

進行方向の計算回数削減による ソーシャルフォースモデルを用いた 人流シミュレーションの高速化

情報科学研究科 情報科学専攻 前川研究室

2281011 片寄颯人



はじめに

人流シミュレーション

歩行者をエージェントでモデル化

駅や商業施設などの避難シミュレーション
→ソーシャルフォースモデル(SFM)

解析人数が増えるほど処理時間が長くなる



はじめに

人流シミュレーション

歩行者をエージェントでモデル化

駅や商業施設などの避難シミュレーション
→ソーシャルフォースモデル(SFM)

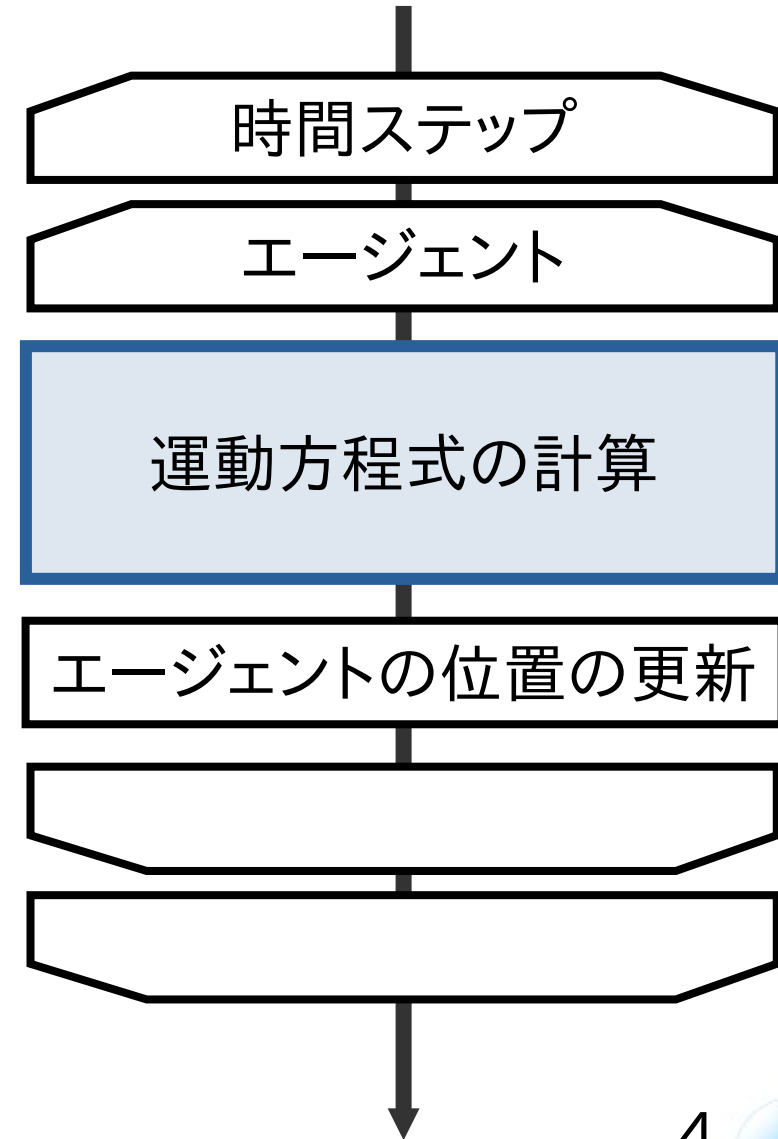
解析人数が増えるほど処理時間が長くなる

本研究の目的

SFMを用いた人流シミュレーションの高速化

SFMの解析手順

反復回数が多くなるほど
解析時間が長くなる



SFMの運動方程式

周囲の人を避ける動きをモデル化して
エージェントの動きを再現

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0 e_i - v_i}{\tau} + \sum_j F_{ij} + \sum_w F_{iw}$$

m_i エージェントの体重

v_i^0 エージェントの希望速度

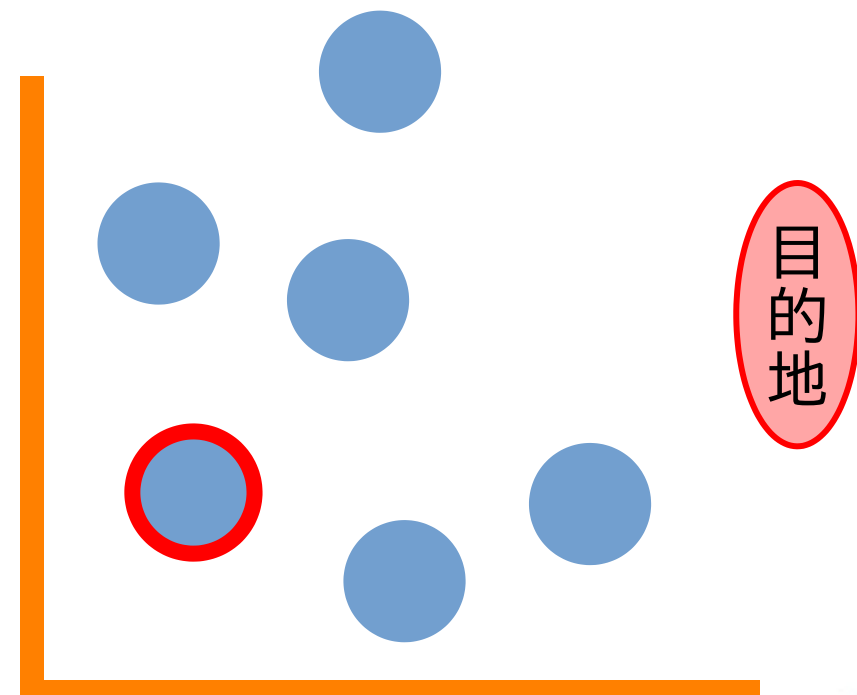
v_i エージェントの歩行速度

τ 時定数

ソーシャルフォースモデル (SFM)

エージェント移動を決定する運動方程式

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0 e_i - v_i}{\tau} + \sum_j F_{ij} + \sum_w F_{iw}$$

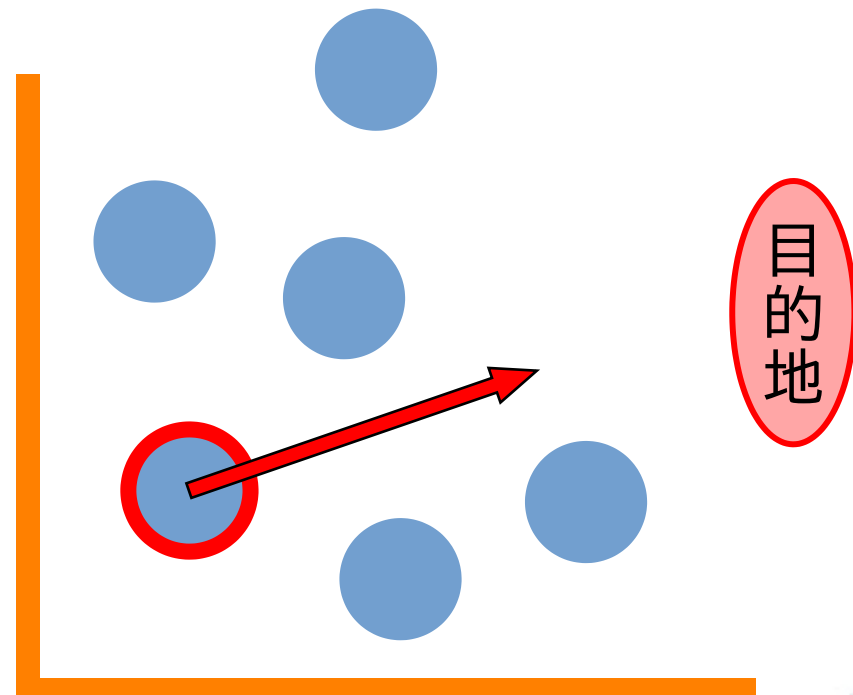


ソーシャルフォースモデル (SFM)

エージェント移動を決定する運動方程式

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0 e_i - v_i}{\tau} + \sum_j F_{ij} + \sum_w F_{iw}$$

目的地



ソーシャルフォースモデル (SFM)

エージェント移動を決定する運動方程式

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0 e_i - v_i}{\tau} + \sum_j F_{ij} + \sum_w F_{iw}$$

目的地

進行方向ベクトル e_i の計算が必要

目的地

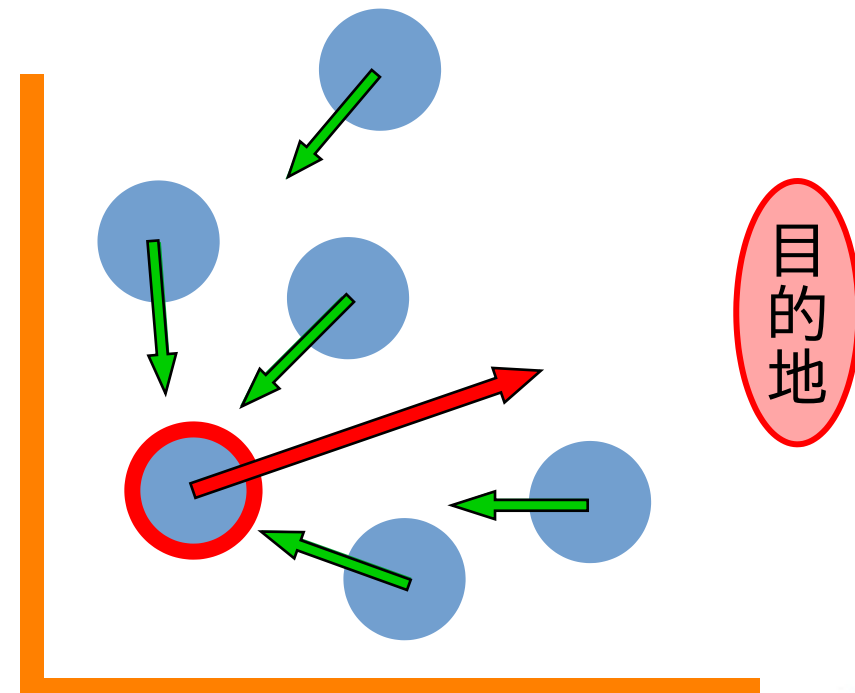
ソーシャルフォースモデル (SFM)

エージェント移動を決定する運動方程式

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0 e_i - v_i}{\tau} + \sum_j F_{ij} + \sum_w F_{iw}$$

目的地

周囲のエージェントを避ける力



ソーシャルフォースモデル (SFM)

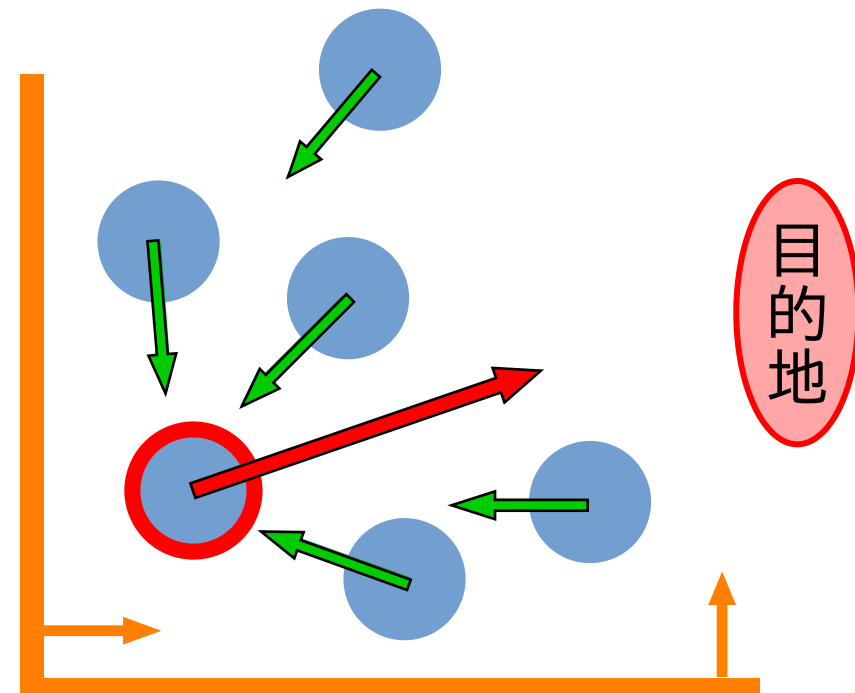
エージェント移動を決定する運動方程式

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0 e_i - v_i}{\tau} + \sum_j F_{ij} + \sum_w F_{iw}$$

