

群集行動マルチエージェントシミュレーションの高速化

吉本達也¹ 宮本孝道² 石坂一久¹ 吉田孝志²

概要: 大規模イベントや災害時に伴う街中の歩行者や車の群集行動を予測し、異常混雑の防止のための誘導指示を可能とするマルチエージェントシミュレーションによる高精度な群集行動シミュレーションの高速化手法を提案する。提案手法は、道路上では複数の歩行者を同一の行動モデルでモデル化できるという特性を利用し、シミュレーションの計算量の削減およびプロセッサのベクトル命令の活用による高速化を行う。提案手法を2万人規模の群集行動シミュレーションに適用し、4.77倍の高速化が可能となることを実証した。これにより、従来1時間程度要していた群集行動シミュレーションを、10分程度のリアルタイム性が必要な誘導指示に適用できることを示した。

Acceleration of Crowd Behavior Multi-Agent Simulation

TATSUYA YOSHIMOTO¹ TAKAMICHI MIYAMOTO²
KAZUHISA ISHIZAKA¹ TAKASHI YOSHIDA²

1. 背景

大規模イベントや災害時に伴う街中の歩行者や車の群集行動を予測し、異常混雑の防止のための誘導指示に活用する群集行動シミュレーションの開発が進んでいる[1]。しかし、大規模な群集行動を複数のシナリオで詳細にシミュレーションする場合、全体で数時間程度の時間を要し、用途はオフラインでの事前シミュレーションに限られていた。これを数十分レベルに高速化すると、スタジアム等でのイベント中での警備員の配置計画に活用でき、また数秒程度に高速化すると、群集の行動に応じたリアルタイムなシミュレーションにより避難誘導などへ活用できる。

群集行動シミュレーションにおいては、シミュレーション対象となる群集の挙動をモデル化し、コンピュータ上で再現する。群集の挙動をモデル化する手法として Social Force モデル[2]、ネットワークモデル[3]、セルオートマトン法[4]など、さまざまな手法が存在する。その中において、群集行動予測に多く用いられるシミュレーション方法の一つとして、マルチエージェントシミュレーション (MAS) がある[5][6]。MAS とは、個々のエージェントに単純な行動モデルを定義し、エージェント間の相互作用の結果、全体の複雑な挙動を再現する高精度なシミュレーション方法である。しかし、個々のエージェントの行動結果に基づいて群集全体の挙動をシミュレーションするため、エージェントごとに行動モデルにしたがった状態の更新処理を行わなければならない。エージェント数に比例して計算時間が増大する。つまり、大規模な群衆行動シミュレーションをリアルタイムに実行するためには、高速化が課題となる。

シミュレーションを高速化するためには、シミュレーションモデルの特徴を活かした計算アルゴリズムの効率化だけでなく、計算機の活用も重要である。現在の計算機での高速化方法として、マルチコアによる並列化や、SIMD 化等のベクトル処理が一般的に活用される[7][8]。MAS では多数のシナリオに対してシミュレーションを行うため、非同期な並列化が可能なマルチコア並列化の利用は容易である。しかし、ベクトル処理はデータを同期的に並列処理するため、MAS では利用が進んでいない。したがって、更なる高速化のためにはベクトル処理の利用が重要である。本稿では、街中の道路を移動する歩行者などの移動体をエージェントとした MAS を対象として、エージェントの行動モデルの特徴を利用したベクトル処理方法を提案する。

本稿は以下のような構成となる。2章では、本稿において対象とする移動体をエージェントとした MAS のモデルの概要を述べる。3章では、MAS をベクトル処理によって高速化するための手法について述べる。4章では、提案手法による群集行動シミュレータの高速化効果について述べる。最後に、5章で本稿のまとめを述べる。

2. 移動体をエージェントとした MAS

本章では、本稿において対象とする移動体をエージェントとした MAS のモデルの概要について説明する。

移動体をエージェントとした典型的な MAS は、エージェント間の衝突判定と判定結果を考慮した位置の更新によって行われる。例えば、直進歩道においては、各エージェントは目的方向へ直進するが、移動速度や方向の違いによる衝突の可能性を判定し、直進だけでなく減速・追い越し・

