

物流倉庫における人・配送ロボット協調のための センサ統合システム

浅井 悠佑^{1,a)} 青木 俊介³ 米澤 拓郎¹ 河口 信夫^{1,2}

概要：コンピューティング技術や深層学習技術の発達に伴い、自走式配送ロボット (AMR) の利用が進んでいる。産業面での利用も加速しており、物流倉庫においては配送ロボットがピッキング作業の一部を担うなど、オートメーション化の流れは必至であると言える。しかし、物流倉庫には棚や商品が所狭しと置かれており、ナビゲーションをする際には多くの死角が存在する。従来の AMR は搭載センサのみを用いてナビゲーションするため、死角の多い環境においては、衝突の危険性がある。死角となりうる箇所を対象に見えない障害物の動きを対象とした手法が提案されているが、死角の場所が建物に依存している場合や実験室環境もしくは簡易なシミュレーションによる検証のみにとどまっている場合が多い。そこで本研究では、これらの問題を解決するために実環境でのセンシングを行い、センシング情報を配送ロボットナビゲーションに利用する仕組みを提案する。死角の位置の変化を念頭に、時間変化を考慮したコストマップ構築手法を提案し、実際の物流倉庫を模したシミュレーション環境において実験、評価を行う。

Fusion of Distributed Sensors for Human and Mobile Robot Collaboration in Logistic Warehouse

1. はじめに

コンピューティング技術や深層学習技術の発達に伴い、自走式配送ロボット (AMR) の利用が進んでいる [1], [2]。人手不足や作業の自動化を目的にオフィス、病院、工場、倉庫などで今後さらなる AMR の活用が進むと考えられる。物流倉庫においては配送ロボットがピッキング作業の一部を担うなど、オートメーション化の流れは必至であると言える。実世界で AMR を利用するには安全かつ効率的な運用が不可欠である。しかし、既存の物流倉庫に AMR を導入する際には建物のレイアウトや作業フローの関係上、安全で効率的な運用を達成するにはいくつもの課題が存在する。物流倉庫には棚や商品が所狭しと置かれており、AMR がこの環境内でナビゲーションする際には多くの死角が存在する。従来の AMR は搭載センサのみを用いてナ

ビゲーションする 경우가ほとんどであるため、死角の多い環境においては、衝突やデッドロックの危険性があると考えられる。

外界にセンサを設置して死角に存在する障害物情報をリアルタイムに共有して補完する手法が提案されている [3], [4]。しかし、これらの手法は実験室環境や簡易なシミュレーションによる検証のみにとどまっている場合が多い。遮蔽物による死角を考慮した研究も存在するが [5], [6]、同様に実環境での利用を想定した研究は多くない。また、産業用 AMR の安全性を定めた標準規格も複数存在するが、いずれも安全性を完璧に担保する仕組みとはなっていない [7]。AMR の利用シーンは多岐に渡るため、利用シーンを限定した上で実環境を想定した実証や実験が必要であると考えられる。

本研究では、これらの問題を解決するために、実環境でのセンシングを行い、センシング情報を配送ロボットナビゲーションに活用する仕組みを提案する。物流倉庫での作業において、人や物の移動が頻繁に発生するエリアを対象に、観測された障害物が遮蔽物によって認識できなくなった場合の時間的変化を考慮したコストマップ構築手法を提案する。実環境からセンシングによって得られたデータを

¹ 名古屋大学大学院 工学研究科
Graduate School of Engineering Nagoya University

² 名古屋大学 未来社会創造機構
Institute of Innovation for Future Society

³ 国立情報学研究所
National Institute of Informatics

a) asayu@ucl.nuee.nagoya-u.ac.jp

参考にシミュレーション環境を構築し、実際の物流倉庫での検証を念頭にシミュレーション上で実験、評価を行う。

本稿の構成は以下の通りである。2 章では関連研究について述べる。3 章では対象となる環境について述べ、本研究における問題の定義を行う。4 章では提案手法について述べ、5 章では実装について、6 章では行った実験について述べる。最後に 7 章では今後の展望について述べる。

2. 関連研究

物流倉庫や工場における AMR と人の協調作業手法はいくつか提案され [3], [8], [9], [10], それらの多くはデジタルツインや簡易的なシミュレーター上で実験が行われている。しかし、AMR と人が作業スペースを完全に共有する環境での運用は未だ研究段階であると言える。それに加えコストや実証環境の不足から、現場での十分な検証がなされているとは言えない。Fukushima らは物流倉庫で AMR と人のピッキング協調作業を行うシステムを提案し、実証実験を行った [1]。実証実験の前後にはシミュレーション上で検証を行ったが、作業者の導線をモデル化する際には、実際の経路ではなく最短距離にするといった簡略化が見られる。

