

千葉工業大学

修士学位論文

格子分割を用いた進行方向計算の削減による  
人流シミュレーションの高速化

令和5年3月

所属専攻 : 情報科学専攻

学生番号・氏名 : 2281011 番 片寄 颯人

指導教員 : 前川 仁孝 教授



## 修士論文要旨

専攻	学生番号	氏名
情報科学	2281011	片寄 颯人
論文題目 格子分割を用いた進行方向計算の削減による 人流シミュレーションの高速化		
キーワード マルチエージェントシミュレーション, 人流シミュレーション		
論文要旨 <p>本論文は, SFM(Social Force Model) を用いた人流シミュレーションを高速化するために, エージェントの進行方向の計算回数を削減する手法を提案する. SFM は, 時間ステップごとに各エージェントの運動方程式を解くことで, 人々の流れを解析する手法である. SFM の運動方程式は, 目的地に向かう力, 周囲のエージェントを避ける力, 障害物を避ける力の合力を用いてエージェントの移動を決定する. SFM を用いた人流シミュレーションは, 解析人数や壁などの障害物数が増えるほど, 解析時間が膨大になるため, 高速化が求められている. 人流シミュレーションのなかでも避難時の解析は, 机や壁などの障害物が多い傾向がある. 机や壁などの固定されている障害物や目的地は, 解析中に座標が変化しないという特徴があり, 目的地までに向かう力を計算するために必要なエージェントから目的地までのベクトルは, エージェントの座標に応じて決定するという特徴がある. そこで, 本論文では, 解析領域を格子状に分割し, 格子領域ごとに進行方向をあらかじめ計算することで, 解析中の進行方向の計算回数を削減する. 評価の結果, 提案手法は, 従来のセル分割法に対して, 許容できる誤差の範囲で解析時間が最大〇〇倍高速することを確認した. (508 文字)</p>		



## Summary of Master's Thesis

Course	Student No.	SURNAME, Firstname
Information and Computer Science	2281011	KATAYOSE Hayato
<b>Title</b>  Speedup of Pedestrian Simulation by Reduction of Direction Calculations using Grid Division		
<b>Keywords</b> Multi Agent Simulation, Pedestrian Simulation, Social Force Model		
<b>Summary</b>  <p>This paper proposes a speedup method by reducing the number of calculations of the agent's traveling direction for the Social Force Model(SFM). The SFM is a method to simulate human behavior based on motion equations. The motion equation of the SFM uses the resultant forces of the force to go to the destination, the force from surrounding agents, and the force avoiding the obstructions to determine the agent's movement. The SFM is required to speed up because analysis time increases as the number of agents and obstacles. fixed obstacles and destinations have characteristics that the coordinates do not change under analysis. Therefore, the proposed method reduces the number of calculations of the agent's traveling direction during analysis by dividing the analysis region into a grid and calculating the direction of travel for each grid region in advance. As a result of the evaluation, the proposed method gives us about [高速化率] times speedup within an acceptable error range.</p>		

# 目次

図一覧	iii
表一覧	v
第1章 はじめに	1
第2章 人流シミュレーション	3
2.1 ネットワークモデルの歩行者モデル**	4
2.2 静的フロアモデル	5
2.3 SocialForceModel(SFM)	6
2.3.1 周囲のエージェントから受ける力	9
2.3.2 周囲の壁から受ける力	11
2.3.3 経路地の設定	12
第3章 一般的な SFM の高速技法	17
3.1 モデルの簡易化	17
3.2 エージェント間の距離の計算回数削減	17
3.2.1 セル分割法	18
3.2.2 視野パラメータを用いた削減手法	19
3.3 単位時間あたりの計算回数	19
3.3.1 エージェントごとの並列性を用いた手法	19
3.3.2 解析領域ごとの並列性を用いた手法	22
3.4 経路選択時の判定回数削減	22
3.4.1 経路選択の単純化	22
3.4.2 経路選択手法の～～	22
3.5 abstract.tex	22
3.6 簡単コマンド	22
3.7 参考文献	23
3.7.1 bib ファイルを使う場合	24

3.7.2 bib ファイルを使わない場合 . . . . .	24
<b>第 4 章 エージェント間距離の削減手法</b>	<b>26</b>
4.1 パターン 1 (セル分割法) . . . . .	28
4.2 パターン 2 . . . . .	28
4.3 パターン 3 . . . . .	29
4.4 パターン 4 . . . . .	29
4.4.1 パターン 5 . . . . .	30
4.5 パターン 6 . . . . .	30
<b>第 5 章 格子分割による進行方向ベクトル計算の削減手法</b>	<b>31</b>
5.1 進行方向計算の前処理 . . . . .	31
5.2 壁を避ける力の前処理 . . . . .	31
<b>第 6 章 評価</b>	<b>36</b>
<b>第 7 章 おわりに</b>	<b>37</b>
<b>謝辞</b>	<b>38</b>
<b>参考文献</b>	<b>39</b>
<b>付 録 A プログラムの説明</b>	<b>40</b>
A.1 節番号のテスト . . . . .	40
A.1.1 項番号のテスト . . . . .	40
A.2 ページレイアウト表示 . . . . .	40

# 図 目 次

2-1	人流シミュレーションの活用例	3
2-2	ネットワークモデルの例	4
2-3	静的フロアフィールドモデルのイメージ	5
2-4	マンハッタン距離を用いた静的フロアフィールドモデルの例	5
2-5	マンハッタン距離を用いた静的フロアフィールドモデルの例	5
2-6	静的フロアフィールドモデルを用いたアーチ現象の再現	5
2-7	フロアフィールドモデルの例	6
2-8	二次元連続座標モデルの例	6
2-9	SFM を用いた人流シミュレーションのフローチャート	8
2-10	SFM における周囲のエージェントから受ける力の計算	11
2-11	他のエージェントから受ける力の範囲を限定するときの例	12
2-12	SFM における周囲の壁から受ける力の計算	13
2-13	SFM でスタック現象が起きる例	13
2-14	経路地を設定する SFM の例	14
2-15	経路地を設定する SFM の例	14
3-1	一次元モデルの例	18
3-2	アーチ現象の例	18
3-3	セル分割法を用いた例	19
3-4	SFM の並列化可能な処理	20
3-5	3 スレッドでの並列化の例	20
3-6	領域分割の例	21
3-7	適当な図を張ってみた	23
3-8	横並び (左)	23
3-9	横並び (右)	23
4-1	影響範囲の例	27
4-2	エージェント 4 の実装パターンごとの近似領域の例	27

4-3 進行方向ごとの近似領域 . . . . .	27
5-1 提案する格子分割の例 . . . . .	32
5-2 経由地がある場合の進行方向の例 . . . . .	32
5-3 提案手法の解析全体のフローチャート . . . . .	33
5-4 提案手法の前処理のフローチャート . . . . .	34
5-5 提案手法の運動方程式計算のフローチャート . . . . .	35

# 表 目 次

2-1 SFMのパラメータ . . . . .	9
4-1 進行方向を分類する条件 . . . . .	28



# 第1章

## はじめに

人が多く集まるイベントなどの場所では，想定よりも多くの人が集まることで，群集事故が発生する恐れがある．群集事故は，人が将棋倒しのように倒れる群衆雪崩や〇〇である．群衆事故を防止するには，急に狭くなる空間を作らないことや階段や出口を広くするなどが有効である（参考文献）．人の滞留や避難時間の予測に人流シミュレーションが用いられている．(?)，(?)，(?)，(?)．

人流シミュレーションは，コンピュータ上で人を運動方程式に基づくエージェントとして解析する手法である．人流シミュレーションのなかでも歩行者の動きの再現には，ネットワークモデルやフロアフィールドモデル，SocialForceModel(SFM)などが広く用いられている．ネットワークモデルは，〇〇である．ネットワークモデルは，災害時における都市部の避難シミュレーションなどの大域的な解析を高速に解析が可能であるが，群集事故の防止に必要な滞留などの解析ができないことが報告されている．このため，建物内などの滞留の解析には，フロアフィールドモデルやSFMがよく用いられている（参考文献）フロアフィールドモデルは，～～の手法である．フロアフィールドモデルを用いた避難シミュレーションは，一つの格子が人の大きさに制約されるため，出口付近で見られる滞留（アーチ現象）の解析精度が低いことが報告されている．高い解析精度が必要な場合は，解析領域を二次元の連続座標として解析するSFMがよく用いられている（参考文献）．SFMは，社会心理学的な要素と物理学的な要素で成り立つ運動方程式をエージェントごとに計算することで，人流の動きを再現する手法である．tion ‘sph’ on page 19 undefined on input line 64.SFMの運動方程式は，目的地に向かう力，周囲のエージェントを避ける力，障害物を避ける力の合力を算出し，エージェントの速度や進行方向を計算する．SFMの運動方程式の計算は，エージェントや障害物の数の増加に応じて解析時間が膨大になることから高速化が求められている．

一般的なSFMの高速化技法として，単位時間あたりの計算回数の増加や，モデルの単純化，エージェント間距離の計算回数の削減が行われている．SFMの単位時間あたりの計算回数の増加には，GPU(Graphics Processing Unit)やMPI(Message Passing Interface)を用いた並列処理を用いることが一般的である．SFMは，エー

