

2023年度 卒業論文

3次元空間における波動方程式による  
追跡行動アルゴリズムに関する研究

指導教員：渡辺 大地 教授

メディア学部 ゲームサイエンスプロジェクト

学籍番号 M0A20110

熊谷 樹

2024年2月

2023年度 卒業論文概要				
論文題目 3次元空間における波動方程式による 追跡行動アルゴリズムに関する研究				
メディア学部 学籍番号：M0A20110	氏名	熊谷 樹	指導 教員	渡辺 大地 教授
キーワード	3次元、波動方程式、追跡行動、ゲーム、経路探索			
<p>近年のコンピューターゲームはオープンワールドゲームなどの普及により、マップの複雑化や大規模化が進んでいる。ゲーム AI の重要な技術として、経路探索や追跡がある。近年のコンピューターゲームでは、プレイヤーなどによる動的なマップ変更が増え、経路探索、追跡アルゴリズムも動的な対応が必要になる場面が増えている。本研究では、波動方程式の波伝搬作用を利用した追跡アルゴリズムを2次元空間から3次元空間へと拡張することを目的とする。このアルゴリズムでは、事前の準備が空間中の各格子点において波の侵入の可否のみでの設定だけでよく、動的なマップ変更への対応にも追加の処理を必要としない。また、波動方程式を利用したシミュレーションでは、波の持つ反射や回折などの現象も実現できるため、複雑なマップでも波がマップ全域に伝搬する。そのため、複雑で大規模なマップや、マップが動的に変化する場合においても利用が可能である。本手法は、マップを格子点で区切り、波動方程式によって各格子点に仮想的に波を伝搬させる。追跡対象から発せられた波が追跡エージェントに到達し、エージェントが波が伝搬してきた方向へ移動することによって追跡を実現する。また、本研究では、空間中に波の伝搬速度を変更したエリアを個別に設定することで、追跡経路の誘導を実現する手法を提案する。検証結果として、3次元空間へと手法を拡張し追跡を実現することができた。また、波の伝搬速度変更では、追跡経路の誘導を実現することが可能であった。</p>				

# 目次

第1章	はじめに	1
1.1	研究背景と目的	1
1.2	論文構成	3
第2章	提案手法	4
2.1	波の伝搬について	4
2.2	追跡アルゴリズム	6
2.3	波の伝搬速度変更エリアについて	9
第3章	検証	10
3.1	追跡検証に利用したマップ	10
3.2	追跡結果	10
3.3	動的なマップ変更への対応	14
3.4	波伝搬速度変更エリアを設定	16
3.5	考察	18
第4章	まとめ	19
	謝辞	20
	参考文献	22

# 目 次

2.1	発生直後の波 . . . . .	7
2.2	波が伝搬する様子 . . . . .	7
2.3	発生直後の波 . . . . .	8
2.4	波が伝搬する様子 . . . . .	8
3.1	マップ上からの図 . . . . .	11
3.2	マップ正面からの図 . . . . .	11
3.3	追跡後上からの図 . . . . .	11
3.4	追跡後正面からの図 . . . . .	11
3.5	移動対象の追跡初期状態、上からの図 . . . . .	12
3.6	移動対象の追跡初期状態、正面からの図 . . . . .	12
3.7	移動対象の追跡途中 1、上からの図 . . . . .	13
3.8	移動対象の追跡途中 1、正面からの図 . . . . .	13
3.9	移動対象の追跡途中 2、上からの図 . . . . .	13
3.10	移動対象の追跡途中 2、正面からの図 . . . . .	13
3.11	移動対象の追跡終了、上からの図 . . . . .	14
3.12	移動対象の追跡終了、正面からの図 . . . . .	14
3.13	動的マップ変更前、上からの図 . . . . .	15
3.14	動的マップ変更前、正面からの図 . . . . .	15
3.15	動的マップ変更直後、上からの図 . . . . .	15
3.16	動的マップ変更直後、正面からの図 . . . . .	15
3.17	動的マップ変更後追跡終了後、上からの図 . . . . .	16
3.18	動的マップ変更後追跡終了後、正面からの図 . . . . .	16
3.19	伝搬速度変更エリア上からの図 . . . . .	17
3.20	伝搬速度変更エリア正面からの図 . . . . .	17
3.21	伝搬速度変更エリア設定後、上からの図 . . . . .	17

3.22	伝搬速度変更エリア設定後、正面からの図 . . . . .	17
------	-------------------------------	----

# 第 1 章

## はじめに

### 1.1 研究背景と目的

近年のコンピューターゲームは Grand Theft Auto V[1] やゼルダの伝説 ブレス オブ ザ ワイルド [2] といった広大なマップをキャラクターが自由に移動できるオープンワールドゲームの普及により、マップの複雑化や大規模化が進んでいる。これにより、ゲーム AI 技術の重要性も高まっている。三宅 [3][4] は、ゲーム AI 技術は大きく、「キャラクター AI」、「メタ AI」、「ナビゲーション AI」に分類することができることを述べている。このうちキャラクター AI やナビゲーション AI の実現に重要となる技術が経路探索である。経路探索アルゴリズムはコンピューターゲームのみならず、カーナビゲーションシステム [5] や災害時の避難経路シミュレーション [6] などにも用いられる重要な理論である。

近年のコンピューターゲームでは、プレイヤーなどによる動的なマップ変更が増え、経路探索、追跡アルゴリズムも動的な対応が必要になる場面が増えている。MINECRAFT[7] では、プレイヤーがフィールド上で自由にオブジェクトの設置や削除を行うことができる。敵エージェントは動的に変化するフィールド上で追跡を行う必要がある。コンピューターゲームではインタラクティブ性を重視するため、経路探索アルゴリズムが動的なマップ変更へ対応することが非常に重要であるといえる。

エージェントの行動を制御する方法として、「Killzone」[8] というゲームではマップ上に等間隔にポイントを敷き詰め、そのポイントを分析・評価しエージェントを移動する手法を用いている。他にも、桑谷ら [9] や佐藤ら [10] はシューティングゲームにおいて敵の弾をよけるルートを経路探索を用いて求める手法を発表している。これらに共通しているのはエージェントを目的箇所まで移動することである。前述の Killzone の例は、あらかじめ各ポイントに様々な情報を蓄積しなければならないため、マップが広大になるにつれ蓄積する情報も膨大なものになる。経路探索を

用いる手法では、マップが複雑化するほど処理が重くなる。必要領域すべてを探索せずに一定まで探索し、その時点での最適解を求める手法もある。しかし、一定までしか探索しないため精度が落ちてしまい、場合により望む結果が出ない場合がある。

コンピューターゲームで利用される代表的な経路探索アルゴリズムとして、Dijkstra[11] が提唱したダイクストラ法や Peter ら [12] が提唱した A\*アルゴリズムがある。これらのアルゴリズムは、エージェントから目的地までのコストを算出し、コストマップ [13] を生成する。そして、コストマップを基に最適な経路を導き出す。しかし、マップ全体のコストマップを参照するため処理が重く、マップの変更や目的箇所の移動によりコストマップの再計算が必要になる。また、ナビゲーションメッシュ [14] という手法も存在する。この手法では、経路の算出にゲーム内の地形やオブジェクトを表すポリゴンメッシュ [15] を用いる。しかし、経路探索に利用するポリゴンメッシュはあらかじめ計算している必要があり、マップの変更が発生するとポリゴンメッシュの再計算が必要になる。

Stenz[16] は動的な変化に対し、関連部分のみを更新するアルゴリズムとして、D\*アルゴリズムを提唱した。D\*アルゴリズムはマップ変化によって最短経路がふさがった場合でも効率的に再探索が行えるアルゴリズムである。しかし、新たな最短経路が出現した場合は効率的な探索を行うことができない。

そこで本論文では、渡辺 [17] が発表した波動方程式の波伝搬作用を利用した追跡アルゴリズムを2次元空間から3次元空間へと拡張することを目的とする。このアルゴリズムでは、事前の準備が空間中の各格子点において波の侵入の可不可のみの設定だけでよく、動的なマップ変更への対応にも追加の処理を必要としない。また、波動方程式を利用したシミュレーションでは、波の持つ反射や回折などの現象も実現できるため、複雑なマップでも波が伝搬する。そのため、複雑で大規模なマップや、マップが動的に変化する場合においても利用が可能である。本手法は、マップを格子点で区切り、波動方程式によって各格子点に仮想的に波を伝搬させる。追跡対象から発せられた波が追跡エージェントに到達し、エージェントが波が伝搬してきた方向へ移動することによって追跡を実現する。

