

動的環境における動作計画のための C-space の予測 時系列データに対応した Vox2C-space

Predicting C-space for motion planning in dynamic environments
Vox2C-space for time series data

木南 貴志 (中部大) 学 坂本健太郎 (中部大) 山内 悠嗣 (中部大)

Takashi KINAMI, Chubu University

Kentaro SAKAMOTO, Chubu University

Yuji YAMAUCHI, Chubu University, yuu@isc.chubu.ac.jp

In dynamic environments, robots must plan their movements while considering obstacles that move within the same space. Configuration space (C-space), which is a multidimensional space representing the robot's posture, is used for motion planning. By creating a path to avoid configuration obstacles in C-space, the robot can generate movements that avoid obstacles in real space. While C-space generally represents the robot's posture at the current time, predicting the future C-space in a dynamic environment can enable motion planning in complex environments. This paper proposes a deep learning network that rapidly predicts the C-space at the subsequent period using time-series information.

Key Words: Motion Planning, Deep Learning, Configuration-space, Time-series data

1 はじめに

人と一緒に作業を行う協働ロボットの需要が拡大し、作業の自動化のために様々な分野に導入されている。人間と同じ空間でロボットが作業をするには、周囲の環境を把握して障害物を避けるだけでなく、作業を行う人間に衝突しない動作が求められる。

動作計画では Configuration-space(C-space) と呼ばれるロボットの姿勢を表現した空間が用いられる。実空間におけるロボットと障害物が衝突するような姿勢は、C-space 上においてコンフィギュレーション障害物として写像されるため、C-space 上でコンフィギュレーション障害物を回避する経路を生成することで、実空間においても障害物を回避するような動作を生成できる。人間のように動きを伴う物体が存在するような動的環境に対応するためには、動きを予測して動的な障害物を避けるようなロボットの動作を計画する必要がある。しかしながら、動的な障害物を考慮した C-space の作成や動作計画法は提案されているが、動的障害物の動きを予測するまでには至っていない。

そこで、本稿では時系列情報から次時刻の C-space を予測する深層学習ネットワークを提案する。既に我々は、3次元の占有グリッドマップとして表現されるボクセルデータから C-space を生成する Vox2C-space[1] と呼ぶ深層学習ネットワークを提案している。本研究では、Vox2C-space を拡張して時系列データに対応させることで次時刻の C-space を高速に予測する。予測した C-space を用いて動作計画を行うことで、動的な環境に対応することが可能となる。

2 関連手法

2.1 静的な環境下における動作計画法

現在に至るまでに、動作計画法に関する多くの手法が提案されている。動作計画法は軌道を生成する過程によって、幾つかの種類に分けられる。

ポテンシャルベース

目標姿勢に引力ポテンシャル、障害物に斥力ポテンシャルを仮想的に設置したポテンシャル場を作成することで目標姿勢までの軌道を生成するポテンシャルベースの手法 [2] が提案されている。この手法は、障害物の回避と目標姿勢への到達を同時に実行できる。一方で、ポテンシャル関数の勾配が 0 になる停留現象が生じる可能性がある。また、3次元以上の多次元空間において膨大な計算量が発生する。

ランダムサンプリングベース

C-space 上にランダムにノードをサンプリングし、ノード間を結ぶことで目標姿勢までの軌道を生成するランダムサンプリングベースの手法が提案されている。代表的な手法としては Probabilistic Road Map(PRM)[3] や Rapidly-exploring Random

Tree(RRT)[4] が存在する。PRM と RRT はどちらもノードをランダムにサンプリングする手法であるが、前者は事前にサンプリングするのに対して、後者は逐次的にサンプリングする。PRM では、事前にランダムにサンプリングされたノード間を結ぶことで軌道を生成する。RRT では、ランダムにサンプリングされたノードと最も近いノードを結ぶことを逐次的に繰り返すことで目標姿勢までの軌道を生成する。RRT は高速に動作し実装も容易なことから、最適性を付与した RRT*[5] などの RRT を拡張した多くの手法が提案されている。また、高次元の C-space においても比較的高速に動作することから多軸ロボットの動作計画にも用いられる。

これらの静的な環境下における動作計画法は、安全が確保された環境においては効率よく動作する。しかし、動的な環境では動的物体を発見した際に軌道の再計画が必要になり無駄のある軌道を生成する。また、障害物が回避できなかった場合は、障害物に衝突する恐れがあり人間に危険が及ぶ可能性も存在する。