1 日本電気株式会社 システムプラットフォーム研究所
NEC System Platform Laboratory

2 日本電気株式会社 データサイエンス研究所
NEC Data Science Laboratory

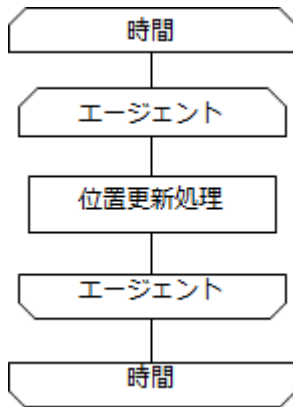


図1 従来のMASの全体フロー

滞留などの行動を選択する。

群集行動シミュレーションは、このような行動を選択する複数のエージェントが街中の歩道を移動する様子をシミュレーションし、混雑や移動時間を予測する。一般的に、MASによる群集行動シミュレーションは、事前に各エージェントの行動モデルやシミュレーション空間を定義し、シミュレーション実行時には、単位時間ごとに各エージェントの位置を計算する。そして、エージェントの位置の更新過程を表示することで、群集行動の予測情報を提供する。

図1において、従来のMASの全体処理フローを示す。全体フローとしては、時間経過に対するループと各時刻における全エージェントに対するループの2重ループが存在し、エージェントループ内で各エージェントの位置更新処理を行う。なお、位置の更新はエージェントの行動モデルに従って行われる。エージェントの行動モデルはエージェントが時間経過と共に移動する位置を決定するためのルールであり、シミュレーション対象となる移動体の行動特性が反映するよう定義される。以降では、本稿で対象とする従来のシミュレーション空間モデルおよびエージェントの行動モデルについて説明する。

2.1 シミュレーション空間モデル

街中の歩行者移動を想定する場合、シミュレーション空間は街中の歩道であり、2次元平面上で表現される。その場合、対象とするシミュレーション空間は複数の歩道区間に分割され、それらの連結構造として表現可能である。エージェントはある初期位置から区間内を一方向に直進し、いくつかの区間を経由することで、目的地へと到達する。

現実的には、歩行者はある一定の存在範囲を持ち、他の歩行者と範囲が重ならないよう一定距離を維持しながら移動する。本稿では、その様子をシミュレーション上で簡易的に表現する方法の一つとして、空間をエージェント1人が存在しうる一定幅（現実の50cm~1mに相当）で均等に分割し、分割された最小単位領域（セル）の中心点をエージェントが移動する方法を対象とする。

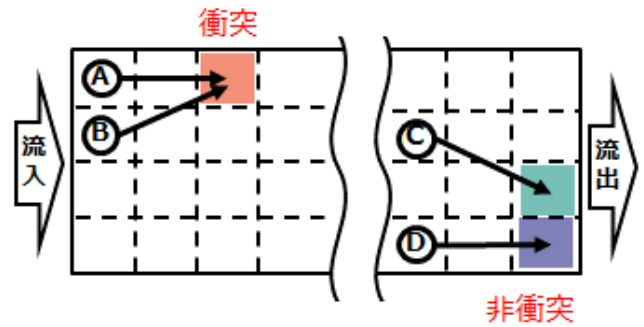


図2 シミュレーション空間モデル
(歩道区間のセル化とエージェントの衝突判定)

図2では、歩道区間のセル化とエージェントの衝突判定について例示している。区間はX軸・Y軸からなる2次元座標空間であり、一定幅のセルに分割される。各エージェントは区間の一端から流入し、単位時間経過ごとにセルの中心点上を移動し、もう一方の端部へと流出し、次の区間へ移る。また、各エージェントの衝突判定はセルの位置の重なりによって判定される。例えば、エージェントAは直進の結果、同一セル上にエージェントBが存在するため、衝突と判定される。一方で、エージェントCとDは異なるセル上に移動するため、衝突と判定されない。

2.2 エージェントの行動モデル

次に、本稿で想定するエージェントの行動モデルについて説明する。図3に、エージェントの行動モデルの実行フローを示す。行動モデルを構成する処理フローとして、(1) 移動方策の選択、(2) 位置の計算、(3) 衝突判定、(4) 移動方策の変更、(5) 位置の更新を挙げている。各フローは以下のような処理内容となる。

- (1) 移動方策の選択
直進・減速・追い越し・滞留などの移動方策の中から1つを選択する。
- (2) 位置計算
単位時間後のエージェントの移動先の位置を計算する。位置の計算はエージェントの現在位置、移動速度、移動方向などの情報に基づいて行われる。なお、(1)において選択された移動方策に応じて位置の計算方法は異なる。
- (3) 衝突判定
計算した移動先の位置に基づいて衝突判定を行う。衝突が発生する場合、(4)に移る。衝突が発生しない場合、(5)に移る。
- (4) 移動方策変更
衝突が発生する場合、移動方策を変更する。そして、新たに決定した移動方策に基づいて(2)の位置の計算フローに戻る。