また、本研究では、空間中に波の伝搬速度を変更したエリアを個別に設定することで、追跡経路の誘導を実現する手法を提案する。

本研究では、3次元空間へと手法を拡張することができた。障害物のある空間において、追跡を実現することが可能であった。また、追跡対象エージェントが移動している場合やマップの構造が動的に変化した場合においてもデータの再構築を行わずに追跡を実現することができた。さらに、事前に波の伝搬速度を変更するエリアを設定することで、追跡経路の誘導を実現することが可能であった。

## 1.2 論文構成

本論文は全4章からなる。第1章は研究背景と目的について述べる。第2章は研究手法について述べる。第3章では、検証内容とその結果について述べる。最後の第4章ではまとめと考察を述べる。



# 第 2 章

## 提案手法

### 2.1 波の伝搬について

波の計算には 3 次元波動方程式を利用する。関数  $w = w(x, y, z, t)$  であらわされる空間において、3次元波動方程式 [18][19] は以下の式で表すことができる。

$$\frac{\partial^2 w}{\partial t^2} = c^2 \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) - \mu \frac{\partial w}{\partial t} \quad (2.1)$$

ここで  $c$  は波の伝搬速度であり、 $\mu$  は粘性による減衰係数である。本手法では減衰減少を表現する必要はないため簡略化し、 $\mu = 0$  とする。

$$\frac{\partial^2 w}{\partial t^2} = c^2 \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \quad (2.2)$$

次に 3 次元波動方程式を離散化し、プログラムで実装できる形式に改める。格子状に並んでいる空間  $w_{i,j,k}$  に対して、格子点間の距離を  $\Delta h$  とする。また、 $w$  の時刻  $k$  における値を  $w_{i,j,k}^k$  と表し、 $w_{i,j,k}^{k+1}$  と  $w_{i,j,k}^k$  の間の時間差を  $\Delta t$  とする。式 (2.2) の左辺から考える。偏微分の定義から

$$\frac{\partial w}{\partial t} = \lim_{\Delta t \rightarrow +\infty} \frac{w(x, y, z, t + \Delta t) - w(x, y, z, t)}{\Delta t} \quad (2.3)$$

と表すことができる。また、 $w(x, y, z, t)$  が  $t$  で 2 階偏微分可能と仮定すると、 $w(x, y, z, t)$  の変化量は、 $t$  の正方向と負方向で同一値とみなせることになる。したがって

$$\frac{\partial w}{\partial t} = \lim_{\Delta t \rightarrow +\infty} \frac{w(x, y, z, t + \Delta t) - w(x, y, z, t)}{\Delta t} = \lim_{\Delta t \rightarrow +\infty} \frac{w(x, y, z, t) - w(x, y, z, t - \Delta t)}{\Delta t} \quad (2.4)$$

も正しいということになる。 $\Delta t$  を十分小さな値と考え、

$$\frac{\partial w}{\partial t} \approx \frac{w(x, y, z, t) - w(x, y, z, t - \Delta t)}{\Delta t} \quad (2.5)$$

とする。 $(\Delta t)^2$  を「 $\Delta t^2$ 」と表記するものとしたとき、

$$\begin{aligned} \frac{\partial^2 w}{\partial t^2} &\approx \frac{\frac{w(x, y, z, t + \Delta t) - w(x, y, z, t)}{\Delta t} - \frac{w(x, y, z, t) - w(x, y, z, t - \Delta t)}{\Delta t}}{\Delta t} \\ &= \frac{w(x, y, z, t + \Delta t) - 2w(x, y, z, t) + w(x, y, z, t - \Delta t)}{\Delta t^2} \end{aligned} \quad (2.6)$$

という式で  $\frac{\partial^2 w}{\partial t^2}$  は近似することができる。

次に式 (2.2) の右辺を考える。式 (2.6) と同じように考えると、

$$\frac{\partial^2 w}{\partial x^2} \approx \frac{w(x + \Delta h, y, z, t) - 2w(x, y, z, t) + w(x - \Delta h, y, z, t)}{\Delta h^2} \quad (2.7)$$

$$\frac{\partial^2 w}{\partial y^2} \approx \frac{w(x, y + \Delta h, z, t) - 2w(x, y, z, t) + w(x, y - \Delta h, z, t)}{\Delta h^2} \quad (2.8)$$

$$\frac{\partial^2 w}{\partial z^2} \approx \frac{w(x, y, z + \Delta h, t) - 2w(x, y, z, t) + w(x, y, z - \Delta h, t)}{\Delta h^2} \quad (2.9)$$

となる。結果として、

$$\begin{aligned} w^*(x, y, z, t) &= w(x + \Delta h, y, z, t) + w(x - \Delta h, y, z, t) + \\ &\quad w(x, y + \Delta h, z, t) + w(x, y - \Delta h, z, t) + \\ &\quad w(x, y, z + \Delta h, t) + w(x, y, z - \Delta h, t) - \\ &\quad 6w(x, y, z, t) \end{aligned} \quad (2.10)$$

としたときに、

$$c^2 \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \approx c^2 \left( \frac{w^*(x, y, z, t)}{\Delta h^2} \right) \quad (2.11)$$

となる。さらに、式 (2.2) に式 (2.6) と式 (2.11) を代入すると、

$$\frac{w(x, y, z, t + \Delta t) - 2w(x, y, z, t) + w(x, y, z, t - \Delta t)}{\Delta t^2} \approx c^2 \left( \frac{w^*(x, y, z, t)}{\Delta h^2} \right) \quad (2.12)$$

となる。

このとき、式 (2.12) は以下のように書き換えることができる。

$$\begin{aligned} & \frac{w_{i,j,k}^{k+1} - 2w_{i,j,k}^k + w_{i,j,k}^{k-1}}{\Delta t^2} \\ &= c^2 \left( \frac{w_{i+1,j,k}^k + w_{i-1,j,k}^k + w_{i,j+1,k}^k + w_{i,j-1,k}^k + w_{i,j,k+1}^k + w_{i,j,k-1}^k + 6w_{i,j,k}^k}{\Delta h^2} \right) \end{aligned} \quad (2.13)$$

これを  $w_{i,j,k}^{k+1}$  について解くことで、

$$\begin{aligned} w_{i,j,k}^{k+1} &= \frac{c^2 \Delta t^2}{\Delta h^2} (w_{i+1,j,k}^k + w_{i-1,j,k}^k + w_{i,j+1,k}^k + w_{i,j-1,k}^k + w_{i,j,k+1}^k + w_{i,j,k-1}^k) + \\ & \quad \left( 2 - \frac{6c^2 \Delta t^2}{\Delta h^2} \right) w_{i,j,k}^k - w_{i,j,k}^{k-1} \end{aligned} \quad (2.14)$$

を得る。

## 2.2 追跡アルゴリズム

本手法では、プレイヤーキャラクターや追跡エージェント等が移動可能な 3D マップデータを準備する。次にマップ上に格子点を配置する。各格子点はマップ内でエージェントが移動可能かどうかの情報と、波の振幅値を表す実数値の情報を持つ。

本研究における追跡アルゴリズムは以下の方針で実現する。

1. 追跡対象を発生源とする波を発生させる。
2. 波動方程式により、マップ内を波が伝搬する。
3. 追跡行動をおこなうエージェントに波が伝搬する。
4. 追跡エージェントは波が伝搬してきた方向へ向かう。

波の発生は、発生位置に最も近い格子点の振幅値に正の一定数を設定する。この場合は、追跡対象に最も近い格子点に正の一定数を設定する。発生した波は徐々にほかの格子点に対し、振幅値が波及していく。これにより波が 3次元マップ全体に広がっていく。

発生直後の波の様子を図 2.1 に、発生から時間が経過し、波が伝搬する様子を図 2.2 に示す。マップ全体に波が伝搬している様子が確認できる。

また、障害物がある状態で波が伝搬する様子を図 2.3、図 2.4 に示す。障害物がある部分では波が遮られていることがわかる。また、波が障害物を回り込んでいる様子も確認できる。

それぞれ第一波が到達している格子点を青く表示している。マップ内の格子点すべてを表示した場合非常に動作が遅くなるため、すべての格子点は表示せずに数個おきに表示している。