エージェントごとの運動方程式の計算に並列性があるため、高い並列性を得ることができ（参考文献 [GPU や MPI を使っている論文]）。GPU を用いた並列化手法は、エージェントごとに必要な進行方向の計算を複数スレッドで並列に計算する。モデルの単純化は、計算負荷が高い SFM の運動方程式の計算を一次元に簡易化することで、解析時間を削減する手法である（参考文献 [一次元化モデル]）。SFM の一次元化手法は、許容できる範囲の誤差で避難完了時間を解析できるが、出口付近などの滞留の再現度が低いことが報告されている（参考文献）。エージェント間距離の計算回数の削減には、影響半径の設定や、セル分割法が提案されている（参考文献）。影響半径は、エージェント間の距離が大きくなるほど相互作用力が大きくなることを利用し、相互作用力が 0 に近似可能な距離を設定する。影響半径を設定することで、影響半径外のエージェントに対する相互作用力が計算不要となる。セル分割法は、解析領域を格子状のセルに分割し、周囲のエージェントに対する影響範囲内外の判定をセル単位で実行する手法である。影響範囲内外の判定には、エージェント間距離の計算が必要となるため複数のエージェントに対する判定処理をまとめて実行することで、エージェント間距離の計算回数を削減する。建物内の避難時におけるシミュレーションは、壁や机といった障害物が多く、障害物を避ける力の計算回数が多い傾向がある。障害物を避ける力は、エージェントの座標と障害物の座標を用いて計算する。壁や机などの障害物は、解析中に座標が変化しない特徴がある。そこで、本論文は、解析領域を格子に分割し、格子領域ごとにあらかじめ進行方向を計算し、メモリ領域に保存することで、解析中の進行方向の計算回数を削減する。

以下の章では、まず、ページフォーマットを示すために、第2章で「あああああ」を述べる。次に、第3章で、本スタイルファイルで定義したコマンドについて述べる。最後に、題7章でまとめる。

## 第2章

# 人流シミュレーション

人流シミュレーションは、コンピュータ上で人の動きを再現する手法であり、図 2-1 に人流シミュレーションの例を示す。図 2-1 中の青色の丸は右側に進む人、緑色の丸は左側に進む人、黄色の四角は壁、青色の四角は障害物である。赤色の障害物は、自動販売機やゴミ箱などの移動が可能である設置物である。図 2-1 の例では、通路が赤色の障害物によって通路が狭くなっているため、人の滞留や混雑が起きているため、赤色の障害物を撤去することで滞留や混雑を防ぐことができる。図 2-1 のような混雑や滞留を発見するためには、実際に多くの人で実験する必要があるため、時間や費用がかかる。一方で、人流シミュレーションは、コンピュータ上で再現できることから、実際に多くの人を用いて実験するよりも、必要な時間や金額を抑えることが可能である。このように、人流シミュレーションの目的は、人の滞留や混雑が起きないように対策することである。このため、人流シミュレーションは、大規模なイベントを企画する企業や大規模な施設を設計、建築する建設業などで活用されている（参考文献）。人流シミュレーションを活用することで、事前に人の流れを予測することが可能になり、地震や火災などの有事のときに、非常灯や看板の配置、警備員の配置などを最適化できるため、適切な誘導が可能になる。人流シミュレーションを用いて人の動きを解析するためには、人の動きを再現するための歩行者のモデル化（歩行者モデル）が必要である。歩行者モデルは、求められる解析精度や解析規模に応じて使い分ける必要があるため、ネットワークモデルやセルオート

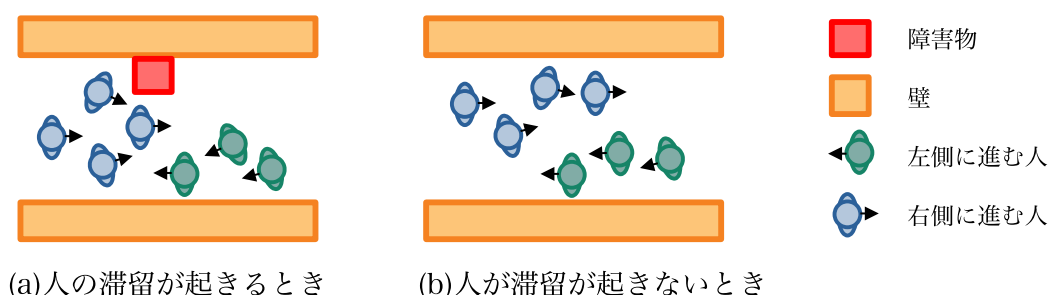


図 2-1：人流シミュレーションの活用例

マトン，SocialForceModel(SFM)などが提案されている．本章では，各歩行者モデルの使用用途や利点，欠点を用いて各手法の立ち位置について述べる．

## 2.1 ネットワークモデルの歩行者モデル\*\*

ネットワークモデルは，人々の移動や行動をネットワーク構造としてモデル化する手法である．図2-2にネットワークモデルの例を示す．図2-2中の(a)は解析対象であり，(b)はネットワークモデルである．図2-2中の(b)の青丸は交差点や道路の末端であり，ノードと呼ばれる．ネットワークモデルでは，ノード間に数十から数百人単位で動かすことで，人の動きを解析する．ネットワークモデルは，解析が高速であるが，モデルの準備に大きな作業が必要になるだけでなく，モデルの定義に知識や経験が必要になることが多い．ネットワークモデルを用いた解析では，都市間の人々の移動や，津波や地震などの災害時における都市の避難シミュレーションのような解析人数が多い解析に用いられることが多い（参考文献）ネットワークモデルで建物内の人の流れを解析する場合は，図??に示すように解析領域内にメッシュ状にノードを配置することで解析できる（参考文献）．一方で，建物内などの避難シミュレーションでは，人の流量が低下する原因や滞留の原因を調査することに用いられることが多いため，ネットワークモデルを用いた場合は，滞留などの再現ができない．このため，人の流量が低下する原因や滞留の原因を突き止めるために解析するときは，静的フロアモデルやSFMなどが用いられることが一般的である．

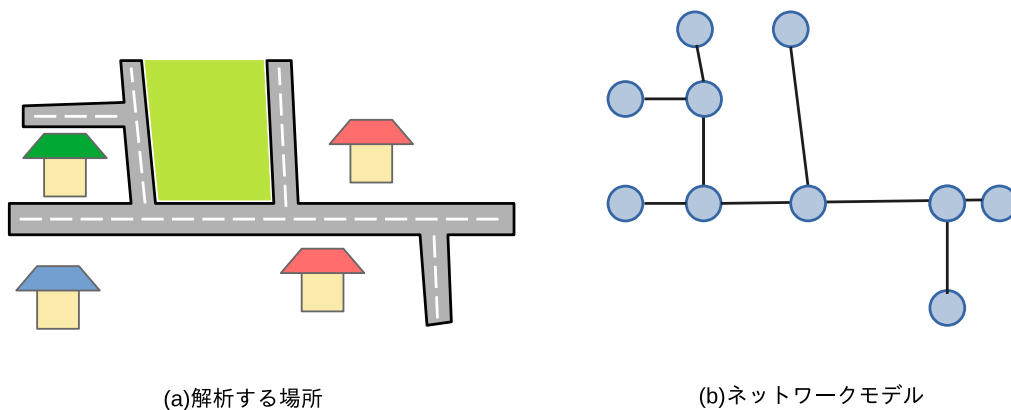


図 2-2 : ネットワークモデルの例

## 2.2 静的フロアモデル

静的フロアモデルは、図 2-7 に示すようなフロアフィールドモデルの空間モデルを用いており、格子ごとに目的地までの距離を設定し、確率を用いてエージェントを移動させることで解析する手法である。図??に静的フロアフィールドモデルのイメージを示す。図??中の格子は解析領域、青丸はエージェント、青色の矢印はエージェントの移動可能な方向である。図??のように、静的フロアフィールドモデルは、図??に室内からの退出時におけるマンハッタン距離を用いた静的フロアフィールドの例を示す。図??中の格子は、各セルを示しており、セル中の数字は各セルの出口までのマンハッタン距離を示す。

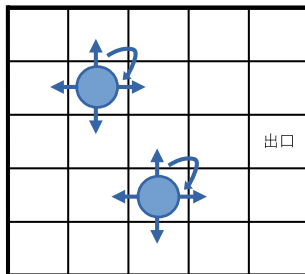


図 2-3 : 静的フロアフィールドモデルのイメージ

6	5	4	3	2
5	4	3	2	1
4	3	2	1	出口 0
5	4	3	2	1
6	5	4	3	2

図 2-4 : マンハッタン距離を用いた静的フロアフィールドモデルの例

6	5	4	3	2
5	4	3	2	1
4	3	2	1	出口 0
5	4	3	2	1
6	5	4	3	2

図 2-5 : マンハッタン距離を用いた静的フロアフィールドモデルの例

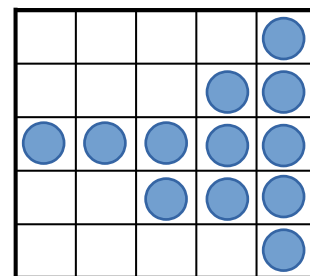


図 2-6 : 静的フロアフィールドモデルを用いたアーチ現象の再現

静的フロアフィールドモデルは、図??に示すように、計算対象のエージェントの周囲のセルのなかから、出口までの距離が小さくなるようなセルを選択することで、出口までの解析が可能となる。静的フロアフィールドモデルの利点は、解析前に各格子の計算を事前にできるため、非常に高速な解析が可能である点である。一方で、