人と協調するための AMR ナビゲーション手法も提案されている。アルゴリズムベースや深層学習ベースの手法があり、それらの多くは動的な環境を考慮した経路計画手法となっている。Berg らは複数エージェント (障害物) の動きを考慮しながら、ロボットとの衝突回避を保証するアルゴリズムである RVO と呼ばれる動作計画法を提案した [11]。しかし、RVO では周囲にある障害物を検知していることが前提である。他の障害物や建物の壁などによって生じるオクルージョンは考慮されないため、星野らは遮蔽障害物の位置、速さ、移動方向を推定し、これらを定量的に扱うためのモデル化を行った [5]。しかしながら、観測可能な動的障害物の位置、速さ、移動方向の動きをもとに遮蔽障害物の動きを推定するため、場所、タイミングが不定の死角が発生する環境においては適応が困難である。

その他にも、センサ範囲と遮蔽領域の境界に注目したナビゲーション手法の研究も提案されている [12], [13]。これらの研究に加え、この境界領域を時間経過とともに膨張させることでオクルージョンに対処する提案もなされている [6]。しかし、遮蔽物によって生じる遮蔽領域のみに焦点が当てられており、遮蔽領域に存在する障害物に対して挙動を変えろといった AMR 側の視点で議論されていない。また、物流倉庫のように多くの棚が並んだスーパーマーケットを対象とした研究も存在する [14]。しかし、環境認識にロボットのセンサのみを利用しており、通路の角など死角が生じる場所すべてで減速しており、死角が多い環境では大幅に稼働効率が低下すると考えられる。

そこで我々は死角が変化しうる環境において、外部セン



図 1: 対象環境である仮置き場。AMR がこの環境を走行中には、上に積み上げられた商品の死角によって作業者と衝突する可能性がある。また、AMR 単体では死角にいる作業者を検知するのは困難。茶色のエリアは AMR 視点での死角。

サを用いながら死角を補完する AMR ナビゲーション手法を提案する。死角の発生によって動的障害物がセンサによって検知できなくなる場合を想定し、観測された障害物が遮蔽物によって認識できなくなった場合の時間的変化を考慮したコストマップ構築手法を提案する。本研究では、AMR を用いた物流倉庫における作業の自動化を前提にしており、対象とする作業、環境については 3 章で述べる。

3. AMR と人の協調問題の定義

本章では、本研究での AMR と人の協調問題の定義を行う。3.1 節では、AMR と人の協調作業を行う対象の環境について述べ、3.2 節では協調作業問題の定義を行う。

3.1 対象環境

本研究で対象とする環境は図 1 に示すような商品の仮置き場である。物流倉庫においては、入荷された商品は棚に格納されるまで仮置き場に置かれる。仮置き場は庫内物流のボトルネックになることもあり、商品が仮置き場に置かれる際には積み上げられることが多く、場所、高さが不定の死角が頻繁に発生する。本研究では仮置き場から各商品の格納場所がある棚まで商品を運搬する作業を AMR が行うと仮定し、特に仮置き場における AMR のナビゲーションを対象とする。実験を行う仮置き場には、人や物の位置を計測するカメラと LiDAR センサが複数設置されており、AMR の死角を一部補完することが可能である。

3.2 問題定義

行うタスクは死角が複数あり、かつ死角が頻繁に変化する仮置き場において、設置したセンサの情報をいながら AMR のナビゲーションを行うことである。図 1 に示すように、仮置き場を AMR が走行する際、上に積み上げられた商品によってできた死角に作業者がいると、作業者、AMR 双方とも互いの存在を認知できないため、衝突の危

険性が高くなる。また、従来の AMR はナビゲーション時に搭載しているセンサのみを用いることが大半である。この図の写真のように外部センサによって人の存在を検知して AMR に伝えることができれば作業者との衝突やヒヤリハットを回避できると考える。

このような問題を解決するために、設置したセンサから障害物の情報を受け取った後、障害物情報を保持するコストマップを更新することでナビゲーションを行う。対象環境では死角の位置が頻繁に変化するため、固定型センサを用いてもすべての障害物が常時検出できるとは限らない。そのため、このような環境においては、検出した障害物が一定時間後に検出できなくなる場合を考慮する必要がある。

4. 提案手法

本章では、提案手法であるシステムの概要、データ処理のパイプライン、データ共有の詳細について述べる。

4.1 システムの全体像

本研究で提案するシステムは図 2 に示すように複数の逐次、並列処理からなる。カメラの歪み補正を行う内部 (intrinsic) パラメータやカメラ-LiDAR 間の点群とピクセル値の関係を表す外部 (extrinsic) パラメータを予め求めておく。はじめに点群、画像においてはそれぞれダウンサンプリング、歪み補正を行う。その後、画像から物体認識技術を用いて人とダンボールを検出する。得られた画像上のバウンディングボックスに沿って四角錐形に点群を切り出すことで、物体の 3 次元位置を導出する。導出した障害物の位置情報を AMR ナビゲーションで用いるコストマップに反映する。その際に、障害物が死角の変化によって検出できなくなる場合を考慮し、コストマップに一定の間、障害物の情報が保持されるようにする。