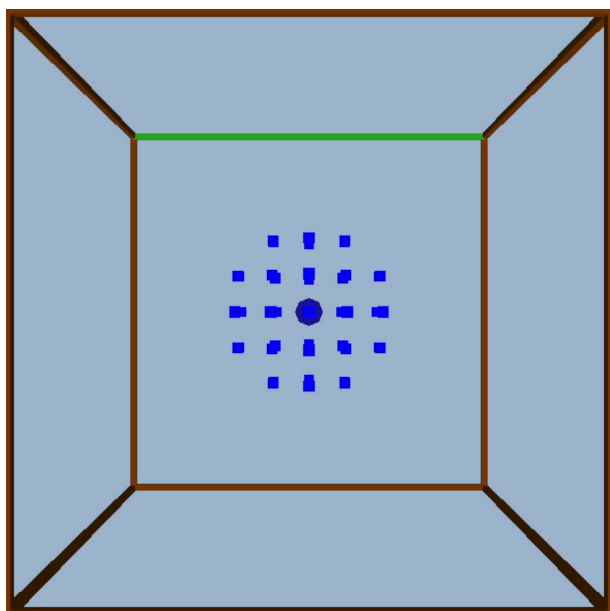


図 2.1 発生直後の波

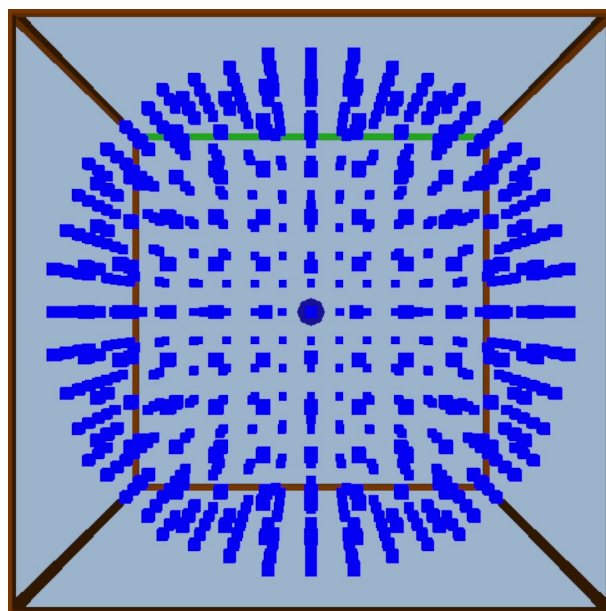


図 2.2 波が伝搬する様子

マップ内に波が広がり、エージェントから最も近い格子点の波の振幅値  $w(x, y, z, t)$  がある閾値を超えた時、エージェントに波が到達したものとする。

エージェントに波が到達したとき、エージェントは波の伝搬方向を算出する。波の伝搬方向は勾配ベクトル [17] によって求めることができる。

$$\nabla w = \left( \frac{\partial w}{\partial x}, \frac{\partial w}{\partial y}, \frac{\partial w}{\partial z} \right) \quad (2.15)$$

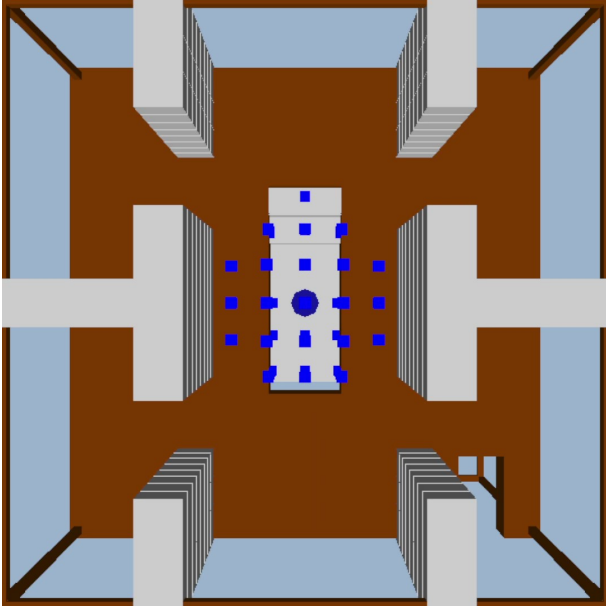


図 2.3 発生直後の波

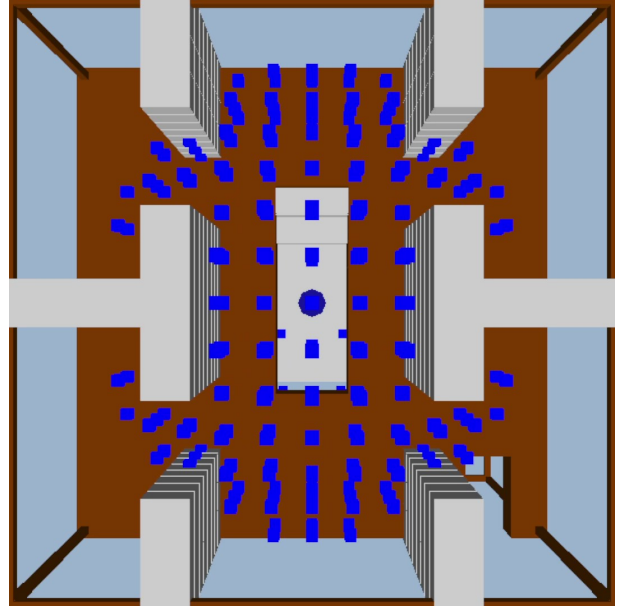


図 2.4 波が伝搬する様子

ここで、偏微分値は「正側と負側の傾きの平均値」で近似するものとし、以下のように算出する。

$$\frac{\partial}{\partial x} w_{i,j,k}^k \approx \left( \frac{w_{i,j,k}^k - w_{i-1,j,k}^k}{\Delta h} + \frac{w_{i+1,j,k}^k - w_{i,j,k}^k}{\Delta h} \right) / 2 = \frac{w_{i+1,j,k}^k - w_{i-1,j,k}^k}{2\Delta h}, \quad (2.16)$$

$$\frac{\partial}{\partial y} w_{i,j,k}^k \approx \left( \frac{w_{i,j,k}^k - w_{i,j-1,k}^k}{\Delta h} + \frac{w_{i,j+1,k}^k - w_{i,j,k}^k}{\Delta h} \right) / 2 = \frac{w_{i,j+1,k}^k - w_{i,j-1,k}^k}{2\Delta h}, \quad (2.17)$$

$$\frac{\partial}{\partial z} w_{i,j,k}^k \approx \left( \frac{w_{i,j,k}^k - w_{i,j,k-1}^k}{\Delta h} + \frac{w_{i,j,k+1}^k - w_{i,j,k}^k}{\Delta h} \right) / 2 = \frac{w_{i,j,k+1}^k - w_{i,j,k-1}^k}{2\Delta h} \quad (2.18)$$

以上から追跡アルゴリズムは、エージェントの位置における  $w(x, y, z, t)$  の絶対値が閾値以上となった時に、 $\nabla w(x, y, z, t)$  の値方向に移動することで実現できる。

第一波がエージェントに到達した後も計算を行った場合、波の上下や壁からの反射波などによって勾配ベクトルが振動してしまう。そのため、エージェントに第一波が到達し、伝搬方向を計算した時点でマップ内のすべての波の  $w$  値を 0 に初期化する。その後、追跡対象位置から再び波を発生させることで勾配ベクトルの振動を回避した。

## 2.3 波の伝搬速度変更エリアについて

現在のアルゴリズムでは波の伝搬速度がマップ全体で一定であるため、エージェントはマップ内のどの場所においても最短経路を選択し追跡が可能である。しかし、コンピューターゲームにおいては追跡エージェントが常に最短経路を選択した場合、難易度が上がってしまうため、難易度調整のために追跡経路を意図的に変更したい場合がある。

そこで、本手法ではマップ内に波の伝搬速度を変更したエリアを設定することにより上記の問題を解決した。式 (2.2),(2.14) において、 $c$  が波が空間中を進む速度を表す。それぞれの格子点の  $c$  の値を  $c_{i,j,k}$  とする。これにより、式 (2.14) は

$$\begin{aligned} w_{i,j,k}^{k+1} = & \frac{c_{i,j,k}^2 \Delta t^2}{\Delta h^2} (w_{i+1,j,k}^k + w_{i-1,j,k}^k + \\ & w_{i,j+1,k}^k + w_{i,j-1,k}^k + \\ & w_{i,j,k+1}^k + w_{i,j,k-1}^k) \\ & + \left( 2 - \frac{6c_{i,j,k}^2 \Delta t^2}{\Delta h^2} \right) w_{i,j,k}^k - w_{i,j,k}^{l-1} \end{aligned} \quad (2.19)$$