2.2 動的な環境下における動作計画法

近年では、動的環境に対応した動作計画法に関する研究が幾つか報告されている。Leven らは、PRM をベースにした Dynamic Roadmaps(DRM)[6] を提案した。DRM は、障害物が存在しないと仮定した環境下でロードマップ法によりグラフを作成する。グラフを生成する際、ノードはランダムにサンプリングされるが、ロボットアームの手先位置や姿勢をどの程度自由に操作できるかを表す可操作度 [7] が低い領域を重点的にサンプリングするように重み付けることで経路が見つかる確率を向上させている。経路探索を行うグラフは C-space 内に生成するが、動的障害物が移動すると C-space 内の衝突領域も動的に変化するため、グラフが複数に分割される可能性がある。経路探索においてグラフが複数に分割されると目標姿勢に辿り着けなくなる可能性があるため、グラフを作成する段階でノード間のエッジが疎である領域にエッジを追加することで動的障害物にロバストなグラフを生成する。軌道を追従する際は、上記のグラフを使用して経路を探索し、動的障害物を観測した場合には部分的にグラフを修正することで、軌道を再計画して動的障害物を回避する。

RRT をベースにした手法として、Dynamic Rapidly-exploring Random Trees(DRRT)[8] が提案されている。DRRT では、最初に C-space 上で RRT と同様の手順で初期姿勢から目標姿勢までの経路を求める。この時、最終的な経路以外の枝分かれした経路も記録する。次に、RRT の探索結果に従って軌道の追従を開始する。この際、動的障害物の移動によりコンフィギュレーション障害物が経路を妨害するようなことがあれば、該当する経路を削除する。経路の削除により目標姿勢までの経路が消滅した場合

には、RRT による探索を再開し、目標姿勢までの経路が見つかったら探索を停止する。上記の手順を軌跡の追従が完了するまで繰り返すことで動的障害物を超えるような動作計画が可能となる。これらの従来法は、グラフを部分的に修正することで動的環境に対応しているため、軌跡を大きく修正する必要がある場面での使用は困難である。

2.3 時系列対応ネットワーク

時系列に対応したネットワークには様々なものが提案されている。その中で代表的な手法に回帰型ニューラルネットワーク (Recurrent Neural Network:RNN) が存在する。RNN は、時刻 $t-1$ の隠れ層からの出力 h_{t-1} を時刻 t の隠れ層に投入する構造になっている。また、隠れ層の内部では時刻 t の入力 x_t と、一時刻前の隠れ層からの出力 h_{t-1} を活性化関数である \tanh 関数に通すことで出力 y_t を求めている。RNN は過去の情報を伝播させていくことで時系列を考慮した学習ができるネットワークとなる。一方で、長期の時系列データになると勾配消失が発生し、上手く学習できないといった欠点も存在する。

Sepp らは RNN の欠点を解決するために Long Short Term Memory (LSTM)[9] を提案した。LSTM は入力ゲート、出力ゲート、忘却ゲートと呼ばれる 3 種類のゲートを保有している。入力ゲートと出力ゲートは、それぞれ $t-1$ のユニットの入力、及び出力をどの程度受け取るかを調整しており、忘却ゲートでは過去のデータを保存するためのユニットである Constant Error Carrousel (CEC) の中身をどの程度残すかを調整する。これらのゲートにより CEC を更新することで、LSTM は長期の時系列データへの対応を可能としている。

LSTM は、長期のデータも学習できるため多くの時系列データに対して有効であるが、ネットワークが比較的大きいため計算コストが高くなる欠点が存在する。そこで、Chung らは Gated Recurrent Unit (GRU)[10] を提案した。GRU は次の時刻への出力が隠れ状態 h の 1 種類のみである。これは、LSTM における記憶と忘却の役割を 1 つにまとめているためであり、ゲートの数もアップデートゲートとリセットゲートの 2 種類に削減されている。このように、LSTM と比較して GRU は小規模なネットワークであるが、LSTM と同等の性能を発揮できるとされている。