4.2 センサ統合による AMR ナビゲーション

環境の変化を得るために複数のカメラ、LiDAR センサを使用する。カメラの画像情報は物体認識を中心に利用され、LiDAR センサは 3 次元的な空間変化や物体の位置を得るために用いられる。

4.2.1 複数の LiDAR センサ情報の統合

設置した各 LiDAR センサの位置、向きを求めるために、本研究では NDT-map matching 手法 [15] を利用する。事前に環境の点群地図を作成し、センサデータとのマッチングによって点群地図との相対的な位置、向きを推定する。例として位置合わせ後の 4 つの LiDAR の点群を図 3 に示す。

4.2.2 動的な障害物の抽出

設置する LiDAR センサ数が増えるに伴って処理する点群数も増加する。そのため、複数の点群データをまとめてリアルタイムで処理するには、処理する点群数の削減が不可

欠である。本研究では固定型センサの特性を活かすために、背景差分を用いて点群数を削減する。事前に動的障害物が無い状態で各センサから点群を取得する。その点群とセンサからリアルタイムで得られた点群の差分を計算することで動的障害物の点群のみを得ることができる。この処理を行うことで扱う点群数も削減される。差分検出の手法としては Octree を用いた。Octree とは点群の配置具合をもとに適応的に空間分割を行うことでボクセル単位の探索や占有度管理が容易になる仕組みである。図 4 に動的障害物である人とダンボールの点群のみを抽出した様子を示す。

4.2.3 画像による物体認識

近年、画像を用いた物体認識技術の向上によって、画像内にある任意の物体の検出は比較的容易である。物流倉庫内に設置したカメラ画像から YOLOv4 [16] を用いて人の認識を行った例を図 5a に示す。また、本研究ではダンボールの認識を行うために約 1800 個のダンボールのアノテーションデータを作成し、Detectron2 [17] を用いて学習した。ダンボール認識の一例を図 5b に示す。

4.2.4 センサフュージョン

抽出した動的障害物の点群と画像から取得した物体認識結果を統合するために、事前に内部 (intrinsic)、外部 (extrinsic) パラメータをそれぞれ取得する。カメラキャリブレーションツールを用いてカメラ歪みパラメータを求め内部パラメータを得る [18]。次に画像の画素とそれに対応する LiDAR 点群の関係を表す外部パラメータを求める。具体的な手法としては、建物の特徴 (壁、棚など) を画像と点群からそれぞれ抽出してマッチングを行う手法を用いる [19]。マッチング精度は点群に色情報を付与することで確認することができる。

画像と点群の対応関係を取得後、画像認識の結果から得られたバウンディングボックスの情報を画像と点群の点群を切り出す。方法としては図 6 に示すような、点群を四角錐状に切り出す Frustum Culling 手法を用いた。Frustum Culling は Point Cloud Library (PCL) を用いて行われ、入力としては、四角錐の垂直、水平方向の角度と四角錐の変換行列を与える。

四角錐の垂直、水平方向の角度は以下の式 (1)、(2) のように与えられる。

$$\theta_v = \arccos(\mathbf{v}_{vu} \cdot \mathbf{v}_{vl}) \quad (1)$$

$$\theta_h = \arccos(\mathbf{v}_{hr} \cdot \mathbf{v}_{hl}) \quad (2)$$

\mathbf{v}_{vu} , \mathbf{v}_{vl} , \mathbf{v}_{hl} , \mathbf{v}_{hr} はそれぞれバウンディングボックス上下左右端方向のベクトルである。

四角錐の平行移動、回転を得るためには、まずカメラの初期位置方向ベクトルとバウンディングボックスの中心点方向ベクトルがなす角度 θ_o を式 (1)、式 (2) と同様に求める。また、四角錐の回転軸ベクトル \mathbf{r} は以下のように求められる。