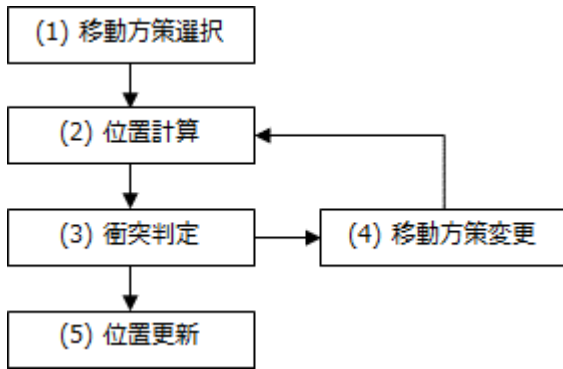


図3 エージェントの行動モデル

(5) 位置更新

衝突が発生しない場合、単位時間後の移動先の位置として決定し、エージェントの位置情報を更新する。

エージェントの行動モデルは群集行動シミュレーションによって目的とする情報（異常混雑、退避ルートの傾向など）を得るために、加味すべき歩行者の行動特性に応じて設計者が定義するものであり、決定的なものではない。しかし、エージェント間の衝突を考慮する場合、エージェントの位置計算後に衝突判定処理は必ず存在するため、前記フローで構成されるものと考えてよい。

なお、各エージェントは健常者・車いす移動者・高齢者のように歩行者としての特徴に応じて分類され、それぞれ同一速度で移動するものとする。また、同一速度・同一方向で移動するエージェントの集合をエージェントクラスと呼ぶ。

3. MASのベクトル処理による高速化

エージェントの衝突を判定するためには、移動先の位置に他のエージェントが存在するかどうか判定しなければならない。言い換えると、エージェントは互いに位置情報を参照し合い、衝突を回避するよう行動を選択し、移動する位置を決定する。つまり、エージェントは互いの位置情報に基づいて自分の移動先を決定するため、依存関係が存在する。このような依存関係のあるエージェントの位置計算に対して、データを同期的に並列処理するベクトル処理を効率的に利用することは難しい。

本章では、上記課題を解決する方法として、移動速度と移動方向が同じ（つまり、同一のエージェントクラスに属する）エージェント間では衝突が起こらないという行動モデルの特徴を利用したベクトル処理方法を提案する。

3.1 ヒストグラムを用いた単一の等速移動エージェント群の存在する区間の検出

街中の歩行者移動を想定した場合、歩行者間の衝突が発

		区間		
		1	2	...
クラス	1	5	0	...
	2	0	4	...
	3	0	2	...

図4 ヒストグラムのデータテーブル

生する状況として、移動速度が異なる歩行者が混在する状況（混成流）と、移動方向の異なる歩行者が混在する状況（対向流）が挙げられる。これらの状況においては、歩行者は前方に移動速度の遅い歩行者がいる場合や、前方に対向者が現れる可能性があり、衝突回避行動が必要となる。つまり、エージェント間で依存関係が存在し、ベクトル処理によって位置計算を行うことは難しい。一方で、区間内のエージェントの移動速度と移動方向が同一である（つまり、すべてのエージェントが同一クラスである）場合、すべてのエージェントが一様に同一方向へと一定速度で直進移動するため、衝突回避行動を必要としない。つまり、エージェント間の依存関係がないため、ベクトル処理を利用することが可能である。そこで、同一クラスのエージェントの集合である等速移動エージェント群が1つだけ存在する区間を検出し、その区間内のエージェントの位置更新処理にベクトル処理を利用する方法を提案する。

本提案手法では、単一の等速移動エージェント群が存在する区間を検出するために、区間内に存在するエージェント数をエージェントクラスごとにカウントし、記録するヒストグラムを作成する。ヒストグラムは図4のような配列データとして管理される。つまり、水平方向に区間情報、垂直方向にエージェントクラス（図中は最大3クラス存在する場合を例示）を持ち、各項目には対象区間内に存在する各クラスのエージェント数が記録される。なお、各項目の値はエージェントが区間へ流入・流出する際に更新される。

単一の等速移動エージェント群が存在する区間の検出は、ヒストグラムを参照し、対象区間の項目のうち、エージェント数が1以上であるクラスが1つのみであるかどうか調べることにより行われる。例えば、図4のヒストグラムの区間1の項を参照した場合、クラス1のエージェントは5人存在し、クラス2, 3のエージェントは共に0人である。よって、区間1には同一クラスのエージェントのみ存在しているため、単一の等速移動エージェント群が存在する区間として検出する。一方、区間2はクラス2のエージェントが4人、クラス3のエージェントが2人存在している。この場合、異なるクラスのエージェントが混在しているため、単一の等速移動エージェント群が存在する区間として検出されない。

