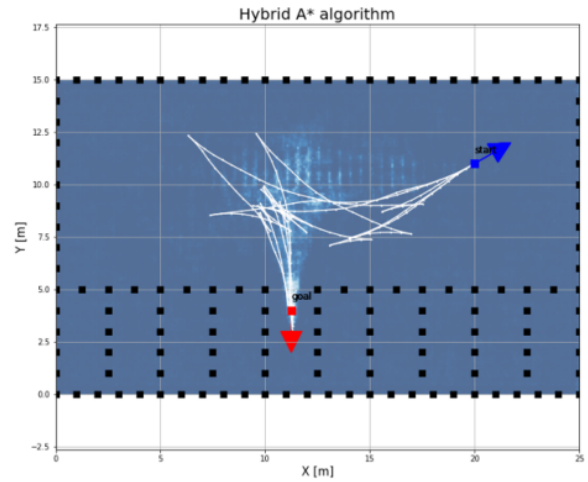
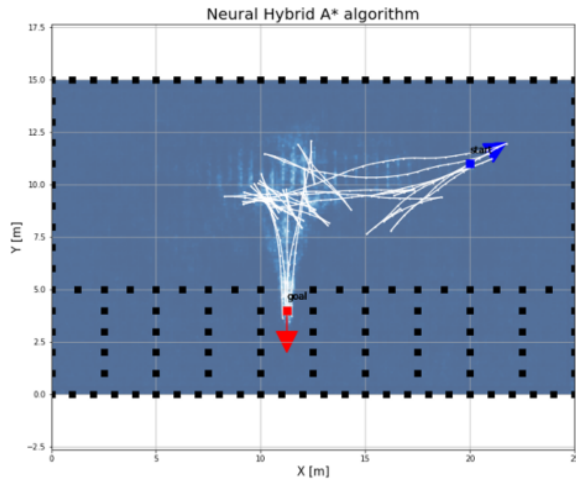


図9. シナリオ3で生成された5つの軌道：(a)提案手法の結果、(b)従来手法の結果。

全てのテストシナリオにおいて、提案するニューラル・ハイブリッドA*法は、従来のハイブリッドA*アルゴリズムと比較して、計算時間とノード数が大幅に削減される。計算時間から見ると、ニューラル・ハイブリッドA*アルゴリズムが自律駐車の問題の状態拡張効率を向上させることがわかる。また、ノードに関しては、提案手法は従来手法よりもはるかにメモリ消費量が少ないことが示された。

V. CONCLUSIONS

本論文では、ディープジェネレーティブネットワークと従来の探索ベースの経路計画アルゴリズムを統合した、自律駐車のための新しいハイブリッド運動計画戦略を紹介する。CVAEネットワークは、障害物情報とともに初期位置とゴール位置が与えられたとき、実現可能な軌道の予測分布を学習するように構築される。提案モデルは、ハイブリッドA*アルゴリズムによって生成された実現可能なパスを持つ複数のマップを学習する。

ニューラル・ハイブリッドA*アルゴリズムによるシミュレーションの結果、ニューラル・ハイブリッドA*アルゴリズムは計算時間とノード数を大幅に削減することがわかった。提案手法は予測分布に基づく拡張戦略を用いるため、他の探索に基づく計画アルゴリズムも適用可能である。今後の研究では、より多様な駐車場シナリオでのシミュレーションを実施する予定である。

REFERENCES

- [1] M. Khalid, K. Wang, N. Aslam, Y. Cao, N. Ahmad, and M. K. Khan, "From smart parking towards autonomous valet parking: A survey, challenges and future works," *Journal of Network and Computer Applications*, p. 102935, 2020.
- [2] Y. Dong, Y. Zhang, and J. Ai, "Experimental test of artificial potential field-based automobiles automated perpendicular parking," *International Journal of Vehicular Technology*, vol. 2016, 2016.
- [3] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How, "Motion planning for urban driving using rrt," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1681–1686.
- [4] J. Vlasak, M. Sojka, and Z. Hanzálek, "Accelerated rrt* and its evaluation on autonomous parking," *arXiv preprint arXiv:2002.04521*, 2020.