静的フロアフィールドは、出口前に形成されるアーチ現象の再現度が低いことが知られている。図2-7に静的フロアフィールドモデルを用いた場合の出口前に形成されるアーチ現象の例を示す。図2-7中の～～～である。静的フロアフィールドモデルは、図2-7のように、格子に一人のみ入ることができることから、動きが格子サイズに制約されるため、出口付近の再現度が低い。フロアフィールドモデルを用いた解析では、〇〇や、を用いることで、解析精度の向上が行われているが、格子サイズの成約から、精度の向上に上限がある。このため、高い解析精度が必要な場合は、SocialForceModel(SFM)のような解析領域を連続座標で解析する手法が用いられることが多い。

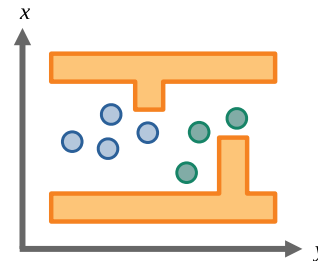
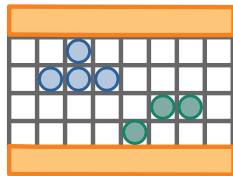


図 2-7 : フロアフィールドモデルの例      図 2-8 : 二次元連続座標モデルの例

## 2.3 SocialForceModel(SFM)

SFM は、人間の社会心理学的な要素と物理的な力を結びつけた動力学モデルであり、近傍のエージェントや壁といった障害物から受ける力によってエージェントの進行方向や速度を解く。本手法は、心理的変数が組み込まれているため、災害時の避難シミュレーションによく用いられる<sup>(?)</sup>。

SFM の解析空間は、二次元連続空間モデルが用いられる。二次元連続空間モデルは、解析領域を分割せずに  $(x, y)$  の連続した座標で解析する手法である。図 2-8 に図 2-7 の例を二次元連続座標で考えた例を示す。図 2-8 中の矢印は座標の  $x$  と  $y$  を示している。SFM は、図 2-7 のフロアフィールドモデルのように格子上のエージェント数に制限がなく、エージェントの位置を座標で考えるため、人流の再現度が高い。SFM におけるエージェント移動は、目的地へ進む力と他のエージェントから受ける力、壁などの障害物から受ける力を用いる運動方程式を用いて求める。式 (2-1) に

SFM の運動方程式を示す．

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0(t)e_i^0(t) - v_i(t)}{t_i} + \sum_{j(\neq i)} f_{ij} + \sum_W f_{iW} \quad (2-1)$$

式 (2-1) 中の総和の記号  $\sum_{j(\neq i)} f_{ij}$  は，エージェント  $i$  以外のすべてのエージェント  $j$  の総和をとることを意味する．同様に， $\sum_W f_{iW}$  は，すべての壁  $W$  の総和をとることを意味する．式 (2-1) 中の  $m_i$  はエージェント  $i$  の体重， $v_i^0(t)$  はエージェントの希望速度， $e_i^0(t)$  は，目的地までの単位ベクトル， $v_i(t)$  は現在の速度ベクトル， $t_i$  は時定数である．式 (2-1) の第一項はエージェントが目的地へ進む力，第二項は他のエージェントから受ける力  $f_{ij}$ ，第三項は壁などの障害物から受ける力  $f_{iW}$  の合力である． $f_{ij}$  と  $f_{iW}$  は，式 (2-2) と式 (2-3) を用いて導出する．

$$f_{ij} = \{A_i \exp[\frac{r_{ij} - d_{ij}}{B_i}] + kg(r_{ij} - d_{ij})\}n_{ij} + \kappa g(r_{ij} - d_{ij})\Delta v_{ij}^t t_{ij} \quad (2-2)$$

$$f_{iW} = \{A_i \exp[\frac{r_i - d_{iW}}{B_i}] + kg(r_i - d_{iW})\}n_{iW} + \kappa g(r_i - d_{iW})(v_i t_{iW})t_{iW} \quad (2-3)$$

表 2-1 に式 (2-2)，(2-3) 中の変数を示す．衝突時関数  $g(x)$  はエージェント同士や壁などに衝突したときに値をとる関数である．式 (2-4) に衝突時関数  $g(x)$  の条件式を示す．

$$g(x) = \begin{cases} 1 & (x < 0) \\ 0 & otherwise \end{cases} \quad (2-4)$$

SFM の衝突時の計算は，条件式である式 (2-4) を用いることで，衝突時のみ計算できる．SFM を用いる人流シミュレーションのフローチャートを図 2-12 に示す．図 2-12 中の目的地へ進む力の計算は，式 (2-1) 中の第一項を用いて算出する．また，他のエージェントから受ける力の計算は，式 (2-2) を用いて算出する．そして，壁などの障害物から受ける力の計算は，式 (2-3) を用いて算出する．SFM を用いた人流シミュレーションは，図 2-12 に示すように，式 (2-1) の運動方程式を積分することで，新しい時間のエージェントの位置と速度を求めることができる．



図 2-9 : SFM を用いた人流シミュレーションのフローチャート



表 2-1 : SFM のパラメータ

$d_{ij}$	エージェント間の距離
$t_{ij}$	エージェント $i$ とエージェント $j$ の衝突面の垂直ベクトル
$n_{ij}$	エージェント $i$ とエージェント $j$ の衝突面の法線ベクトル
$r_i$	エージェント $i$ の体の半径
$r_{ij}$	エージェント $i$ とエージェント $j$ の体の半径の和
$t_{iW}$	エージェント $i$ と壁 $W$ の衝突面の垂直ベクトル
$n_{iW}$	エージェント $i$ とエージェント $W$ の衝突面の法線ベクトル
$A_i$	エージェント $i$ のインタラクション作用
$B_i$	エージェント $i$ の反発作用
$k$	衝突時の反発力係数
$\kappa$	衝突時の摩擦係数
$\Delta v_{ij}$	エージェント $i$ とエージェント $j$ の接線速度の差
$g(x)$	衝突時間関数

### 2.3.1 周囲のエージェントから受ける力

他のエージェントから受ける力は、解析領域全体に存在する他のエージェントから受ける。このため、SFM は、解析する人数が増えると他のエージェントから受ける力の計算時間が長くなる。他のエージェントから受ける力の計算負荷を削減するために、SFM を用いる人流シミュレーションでは、他のエージェントから受ける力を計算する範囲を限定することが多い<sup>(?)</sup><sup>(?)</sup>。他のエージェントから受ける力を計算する範囲を限定することで、遠くに存在するエージェントから受ける力を 0 に近似することができる。図 4-1 に他のエージェントから受ける力の計算範囲の例を示す。図 4-1 の赤丸は他のエージェントから受ける力を計算するエージェント、黒丸はエージェント 4 が計算するときの他のエージェント、オレンジ色の点線は他のエージェントから受ける力の範囲を示す。図 4-1 のエージェント 4 は、オレンジ色の点線内に存在するエージェント 0, 1, 2, 6, 7, 10 の合計 5 人から力を受ける。本論文では、近くのエージェントから受ける力の範囲を限定する SFM を前提として述べる。近くのエージェントから受ける力の範囲は、図 4-1 のように、計算するエージェントの半径数メートルの範囲である。このため、SFM では、他のエージェントが近くのエージェントから受ける力の範囲に存在するか判定が必要である。この範囲に存在するかの判定は、エージェント  $i$  とエージェント  $j$  とのエージェント間の距離  $d_{ij}$

の算出が必要である．本論文では，式(2-5)を用いてエージェント間の距離  $d_{ij}$  を求め，他のエージェントから受ける力の範囲内であるかどうか判定する．

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2-5)$$

式(2-5)中の  $x_i$  と  $y_i$  はエージェント  $i$  の座標  $(x_i, y_i)$ ， $x_j$  と  $y_j$  はエージェント  $j$  の座標  $(x_j, y_j)$  である．エージェント  $i$  は，式(2-5)で求めたエージェント距離  $d_{ij}$  が他のエージェントから受ける力の範囲内であれば，エージェント  $j$  から式(2-2)を用いて算出した力を受ける．他のエージェントから受ける力の範囲の半径を  $R$  としたとき，エージェント  $i$  の他のエージェント  $j$  が範囲内にいるかどうかの判定式を式(2-6)に示す．

$$R \geq d_{ij} \quad (2-6)$$

エージェント  $i$  は，式(2-6)の条件を満たす他のエージェント  $j$  から式(2-2)で求める力を受ける．

他のエージェントから受ける力の範囲を限定する SFM のフローチャートを図??に示す．図??のフローチャートでは，各エージェントに対しエージェント間の距離を計算し，範囲内であるか判定することで，他のエージェントから受ける力の範囲を限定している．SFM を用いる人流シミュレーションは，他のエージェントから受ける力の範囲を視野の範囲にすることで，視野を再現できる．