となる。波伝搬の際に各格子点の  $c$  の値を利用し、計算を行うことで波の伝搬速度が異なるエリアを設定することが可能となる。

## 第 3 章

# 検証

### 3.1 追跡検証に利用したマップ

検証用のシステムは、Fine Kernel ToolKit [20](以下 FK) 上に C# で実装した。マップは  $xyz$  空間にあり、格子点数は 1 辺 50 個、合計 125,000 個である。

図 3.1、図 3.2 は今回検証に使用したマップを示したものである。上下の階層構造になっており、それぞれの階層にいくつかの小部屋を設置している。

マップ内のエージェントについては、青い円錐型のエージェントが操作プレイヤーであり、赤い円錐型エージェントが追跡エージェントである。実行を開始すると追跡エージェントが操作プレイヤーから発せられた波を基に追跡を開始する。また、マップ内の茶色、灰色の部分は侵入不可能として設定しており、追跡エージェントは通過することができず、追跡経路でも障害物として扱われる。また、今回の検証では、操作プレイヤーエージェント、追跡エージェントどちらも重力を考慮していないため 3 次元空間内で全方向に自由に動けるものとしている。

### 3.2 追跡結果

図 3.3、図 3.4 はプレイヤーキャラクターが移動を行わずに待機している状態で追跡を行った結果を示したものである。初期設定として中央上側に追跡エージェントを配置し、右下の小部屋部分に追跡対象であるプレイヤーエージェントを配置した。追跡エージェントの移動経路はマップ内に赤く表示している。3 次元マップでも追跡が行われていることがわかる。また、追跡エージェントの移動経路から、今回のマップのように小部屋が複数あり、階層が分かれている複雑なマップでも、障害物を回避した効率の良い経路を選択し、追跡が実現できていることが確認できる。

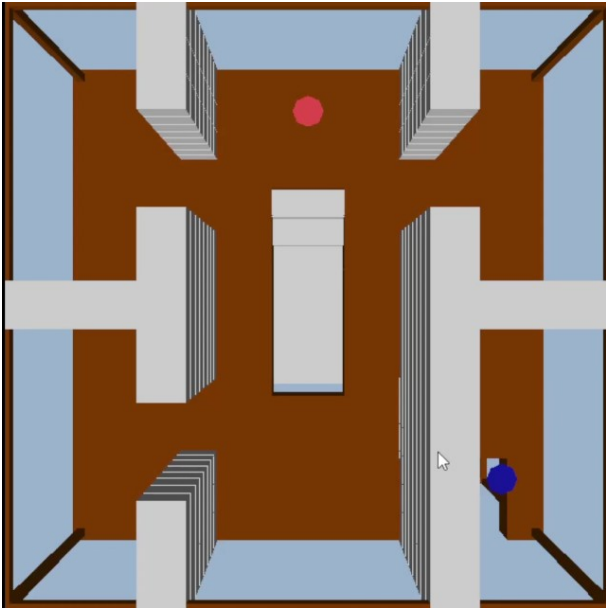


図 3.1 マップ上からの図

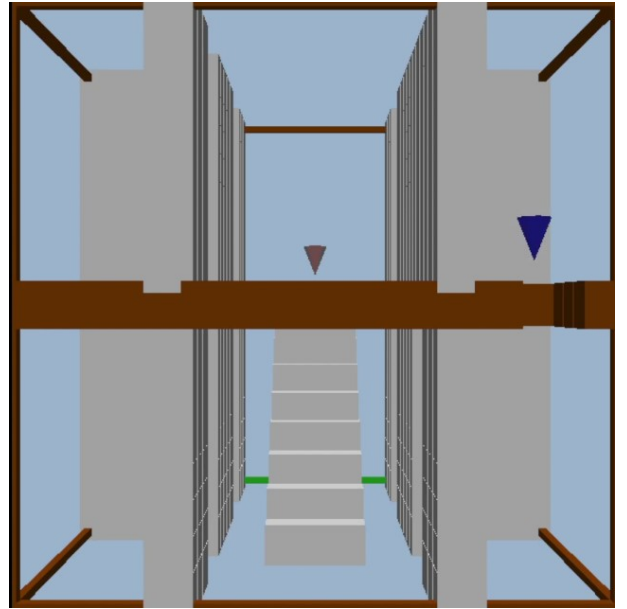


図 3.2 マップ正面からの図

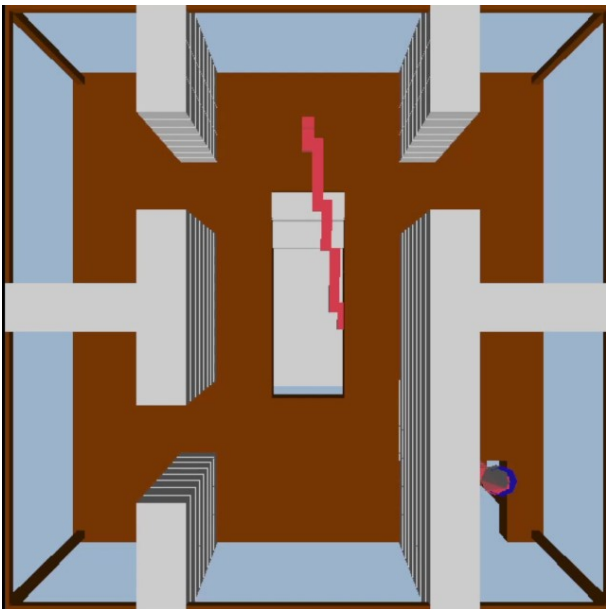


図 3.3 追跡後上からの図

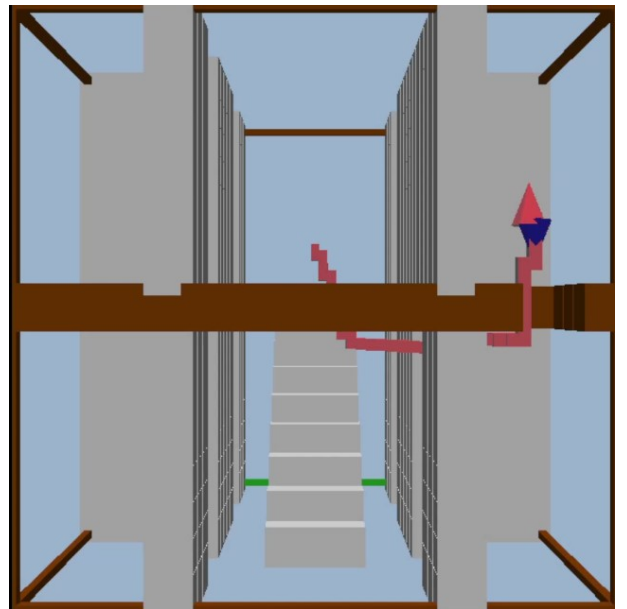


図 3.4 追跡後正面からの図

また、図 3.5、図 3.6 は移動するプレイヤーキャラクターを追跡する際の初期状態を示した図である。初期設定として右下の小部屋部分に追跡エージェントを配置し、中央下側にプレイヤーエージェントを配置した。プログラムが実行されると、プレイヤーは移動を行い追跡エージェント



トから遠ざかる。まず、プレイヤーがマップ中央部まで進んだ様子を図 3.7、図 3.8 に示す。次に、マップ中央の階段部分からマップ内一回部分へプレイヤーが下った様子を図 3.9、図 3.10 に示す。最後に、プレイヤーが追跡エージェントの初期位置まで移動し、追跡を終了した様子を図 3.11、図 3.12 に示す。これらの結果から、追跡対象が移動した際にもデータの再構築を行わずに追跡ができていることがわかる。