$$\mathbf{r} = \mathbf{v}_o \times \mathbf{v}_c \quad (3)$$

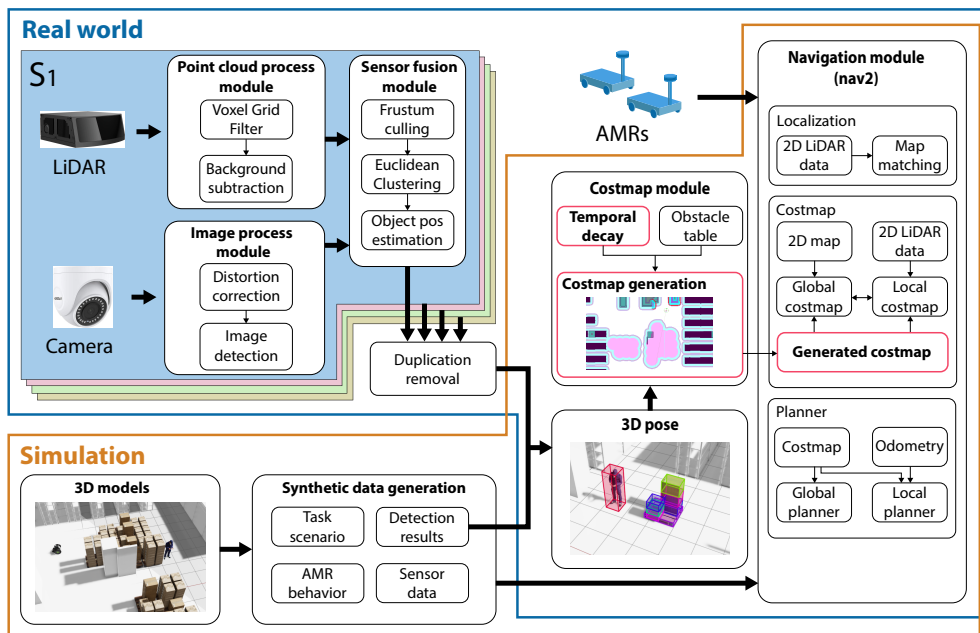


図 2: 提案手法の概要. それぞれのセンサノード (S_1, S_2, S_3, S_4) はカメラと LiDAR からなる.

これらを用いて, 回転行列を求める. $\mathbf{r} = (r_x, r_y, r_z)$, θ_o を用いて,

$$R = \begin{bmatrix} \cos \theta_o + r_x^2 c & r_x r_y c - r_z \sin \theta_o & r_x r_z c + r_y \sin \theta_o \\ r_y r_x c + r_z \sin \theta_o & \cos \theta_o + r_y^2 c & r_y r_z c - r_x \sin \theta_o \\ r_z r_x c - r_y \sin \theta_o & r_z r_y c + r_x \sin \theta_o & \cos \theta_o + r_z^2 c \end{bmatrix}$$

となる. また, c は $(1 - \cos \theta_o)$ と定義した. また, カメラは x , y , z 方向には移動しないため, 平行移動行列は $T = (0 \ 0 \ 0)^T$ とすることができる. 求めた行列 R , T を用いると, 四角錐の変換行列 P は以下ようになる.

$$P = \begin{bmatrix} R_{00} & R_{01} & R_{02} & T_0 \\ R_{10} & R_{11} & R_{12} & T_1 \\ R_{20} & R_{21} & R_{22} & T_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

求めた四角錐の垂直, 水平方向の角度と四角錐の変換行列を用いることで, バウンディングボックス情報を元にした点群の切り出しが可能となる.

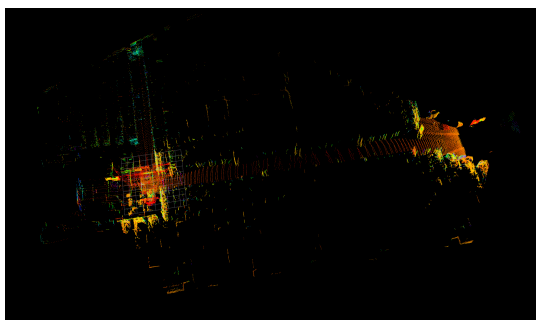


図 3: 物流倉庫内に設置した複数の LiDAR の点群データ (位置合わせ後).

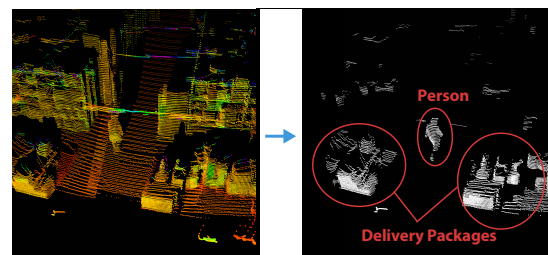


図 4: 動的障害物の点群抽出の概要.



(a) 作業者の認識.

(b) ダンボールの認識.

図 5: 画像による物体認識の例.

切り出した点群を元に動的障害物の位置を推定する. 本研究では, ユークリッドクラスタリングを用いることで点群をいくつかのクラスに分ける. この手法はシンプルだが高速であるため, リアルタイム処理を行う上では有用である. パラメータの調整によって切り出した動的障害物の点群をクラスタリングするとほとんどの場合はクラスは 1 つに限定される.