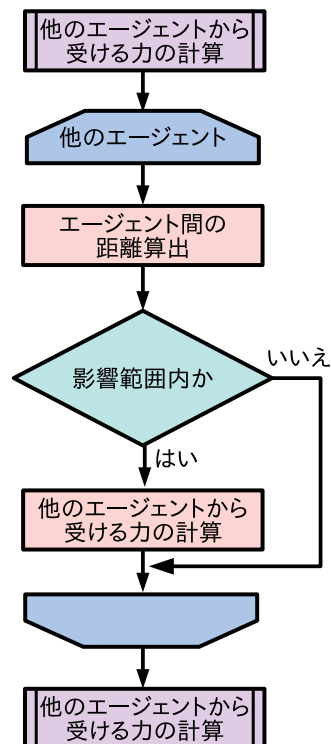


図 2-10 : SFM における周囲のエージェントから受ける力の計算

### 2.3.2 周囲の壁から受ける力

周囲の壁から受ける力は、エージェントの周囲の障害物を避けるために受ける力である。SFM を用いた人流シミュレーションは、壁や机などの障害物を粒子として計算することが一般的である（参考文献）。図??に壁を粒子化した例を示す。図??中の 色の丸はエージェント、 色の丸は壁粒子である。図??のように、～～である。

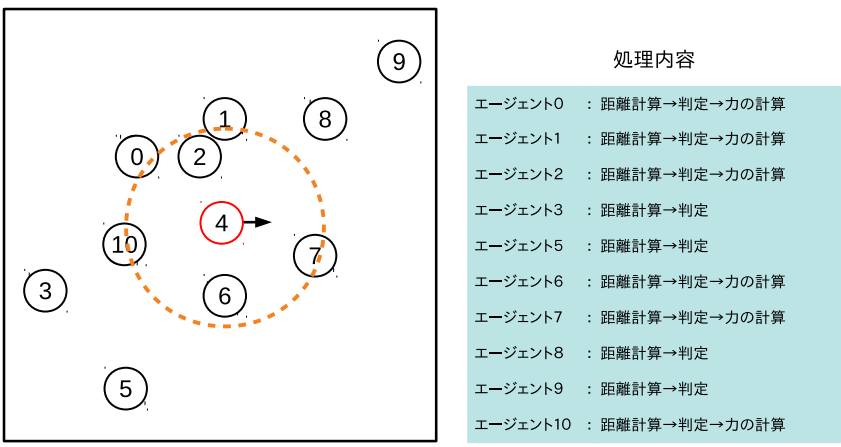


図 2-11 : 他のエージェントから受ける力の範囲を限定するときの例

2.3.3 経由地の設定

SFM は、エージェントと目的地の間に障害物が存在するとスタック現象が発生することが報告されている（参考文献）。スタック現象は、エージェントが動かなくなる現象のことであり、障害物から受ける力と目的地に向かう力の関係により生じる。図 2-13 に SFM でスタック現象が生じる例を示す。図 2-13 中の緑色の丸はエージェント、矢印はエージェントの進行方向、オレンジ色の四角は机などの障害物である。図 2-13 中のエージェント A とエージェント B は解析領域右上の出口に向かうため、エージェント A が机などの障害物に向かって進む。この場合は、エージェント A が机に向かって進み続けることや机の上を歩くなどの想定しない動きをすることがある。このため、障害物が多く存在するような解析では、出口 (目的地) だけでなく、目的地までの道のりを示す経由地を設定することで、エージェントのスタック現象や想定しない動きを防ぐことができる。図 2-14 に経由地を設定する例を示す。図 2-14 中の緑色の丸はエージェント、四角は障害物、青色の四角は経由地、赤色の四角は目的地を示す。図 2-14 の例では、エージェント A は、経由地を通ったあとに目的地に進むため、図 2-13 のように机に進むことが防げる。教室などの障害物多い解析では、図 2-15 に示すように、複数の経由地を設定する必要がある。図 2-15 の例では、エージェントは一番近くの経由地から目的地までの道のりを辿る。目的地までの道のりを決定する手法は、ダイクストラ法などのグラフ理論で用いられる手法が使われることが多い（参考文献）。ダイクストラ法は、

ああああああああああああああああああああああああああああああああああ

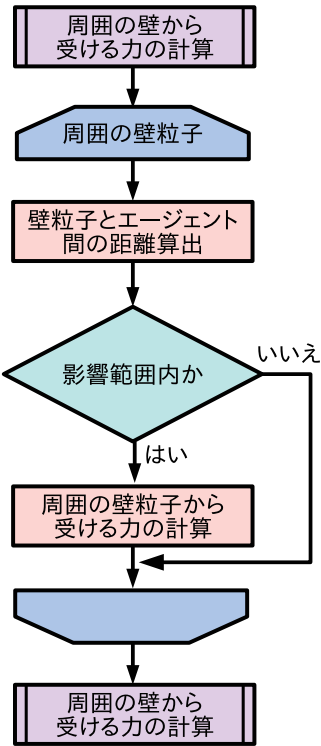


図 2-12 : SFM における周囲の壁から受ける力の計算

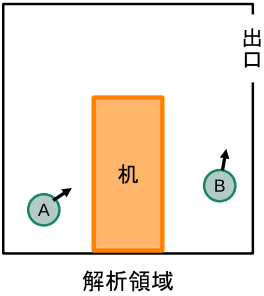


図 2-13 : SFM でスタック現象が起きる例

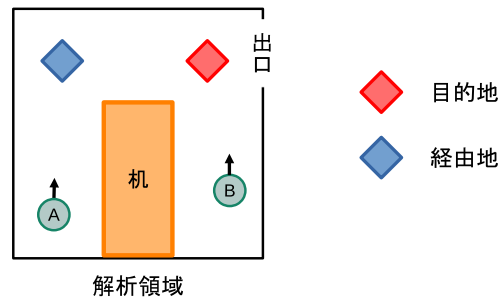


図 2-14 : 経由地を設定する SFM の例

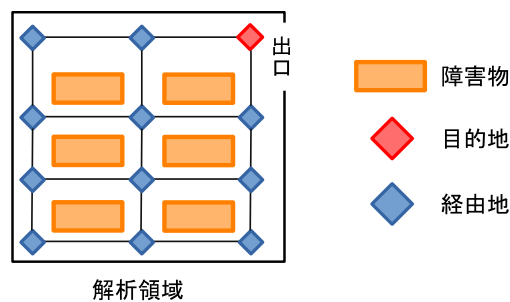


図 2-15 : 経由地を設定する SFM の例

[illegible]

[illegible]

[illegible]



## 第3章

# 一般的なSFMの高速技法

SFMを用いた人流シミュレーションは、解析人数が多くなるほど計算負荷が膨大になるため、解析に時間がかかる。SFMの解析時間を削減するために、モデルの単純化（参考文献）やエージェント間距離の計算回数削減手法（参考文献）、単位時間あたりの計算回数の増加手法（参考文献）、経路選択時の判定回数の削減手法（参考文献）などが提案されている。本章では、SFMの各高速化手法について述べる。

### 3.1 モデルの簡易化

SFMの簡易化手法は、SFMの計算負荷を削減するために、エージェント同士や壁や机などの障害物から受ける力、進行方向を単純化する手法である。SFMの簡易化手法の一つに一次元歩行者モデルがある（参考文献）。一次元歩行者モデルは、エージェントの動きを  $x$  や  $y$  のみにする手法である。図 3-1 に避難シミュレーション時の SFM と一次元歩行者モデルの例を示す。図 3-1 中の (a) は SFM などの二次元連続空間モデルを示し、(b) は一次元連続歩行者モデルの例を示す。一次元歩行者モデルは、図 3-1 のように、エージェントの動きを算出する式を一次元に変更することで、計算負荷を削減できる。本手法は、人の流れ（流量）を解析する場合では、高速かつ許容できる誤差の範囲で解析できることが報告されている（参考文献）。一方で、一次元でエージェントの動きを再現するため、人の押し合いや図 3-6 のようなアーチ現象などを再現できない。このため、人の押し合いやアーチ現象を再現したい場合には、一次元歩行者モデルなどのモデルを簡易化しない高速化技法を用いることが望ましい。

### 3.2 エージェント間の距離の計算回数削減

SFM は、解析人数が増加するほど周囲のエージェントから受ける力の計算に必要な周囲のエージェントが影響範囲内外かの判定の回数が増加する。周囲のエージェントが影響範囲内外かの判定は、ルートなどを用いてエージェント間距離  $d_{ij}$  を求め

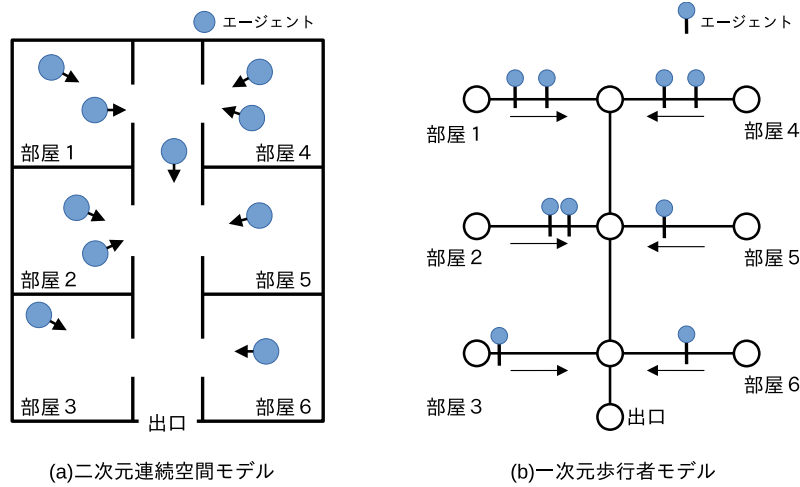


図 3-1 : 一次元モデルの例

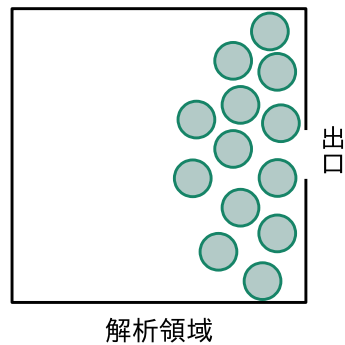


図 3-2 : アーチ現象の例

必要があるため、特にエージェント間距離  $d_{ij}$  の計算に時間がかかる．このため、SFM を用いた人流シミュレーションの解析時間を削減するためには、エージェント間距離  $d_{ij}$  の計算回数を削減することが有効である．エージェント間距離の計算回数削減手法にセル分割法や視野パラメータを用いた削減手法がある．