時系列に対応したネットワークは画像生成の分野にも応用されている。Wu らは、物体の各パーツを時系列データとして扱うことで高精度に画像を生成する PQ-NET[11] を提案した。PQ-NET は、3 次元画像として表現された物体をパーツごとに生成することで複雑な 3 次元画像を高精度に生成できる。

2.4 提案手法の概要

本研究では、Vox2C-space[1] への入力を時系列のボクセルとし、GRU を導入した時系列ネットワークにより次時刻の C-space を高速に予測する。本研究の貢献点は以下の通りである。

- 未来の C-space を予測
本研究では Vox2C-space の高速かつ完全な C-space を生成する利点を活かしつつ、次時刻の C-space を予測することが可能となる。
- 動的環境への適応
予測した未来の C-space を用いて動作を計画することで動的環境に対応することができる。また、本研究では C-space を用いる従来の動作計画法をそのまま適用可能である。

3 提案手法

3.1 時系列に対応した Vox2C-space

提案する時系列に対応した Vox2C-space は、図 1 のように占有グリッドマップとして表現される 2 値のボクセルから未来の C-space を生成することを目的としたネットワークである。時系列に対応した Vox2C-space は 2 段階で学習が行われる。1 段階目では、図 2 に示す step1 のようにエンコーダ E がボクセルの特徴量 z を抽出し、デコーダ D が z の情報を元に C-space を生成するように学習する。

2 段階目では、学習したエンコーダ・デコーダを用いて、時系列ネットワークを学習する。今回は、時系列ネットワークに GRU を採用した。図 2 に示す step2 のように、GRU は n 時刻分の特徴量 $Z = \{z_1, z_2, \dots, z_{n-1}, z_n\}$ を入力した時、出力が特徴量

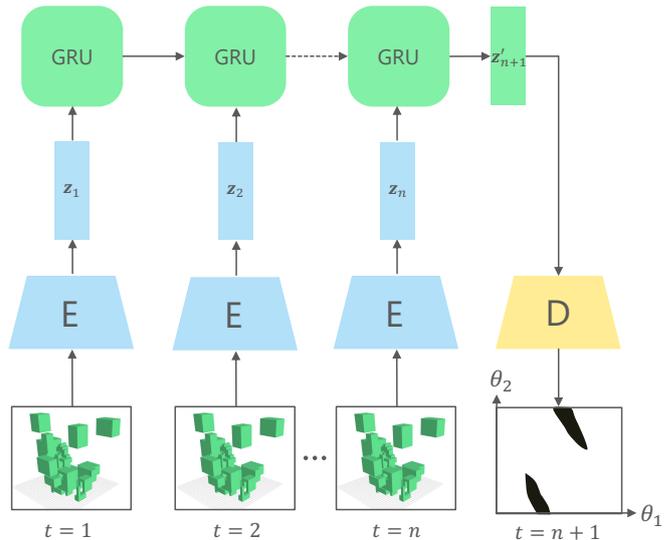
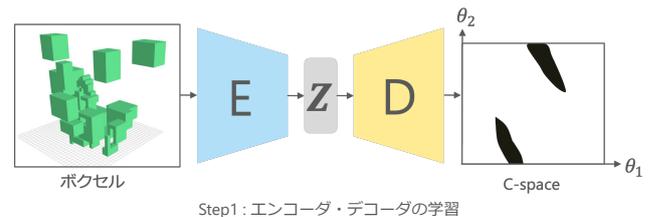
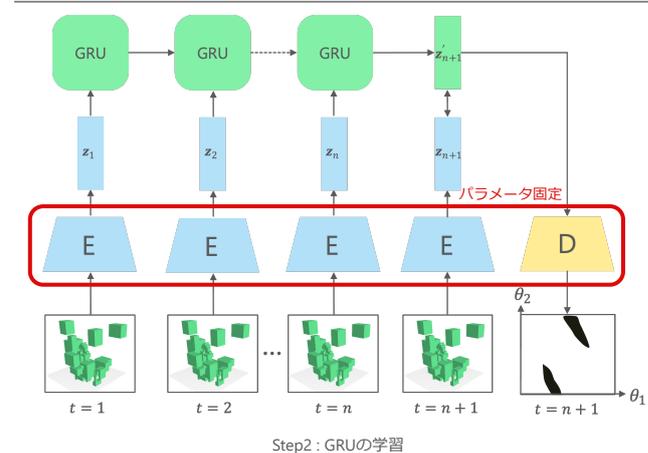


Fig.1 Network overview.

z_{n+1} となるように学習する。ここで、 z_i は学習済みエンコーダに i 時刻目のボクセルを入力した時に得られる特徴量である。なお、GRU 部分を学習の際は、エンコーダ・デコーダのパラメータは更新しない。



Step1: エンコーダ・デコーダの学習



Step2: GRUの学習

Fig.2 Learning process.