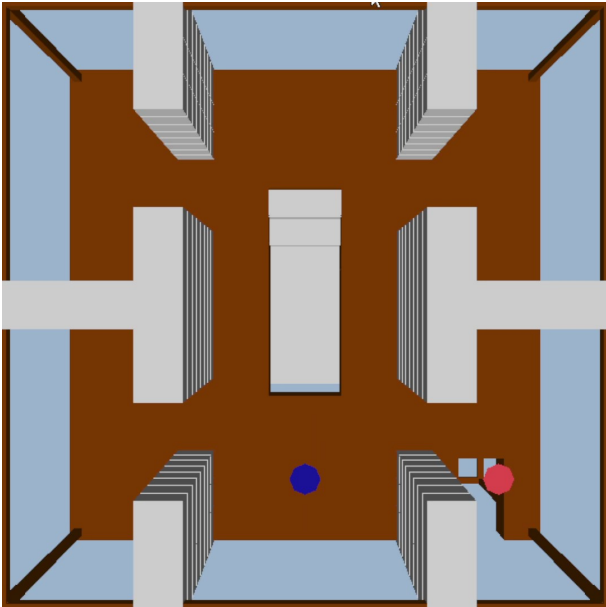


図 3.5 移動対象の追跡初期状態、上からの図

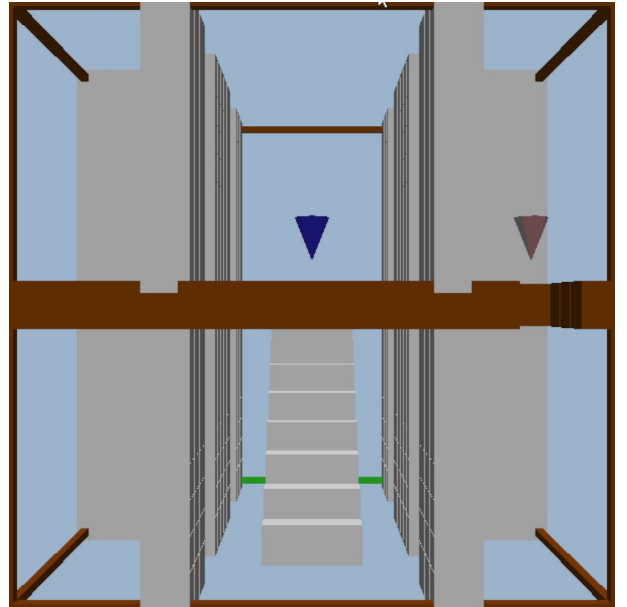


図 3.6 移動対象の追跡初期状態、正面からの図

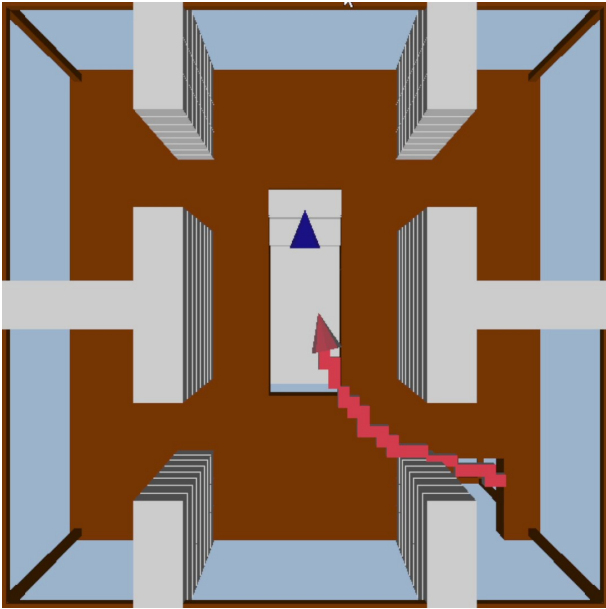


図 3.7 移動対象の追跡途中 1、上からの図

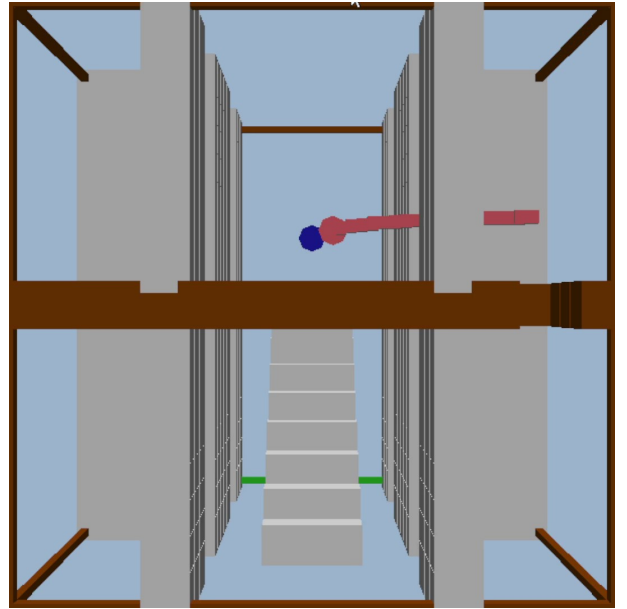


図 3.8 移動対象の追跡途中 1、正面からの図

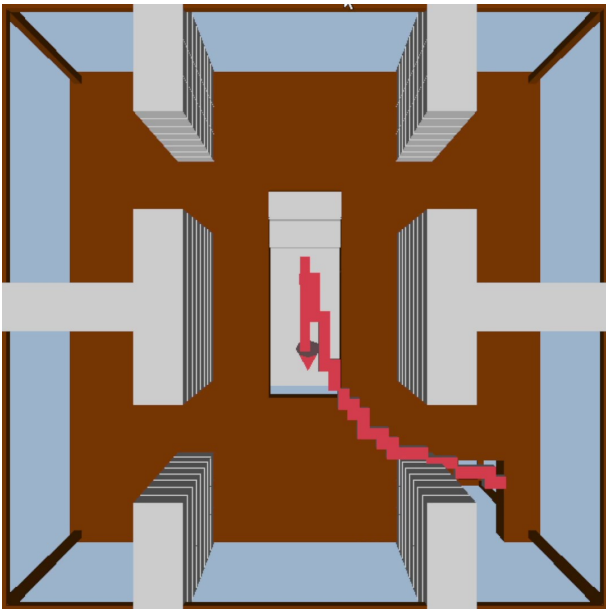


図 3.9 移動対象の追跡途中 2、上からの図

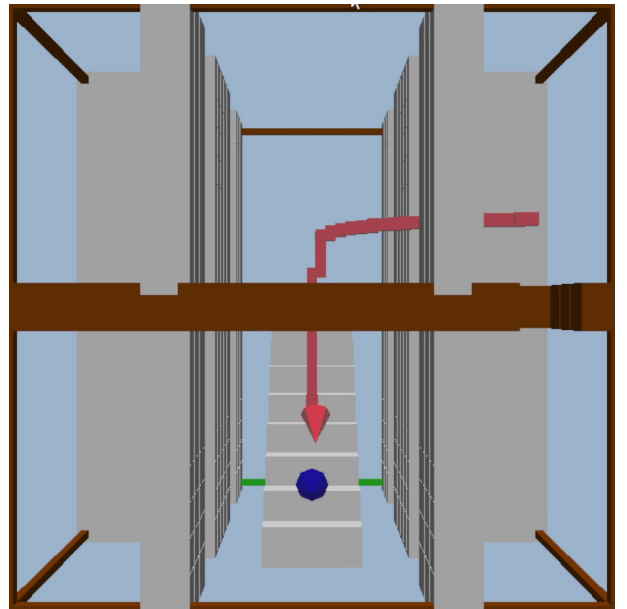


図 3.10 移動対象の追跡途中 2、正面からの図

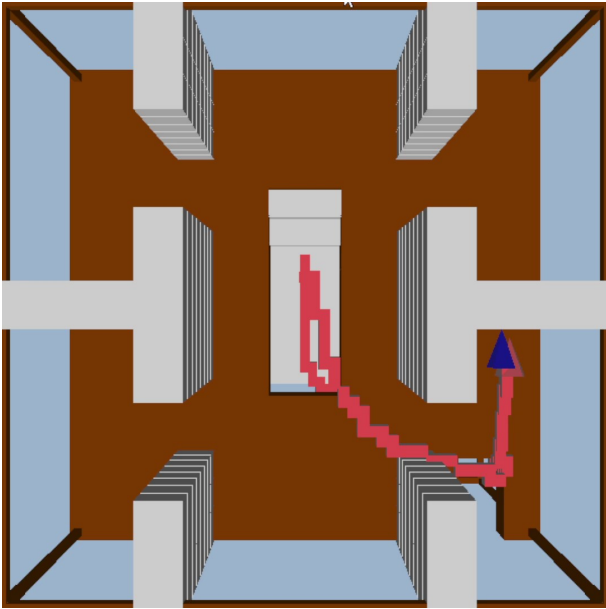


図 3.11 移動対象の追跡終了、上からの図

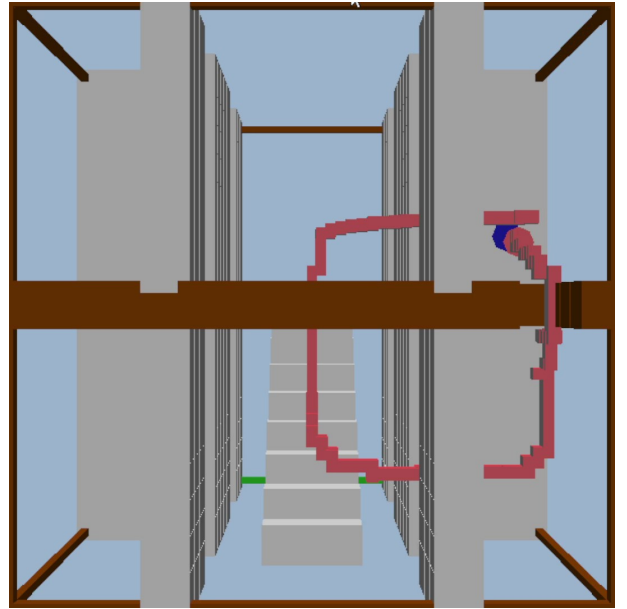


図 3.12 移動対象の追跡終了、正面からの図

### 3.3 動的なマップ変更への対応

図 3.13、図 3.14 は、動的なマップ変更を行った際の動作を検証する際に利用したマップである。図 3.1、図 3.2 と異なり、初期状態ではプレイヤーの左側の壁に空間があり通行可能となっている。追跡処理を実行中にこの空間を閉じることで追跡エージェントが経路の変更が必要な状況とする。また、今回の検証では追跡アルゴリズムには処理を追加しておらず、動的にマップを変更できる機能のみを追加している。

実際に実行中にマップを変更した状態を示しているのが図 3.15、図 3.16 である。実行直後はプレイヤーの位置に直線的に向かっているが、マップ変更しプレイヤー左側の空間が閉じた後は、効率の良い経路が変更されたため追跡エージェントが方向を変えていることがわかる。