目的地

周囲のエージェントを避ける力

障害物を避ける力



SFMの計算時間の削減

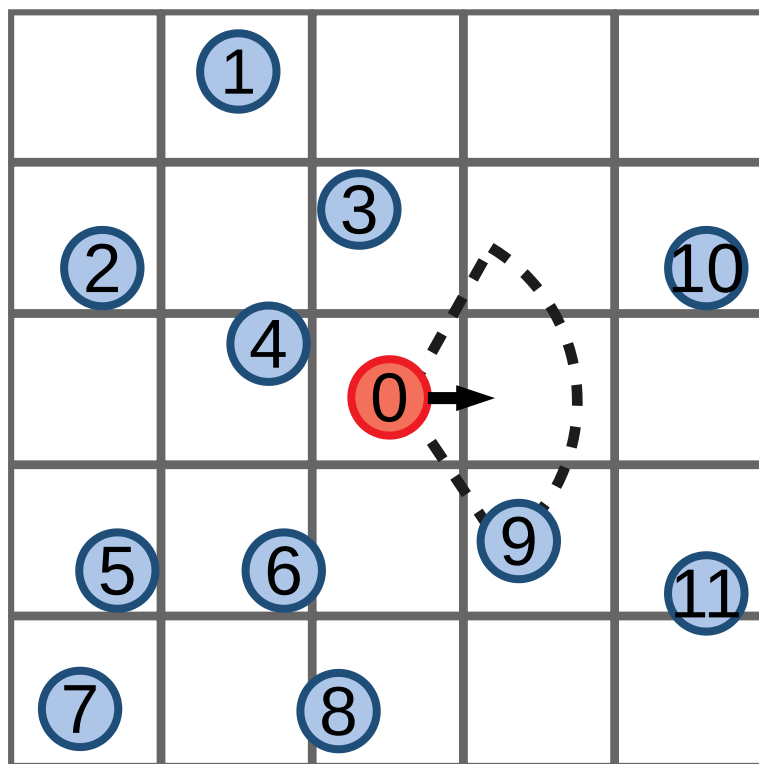
運動方程式 =  +  + 

エージェントを避ける力の計算

エージェント数が増えるほど計算回数が増える
→ **セル分割法**による計算回数削減

セル分割法を用いたSFM

解析領域を格子状に分割することで
エージェント間距離を計算するエージェント数を削減

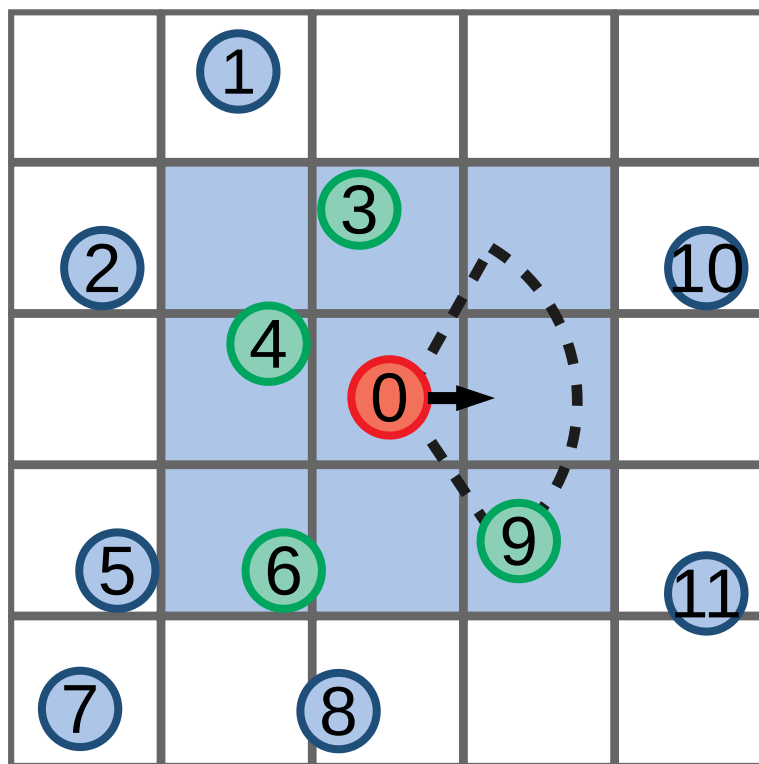


0 計算対象の
エージェント

-- 視野範囲

セル分割法を用いたSFM

解析領域を格子状に分割することで
エージェント間距離を計算するエージェント数を削減

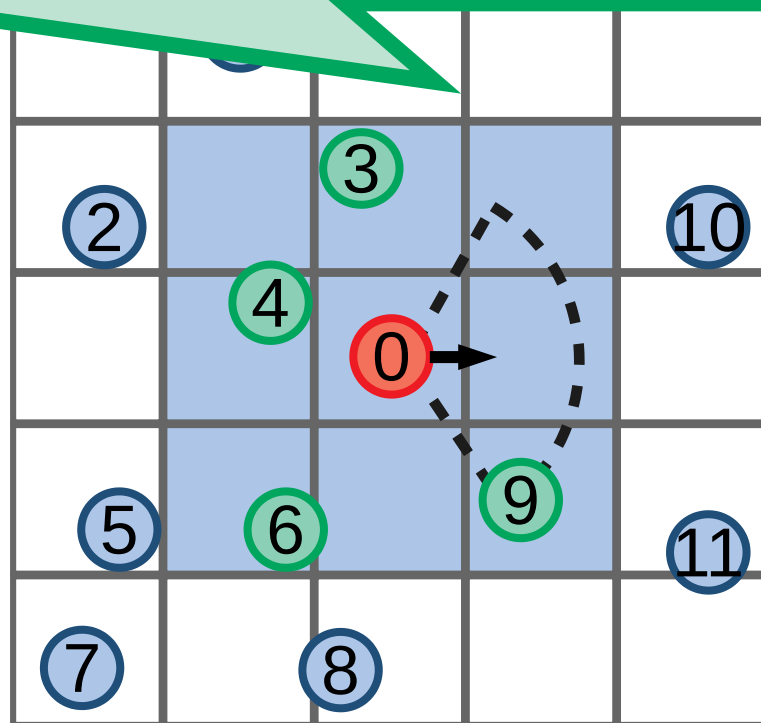


⑦ 計算対象の
エージェント

-- 視野範囲

セル分割法を用いたSFM

エージェント3: 距離と角度の計算→判定
エージェント4: 距離と角度の計算→判定
エージェント6: 距離と角度の計算→判定
エージェント9: 距離と角度の計算→判定→力の計算