### 3.2.1 セル分割法

セル分割法は、水や空気などの動きを解析できる MPS 法<sup>(?)</sup> や銀河系などの圧縮性流体に用いられる SPH 法<sup>(?)</sup> などの粒子法によく用いられている．粒子法は、近傍の粒子と相互作用する力を計算し、粒子の行動を決定する．このため、セル分割法は、粒子法と同じように近傍のエージェントとの相互作用力を計算する SFM に対

しても用いられる。本手法は、解析領域を格子上のセルに分割し、計算するエージェントの存在するセルと近傍のセルに存在するエージェントに対して他のエージェントから受ける力の範囲であるか判定し、範囲内であれば他のエージェントから受ける力を計算する手法である。図 3-3 にセル分割法を用いる SFM の例を示す。図 3-3 中の赤丸は他のエージェントから受ける力の計算をするエージェント、黒丸は他のエージェント、四角は解析領域を格子状に分割したセルである。この例では、エージェント 4 の行動を更新する際に青色のセル内に存在するエージェントのみを参照するため、エージェント番号 3, 5, 9 の計算を削減できる。このとき、視野を用いる SFM は、速度計算するエージェントの進行方向前方に存在するエージェント情報を用いて計算するため、エージェントの進行方向後方のセルに存在するエージェント情報は不要になる、このため、視野を用いる SFM では、視野を考慮し、参照するセルを視野範囲に近づけることで、計算回数を減らすことができる。

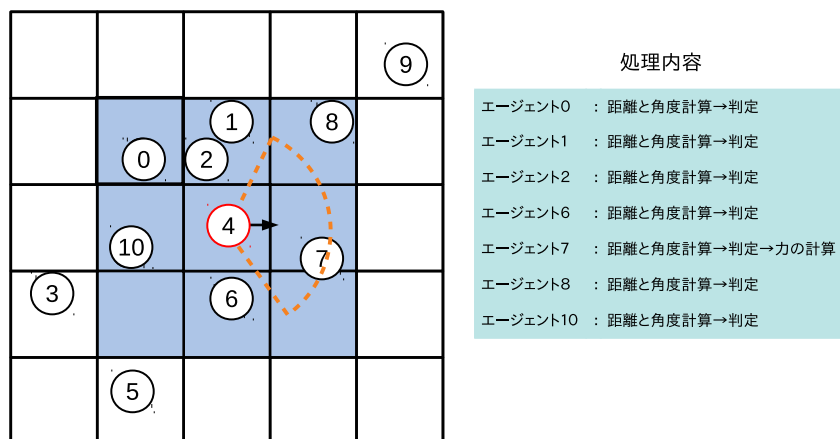


図 3-3 : セル分割法を用いた例

### 3.2.2 視野パラメータを用いた削減手法

視野パラメータを用いた SFM では、

## 3.3 単位時間あたりの計算回数

### 3.3.1 エージェントごとの並列性を用いた手法

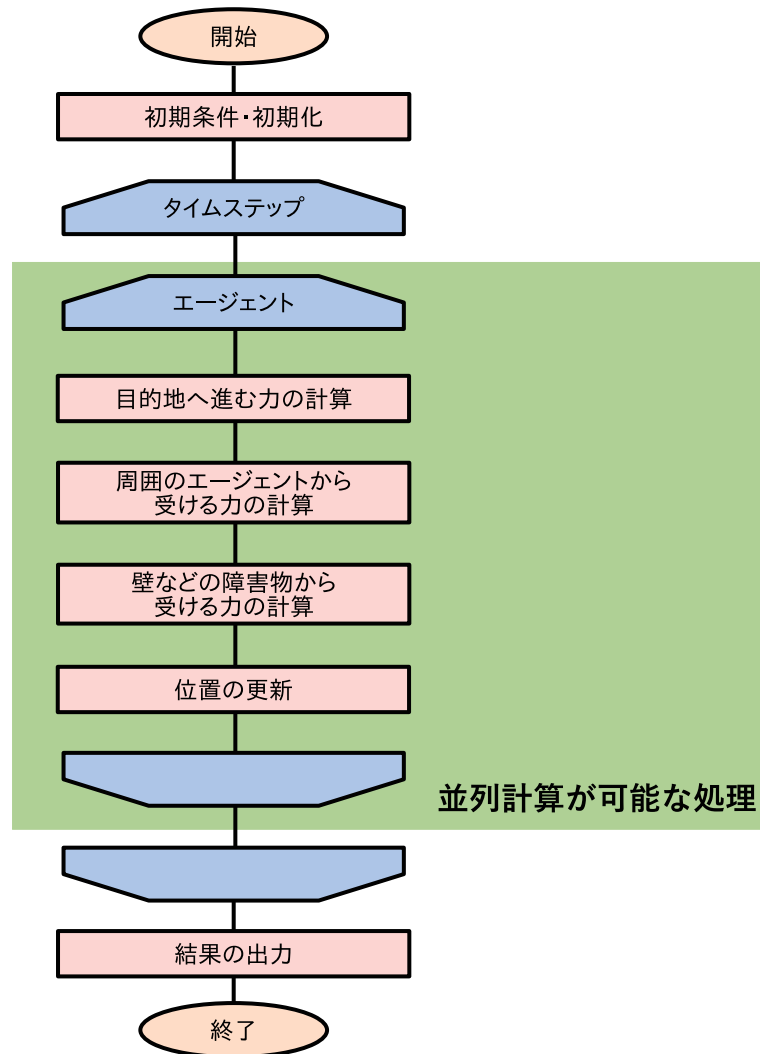


図 3-4 : SFM の並列化可能な処理

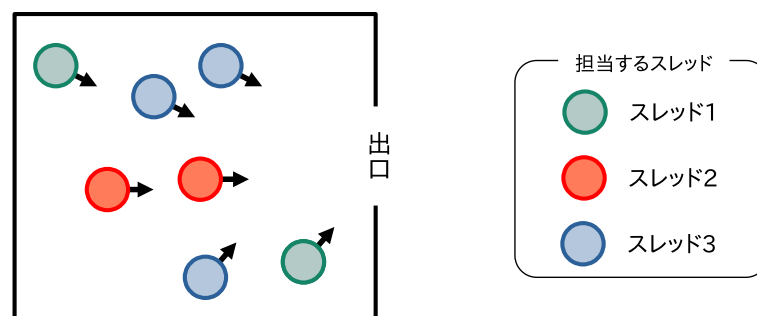


図 3-5 : 3 スレッドでの並列化の例

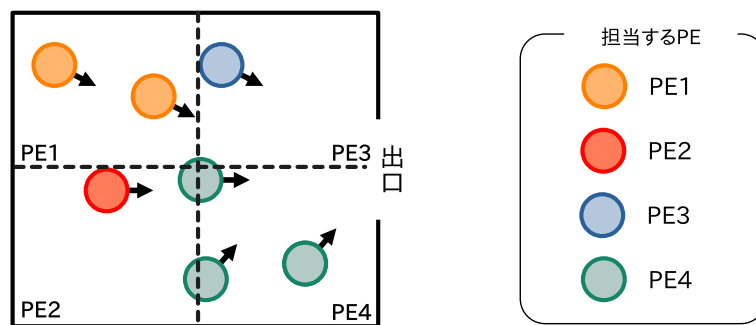


図 3-6 : 領域分割の例

### 3.3.2 解析領域ごとの並列性を用いた手法

## 3.4 経路選択時の判定回数削減

### 3.4.1 経路選択の単純化

### 3.4.2 経路選択手法の ～ ～

スタイルファイルの使い方について少し述べます．普通に利用する分には，拡張子が.tex ファイルのみを編集するだけで事が足りるように設計しました．ただし，拡張子が.sty ファイルの書き換えは制限しません．自由に改変してください．

## 3.5 abstract.tex

abstract.tex を書き換えると表紙およびアブストラクトを生成します．abstract.tex 内のコメントにしたがって書き換えを行ってください．卒論にはアブストラクトが不要です．修士のみアブストラクトを作成してください．

また，アブストラクトの設定は shuronABS.sty に書いてあります．表紙の設定は penguin.sty に書いてあります．困ったときはこれらのファイルを変更してください．

## 3.6 簡単コマンド

penguin.sty の 294 行目以降には，ショートカットコマンドを記述しました．気が向いたら使ってやってください．あくまでショートカットコマンドなので，penguin.sty のコマンドを使わなくても同じ機能を実現することができます．

- `\owata`
- `\ol{ 数式 }`
- `\fig{ タイトル }{ ファイル名 }{ 図の横幅 [cm] }`
- `\doublefig{ タイトル 1 }{ ファイル名 1 }{ 図の横幅 1[cm] }{ 図と図の間隔 [cm] }{ title_2 }{ file_name2 }{ size_2[cm] }`
- `\figref{ fig: ラベル }`