3.2 ネットワークアーキテクチャ

図 3 に時系列に対応した Vox2C-space のネットワークアーキテクチャを示す。時系列に対応した Vox2C-space は、占有グリッドマップとして表現された 2 値のボクセルをエンコーダに入力することで n 時刻分の特徴量を抽出する。そして、得られた n 時刻分の特徴量を GRU に入力することで、 $n+1$ 時刻分の特徴量を得る。最後に、この特徴量をデコーダに入力することで $n+1$ 時刻目の C-space を生成する。エンコーダは、 $32^3 \times 1$ のボクセルを入力として、3 次元畳み込み層、インスタンス正規化層、ReLU、プーリング層を用いて $4^3 \times 512$ の特徴量を抽出する。エンコーダから出力された特徴量は 3 次元であるため、特徴量を 32,768

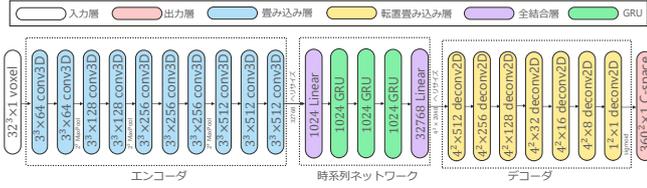


Fig.3 Network architecture.

次のベクトルに平滑化する．次に，平滑化したベクトルを入力として，全結合層，ReLU，GRUを用いて特徴量を抽出する．出力された特徴量は1次元であるため，特徴量を $4^2 \times 2,048$ にリサイズする．最後に，リサイズした特徴量を入力として，転置畳み込み層，インスタンス正規化層，ReLUと続き，最後にシグモイド関数を通して $n+1$ 時刻目のC-spaceを生成する．出力されるC-spaceは，非衝突領域 C_{free} が0，衝突領域 $C_{obstacle}$ が1として表現される．

3.3 損失関数

損失関数は，エンコーダ・デコーダ学習時とGRU学習時で異なるものを使用する．エンコーダ・デコーダ学習時の損失関数は生成されたC-spaceと真のC-spaceのバイナリクロスエントロピー誤差の平均値として定義され，C-spaceのセルの総数を N ，衝突領域 $C_{obstacle}$ である確率を p_i ，それに対応する真値を gt_i とした時，式(1)で表現される．

$$L_{enc} = \frac{1}{N} \sum_{i=1}^N [gt_i \log(p_i) + (1 - gt_i) \log(1 - p_i)] \quad (1)$$

GRU学習時の損失関数は，学習済みエンコーダに $n+1$ 時刻のボクセルを入力した時に得られる特徴量 z_{n+1} と，GRUから出力される特徴量 z'_{n+1} より式(2)のように定義される．

$$L_{gru} = \|z'_{n+1} - z_{n+1}\|_2 \quad (2)$$

ここで， z_{n+1} は正解値として扱っており，学習時のみ計算することに注意が必要である．また，GRU学習時はエンコーダ・デコーダの重みは更新されない．

4 評価実験

提案手法の有効性を確認するために，生成される未来のC-spaceの精度をVox2C-spaceと比較する．Vox2C-spaceは時系列に対応していないため，時刻 t のボクセルを入力したときに， $t+1$ のC-spaceを出力するように学習する．提案手法は連続した3時刻のボクセルを入力した時に $t+1$ のC-spaceを出力するように学習する．