0 計算対象のエージェント

-- 視野範囲

SFMの計算時間の削減

$$\text{運動方程式} = \text{[Red Box]} + \text{[Green Box]} + \text{[Orange Box]}$$

エージェントを避ける力の計算

エージェント数が増えるほど計算回数が増える
→セル分割法による計算回数削減

さらなる高速化には

障害物を避ける力

目的地

の計算回数削減が必要

提案手法の方針

$$\text{運動方程式} = \text{[Red Box]} + \text{[Green Box]} + \text{[Orange Box]}$$

障害物を避ける力の計算

障害物は座標が変化しない



目的地

進行方向ベクトル e_i はエージェントの座標に応じて決まる

提案手法の方針

$$\text{運動方程式} = \text{[Red Box]} + \text{[Green Box]} + \text{[Orange Box]}$$

障害物を避ける力の計算

障害物は座標が変化しない



目的地

進行方向ベクトル e_i はエージェントの座標に応じて決まる

提案手法

目的地

障害物を避ける力

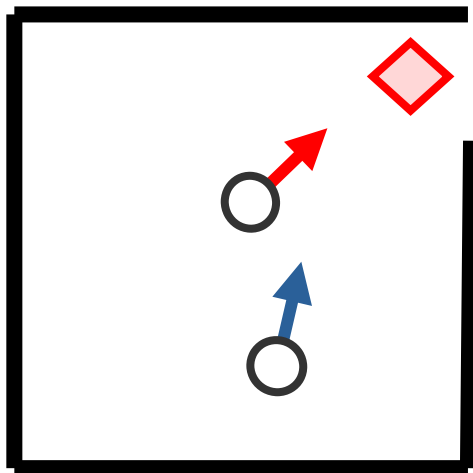
の計算を簡略化

→ 演算回数を削減による高速化を目指す

提案手法

解析領域を格子状に分割

あらかじめ格子ごとのエージェント進行方向を設定



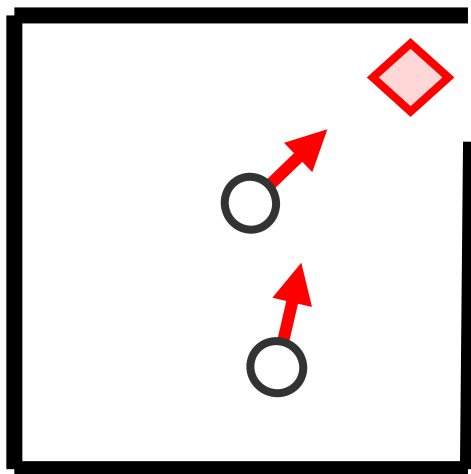
解析領域

◇ 目的地
○ エージェント

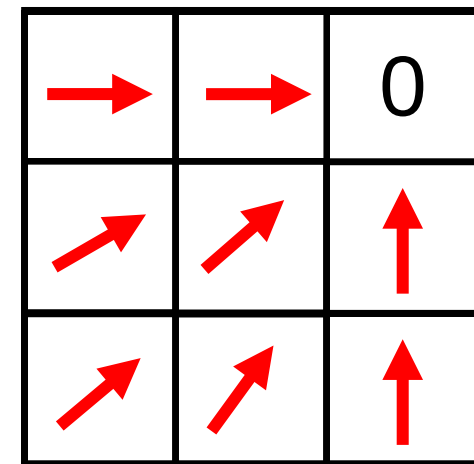
提案手法

解析領域を格子状に分割

あらかじめ格子ごとのエージェント進行方向を設定



解析領域

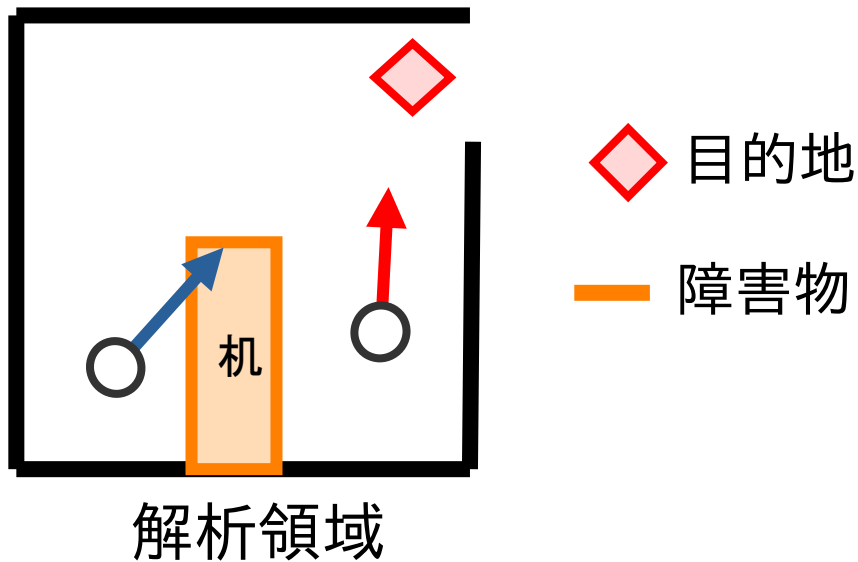


格子ごとの進行方向

提案手法

解析領域を格子状に分割

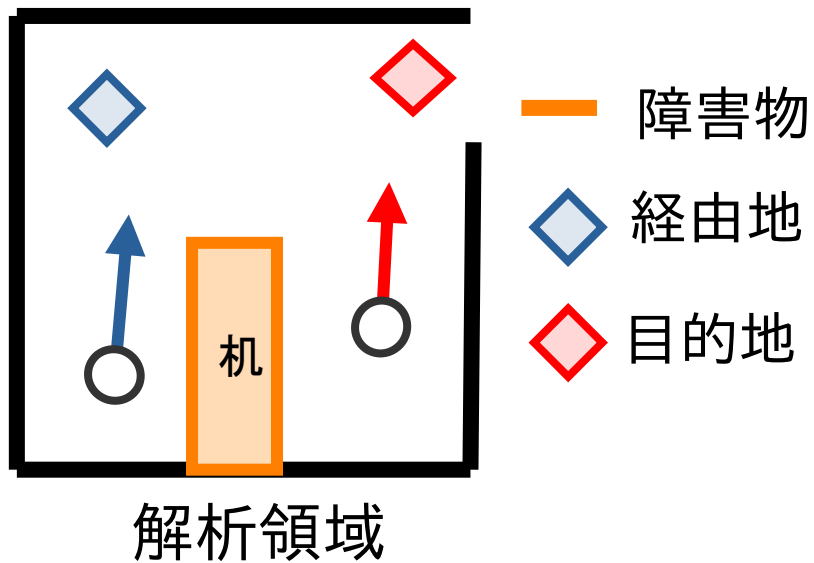
あらかじめ格子ごとのエージェント進行方向を設定



提案手法

解析領域を格子状に分割

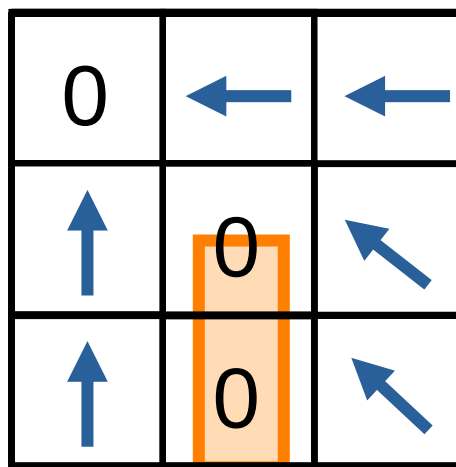
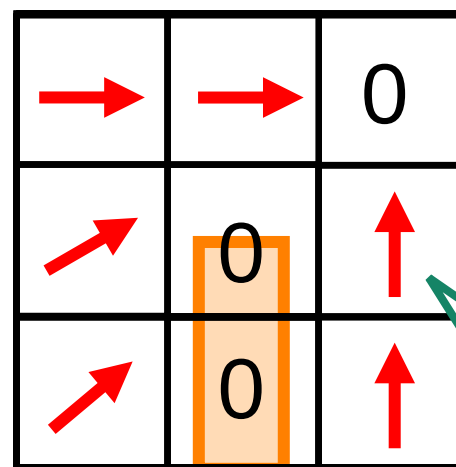
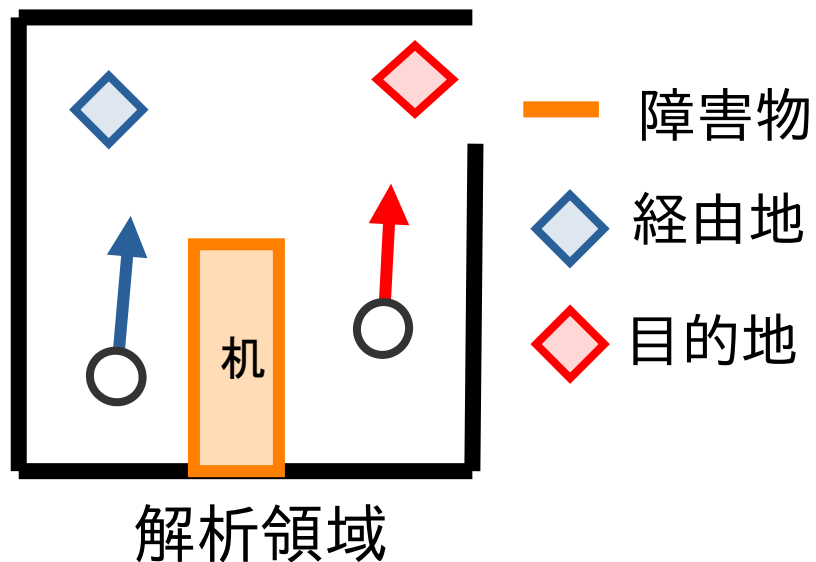
あらかじめ格子ごとのエージェント進行方向を設定



提案手法

解析領域を格子状に分割

あらかじめ格子ごとのエージェント進行方向を設定



各格子の計算

障害物を
避ける力

目的地
向かう力

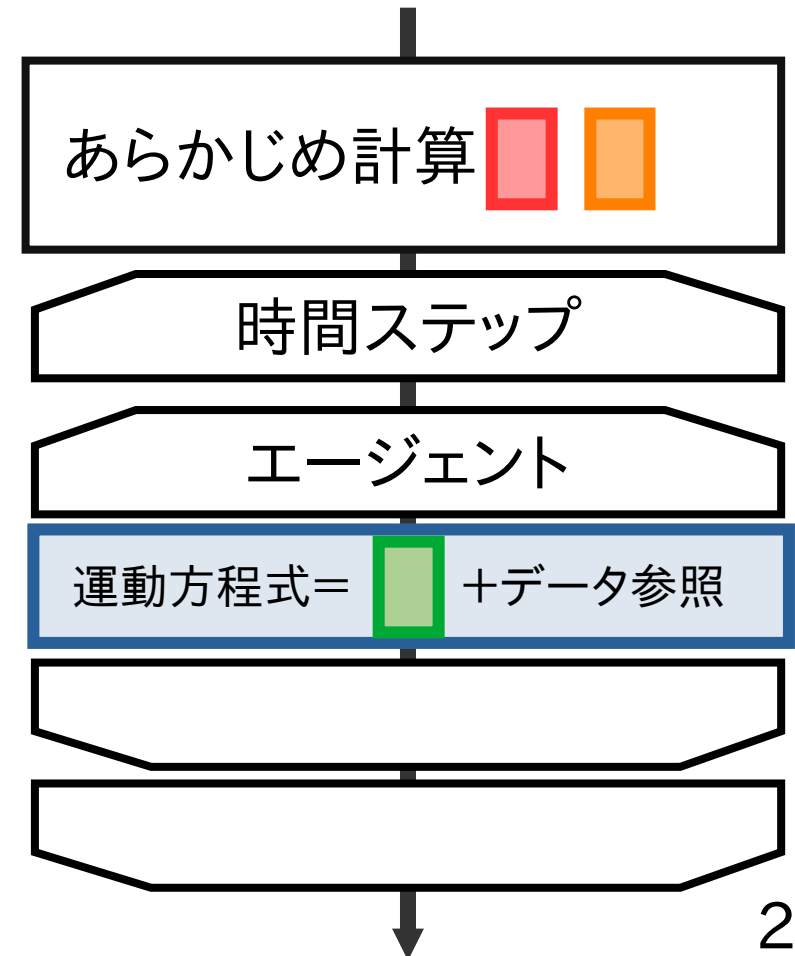
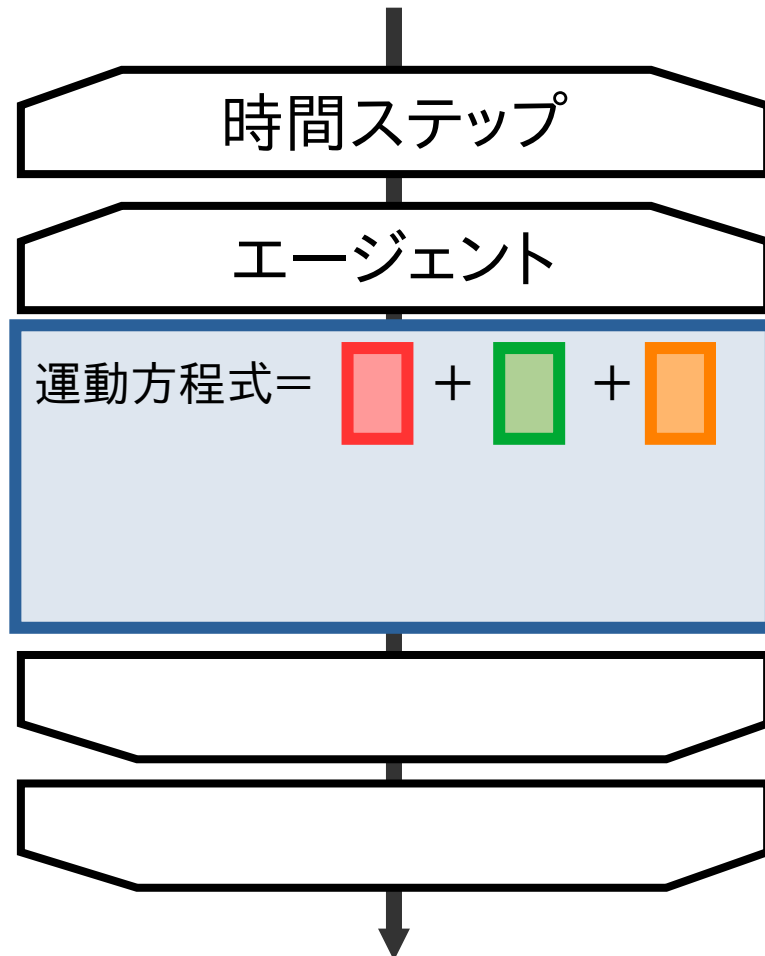
提案手法を用いたSFM