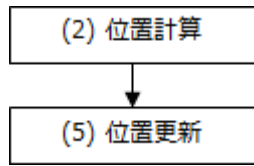


図5 等速移動エージェント群の行動モデル

3.2 等速移動エージェント群の位置計算の簡略化

単一の等速移動エージェント群が存在する区間が検出された場合、区間内のエージェントの位置計算は衝突判定を必要としないため簡略化される。つまり、等速移動エージェント群のすべてのエージェントは移動方策として直進移動をとり、衝突判定を必要とせず、同一の移動速度、方向で直進するものとして位置計算を行ってもよい。よって、図5のように、等速移動エージェント群の行動モデルは、図3の(1) 移動方策選択、(3) 衝突判定、および(4) 移動方策変更が実行されず、移動方策として直進移動を選択した場合の(2) 位置計算および(5) 位置更新のみ実行すればよい。以上から、行動モデルが簡略化され、位置更新処理にかかる計算量を削減することが可能となる。

3.3 ベクトル処理の利用

等速移動エージェント群における各エージェントはエージェント間で衝突が発生することなく一様に直進移動し、エージェント間の依存関係が存在しない。つまり、等速移動エージェント群における各エージェントの位置計算処理は、互いに干渉することなく図5の行動モデルに従って行われる。また、一様に直進移動するため、位置計算の計算方法が同一であり、異なるエージェントのデータ（現在位置）に対して同一の位置計算を行うことになる。よって、異なるデータに対して同一の計算を同期的に並列処理するベクトル処理を利用することが可能となる。

提案手法によるMASの全体フローを図6に示す。図1に示した従来フローと異なり、提案手法では各時刻において区間ごとにループ処理を行う。区間ループ内では、ヒストグラムを参照し、単一の等速移動エージェント群が存在するかどうか判定する。単一の等速移動エージェント群が存在する場合、その区間内に存在するエージェントに対して図5に示した等速移動エージェント群に対する行動モデルに従って位置更新処理を行い、かつ、ベクトル処理を利用可能となる。単一の等速移動エージェント群が存在しない区間においては、異なるクラスのエージェントが混在し、混成流や対向流が発生しているため、従来通りの図3の行動モデルに従った位置更新処理が行われる。また、エージェントの位置更新フローの終了後、エージェントの更新後の位置情報にしたがってヒストグラムを更新する。例えば、クラス1に属するエージェント1人がある区間Aから流出し、次の区間Bに流入した場合、ヒストグラムのデ

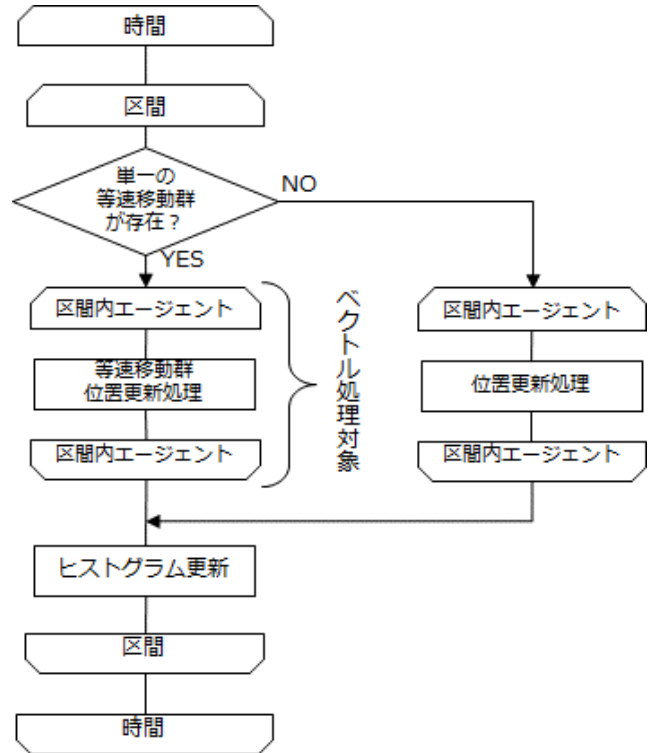


図6 提案手法におけるMASの全体フロー

ータテーブルの区間A・クラス1の項目から1減らし、区間B・クラス1の項目に1加えることで更新される。

4. 高速化効果の検証

本章では、大規模スタジアムから周辺駅まで2万人規模の歩行者が退場する様子をシミュレーションする群集行動シミュレーション[1]を例に用い、提案手法によって得られる高速化効果を検証する。つまり、図1の処理フローで構成されていた従来の群集行動シミュレーションを図6の処理フローに変更し、等速移動エージェント群の位置更新処理の計算量削減効果とベクトル処理による高速化効果を示す。ベクトル処理の評価環境としては、Intel Xeon E5-2650V2 (32コア) プロセッサ、Intel Parallel Studio XE 2017 Composer Edition for C++ Linux コンパイラを用い、AVX方式によるSIMD命令を利用した。なお、今回はベクトル処理による高速化効果の評価するために、マルチコアによる並列化は行わず、同一コア上での実装となる。