4.1 データセット

ロボットシミュレータGazeboを用いて，データセットを作成する．ロボットには1軸目が z 軸周り，2軸目が y 軸周りに回転するロボットアームを使用する．また，各軸の回転範囲は $[-180, 180]$ となる．シミュレーション空間にロボットと動的障害物を生成し，3次元データを点群として撮影することでボクセルとC-spaceを作成する．撮影の際は，上空に設置した4つのカメラを使用し， $2.3[m^3]$ の範囲を撮影する．動的障害物は，各辺の長さの範囲が $[0.1m, 0.5m]$ の直方体として表現し，範囲内のランダムな位置に3個生成する．また，動的障害物の速度の範囲は，ウォーキング～ランニング程度の速度を想定し， $[2.5km/h, 9.0km/h]$ に設定した．データセットの例を図4に示す．データセットはボクセルとC-spaceをペアとして，学習用を6,000シーン，評価用を2,000シーン作成した．

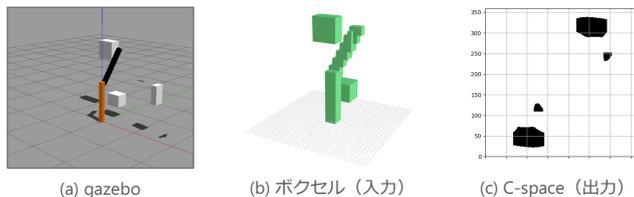


Fig.4 Example of dataset.

4.2 結果

各手法のIoUを表1に示す．表1の結果より，Vox2C-spaceと比較して提案手法はIoUが5%向上していることが分かる．次に，提案手法によって生成された未来のC-spaceを図5に示す．シーン1の結果を見ると，未来のC-spaceの大きな形状が捉えられていることが分かる．一方で，シーン2のような小さな障害領域が多数ある場合，数時刻遅れてから真値と同じようなC-spaceが生成されている．同様に，シーン3のような急激に障害物領域が変化するような場面においても，数時刻遅れてから真値と同じようなC-spaceが生成されていることが分かる．

Table 1 Comparison of IoU for each method.

ネットワーク	Vox2C-space	提案手法
IoU	0.51	0.56

4.3 考察

表1から，提案手法の方がIoUが高いことが分かる．これは，時系列に対応したVox2C-spaceにGRUを採用したことで，複数時刻のボクセルから時系列情報を抽出できたことが要因だと考えられる．

図5を見ると，3,4シーン目ではC-spaceの生成が数フレーム遅れる傾向にあるが，これはロボットと障害物の位置関係が要因と考えられる．シーン3のようなコンフィギュレーション障害物が離れた位置に複数ある際，実空間においてはロボットとわずかに衝突するような位置に動的障害物が存在していることが多い．また，シーン4のような急激な変化が起こる場面では，実空間では障害物同士が重なったり，ロボットと障害物が非常に近い位置にあると考えられる．このようなシーンではロボットと障害物，及び障害物同士の位置関係がより重要になる．しかし，今回は分解能が比較的低いボクセルを入力したため，このような微妙な位置関係を表現しきれなかった可能性がある．そのため，ボクセルの分解能を高くすればロボットと障害物の位置関係がより鮮明になり，C-spaceの生成が遅れるといった問題も軽減できると考えられる．

5 おわりに

本研究では，時系列に対応したVox2C-spaceを提案した．提案手法を用いて高速に次時刻のC-spaceを予測することで，動的環境へ適応することができる．今後は，C-spaceの長期的な予測を行う時系列ネットワークを検討する予定である．

参考文献

- [1] 木南 貴志，山内 悠嗣，“Vox2c-space:動作計画のための機械学習に基づくc-spaceの生成”，日本ロボット学会学術講演会，2022．
- [2] Y. Koren, and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation”, *ICRA*, pp. 1398–1404, 1991.
- [3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces”, *IEEE Trans. Robot.*, vol. 12, pp. 566–580, 1996.
- [4] S. M. Lavalle, “Rapidly-exploring random trees: A new tool for path planning”, *technical report*, 1998.
- [5] S. Karaman, and E. Frazzoli, “Incremental sampling-based algorithms for optimal motion planning”, *Robotics: Science and Systems*, 2010.
- [6] P. Leven, and S. Hutchinson, “A framework for real-time path planning in changing environments.” *IJRR*, vol. 21, no. 12, pp. 999–1030, 2002.
- [7] 吉川 恒夫，“ロボットアームの可操作度”，日本ロボット学会誌，vol. 2, no. 1, pp. 63–67, 1984．
- [8] D. Ferguson, N. Kalra, and A. Stentz, “Replanning with rrt”, *ICRA*, pp. 1243–1248, 2006.
- [9] S. Hochreiter, and J. Schmidhuber, “Long short-term memory”, *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [10] J. C. andÇağlar Gülçehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling”, *EMNLP*, pp. 1724–1734, 2014.
- [11] R. Wu, Y. Zhuang, K. Xu, H. Zhang, and B. Chen, “Pq-net: A generative part seq2seq network for 3d shapes”, *CVPR*, 2020.