解析中は配列を参照

目的地

障害物を避ける力

の計算を削減



提案手法を用いたSFM

解析中は配列を参照

目的地

障害物から受ける力

の計算を削減

エージェントを避ける力

→解析中に計算が必要



エージェントを避ける力の
計算時間を削減
→さらなる高速化が見込める

あらかじめ計算



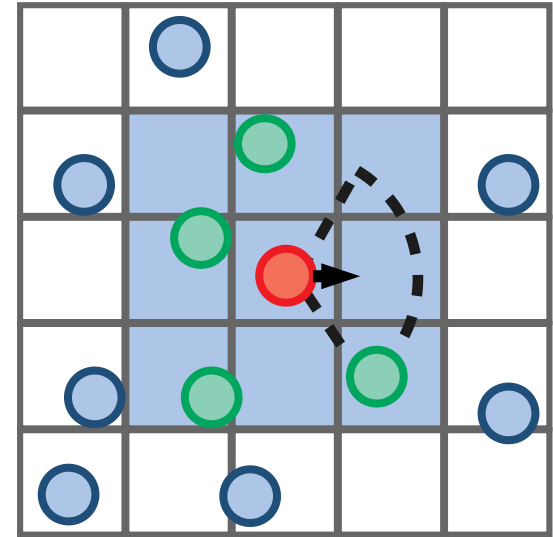
時間ステップ

エージェント

運動方程式 =  + データ参照

エージェントを避ける力の計算時間削減

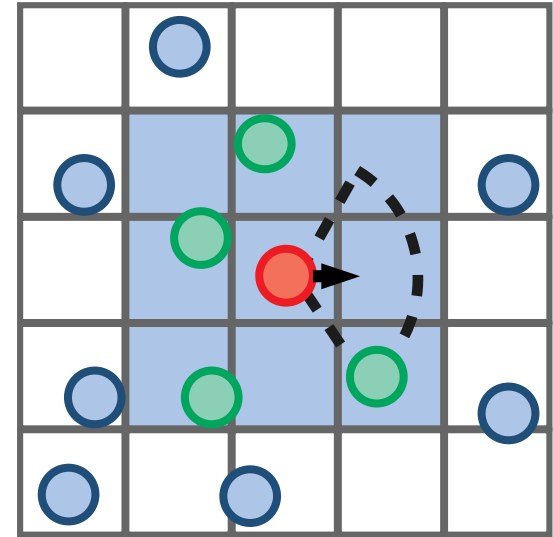
単純なセル分割法
影響範囲外のエージェント間距離も計算
→無駄な処理



エージェントを避ける力の計算時間削減

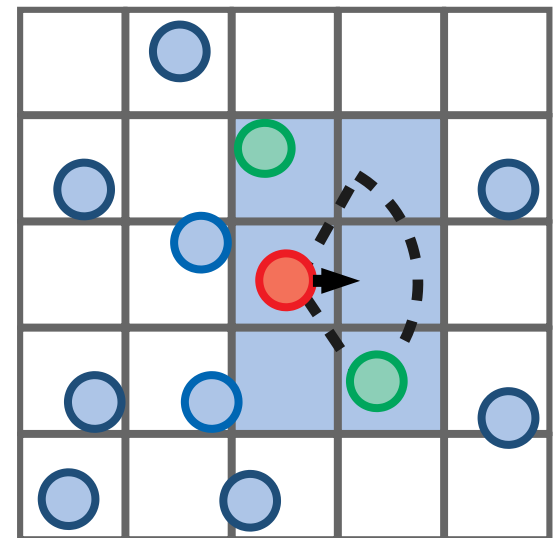
単純なセル分割法

影響範囲外のエージェント間距離も計算
→無駄な処理



提案手法

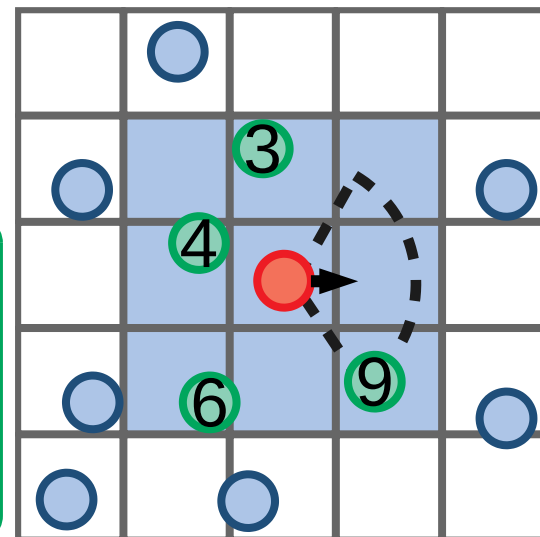
影響範囲に合わせた近似領域を設定
→エージェント間距離の計算回数を削減



近似領域の高精度化

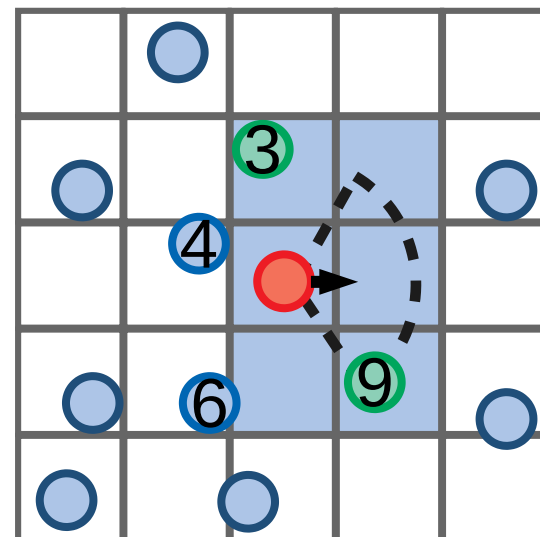
単純なセル分割法

エージェント3: 距離と角度の計算→判定
エージェント4: 距離と角度の計算→判定
エージェント6: 距離と角度の計算→判定
エージェント9: 距離と角度の計算→判定→力の計算



提案手法

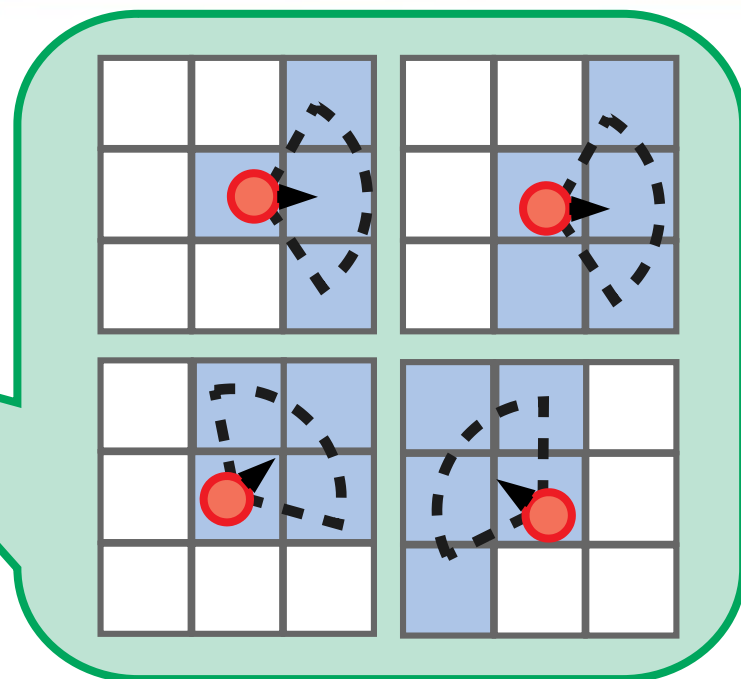
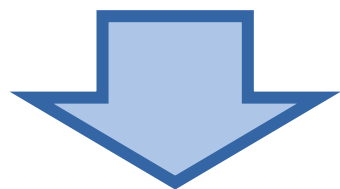
エージェント3: 距離と角度の計算→判定
エージェント9: 距離と角度の計算→判定→力の計算



近似領域の高精度化

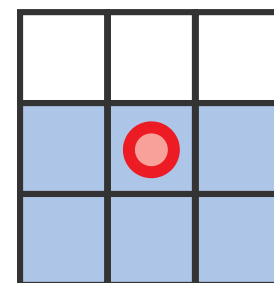
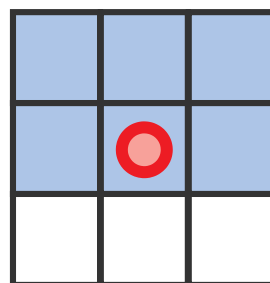
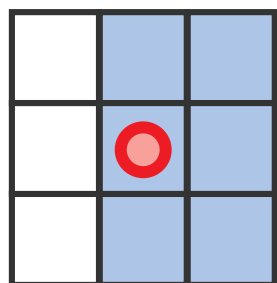
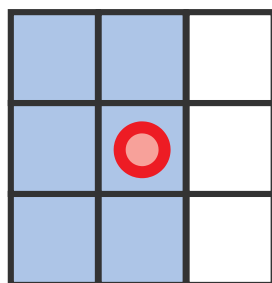
近似領域の設定方法

座標や進行方向に応じて数多く存在
→セルの選択に時間がかかる



近似領域の形状は**四角形を維持**

→範囲内に含まれるかどうかを**高速**に判定可能





近似領域の設定方法

進行方向(上下左右)の分類方法

- ・エージェントの進行方向ベクトルや視野角
 - ・セル分割法のセルの座標
- さまざまな方法が考えられる

本研究

近似領域の設定を6パターン設定

パターン1	セル分割法
パターン2	エージェントの進行方向
パターン3	エージェントの進行方向と視野座標
パターン4	エージェントの座標と視野座標
パターン5	視野座標とセルの座標
パターン6	エージェントの進行方向



評価

提案手法の有効性を評価するために
SFMを用いた人流シミュレーションを行う

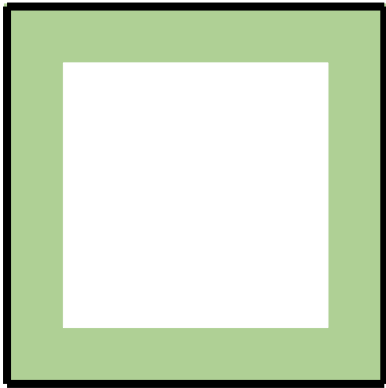
既存手法	セル分割法を用いたSFM
提案手法	<ul style="list-style-type: none">・ 格子分割を用いた進行方向の計算回数削減・ エージェント間距離の計算回数削減

測定内容

- ・ 進行方向計算中の演算回数
- ・ シミュレーション時間
- ・ 提案手法が生じる誤差

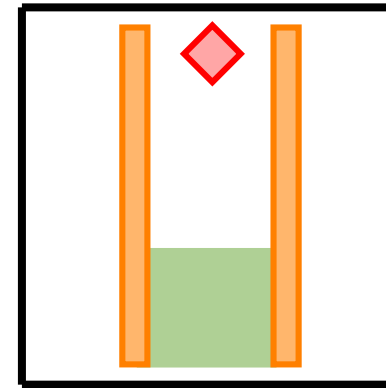
エージェントの初期配置

交差の配置

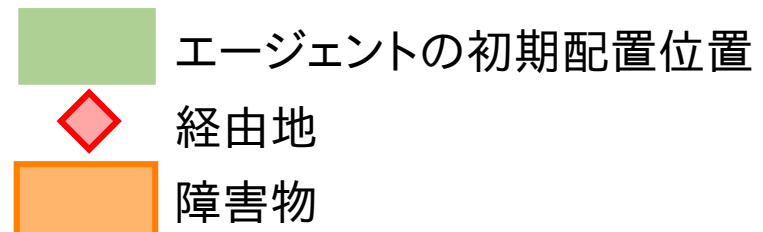


解析領域	300m × 300m
エージェント数	可変
壁粒子数	0
1エージェントあたりの経由地	0

直進の配置



解析領域	50m × 50m
エージェント数	40
壁粒子数	可変
1エージェントあたりの経由地	1





測定条件

関連研究を参考にパラメータを設定

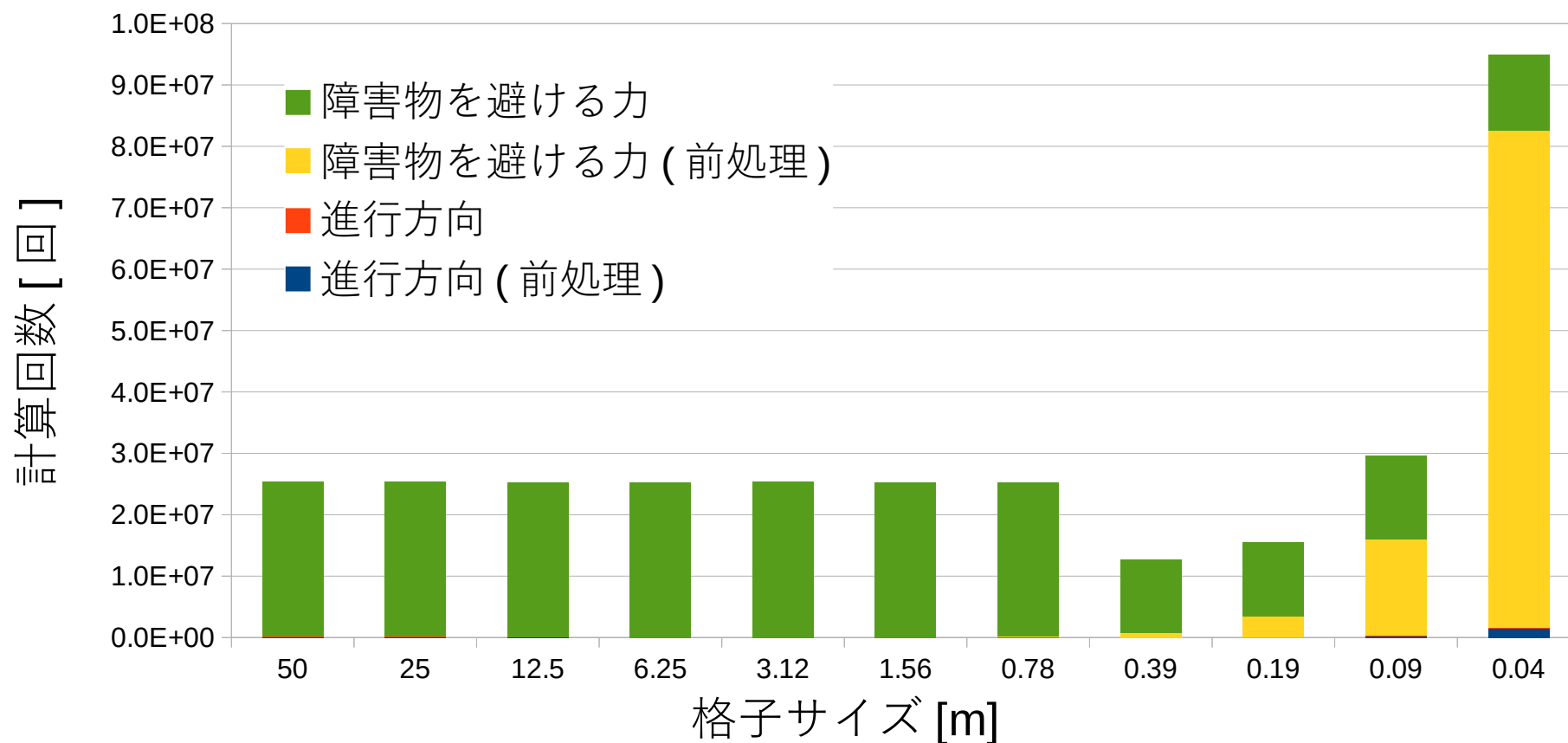
A_i	2000 N
B_i	0.08
k	$1.2 \times 10^5\text{ kg s}^{-2}$
K	$2.4 \times 10^5\text{ kg m}^{-1}\text{ s}^{-2}$
$v_i^0(t)$	1.4 m/s
t_i	0.0001 s
r_i	0.25 m
格子サイズ	$50.00\text{ m} \sim 0.19\text{ m}$