複数のクラスに分けられた場合は, それぞれのクラス

タの位置の平均を推定位置とする。また、複数のセンサを用いて障害物の位置推定を行っているため、同一の障害物を複数のセンサで検出する場合も多々ある。そのため、同じ種類の障害物が一定のユークリッド距離 d_t 以内の障害物は同一の物とみなすことで複数検知を防ぐ。複数検知が行われたと判断した場合、画像認識の確度が高い結果を優先する。また、 d_t は検知したい物体によって変化させる。

4.2.5 障害物の時間的変化を考慮したコストマップの構築

推定した障害物の位置を元に AMR ナビゲーションで用いられるコストマップを構築する。コストマップはナビゲーションの際の経路探索や速度指令値の決定に用いられる。

障害物の存在によってマップ上に生成されるコストを時間変化に伴って減衰させる。時間変化による障害物のコスト減衰は以下のように記述できる。

$$r_{t+\Delta} = \begin{cases} r_t - f\Delta & (r_t \geq 0) \\ r_t & (r_t < 0) \end{cases} \quad (4)$$

式 (4) で示した r は時間経過にともない 0 に近づく値である。ここで $r_0 = 1$ とする。減少の度合いを表す f は式 (5) のように定義される。

$$f = \frac{\alpha}{d_{r-o}} \quad (5)$$

d_{r-o} はロボットと障害物の距離を表す。また、 α は減少速度を決定する定数である。

本研究では 2 次元平面上で表されるコストマップを扱う。障害物の情報を 2 次元平面上へ投影する際には、障害物の形を円もしくは四角形で近似する。ここで、2 次元平面上の任意の点を $p = (x, y)$ とし、2 次元平面上における地図に対応する点の集合を $M(p)$ とする。認識障害物が 2 次元平面上に投影される点の集合を S とすると、地図上の障害物の座標の集合は以下のように表すことができる。

$$\{p \in S \mid M(p)\}$$

ナビゲーションに用いるコストマップは固定センサで認識

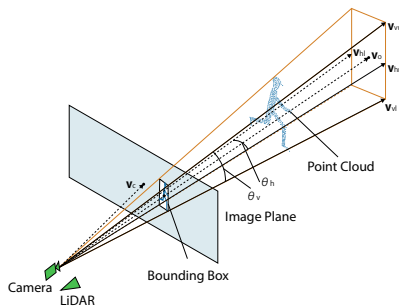


図 6: バウンディングボックスの形に沿って四角錐上に点群を切り出す手法の概要。

した障害物情報のほか、事前に作成した地図とロボットのセンサ情報も統合される。ここで、地図とロボットのセンサ情報をもとに計算した時間 t におけるコストを $C(p)_t$ とすると、認識した障害物の時間的変化を考慮したコストは以下ようになる。

$$C(p)_{t+\Delta} = \begin{cases} C(p)_t + C_{lethal} * r_{t+\Delta} & \{p \in S \mid M(p)\} \\ C(p)_t & \{p \notin S \mid M(p)\} \end{cases} \quad (6)$$

式 (6) は地図と AMR のセンサによって計算されたコストに認識した障害物のコストを付与する計算を行っている。また、 C_{lethal} は障害物が存在する際のコストを表す。コストが C_{lethal} の座標にはロボットは移動することはできない。コストマップの仕様上において、 $C_t > C_{lethal}$ であるときは $C_t = C_{lethal}$ として扱う。

5. 実環境でのセンシングとシミュレーション環境構築

本章では、実験を行ったセンシング環境と実環境を元にした AMR シミュレーション環境の詳細を述べる。

5.1 実験環境

実験を行う環境として図 7 で示す物流倉庫の 3 階フロアを対象とする。フロアのサイズとしてはおよそ 70m x 60m ほどあり、仮置きエリア、格納エリア (棚があるエリア)、出荷エリアの 3 ヶ所に大別される。仮置きエリアは入荷・検品された商品を一時保管する場所である。このフロアで行われる作業の流れを示す。作業者は仮置きエリアに置かれた商品のバーコードをスキャンする。スキャンされた商品は作業者がスキャナに表示された指定の棚に商品を格納することで在庫となる。注文を受けた商品は対応する棚から作業者によってピッキングされ出荷エリアを通して出荷される。

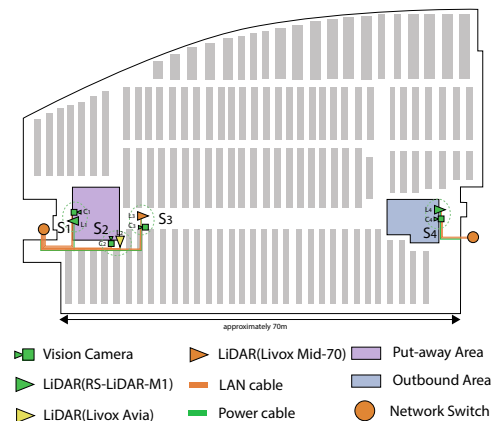


図 7: センサ設置場所と、ネットワークやケーブルの構成。2つのネットワークスイッチは共通のネットワークに接続されている。

表 1: LiDAR のスペック

(L_1 : RS-LiDAR-M1 L_2 : Livox Avia L_3 : Livox Mid-70)

	L_1	L_2	L_3
range (m)	200	450	90
horizontal FOV (°)	120	70	70
vertical FOV (°)	25	77	70
points (pts/s)	750,000	240,000	100,000