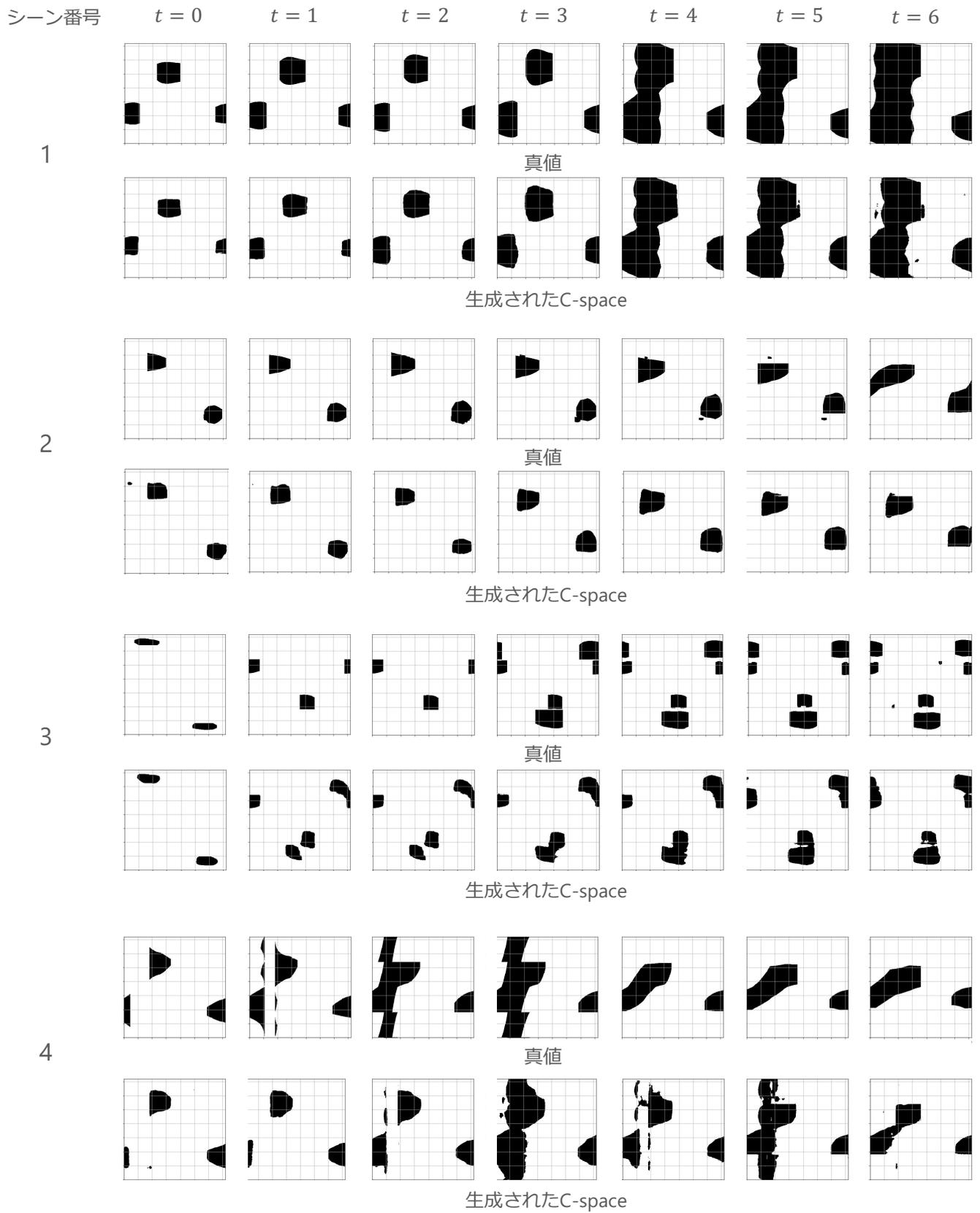


Fig.5 Generated future C-space.