評価環境

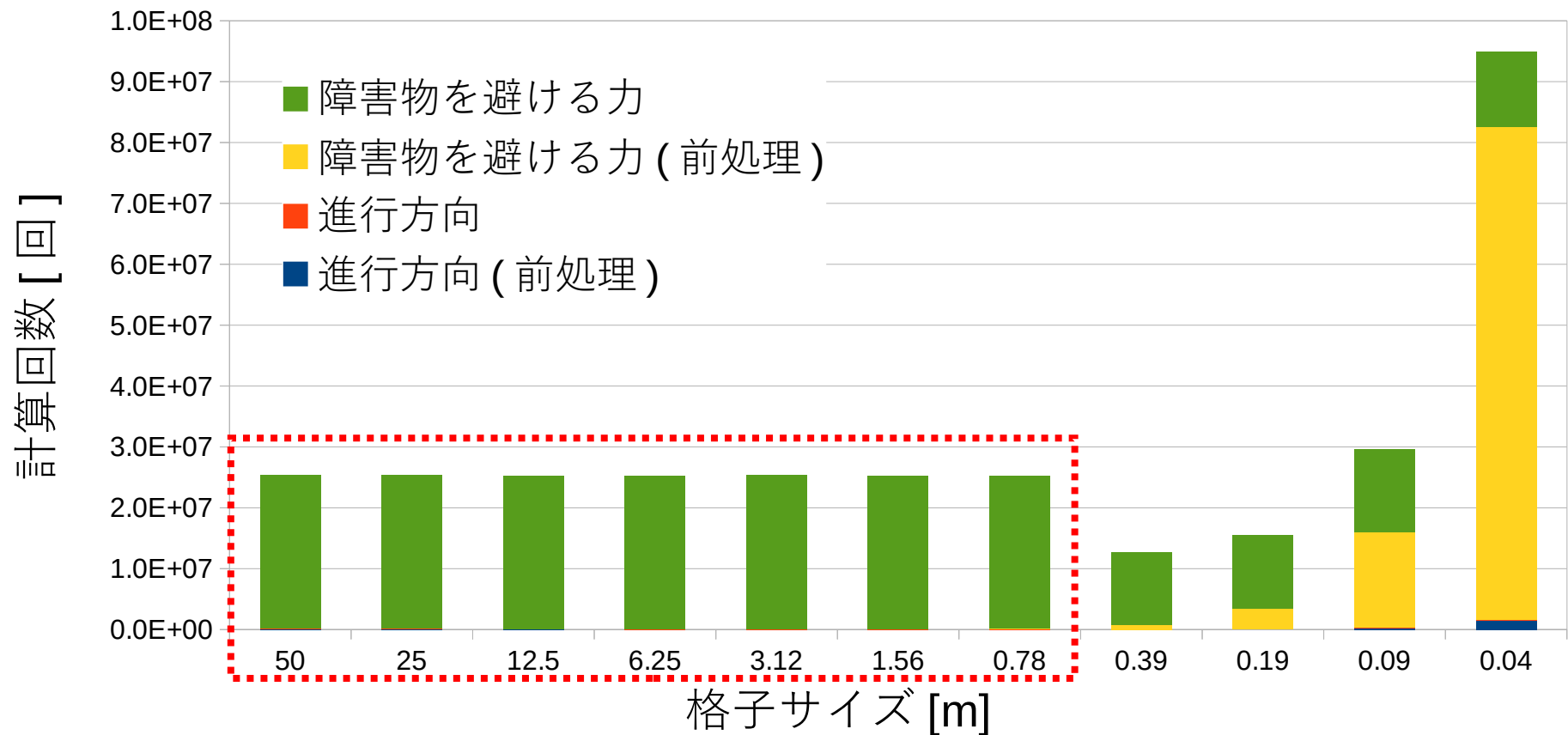
	マシン1	マシン2
CPU	Intel Xeon E5-2687W v2	Intel Xeon E5-2667W v2
Memory	64GB	32GB
OS	Linux 4.12.9	Linux 6.5.8
GCC version	7.2.0	13.2.0
最適化 オプション	-O3	

直進の配置 (通路幅2m) の計算回数



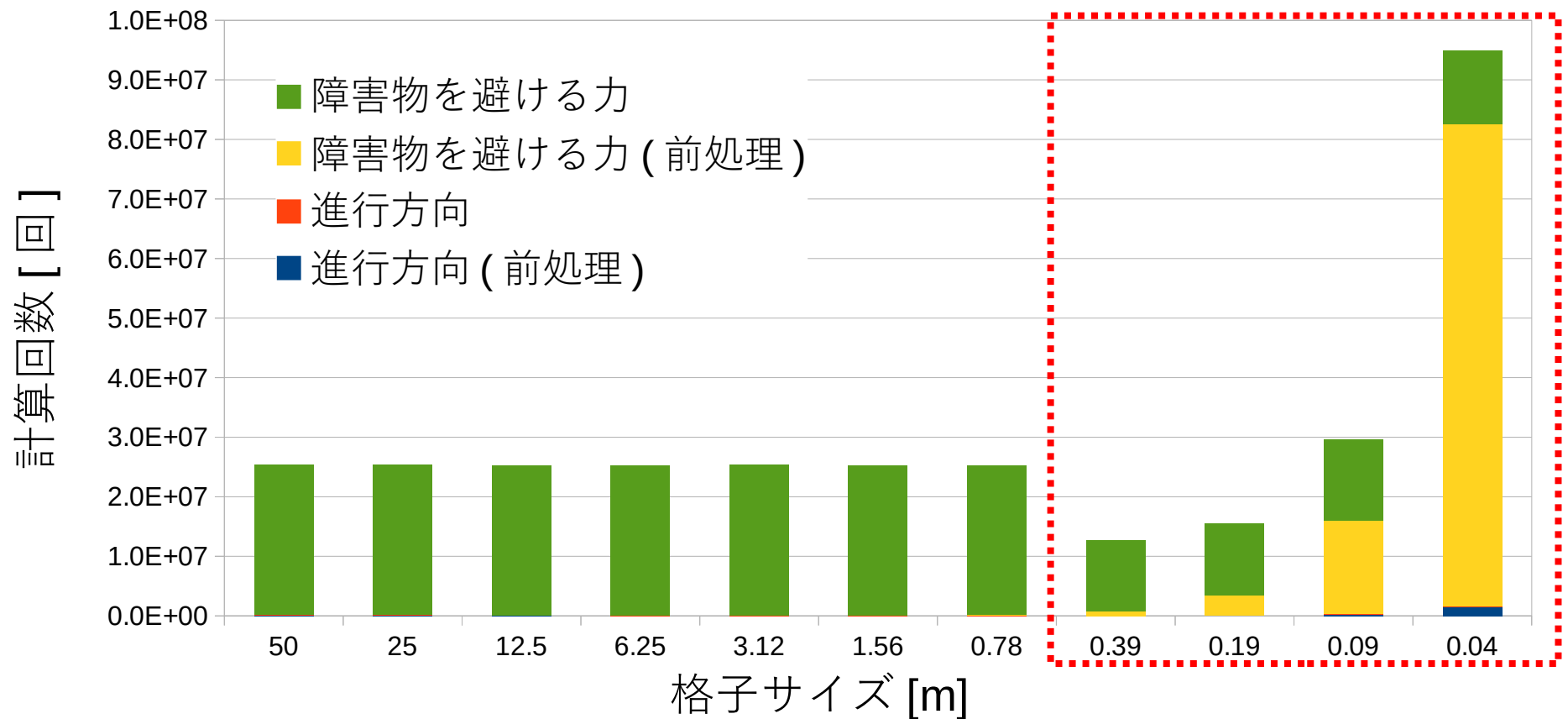
格子サイズ50mから0.78mまで
エージェントが通る場所が個別計算の格子に設定
→計算回数が削減できない

通路幅2mのときの計算回数



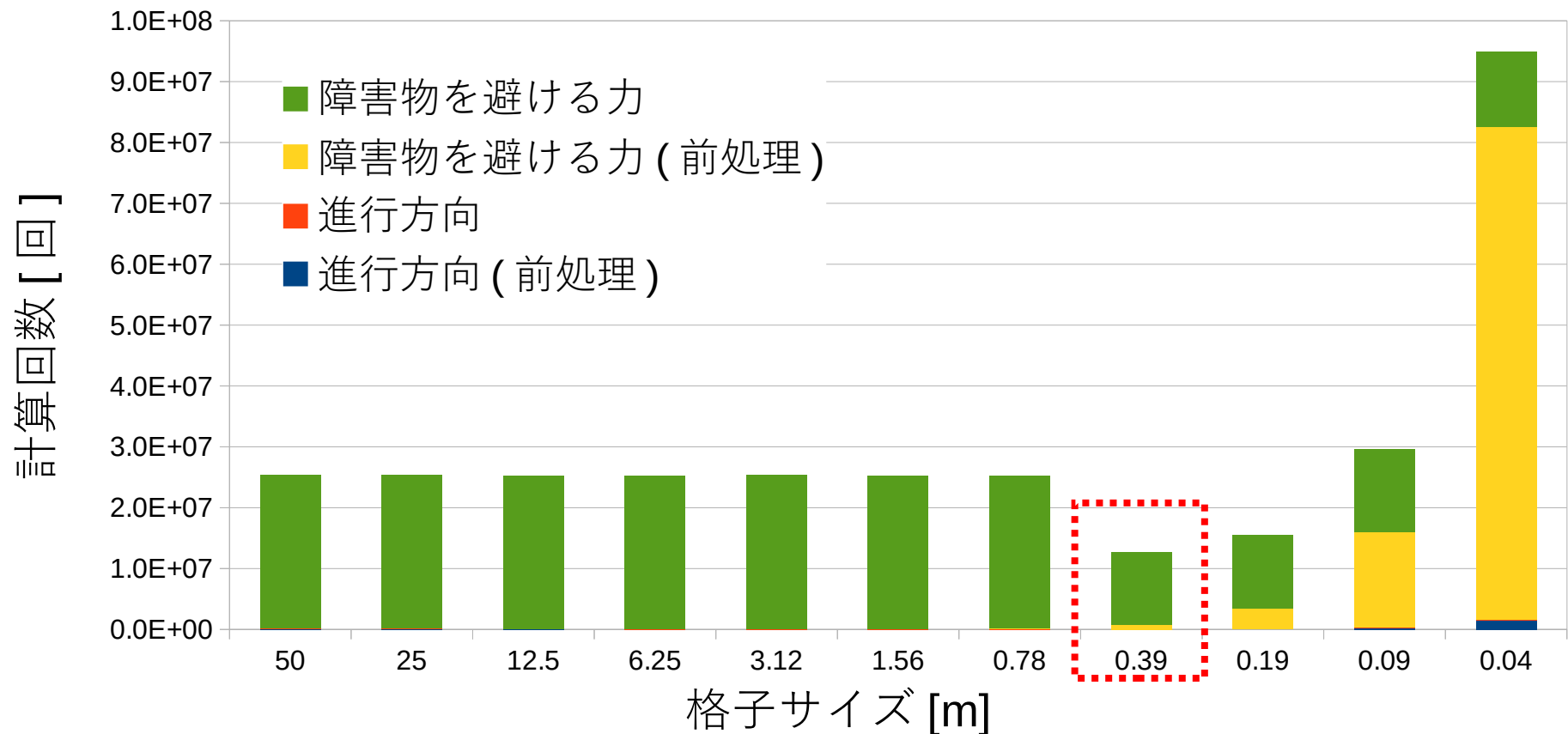
格子サイズ50mから0.78mまで
エージェントが通る場所が個別計算の格子に設定
→計算回数が削減できない