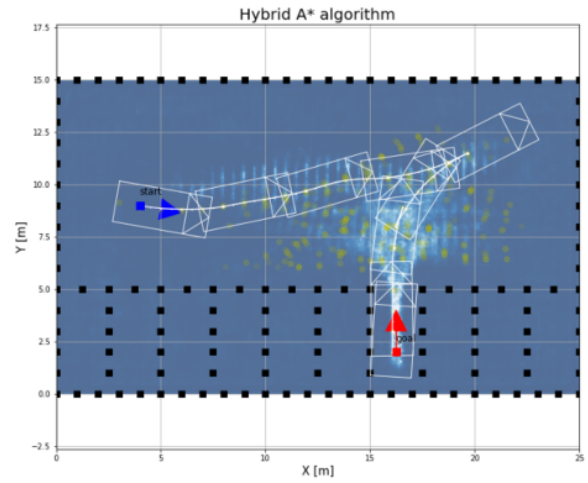
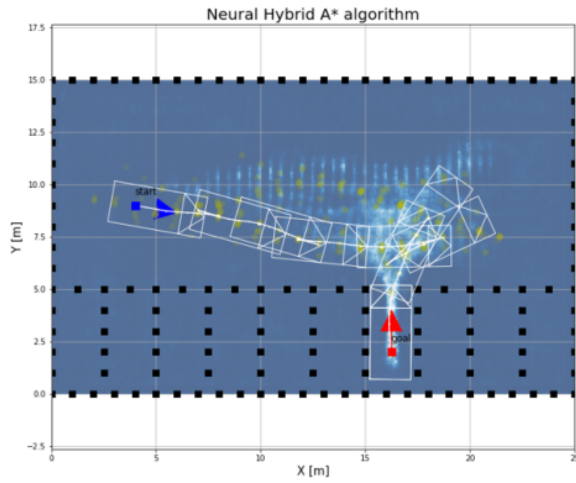


図10. シナリオ4における提案手法と従来手法の比較。

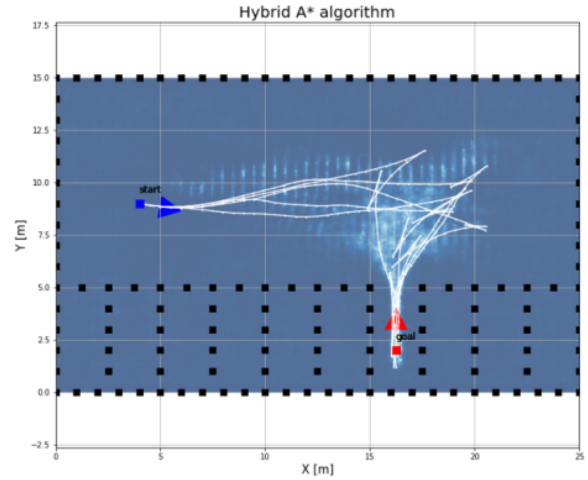
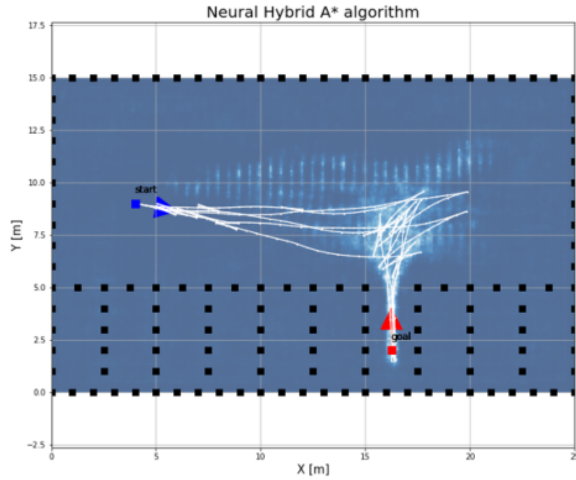


図11. シナリオ4で生成された5つの軌道: (a)提案手法の結果、(b)従来手法の結果。

- [5] R. Benenson, S. Petti, T. Fraichard, and M. Parent, "Integrating perception and planning for autonomous navigation of urban vehicles," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 98–104.
- [6] L. Cheng, C. Liu, and B. Yan, "Improved hierarchical a-star algorithm for optimal parking path planning of the large parking lot," in *2014 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2014, pp. 695–698.
- [7] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.
- [8] O. Sharma, N. C. Sahoo, and N. Puhon, "Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: A state-of-the-art survey," *Engineering Applications of Artificial Intelligence*, vol. 101, p. 104211, 2021.
- [9] A. Bicchi, A. Marigo, and B. Piccoli, "On the reachability of quantized control systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 546–563, 2002.
- [10] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*. Munich Germany, 2005, pp. 1–7.
- [11] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7087–7094.
- [12] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, "Neural rrt*: Learning-based optimal path planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, 2020.
- [13] N. Chen, M. Karl, and P. Van Der Smagt, "Dynamic movement primitives in latent space of time-dependent variational autoencoders," in *2016 IEEE-RAS 16th international conference on humanoid robots (Humanoids)*. IEEE, 2016, pp. 629–636.
- [14] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, pp. 3483–3491, 2015.
- [15] X. Feng, Z. Cen, J. Hu, and Y. Zhang, "Vehicle trajectory prediction using intention-based conditional variational autoencoder," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3514–3519.
- [16] A. Bhattacharyya, M. Hanselmann, M. Fritz, B. Schiele, and C.-N. Strachle, "Conditional flow variational autoencoders for structured sequence prediction," *arXiv preprint arXiv:1908.09008*, 2019.
- [17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.