- `\tabref{ tb: ラベル }`

図 3-7 に、`\fig` コマンドを用いて図を貼る例を示します。図 3-7 は、`\fig{ 適当な図を張ってみた }{ulysses16}{5}` で貼り付けています。図 3-7 では、図の横幅が 5cm になるように大きさ指定をしています。

また、図 3-8 と図 3-9 は、`\doublefig` コマンドを用いて図を並べた例です。これらの図は、`\doublefig{ 横並び (左) }{test1}{2.5}{0.5}{ 横並び (右) }{test2}{2.5}` で貼り付けています。`\doublefig` コマンドは、図のタイトル高さを自動調節する機能を持っていません。このため、タイトルの高さは手動で調節してください。

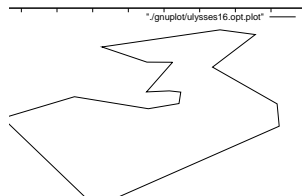


図 3-7 : 適当な図を張ってみた



図 3-8 : 横並び (左)



図 3-9 : 横並び (右)

図を入れる時には、段落と段落の間に入れてください。決して文の途中に図が入ることがあってはいけません。もし、図を参照しているページと図のページが離れてしまった場合は、段落の長さが適切でない可能性があります。フォーマットを変えるのではなく、本文の構成を見直しましょう。

### 3.7 参考文献

参考文献を参照する文の例です<sup>(1)</sup>。参考文献の書き方には、`bib` ファイルを使う方法と使わない方法の 2 通りがあります。好きな方を選択し、`makefile` と `main.tex`

を書き換えてください。

### 3.7.1 bib ファイルを使う場合

main.tex と makefile の書き換えは必要ありません。bibfile.bib に参考文献の記述例があります。cinni や IEEE などでは文献の bibtex 情報が用意されているので、そのファイルをコピーして使えるのが強みです。また、本方式を使うと、人力で参考文献情報をソートする必要が無いのでありがたいです。ただし、参考文献が1つも参照されていないとエラーが生じる模様です<sup>(2)</sup>。

以下に FAQ を載せておきます。

- bib ファイルって何？bibtexって何？  
使い方は google 先生に聞いてください。
- 参考文献情報を書き換えてもコンパイル結果に反映されない  
main.bbl ファイルを消去してから再コンパイルしてください。
- 参考文献スタイルを変更したい  
参考文献のフォーマットを決めるファイルは、sty/ipsjunsrt.bst です。本ファイルは、情報処理学会のスタイルファイルです。
- 名字が1文字の人の表示がおかしい  
情報処理学会フォーマットの仕様です。論文提出直前に bbl ファイルを直接編集してください。
- bib ファイルでエラーが出る  
大抵の場合はカンマ忘れが原因です。次点で参照タグ名の重複かな？

### 3.7.2 bib ファイルを使わない場合

自力で thebibliography の中身を書くパターンです。bibfile.tex に記述例があります。記述した通りに表示されるため、直感的には分かりやすいです。ただし、人力での作業量が多くなるので、この方式を使う場合は頑張ってください。

本方式を用いる場合は、以下のファイルの書き換えが必要です。

- main.tex 61行目 (`\bibliography{bibfile}`) をコメントアウトし、62行目 (`\input{bibfile}`) のコメントアウトをはずしてください



- `makefile` # 記号でコメントアウトしてください
- `bibfile.tex` ここに参考文献を書いてください．参考文献は，本文中での参照順番に手動で並び替えが必要です．

## 第4章

# エージェント間距離の削減手法

SFM の運動方程式では，エージェント間距離が大きいほどエージェント間に働く相互作用力が小さくなる．このため，SFM では，影響半径  $R_c$  を用いてエージェント間距離の計算回数を削減するのが一般的である．図 4-1 に，エージェント 4 の影響半径の例を示す．図 4-1 中の○はエージェントであり，オレンジ色の点線で囲まれた領域がエージェント 4 の影響範囲である．影響半径  $R_c$  は影響範囲の半径であり，影響範囲外から受ける力を 0 とすることで，運動方程式を計算する際に演算を省略することができる．一方，視野を用いた SFM では，視野範囲外の相互作用力を 0 とする．図 4-1 の例で，エージェント 4 が図中矢印の方向に移動する際には，エージェント 4 の視野は，図中の緑色の領域のように設定され，視野  $\subseteq$  影響範囲のような関係となる．

SFM で用いられるエージェント間距離の計算回数削減手法は，影響範囲が円形であることを前提としており，影響範囲が視野のように扇形の場合を想定していない．例えば，図 4-1 のエージェント配置では，半径  $R_c$  内の白い領域に存在する 5 つのエージェントが視野外にあるにも関わらず，これらのエージェントに対するエージェント間距離の計算が必要となる．人の視野角  $\theta$  は  $\pi$  以下であるため，視野を用いた SFM に一般的な SFM のエージェント間距離の計算回数削減手法を用いると，影響範囲を円形に絞り込みをした上で，扇形に絞り込みを行う操作が必要となる．このため，視野範囲外となる半径  $R_c$  内のエージェントが多く存在するほど，相互作用力を 0 として計算するエージェントに対するエージェント間距離の計算回数が増える．そこで，本論文では，視野形状である扇形に近似した領域を設定し，近似領域外のエージェントに対するエージェント間距離の計算回数を削減することで，視野を用いた SFM を高速化する．

提案手法では，近似領域の算出に必要な計算時間を最小限に抑えるために，長方形で視野範囲を近似する．これに合わせて，視野範囲はエージェントの進行方向前方に存在するため，エージェントの進行方向を長方形の辺数に合わせて上下左右の 4 パターンに分類し，エージェントの進行方向に応じた近似領域を設定する．エージェントの進行方向の分類方法によりシミュレーション時間が変化する可能性があ

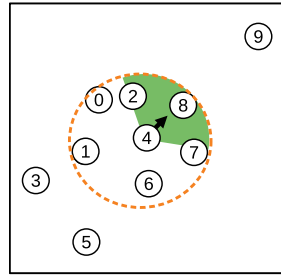


図 4-1 : 影響範囲の例

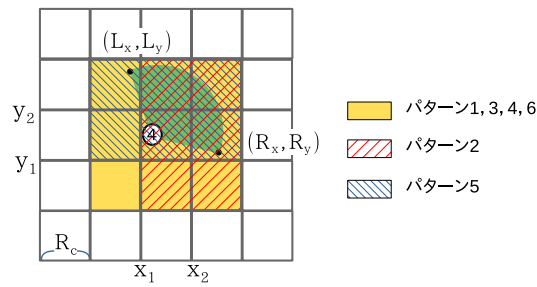


図 4-2 : エージェント 4 の実装パターンごとの近似領域の例

るため，本論文では，既存手法であるセル分割法を含む 6 パターン実装し，その有効性を評価する．提案手法であるパターン 2～6 は，いずれの手法もパターン 1 のセル分割法の考え方を基にしており，時間ステップごとに影響範囲の近似領域を設定した上で，近似領域内で影響範囲内となるエージェントを相互作用力を計算する対象として設定する．表 4-1 に，エージェントの進行方向を分類する条件を示す．ただし，表 4-1 で条件に数式が 2 つ記述されているものは，両条件を満たす必要があることを表す．また，図 4-2 に，図 4-1 のエージェント 4 に対して表 4-1 の各実装パターンが設定する視野範囲の近似領域を示す．

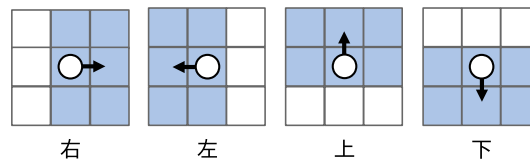


図 4-3 : 進行方向ごとの近似領域

表 4-1 : 進行方向を分類する条件

	右	左	上	下
パターン 2	$\frac{1}{\sqrt{2}} < e_x \leq 1$	$-1 \leq e_x < \frac{-1}{\sqrt{2}}$	$\frac{-1}{\sqrt{2}} < e_x < \frac{1}{\sqrt{2}}$	$\frac{-1}{2} < e_x < \frac{1}{2}$
パターン 3	$\frac{-1}{2} < e_y < \frac{1}{2}$	$\frac{-1}{2} < e_y < \frac{1}{2}$	$\frac{1}{\sqrt{2}} < e_y \leq 1$	$-1 \leq e_y < \frac{-1}{\sqrt{2}}$
パターン 4	$R_x \geq A_x$ $L_x \geq A_x$	$R_x < A_x$ $L_x < A_x$	$R_y \geq A_y$ $L_y \geq A_y$	$R_y < A_y$ $L_y < A_y$
パターン 5	$R_x \geq x_1$ $L_x \geq x_1$	$R_x < x_2$ $L_x < x_2$	$R_y \geq y_1$ $L_y \geq y_1$	$R_y < y_2$ $L_y < y_2$
パターン 6	$\cos(\frac{1}{2}\theta_{view}) \leq e_y$	$e_y \leq -\cos(\frac{1}{2}\theta_{view})$	$\sin(\frac{1}{2}(\pi - \theta_{view})) \leq e_x$	$e_x \leq \sin(\frac{1}{2}(\pi - \theta_{view}))$