本検証では、各エージェントに係る位置更新処理の実行時間を単位時間毎に計測し、従来手法に対する提案手法の総実行時間の割合を算出した。図7では、従来の実行時間を100%とし、提案手法によって計算量削減およびベクトル処理を位置更新処理に利用した場合の実行時間の割合を示している。図7に示すように、等速移動エージェント群の位置更新処理は提案手法によって簡略化され、実行時

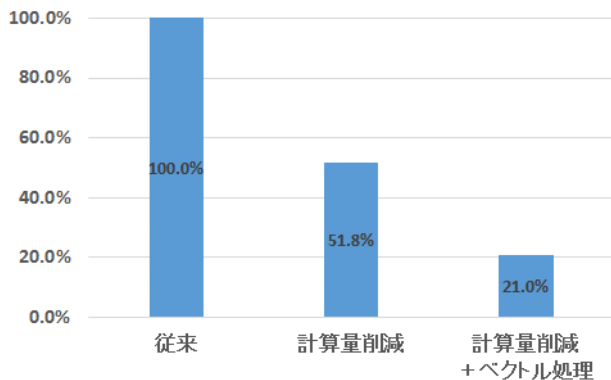


図7 提案手法による位置更新処理の高速化結果

間は51.8%まで削減されている。つまり、1.93倍の高速化結果となった。さらに、ベクトル処理を利用することにより、実行時間は51.8%から21.0%まで低減し、2.47倍の高速化結果となった。つまり、計算量削減とベクトル処理による高速化効果を合わせると、同一コア上で4.77倍の高速化結果となった。

以上より、提案手法を用いることで従来1時間程度要していたシミュレーションを10分程度まで短縮可能とした。その結果、従来は事前シミュレーションに限られていたが、スタジアムからの観客退場中での警備員配置計画作成など、リアルタイムでの用途が拡大する。

5. まとめ

本稿では、移動体をエージェントとするMASにおいて、等速移動エージェント群では衝突判定に伴うエージェント間の依存関係を見捨てることを利用し、位置計算の計算量の削減およびベクトル処理を利用し、MASを高速化する手法を提案した。さらに、2万人規模の群集行動シミュレーションに本提案手法を適用することで4.77倍の高速化が可能となることを示し、群集行動シミュレーションの用途拡大に貢献した。

本提案手法では、単一の等速移動エージェント群の存在する区間に対してのみ、エージェントの位置更新処理の計算量削減およびベクトル処理の利用を可能とした。しかし、区間内に異なる等速移動エージェント群が複数存在しても、位置関係に応じて衝突が発生しない場合もありうる。今後の課題として、そのような場合においてもベクトル処理を利用可能とする手法を提案することが挙げられる。

参考文献

- [1] 吉田孝志, 前野義晴: センシングとシミュレーションによる街なか移動支援, 計測自動制御学会 社会システム部会研究会 (2016).
- [2] Helbing, D., Farkas, I. J., and Vicsek, T.: Simulating dynamic features of escape panic, *Nature* 407(July), pp.487-490 (2000).

- [3] 山出真也, 白山晋: 複雑ネットワークを用いた新しい群集シミュレーション手法, 情報処理学会論文誌 数理モデル化と応用(TOM), Vol.1, No.1, pp.136-148 (2008).
- [4] Blue, V. J., and Adler, J. L.: Cellular automata microsimulation for modeling bi-directional pedestrian walkways, *Transportation Research Part B: Methodological*, Vol.35, Issue 3, pp.293-312 (2001).
- [5] Xiaoshan, B. et al: A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations, *AI & Society*, Vol.22, Issue 2, pp.113-132 (2007).
- [6] Masad, D. and Jacqueline K.: Mesa: An Agent-Based Modeling Framework, *Proc. The 14th PYTHON in Science Conference* (2015).
- [7] Patterson, D. A.: *Computer Architecture: a quantitative approach*, Elsevier (2011).
- [8] Aaby, B. G., Perumalla, K. S., and Seal, S. K.: Efficient Simulation of Agent-Based Models on Multi-GPU and Multi-Core Clusters, *Proc. The 3rd International ICST Conference on Simulation Tools and Techniques*, pp.29 (2010).