追跡が終了した状態を示しているのが図 3.17、図 3.18 である。追跡アルゴリズムへ処理を追加せずとも動的なマップ変更への対応が可能であることが確認できる。

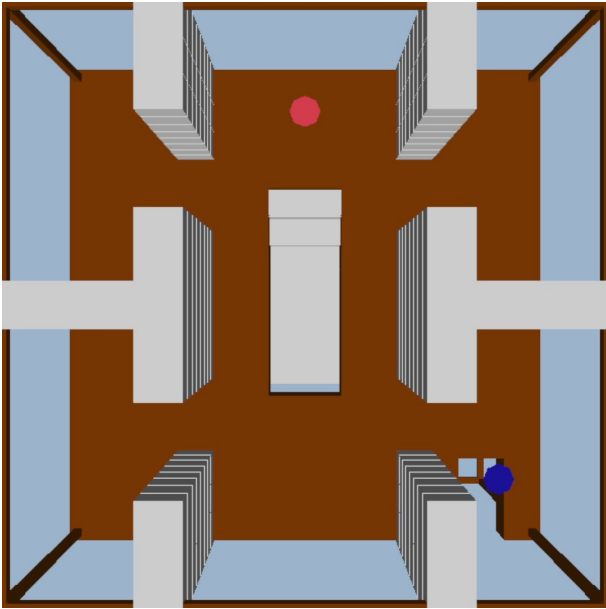


図 3.13 動的マップ変更前、上からの図

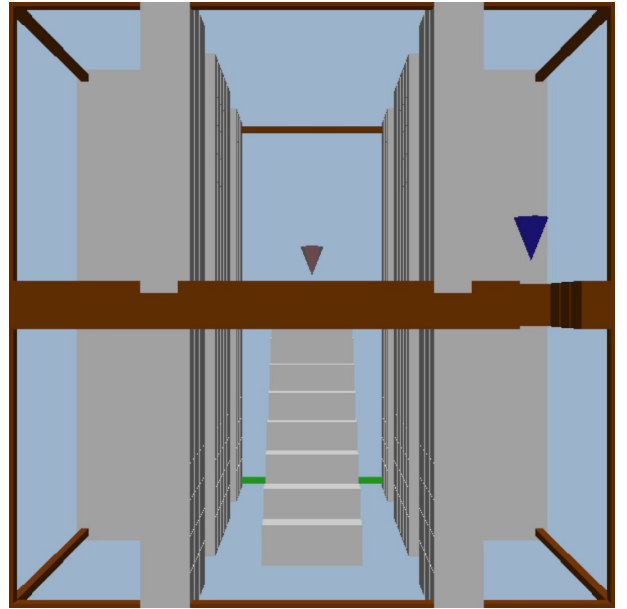


図 3.14 動的マップ変更前、正面からの図

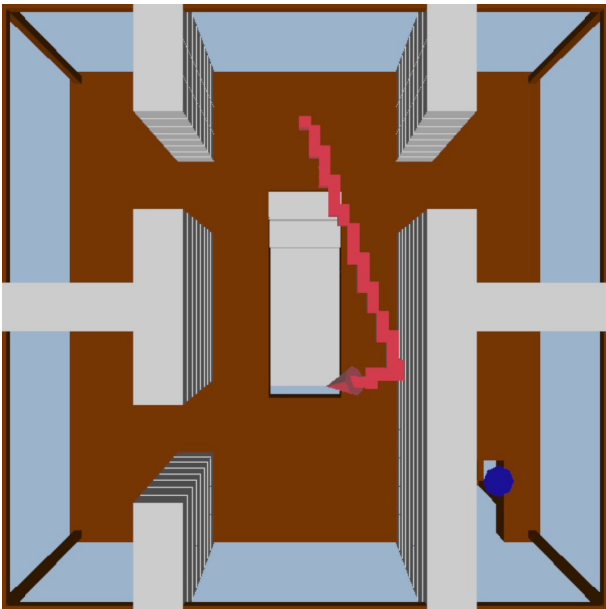


図 3.15 動的マップ変更直後、上からの図

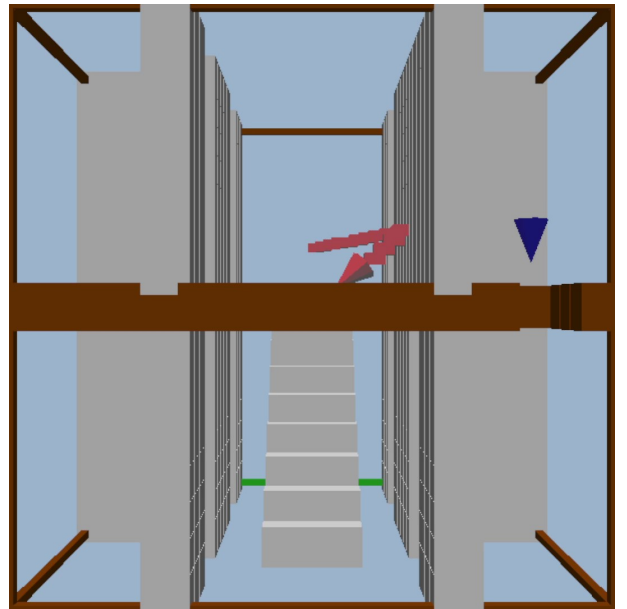


図 3.16 動的マップ変更直後、正面からの図

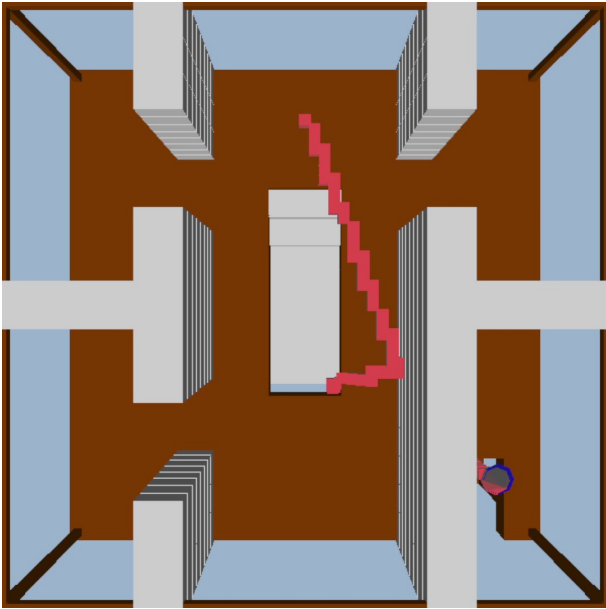


図 3.17 動的マップ変更後追跡終了後、上からの図

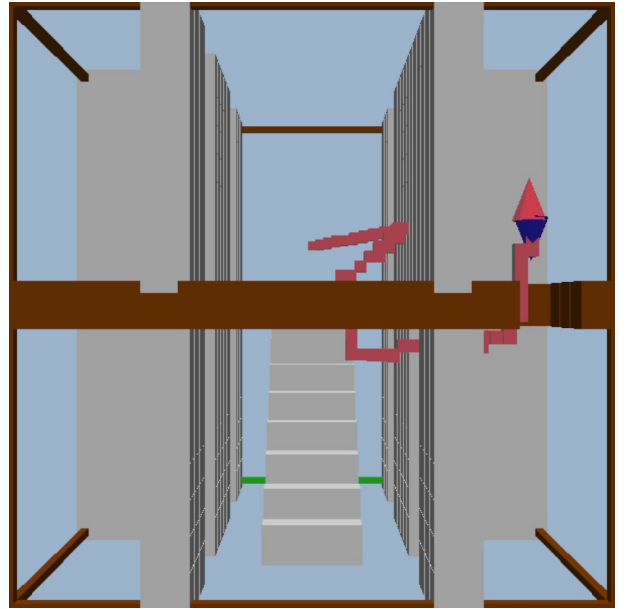


図 3.18 動的マップ変更後追跡終了後、正面からの図

### 3.4 波伝搬速度変更エリアを設定

波の伝搬速度を変更し、追跡経路を誘導する検証を行う。事前の準備として、マップ内に波の伝搬速度を変更するエリアを設定する。図 3.19、図 3.20 は今回の検証で波の伝搬速度を変更したエリアを可視化したものである。図内のマゼンタ色の空間内は波の伝搬速度を遅く設定した。したがって、マゼンタ色の空間内は各エージェントは通行可能であるが、波の伝搬が遅いため経路として選択されにくくなっている。

この状態で追跡処理を実行した様子を示したものが図 3.21、図 3.22 である。図 3.3、図 3.4 の追跡結果と比較すると、マップ中央部での経路が変わっていることが確認できる。そのため、追跡エージェントが事前に設定した伝搬速度を変更したエリアを避けて追跡経路を選択したことが確認できる。

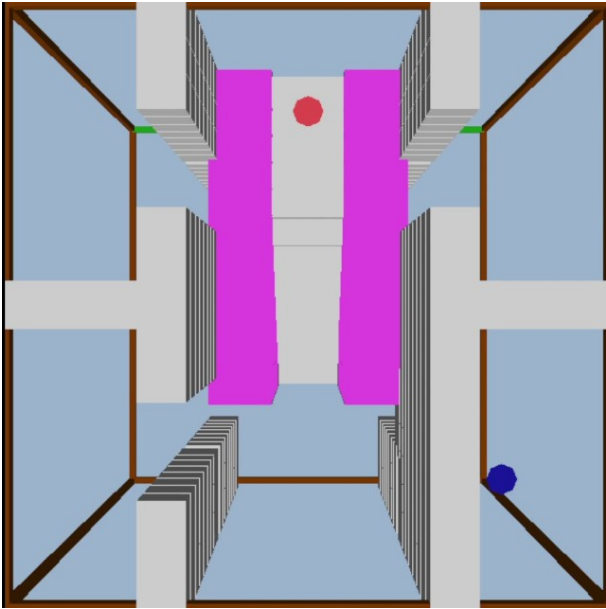