通路幅2mのときの計算回数



格子サイズが大きくなるほど前処理にかかる時間が多くなる
→格子数が多くなるほど前処理中の計算回数が多くなるため

通路幅2mのときの計算回数

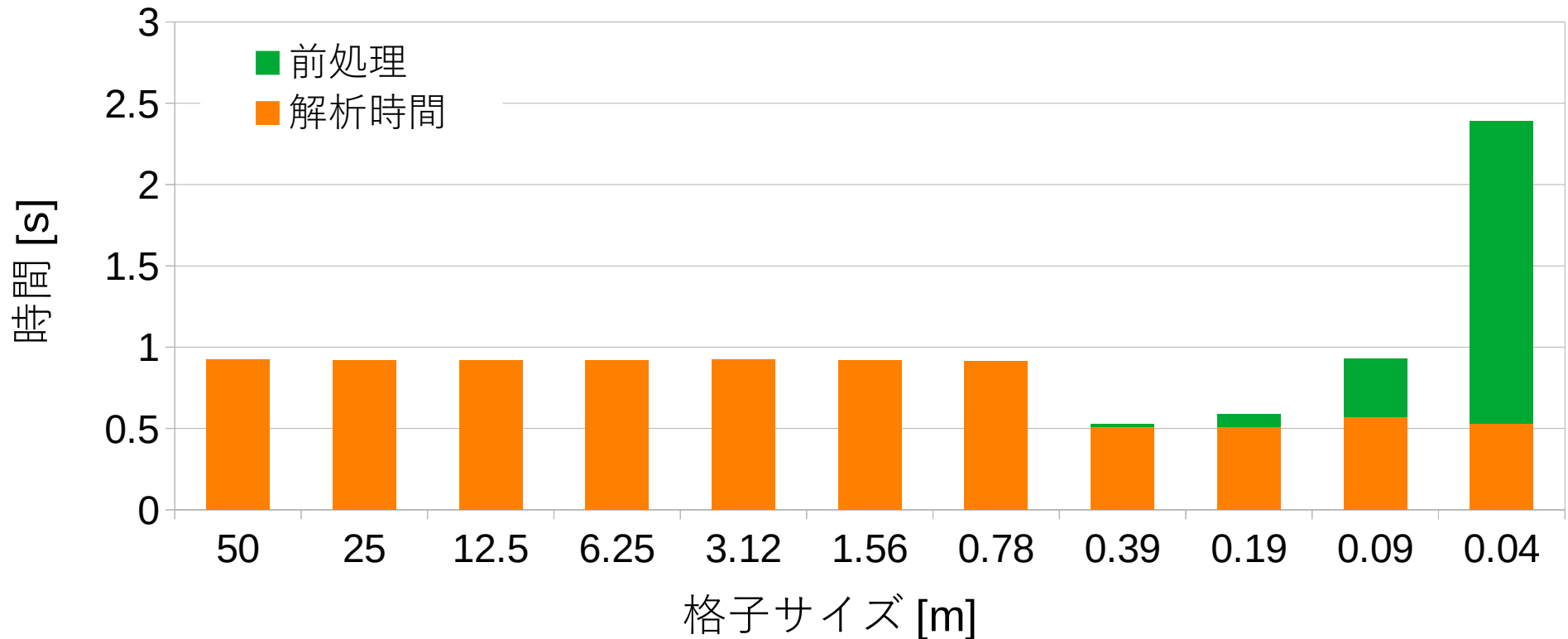


格子サイズ0.39

本測定下において一番計算回数が少ない

- エージェントが通る場所が個別計算する格子にならない
- ・前処理の計算回数が低い

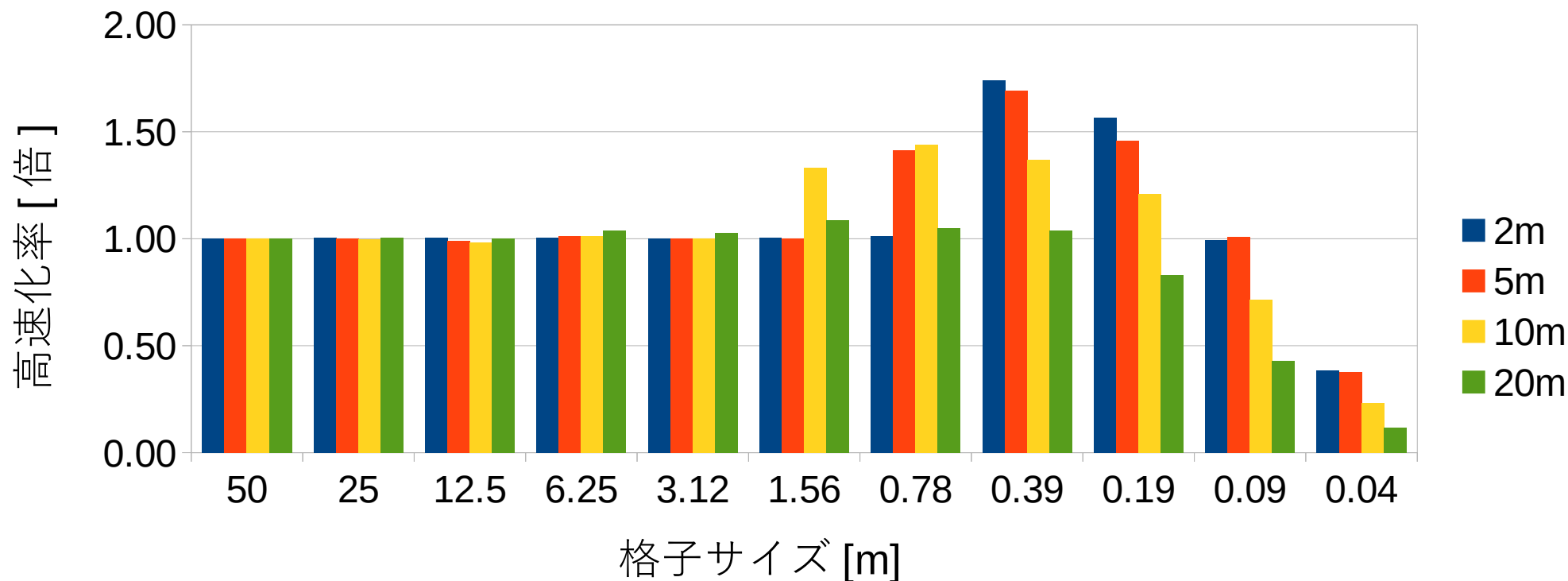
通路幅2mのときの実行時間



格子サイズ0.39mのときに最もはやく解析できる

格子サイズが小さくなるほど前処理の時間が多くなる
→格子数が多くなるほど計算回数が多くなるため

通路幅を変えたときの高速化率



最大高速化率

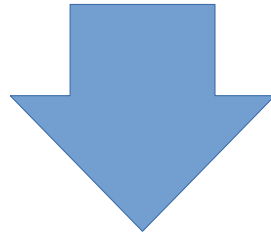
2m:1.75倍 5m:1.69倍 10m:1.45倍 20m:1.11倍

通路幅が狭いほど高速化率が高くなる

通路幅が狭いほど壁粒子の密度が高くなり
解析中の演算回数が削減できるためであると考えられる

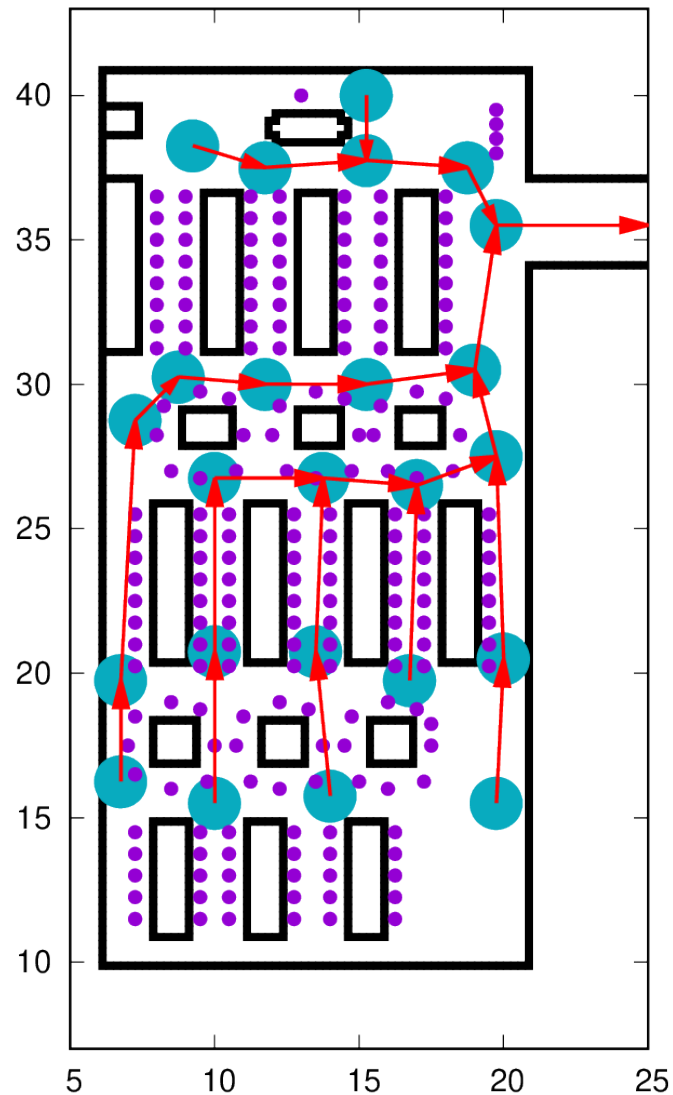
格子分割を用いた計算回数削減

通路を再現した配置
計算回数を○%削減
シミュレーション時間を最大1.7倍高速化



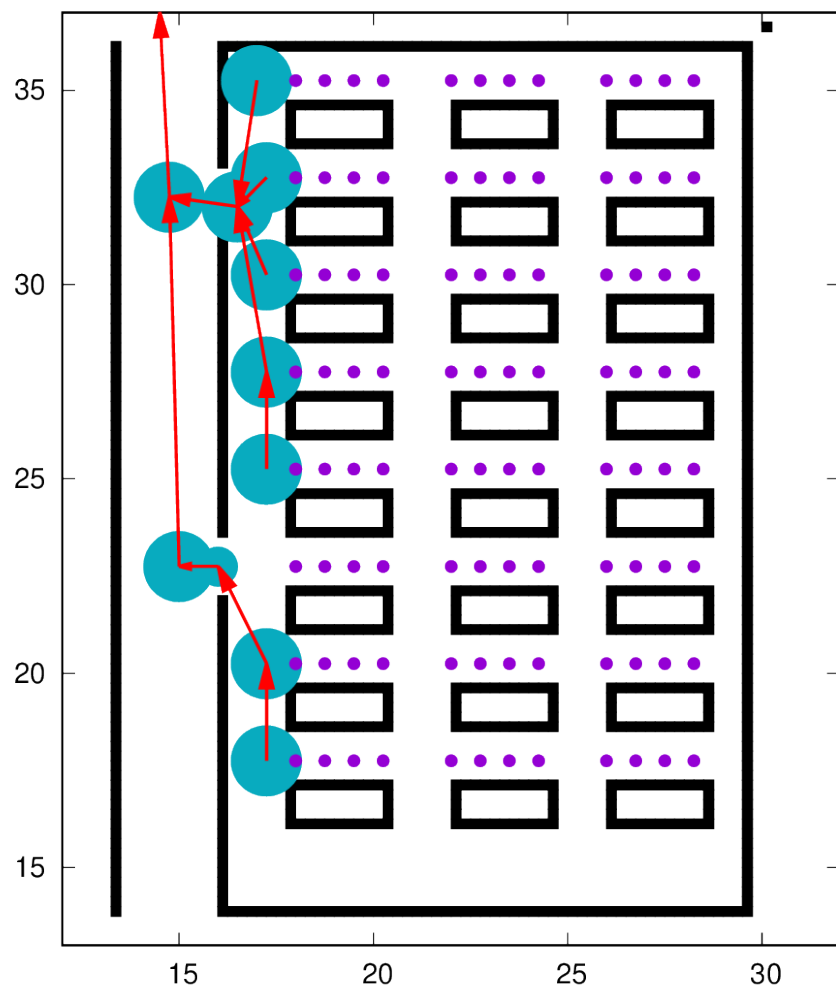
実問題に対しても有効であるか明らかにする
本評価では演習室と教室を再現した配置を用いて
避難シミュレーションを行う