5.2 ハードウェア

固定型センサはカメラと LiDAR を対にした形からなる。カメラは H.View 4K PoE Dome Camera を使用し、LiDAR については 3 種類の異なる LiDAR を設置する。複数種類の LiDAR を用いた理由のひとつとして、センシング範囲や点群密度、価格など様々な要素を考慮することが挙げられる。各 LiDAR の性能を表 1 に示す。図 7 に示すように、物流倉庫のフロア内に計 4 ペアのカメラと LiDAR を設置する。これらのカメラ、LiDAR はいくつかのスイッチを経由して同一ネットワーク上のコンピュータに接続されている。実験に用いるコンピュータは、AMD Ryzen 9 5950X CPU、128GB メモリ、NVIDIA GeForce 3090 GPU を搭載している。センサデータのやりとりは主に Docker イメージ上に構築された Robot Operating System (ROS) / Robot Operating System 2 (ROS2) を用いて行われる。

5.3 センサの設置とキャリブレーション

対象の物流倉庫フロアでの作業の大半は仮置きエリアと出荷エリアで起こる。したがって格納エリアに比べると、仮置きエリアや出荷エリアは人やモノの動きが活発であるといえる。図 7 に示すようにこれら 2 つのエリアとフロアのメインとなる通路をカバーするようにセンサを設置する。センサの位置合わせは 4.2.1 節で解説した手法を用いる。あらかじめ LiDAR センサ (Velodyne VLP-16) を搭載した AMR を庫内でマニュアル走行して点群地図を作成した。その地図とセンサデータをもとに点群地図との相対位置を導出した結果を図 3 に示す。

カメラと LiDAR センサの対応関係を導出するキャリブレーションは 4.2.4 節で解説した手法を用いる。設置したそれぞれの LiDAR センサのデータを物が動いていない状態で 15 秒ほど取得する。記録中に物が動いてしまった場合は再度記録を取り直す。理由として、点群データを記録中に動く物があると画像と点群の特徴点マッチングの際に失敗してしまう可能性があることが挙げられる。

5.4 AMR ナビゲーションのシミュレーション

本節では実験環境における AMR シミュレーションの構築について解説する。

5.4.1 物流倉庫デジタルツイン

デジタルツインとは、物理空間にあるデータを仮想空間

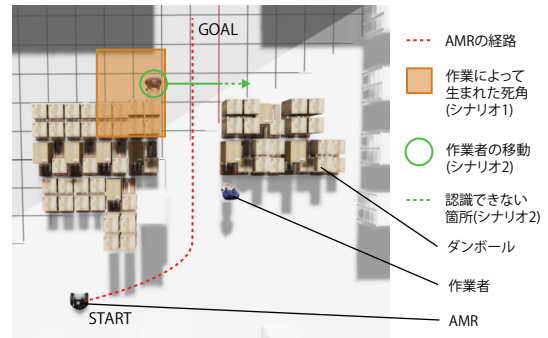


図 8: シミュレーター上の環境と実験シナリオ

にデジタルコピーとして再現する技術である。物理空間を再現するために、実験の対象となる物流倉庫のフロアの 3 次元モデルを作成する。3 次元モデルは色付き点群地図や写真を参考にしながら、棚や柱のレイアウトを再現するように手動で作成する。また、動的障害物である人、ダンボール、パレットの 3 次元モデルは後述する NVIDIA Isaac Sim に付属する既存のモデルを用いた。実験環境に設置したセンサから得た障害物の情報を参考にしながら、仮置き場での人やモノの配置を再現する。AMR のナビゲーション実験は 3 次元シミュレーターである NVIDIA Isaac Sim 上で行う。NVIDIA Isaac Sim はロボティクス向けのシミュレーションプラットフォームで、ROS/ROS2 との連携や仮想センサデータの生成、物体認識のグラウンドトゥールズ生成といった機能を有するため、本研究での実験趣旨に合っていると判断した。

5.4.2 AMR ナビゲーション

AMR を用いたナビゲーション実験には ROS2 フレームワーク上で動作する Nav2 ライブラリをベースに用いる。Nav2 ライブラリはナビゲーションに必要な自己位置推定、コストマップ生成、経路探索、コントロール機能を備える。また、各機能のモジュール化がなされているため、プラグインを実装することによって独自の実装に一部置き換えることが可能である。本研究では、時間経過を考慮した障害物情報を受け取りコストマップを生成する機能をプラグインとして実装した。

6. 実験

本章では、構築したシミュレーション環境を用いて AMR のナビゲーション実験を行う。

6.1 実験シナリオ

図 8 に示すような 2 つのシナリオを想定して実験を行う。両シナリオとも AMR のスタート、ゴール位置は同じである。ベースライン手法として、我々が作成したプラグインを用いない状態でナビゲーション実験を行い比較を行う。