図 3.19 伝搬速度変更エリア上からの図

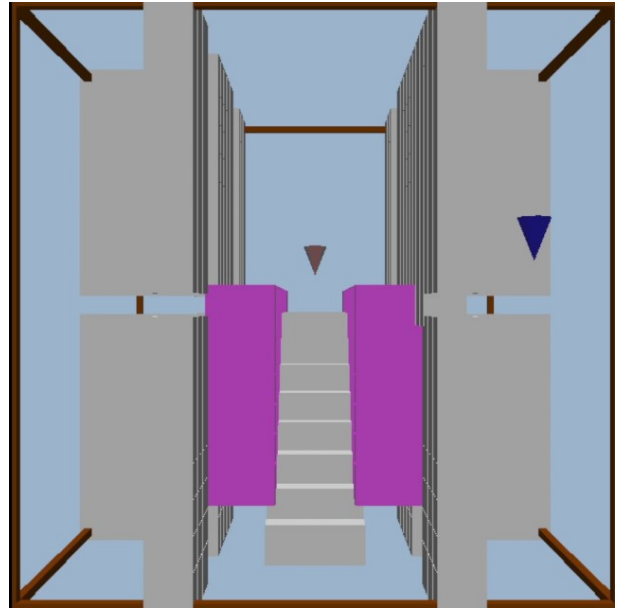


図 3.20 伝搬速度変更エリア正面からの図

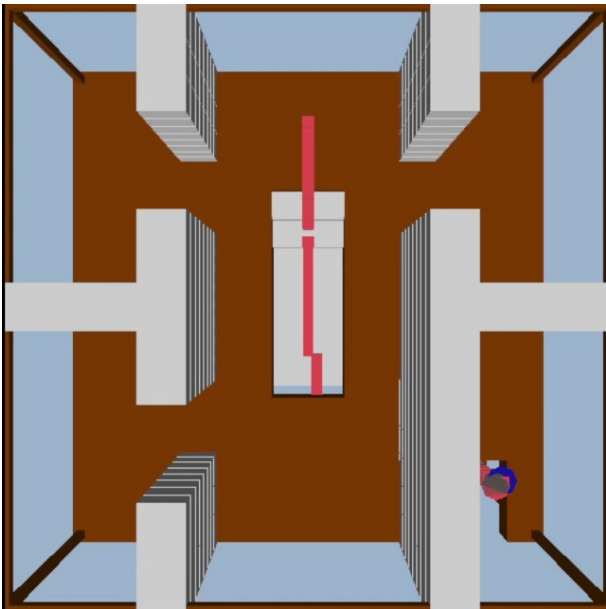


図 3.21 伝搬速度変更エリア設定後、上からの図

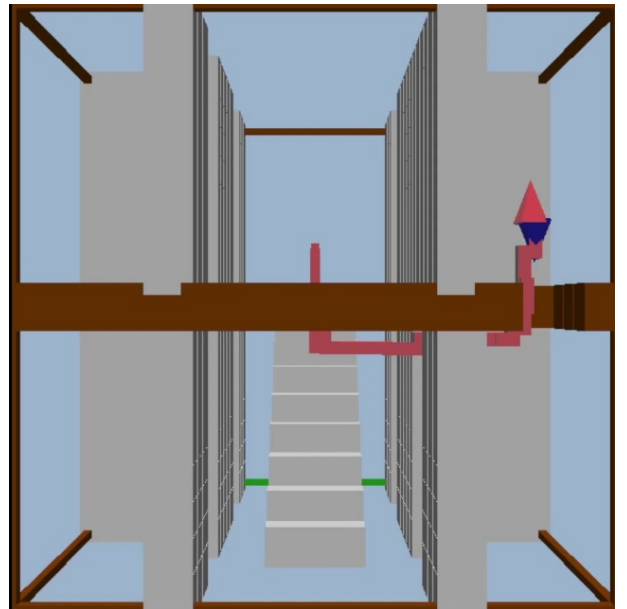


図 3.22 伝搬速度変更エリア設定後、正面からの図

### 3.5 考察

すべての検証で本手法を用いて追跡エージェントを追跡対象へと到達させることができた。追跡対象が移動する場合においても、追跡対象が移動しない場合と同様にデータの再構築を行わずに追跡が可能であった。マップが動的に変化する場合でも追跡アルゴリズムに処理を追加することなく追跡が可能であった。の伝搬速度を変更するエリアを設定することで、常に最適な経路を選択するエージェントではなく、追跡経路をある程度誘導設定することが可能であった。

今回は重力や床などを考慮せずに移動を行うよう設定をしている。追跡エリアや追跡アルゴリズムを調整することで重力があり、床を移動することも可能である。しかし、本手法が最も向いていると考えられるのは3次元空間を自由に移動できるエージェント同士の追跡である。そのため、コンピューターゲームのシチュエーションでは、空中で移動しあう航空機同士の追跡や宇宙空間での追跡に活用できるのではないかと考える。

今回のマップではそれぞれの追跡が動作していたが、細い道が入り組んだ迷路のようなマップでは本手法は効果的ではないと考えられる。理由としては、細い道では波が両側の壁に反射を繰り返して、勾配ベクトルが乱れてしまうためである。また、波の伝搬速度が十分に早くない場合、追跡エージェントの追跡方向の更新回数が減ってしまい、追跡時に無駄な経路をとってしまう現象が確認できた。入り組んだマップでは追跡対象から追跡エージェントまで波が到達するのに時間がかかってしまうことも考えられる。波の伝搬速度を十分早く設定することが追跡の安定に重要ではないかと考える。