演習室の配置



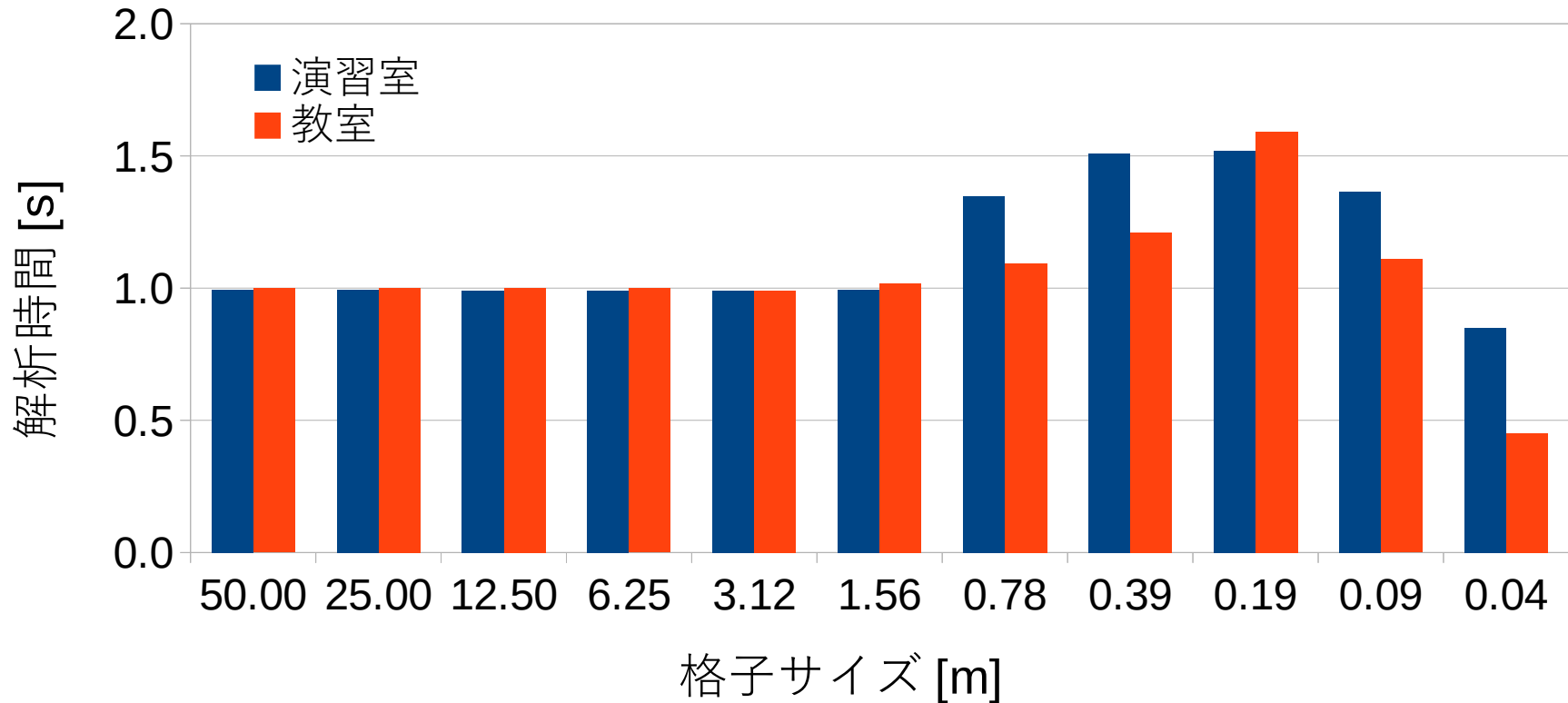
解析領域	50m×50m
経由地	26個
エージェント数	204人
壁粒子数	1454個

教室の配置



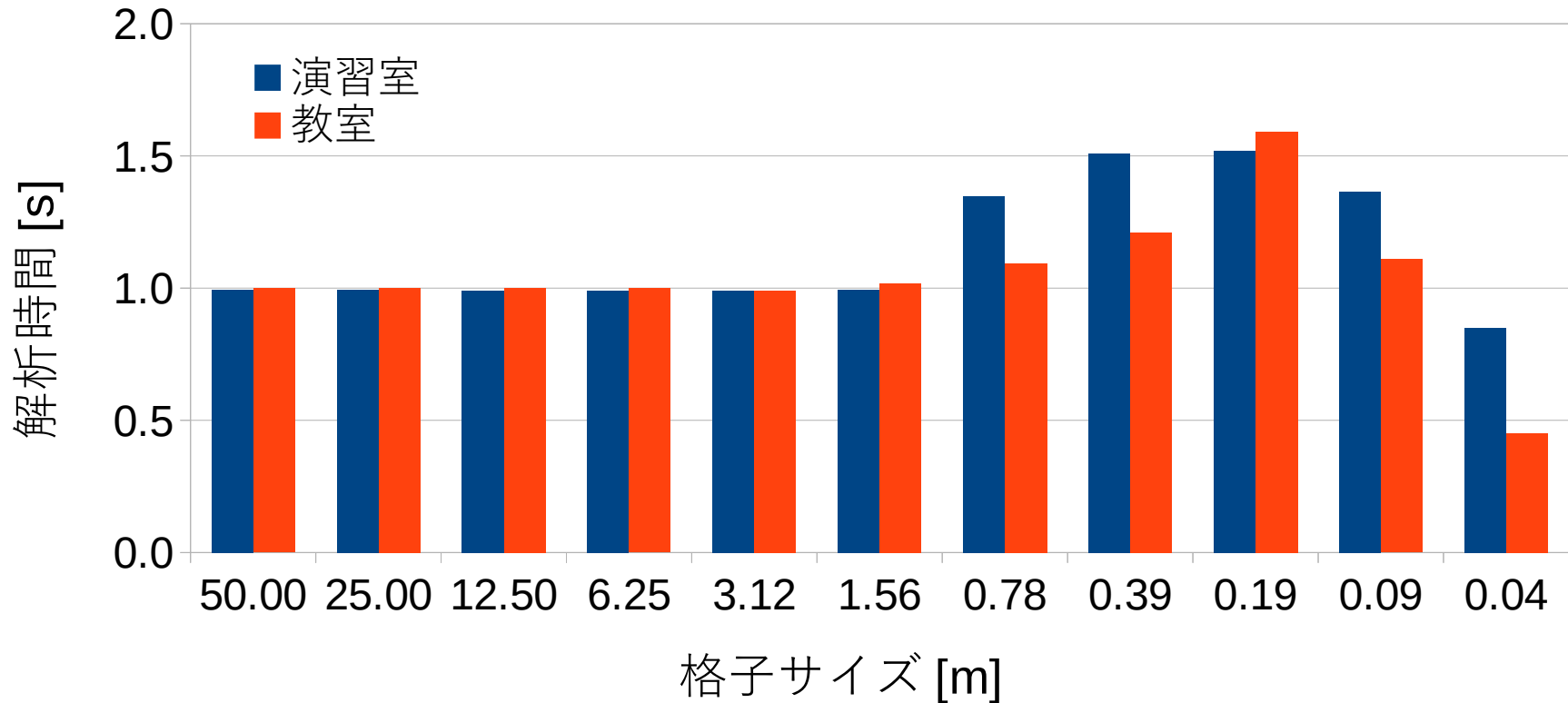
解析領域	50m×50m
経由地	12個
エージェント数	96人
壁粒子数	1037個