シナリオ 1: 作業による死角の発生

図 9 に示すように、作業者がダンボールを上積み上

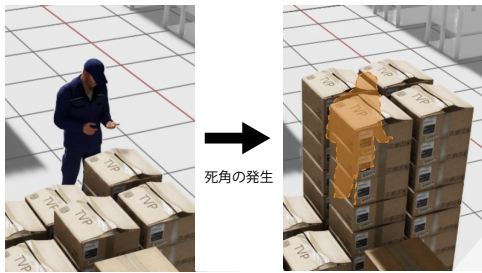


図 9: 作業によって生まれた新たな死角 (シナリオ 1)

げることで死角が新たに生まれるシナリオである。今回の実験では、ナビゲーション開始から 10 秒後にダンボールが積み上がり作業者が認識できない状態になるとする。

シナリオ 2: 死角への作業者の移動

認識可能な場所から死角へ作業者が移動するシナリオである。はじめはセンサによって認識できていたが、移動によって認識ができなくなったと想定する。図 8 の矢印の波線部分がセンサで認識できない箇所である。作業者が移動するタイミングはナビゲーション開始から 10 秒後とし、移動速度は約 1.0m/s とする。

6.2 ロボットの仕様と各種パラメータ

ロボットは NVIDIA Isaac Sim において始めから付属している「Carter」を利用する。このロボットは差動二輪式であり、筐体の半径は 0.36m である。サンプルとして Nav2 のナビゲーションを行うプログラムが利用可能であり、このプログラムに我々が作成したコストマップのプラグインを付与する形でナビゲーション実験を行った。パラメータは新たに追加したプラグイン以外はサンプルと同じものを利用した。なお、コストマッププラグインにおける α の値は 1.0 を指定した。

6.3 結果

図 10 にそれぞれシナリオ毎の速度変化と走行経路の変化を示す。シナリオ 1 では走行経路、シナリオ 2 では速度変化に比較的大きな変化が現れた。シナリオ 1 では、途中で死角に存在する作業者の認識ができなくなったにも関わらず、走行経路から AMR が回避行動を取ったことがわかる。これは、認識ができなくなったあとでも、一定時間コストマップ上に障害物の情報が残っていたからである。提案手法を用いてのシナリオ 2 では、走行経路上を遮る形で作業者が移動するため経路が再計算されその場回転を行った。そして、別の経路を進もうとしたタイミングでコストマップ上の障害物の情報がクリアされたため、再度、初期経路に戻る挙動をした。ベースライン手法においては、距離の関数上、作業者の動きがコストマップに反映されず、作業者の移動は無かったかのような挙動を示した。なお、到着時間についてはいずれも従来手法のほうが早かった。

7. 今後の展望

本研究では、死角の位置変化を念頭に、時間変化を考慮したコストマップ構築手法を提案した。また、センシング情報を配送ロボットナビゲーションに利用するとともに実世界での利用を前提とした仕組みを構築した。また、実際の物流倉庫内にセンサを設置し、人や物の動きをセンシングすることでシミュレーションに活用する方針を示した。それに加え、実際の物流倉庫を模したシミュレーション環境において実験、評価を行った。今回はシミュレーションに際してシナリオを手動で構築したが、ダイナミックに変化する環境においては、実環境をベースにリアルなシミュレーションを多様な条件化で行うことが求められる。また、コストマップにおける障害物のコスト減衰も実際の人の動きを反映しているとはいえず、手法の改善が見込まれる。

謝辞 本研究の一部は NICT 委託研究 01701, 科研費 K18422 に支援いただいています。また、実験環境提供にご協力いただきましたトラスコ中山株式会社に感謝いたします。

参考文献

- [1] Yuto Fukushima, Yusuke Asai, Shunsuke Aoki, Takuro Yonezawa, and Nobuo Kawaguchi. Digimobot: Digital twin for human-robot collaboration in indoor environments. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 55–62. IEEE, 2021.
- [2] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 797–803. IEEE, 2010.
- [3] Elena Cardarelli, Valerio Digani, Lorenzo Sabattini, Cristian Secchi, and Cesare Fantuzzi. Cooperative cloud robotics architecture for the coordination of multi-agv systems in industrial warehouses. *Mechatronics*, Vol. 45, pp. 1–13, 2017.
- [4] Nathalie Majcherczyk, Daniel Jeswin Nallathambi, Tim Antonelli, and Carlo Pinciroli. Distributed data storage and fusion for collective perception in resource-limited mobile robot swarms. *IEEE Robotics and Automation Letters*, Vol. 6, No. 3, pp. 5549–5556, 2021.
- [5] 星野智史, 芳川知樹. 遮蔽領域における障害物をも考慮した移動ロボットの動作計画法. 日本機械学会論文集, pp. 18–00353, 2019.
- [6] Sara Bouraine, Thierry Fraichard, and Hassen Salhi. Provably safe navigation for mobile robots with limited field-of-views in dynamic environments. *Autonomous Robots*, Vol. 32, No. 3, pp. 267–283, 2012.
- [7] Jeremy A Marvel and Rick Norcross. Implementing speed and separation monitoring in collaborative robot workcells. *Robotics and computer-integrated manufacturing*, Vol. 44, pp. 144–155, 2017.
- [8] Matthias De Ryck, Mark Versteyhe, and Frederik De-brouwere. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *Journal of Manufacturing Systems*, Vol. 54, pp. 152–173, 2020.
- [9] Daniel Teso-Fz-Betoño, Ekaitz Zulueta, Unai Fernandez-Gamiz, Iñigo Aramendia, and Irantzu Uriarte. A free