## 第 4 章

# まとめ

本研究では、渡辺が発表した手法を基に、3次元空間内における波動方程式を用いた追跡行動アルゴリズムを提案した。提案手法では、3次元波動方程式を利用し2次元空間での追跡アルゴリズムから3次元空間へとアルゴリズムを拡張することができた。

検証マップでの追跡では、障害物のあるマップでも追跡アルゴリズムが適切に動作していることが確認できた。また、追跡対象が移動した場合でもデータの再構築や追加の処理を行うことなく追跡が可能であった。さらに、追跡処理を実行中にマップが動的に変化した場合でも追跡アルゴリズムには処理を追加せずにプレイヤーを追跡し続けることが可能であった。

本研究では3次元への拡張のほかに、波の伝搬速度を変更したエリアを設定し追跡経路の誘導を行った。検証マップでは、伝搬速度を変更したエリアを設定することで、追跡経路が変更されたことが確認できた。

今後の展望として、検証マップでは、十分リアルタイムでの実行が可能であったが、実行速度の計測は行っていないため、計測を行い既存手法と比較したいと考えている。また、新たに提案した波の伝搬速度を変更することによって追跡経路を誘導する機能については、有効に活用できる状況の考察を行っていきたい。問題点としては、マップサイズを大きくした際やマップ内の格子点間の間隔を狭くした際にマップ内の格子点数が多くなるが、非常に多くのメモリを必要とする恐れがあることが挙げられる。実際にコンピューターゲームなどで本研究を応用する際には、格子点を配置するエリアを柔軟に設定するとよいと考える。



# 謝辞

本論文を執筆するにあたり、ご指導いただきました渡辺先生、阿部先生に深く感謝いたします。渡辺先生には、2年次の先端メディア学から指導して頂いたことに感謝いたします。今回の研究テーマも先端メディアゼミナール II で学会発表を目標に研究を始めたテーマでした。テーマを考えるために、様々な論文を読んでいる中、渡辺先生の論文を読ませていただき、2次元から3次元にできるのではないかと考えたのが始まりです。3年次には無事に初めての学会である NICOGRAPH でポスター発表をすることができ、とても良い経験となりました。また、4年次の DCC 研究発表会では口頭発表まで行うことができました。そして、3年次の NICOGRAPH、4年次の DCC 研究発表会、卒業論文全てで締め切り直前での提出となってしまったことをここで謝罪させていただきます。

阿部先生にも多くの助力をいただきました。特に NICOGRAPH のために移動した後の食事の際に、大学院への進学を勧めてくださったのが印象に残っています。その時点では大学院への進学は全く考えていませんでしたが、お話をきっかけに大学院への進学について家族と相談するなど選択肢の一つとして考えるようになりました。結果としては大学院へは進学せず、就職することを選択しましたが、今後社会人になってから大学院へと進むという選択肢を考えることができるようになりました。

渡辺先生、阿部先生と共に参加した DCC 研究発表会は島での開催で交通手段が不安定でした。行きでは飛行機が天候確認のため飛ぶかどうか中々わからず、帰りは飛行機が故障してしまい船で帰ることになりました。そのような中交通手段の対応をしていただきありがとうございました。このような経験は今までなかったため非常に新鮮で、記憶に残る学会となりました。

多くの助言をしていただいた研究室の先輩方に感謝いたします。渡辺先輩、山本先輩には SATA として多くの授業でご一緒させていただきました。SA 業務についてだけでなく、好きなゲームやコンテンツについて話すことができ、とても楽しく業務を行うことができました。その際に研究や就職活動について多くのアドバイスもしていただき非常に参考になりました。本当にありがとうございました。

また、初めての学会参加であった NICOGRAPH でご一緒した先輩方に感謝いたします。学会メンバーに同学年がいないため緊張していた中、声をかけていただきありがとうございました。学会を通して、研究室内の交流が非常に楽しいものであるということを知ることができました。

また、研究室の同学年のみんなにも感謝します。研究についてだけでなく、授業について情報共有したり、一緒にゲームをして楽しんだり、どうでもいいような話で笑いあったりしてくれてありがとう。なかなか研究室で集まる機会はなかったけれども、オンラインでも交流できていたからこそ最後まで走りきることができました。本当にありがとう。

最後にこれまで支えてくれた家族に感謝します。コロナ禍で始まった大学生活で予想通りの大学生活が始まらない中、上京して生活することになり物凄く不安にさせてしまったと思います。思い返すともっと連絡をこまめにするべきだったと反省しています。冬の仕送りの中に温かいスープの元が入っていた時、ものすごく気にしてくれているんだなと感ずることができました。ここまで無事に成長し大学卒業まですることができたのは家族みんなのおかげです。本当にありがとうございます。

# 参考文献

- [1] Rockstar Games. Grand theft auto v. <https://www.rockstargames.com/jp/gta-v>. 参照: 2024.1.17.
- [2] Nintendo. ゼルダの伝説 ブレス オブ ザ ワイルド — nintendo switch / wii u. <https://www.nintendo.co.jp/zelda/totk/index.html>. 参照: 2024.1.17.
- [3] 三宅陽一郎. デジタルゲームにおける人工知能技術の応用. 人工知能学会論文誌, Vol. 23, No. 1, pp. 44–51, 2008.
- [4] 三宅陽一郎. デジタルゲームにおける人工知能技術の応用の現在. 人工知能学会論文誌, Vol. 30, No. 1, pp. 45–64, 2015.
- [5] 独立行政法人工業所有権情報・研修館. 平成 16 年度特許流通支援チャートカーナビ経路探索技術. <https://www.inpit.go.jp/blob/katsuyo/pdf/chart/fdenki22.pdf>. 参照: 2024.1.17.
- [6] 国土交通省. 徒歩及び車による時系列水害避難行動シミュレーション. <https://www.mlit.go.jp/plateau/use-case/uc22-039/>. 参照: 2024.1.17.
- [7] Microsoft. 公式サイト. <https://www.minecraft.net/ja-jp>. 参照: 2024.1.17.
- [8] 三宅陽一郎. Killzone における npc の動的な制御. <https://www.slideshare.net/youichiromiyake/killzonenpc>. 参照: 2023.12.24.
- [9] 桑谷拓哉, 橋本剛. 熟練プレイヤーレベルを目指す弾幕シューティング ai の開発. 情報科学技術フォーラム, Vol. 12, No. 2, pp. 383–384, 2013.
- [10] 佐藤直之, Sila Temsirirkkul, Luong Huu Phuc, 池田心. Influence map を用いた経路探索による人間らしい弾避けのシューティングゲーム ai プレイヤ. 情報処理学会ゲームプログラミングワークショップ 2016 論文集, Vol. 2016, pp. 57–64, 2016.
- [11] E.W.Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, Vol. 1, pp. 269–271, 1959.
- [12] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuris-

- tic determination of minimal cost paths. *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, pp. 100–107, 1968.
- [13] 中宮正樹, 岸野泰恵, 寺田努, 西尾章治郎. コストマップを用いた移動型センサノードの経路探索手法. *情報処理学会論文誌*, Vol. 49, No. 3, pp. 1374–1386, 2008.
- [14] Unity Technologies. ナビゲーションと経路探索. <https://docs.unity3d.com/ja/2021.3/Manual/Navigation.html>. 参照: 2024.1.17.
- [15] KADOKAWA Game Linkage. いかにしてサーバーはモンスターを歩かせるのか? 『ファイナルファンタジー xiv: 新生エオルゼア』の経路探索テクニック【sqex オープンカンファレンス 2012】. <https://www.famitsu.com/news/201211/29025006.html>. 参照: 2024.1.17.
- [16] Anthony Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 3310–3317 vol.4, 1994.
- [17] 渡辺大地. 波動方程式による波伝搬作用を利用した追跡行動アルゴリズム. *芸術科学会論文誌*, Vol. 20, No. 1, pp. 1–9, 2021.
- [18] 恒藤敏彦. 弾性体と流体. 岩波書店, 1994.
- [19] Eric Lengyel. ゲームプログラミングのための 3D グラフィックス数学. ボーンデジタル, 2002.
- [20] Fine Kernel Project. Fine kernel toolkit. <https://gamescience.jp/FK>. 参照: 2023.12.24.