実問題における高速化率



$$\text{高速化率[倍]} = \frac{\text{格子サイズ0の実行時間[s]}}{\text{格子サイズnの実行時間[s]}}$$

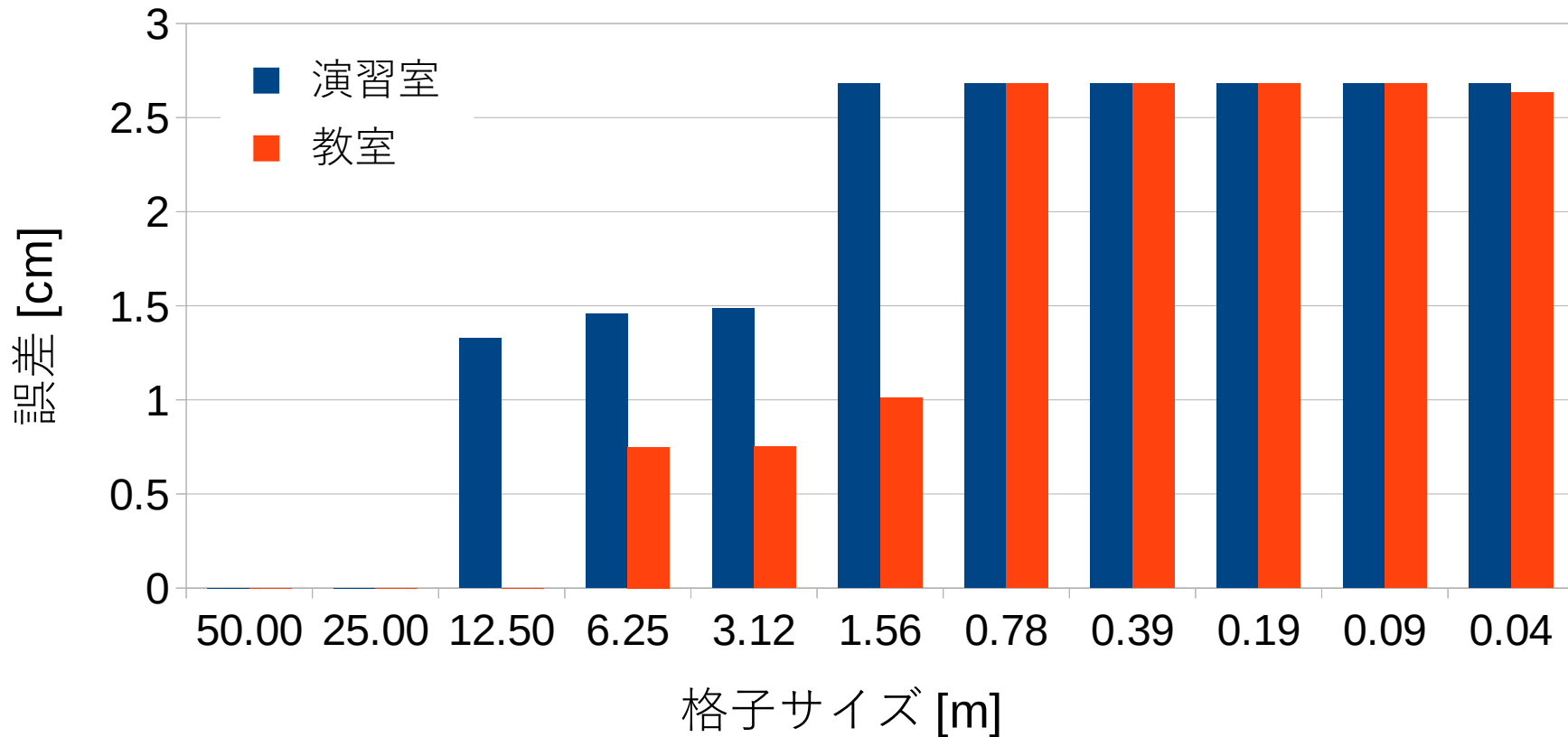
実問題における高速化率



提案手法の高速化率

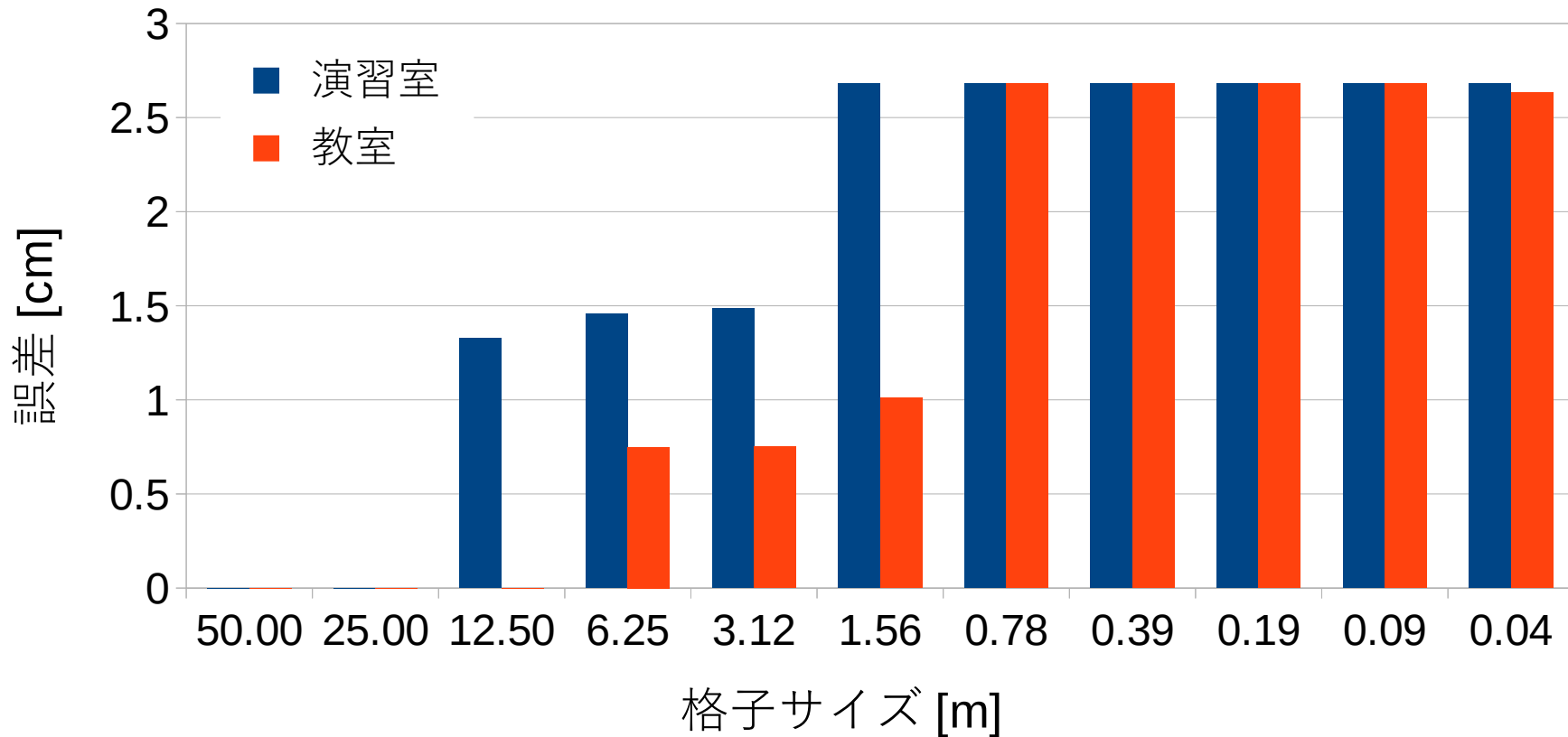
演習室の配置で最大1.50倍, 教室の配置で1.55倍
→ 実問題に対しても解析時間の短縮が可能

実問題における解析精度



誤差の算出方法を示す！

実問題における解析精度




提案手法の誤差

演習室と教室の配置: 最大2.5cm

解析領域50m×50mに対して小さな値

→許容できる範囲の誤差で高速化できる



提案手法

格子分割を用いた進行方向の計算回数削減
計算回数を○%削減
シミュレーション時間を○倍高速化
生じる誤差が○cm
→人流シミュレーションに対して有効

解析中のエージェント間距離の計算回数を削減
→さらなる高速化が見込める



エージェント間距離の計算回数削減

エージェント間距離の計算回数削減による

エージェント間距離の計算回数[10^{10} 回]

括弧内の数字は削減率

人数	パターン					
	1(既存)	2	3	4	5	6
3,000	5.1	3.9 (24%)	4.0 (23%)	4.4 (15%)	4.1 (21%)	4.4 (15%)
5,000	14.4	10.9 (24%)	11.1 (23%)	12.2 (15%)	11.4 (21%)	12.2 (15%)
7,500	33.1	25.2 (24%)	25.8 (22%)	28.3 (15%)	26.7 (20%)	28.3 (15%)



パターンごとの実行時間[s]

人数	パターン					
	1(既存)	2	3	4	5	6
3,000	2,636	2,123	2,140	2,307	2,184	2,292
5,000	7,435	5,941	6,016	6,463	6,162	6,453
7,500	1,7198	13,730	13,985	15,048	14,931	15,036

パターンごとの実行時間[s]

人数	パターン					
	1(既存)	2	3	4	5	6
3,000	2,636	2,123	2,140	2,307	2,184	2,292
5,000	7,435	5,941	6,016	6,463	6,162	6,453
7,500	1,7198	13,730	13,985	15,048	14,931	15,036

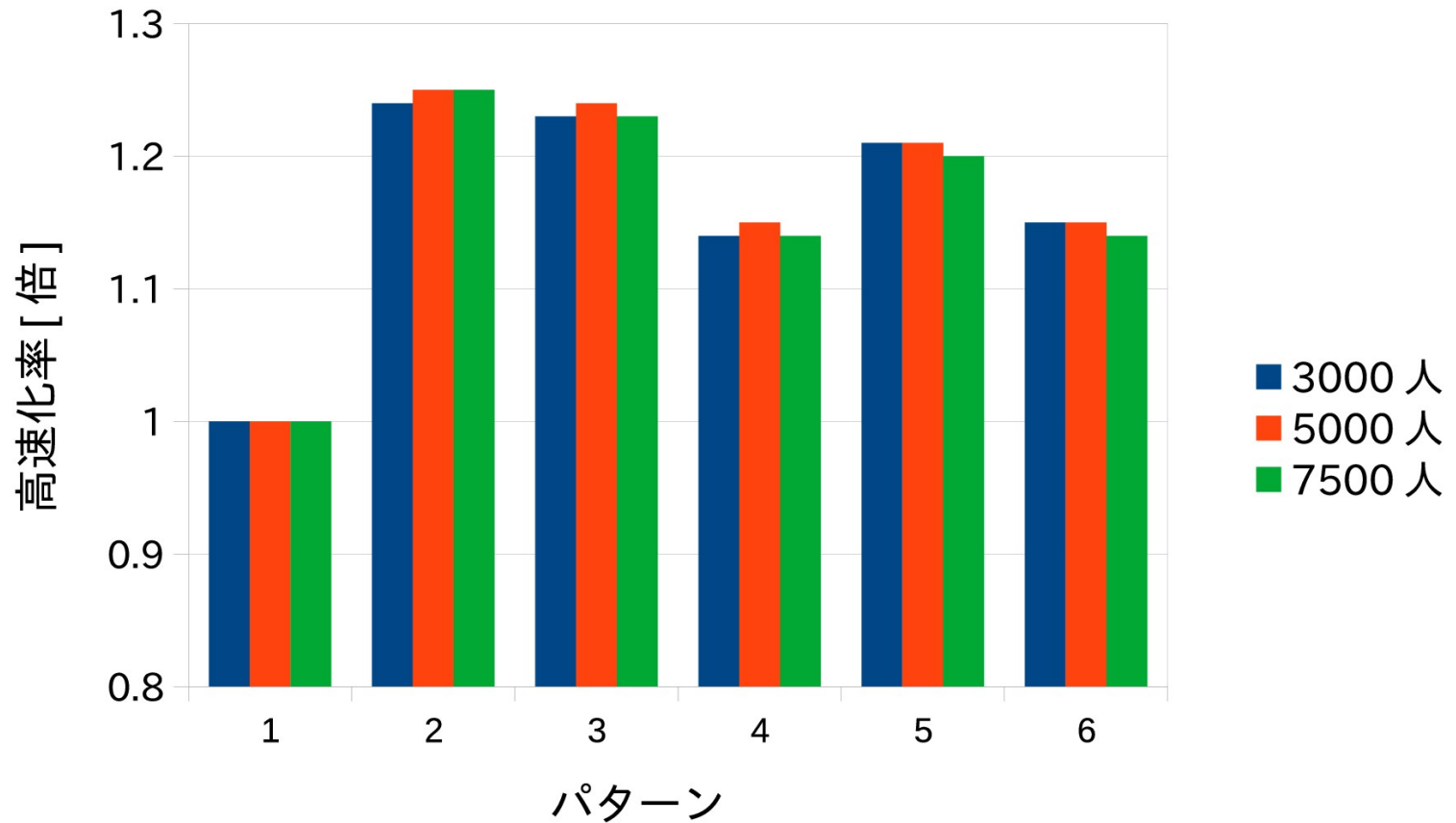
一番高速な手法


パターン1(進行方向ベクトル e を用いた手法)

既存手法と同精度で高速な手法


パターン2(進行方向ベクトル e と視野座標を用いた手法)

パターンごとの高速化率[倍]





実問題に対する実行時間

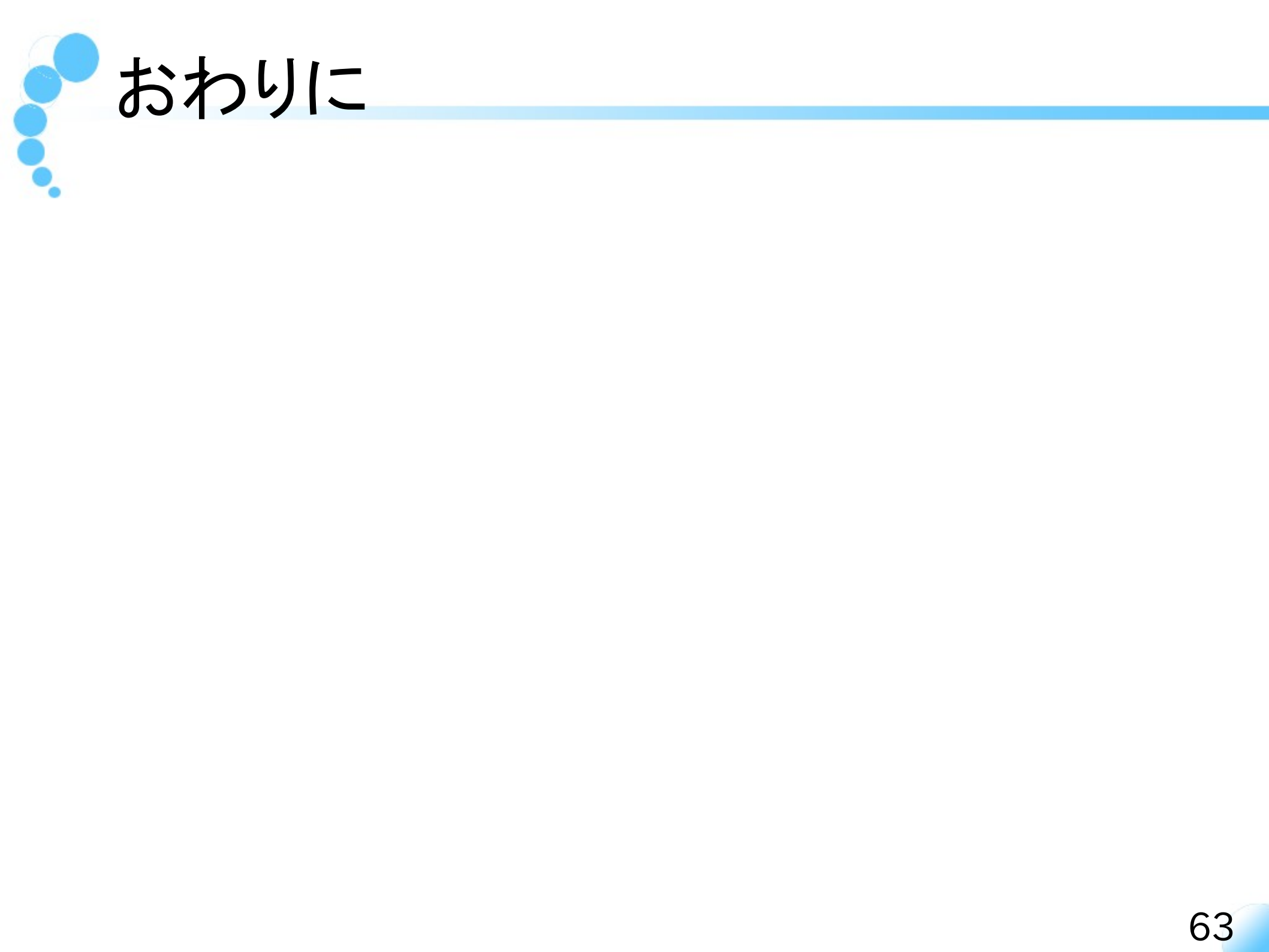


おわりに

SFM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計算中の演算回数を削減

提案手法

格子分割を用いた進行方向計算回数削減で最大1.9倍
エージェント間距離の計算回数削減で最大1.3倍
→提案手法よりも高速に解析できることを確認



おわりに