## 4.1 パターン 1 (セル分割法)

パターン 1 は，一般的なセル分割法<sup>(?)</sup>，<sup>(?)</sup> によるエージェント間距離の計算回数削減を行う．図 4-2 では黄色の領域がセル分割法による近似領域を表す．セル分割法は，図 4-2 のように解析領域を格子状のセルに分割し，影響半径  $R_c$  の円内に重なるセルを近似領域とする手法である．本手法は，分割するセルのサイズを影響半径と同じ距離に設定することで，近似領域をエージェントの近傍 9 セルに限定することができる．本例では，エージェント 4 の近傍 9 セルは黄色のセルであり，それ以外のセルに属するエージェント 3，5，9 に対するエージェント間距離の計算を削減できる．なお，セル分割法は相互作用力を計算する範囲を円形で想定した手法であるため，近似領域が正方形となることから，パターン 1 の実装では進行方向の推測は行わない．また，本論文では，セル分割法のなかでもメモリ使用量を抑えることができると思われる連結リスト法<sup>(?)</sup>，<sup>(?)</sup> を用いる．

## 4.2 パターン 2

パターン 2 は，エージェントの進行方向を表す単位ベクトル  $e_i$  を参照することで，パターン 1 の近似領域を視野範囲に近づける手法である．本手法は，表 4-1 の条件に基づいて単位ベクトル  $e_i$  が示す進行方向を上下左右を  $\frac{1}{2}\pi$  の均等な角度で分割し，セル分割法の近似領域から図 4-3 のように進行方向の反対側にある 3 セルを除外する．図 4-2 の例では，エージェントの進行方向は右と判定され，図 4-3 に示す右方向の青いセルを近似領域に設定する．このため，白色のセルに存在するエージェン

トに対する距離計算を削減できる．パターン 2 は，進行方向を必ず上下左右のいずれかに設定するため，常に近似領域が 6 セルとなり，セル分割法の 9 セルに比べて近似領域の面積を  $\frac{2}{3}$  に削減できる．一方で，図 4-2 のように視野範囲全体が近似領域内に含まれる保障がないため，本来相互作用力の計算が必要なエージェントに対する計算を削減し，誤差が発生する可能性がある．

### 4.3 パターン 3

パターン 3 は，パターン 2 で発生する誤差を防ぐために，近似領域外に視野範囲が存在しないかを判定する処理を追加し，セル分割法と同じ精度を保つ手法である<sup>(?)</sup>．本手法は，パターン 2 の近似領域を導出したのち，視野の座標  $(L_x, L_y)$ ， $(R_x, R_y)$  が近似領域内のセルであればパターン 2 の近似領域を用いて相互作用力を計算し，そうでなければパターン 1 の近似領域を用いて相互作用力を計算する．図 4-2 のエージェントは， $L_x, L_y$  がパターン 2 の近似領域の外にあるため，パターン 3 ではパターン 1 と同様に近傍 9 セルを近似領域とする．エージェントの座標は時間ステップごとに変化するため，パターン 3 を用いると，進行方向が変化しないエージェントに対してもすべての時間ステップでパターン 1 よりもエージェント間距離の計算を削減できるとは限らない．一方で，パターン 3 は，パターン 1 のセル分割法と同じ精度のシミュレーション結果を得ることができる．

### 4.4 パターン 4

パターン 4 は，視野の左右両端の座標  $(L_x, L_y)$ ， $(R_x, R_y)$  を用いて近似領域を設定する手法である．視野範囲は進行方向の前方に存在するため，視野の左右両端の座標はエージェント座標から見て必ず進行方向側となる．この特性を利用し，パターン 4 では，表 4-1 のように，エージェント座標  $(A_x, A_y)$  と視野座標  $(R_x, R_y)$ ， $(L_x, L_y)$  の大小関係を用いて進行方向を判定する．図 4-2 の例では， $R_x \geq A_x$  および  $L_x \geq A_x$  が成り立つため，エージェントの進行方向は上と判定される．進行方向が求まると，近傍 9 セルから進行方向反対側の 3 セルを除外した 6 セルを近似領域とする．表 4-1 のいずれの条件にも当てはまらない場合は，パターン 1 (セル分割法) と同様に近傍 9 セルを近似領域とする．

#### 4.4.1 パターン5

パターン5は、セル分割法のセル座標と視野範囲の左右両端の座標  $(L_x, L_y)$  ,  $(R_x, R_y)$  を用いて近似領域を設定する手法である。本手法は、運動方程式を算出するエージェントが所属するセルの左上座標  $(x_1, y_2)$  および右下座標  $(x_2, y_1)$  を用いて、図4-3の4パターンの水色の領域内に視野が収まっているかを判定する。図4-2の例では、 $R_y \geq y_1$  および  $L_y \geq y_2$  が成り立つため、進行方向は上と分類される。本手法は、エージェントのセル上の位置に応じて進行方向を分類するため、ベクトル  $e_i$  の表す進行方向が同じエージェントでもエージェント座標に応じて異なる方向に分類される可能性がある。本手法も表4-1のいずれの条件にも当てはまらない場合は、パターン1（セル分割法）と同様に近傍9セルを近似領域とする。

#### 4.5 パターン6

パターン6は、パターン2で設定した上下左右の角度範囲を、シミュレーション誤差の出ない範囲で設定する手法である。パターン6では、図4-3中の水色の領域である近似領域内に視野範囲がすべて収まる進行方向を静的に計算し、エージェントの進行方向を表すベクトル  $e_i$  の閾値を定める。視野角を  $\theta_{view}$  とおくと、エージェント座標がどの位置であっても上方向と判定される進行方向の角度は  $\frac{1}{2}(\pi - \theta_{view})$  から  $\frac{1}{2}(\pi + \theta_{view})$  の間であり、これを用いて進行方向ベクトル  $e_i = (e_x, e_y)$  の範囲が  $\sin(\frac{1}{2}(\pi - \theta_{view})) \leq e_x$  という条件を設定できる。パターン6は、パターン3~6のように視野座標  $(R_x, R_y)$  および  $(L_x, L_y)$  を算出する必要がないため、他の分類条件よりも高速に進行方向を分類することができる。一方で、進行方向が特定の方向である場合は、エージェント座標に関係なくパターン1（セル分割法）と同じ近似領域を設定するため、エージェント間距離の計算回数は、パターン1に次いで多くなると考えられる。

## 第5章

# 格子分割による進行方向ベクトル計算の削減手法

SFM の運動方程式は，エージェントの位置が変わるたびに，目的地までのベクトルを表す  $e$  や周囲のエージェントを避ける力  $f_{ij}$ ，障害物を避ける力  $f_{iw}$  の再計算が必要となる．周囲のエージェントを避ける力  $f_{ij}$  の計算は，時間ステップごとにすべてのエージェントの位置が変化するため，解析中のみに計算が可能である．一方，ベクトル  $e$  や障害物を避ける力  $f_{iw}$  は，エージェントの位置に応じて決定し，壁や机などの障害物の座標が固定であることから，解析前に計算が可能である．そこで，本論文では，格子状に分割した領域ごとにベクトル  $e$  と障害物を避ける力  $f_{iw}$  をあらかじめ計算することで，解析中の進行方向の再計算を削減する．

### 5.1 進行方向計算の前処理

### 5.2 壁を避ける力の前処理

図 5-1 に格子分割した進行方向の例を示す．図 5-1 中の  $\bullet$  はエージェントであり，エージェント A，B が 1 つの目的地に向かう様子である．図中のエージェント B は，エージェント A の位置に移動したとき，エージェント A と同じ進行方向になる．このため，解析領域を分割し，格子ごとに進行方向を算出することで，解析中の進行方向の再計算が削減できる．格子ごとの進行方向は，格子の中心座標から目的地までのベクトル  $e$  を求め，配列に格納する．同様に，壁から受ける力  $f_{iw}$  は，解析中に机や壁が動かない特徴を利用し，格子ごとの中心座標を用いて， $f_{iw}$  をあらかじめ算出する．目的地や障害物が存在する格子では，正しく目的地に進むように，0 ベクトルを格納する．0 ベクトルが格納されている格子中に存在するエージェントは，エージェントごとに本来の SFM と同様に進行方向を計算する．目的地の他に経由地が複数存在する場合は，図 5-2 に示すように，経由地ごとに進行方向を設定する．図 5-2 の例では，目的地に向かっているエージェントは，目的地の進行方向が格納

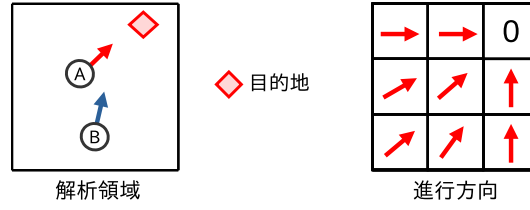


図 5-1 : 提案する格子分割の例

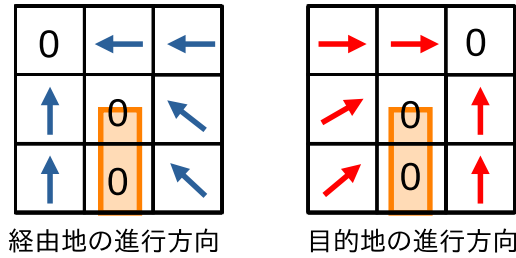


図 5-2 : 経由地がある場合の進行方向の例

されている配列を参照し，経由地に向かっているエージェントは，経由地の進行方向が格納されている配列を参照する．

図 5-3 に提案手法の解析全体のフローチャートを示す．図 5-3 中の前処理は，提案手法に必要である格子ごとの進行方向や壁を避ける力を算出する．図 5-5 に提案手法の前処理のフローチャートを示す．図 5-5 に示すように，提案手法は，進行方向のベクトル  $e$  を格納する配列と壁を避ける力  $F_W$  を格納する配列が必要となる．各配列の計算は，図 5-5 のように格子ごとにあらかじめ計算する．進行方向のベクトル  $e$  を格納する配列中の要素は，図 5-2 のように経由地ごとに違うため，それぞれの経由地で計算する必要となる．このため，進行方向ベクトル  $e$  を格納する配列の前処理は，経由地数の多さや格子サイズの細かさに応じて計算回数が増加する．図 5-5 中の壁を避ける力  $f_{iw}$  を格納する配列は，経由地が変わっても変化しないため，解析領域全体で一つとなる．このため，壁を避ける力  $f_{iw}$  を格納する配列の前処理は，格子サイズの細かさに応じて計算回数が増加する．



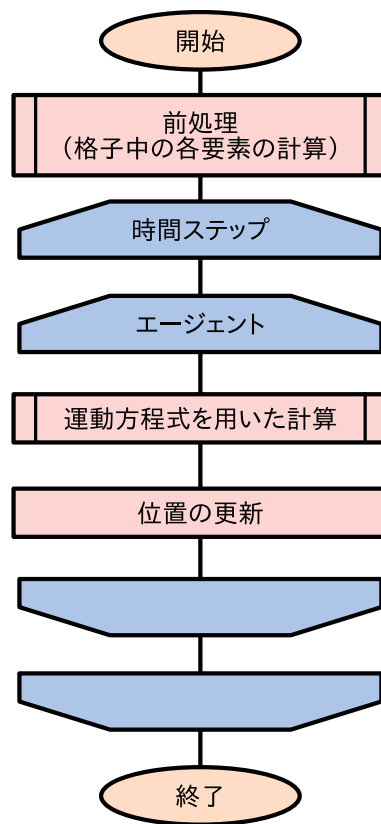


図 5-3 : 提案手法の解析全体のフローチャート

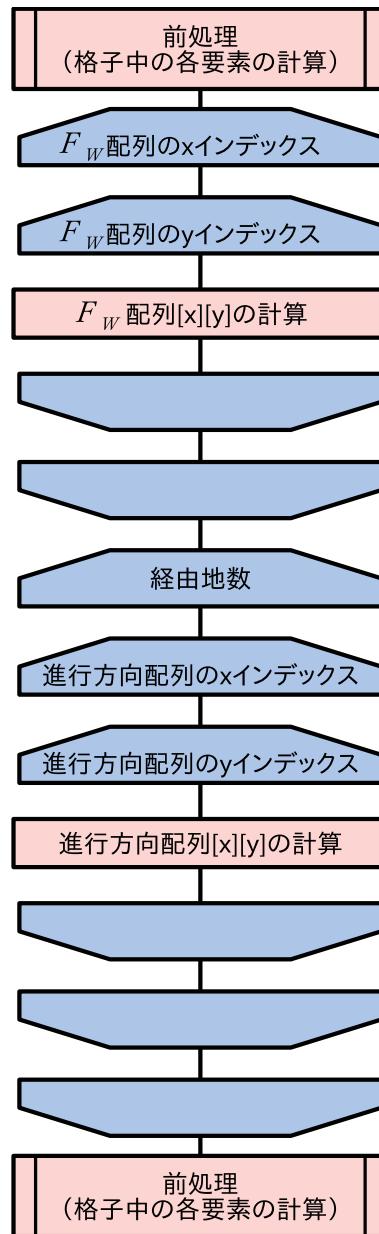


図 5-4 : 提案手法の前処理のフローチャート

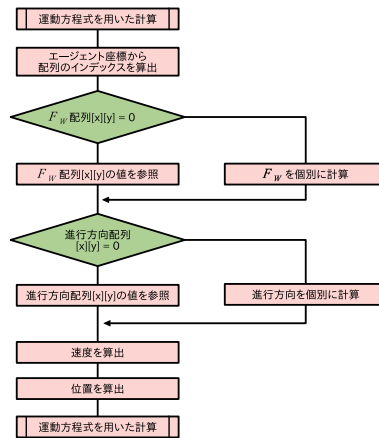


図 5-5 : 提案手法の運動方程式計算のフローチャート

# 第6章

## 評価

使い方についての FAQ 的な感じで

- デフォルトの章構成が気に食わない  
main.tex 中の `\input` を消去してください . 以下の 2 通りの方法で対処できます .
  - 1 `\input`
  - 2 不要な chapter の書かれたファイルを
- 章ごとにファイルを分けるのがめんどい  
第??節や第??節やを参照してください .
- 参考文献が更新されない  
第 3.7 節を参照してください .
- アブストラクトページに工大マークが表示されない (修士のみ)  
コンパイル環境によっては起こるらしい .

## 第7章

### おわりに

\(^o^)/

# 謝辞

aa aa

2023 年 12 月 24 日

## 参考文献

(1) : (1983).

(2) : .

# 付 録 A

## プログラムの説明

付録には，添付するソースコードの説明を書いてください．データ構造や主要な変数の説明は本文中で述べてあると思います．本文で述べたことを一覧形式でまとめる分には構いませんが，まったく同じことを書くのはよくありません．このため，本文中では書けない実装の話（コンパイル方法や，測定条件の変更方法，入出力フォーマットなど）を中心に書きましょう．

また，付録のページは，本文中で邪魔になった定義とか証明とかの避難場所としても利用可能です．

### A.1 節番号のテスト

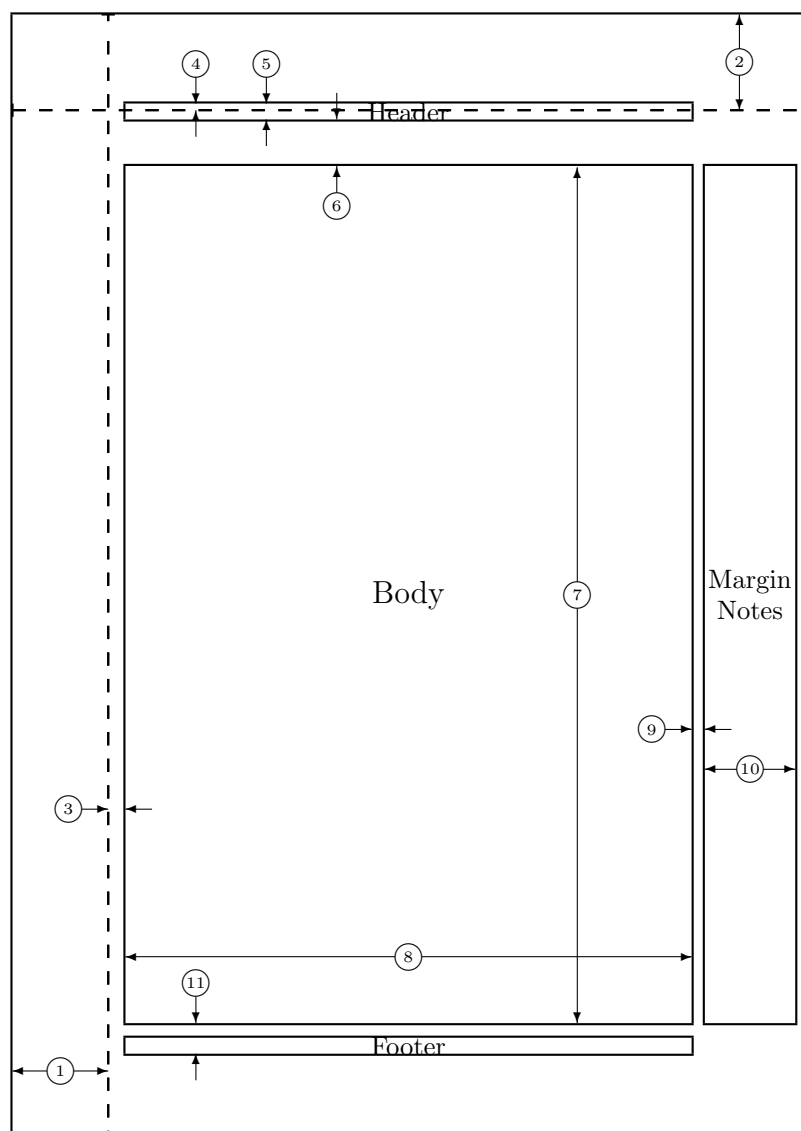
#### A.1.1 項番号のテスト

付録では，こんな風に章番号が表示されます．付録 A, 付録 B というように，付録の chapter にも章番号をつけたい場合は，main.tex66 行目の `\appendix` を `\appendixes` に変更してください．

### A.2 ページレイアウト表示

`tex` の機能を使ってページレイアウトの情報を表示する．





- |                         |                                  |
|-------------------------|----------------------------------|
| 1 one inch + \hoffset   | 2 one inch + \voffset            |
| 3 \oddsidemargin = 13pt | 4 \topmargin = -5pt              |
| 5 \headheight = 12pt    | 6 \headsep = 35pt                |
| 7 \textheight = 645pt   | 8 \textwidth = 426pt             |
| 9 \marginparsep = 10pt  | 10 \marginparwidth = 68pt        |
| 11 \footskip = 23pt     | \marginparpush = 7pt (not shown) |
| \hoffset = 0pt          | \voffset = 0pt                   |
| \paperwidth = 597pt     | \paperheight = 845pt             |