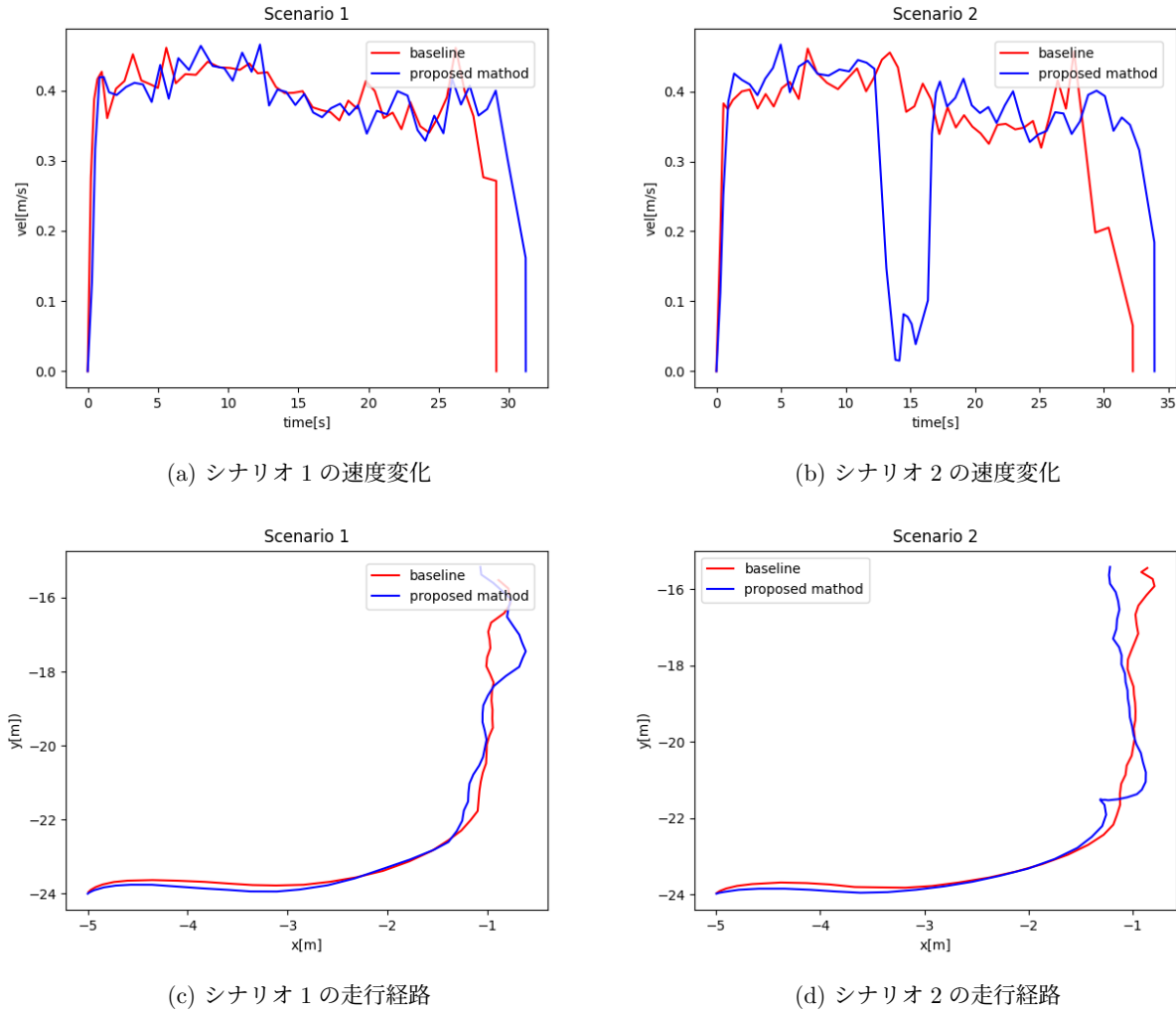


図 10: シナリオ毎の速度変化と走行経路

navigation of an agv to a non-static target with obstacle avoidance. *Electronics*, Vol. 8, No. 2, p. 159, 2019.

- [10] Aniruddha Singhal, Harsh Vardhan Singh, Aarathi Penumatsa, Nakul Bhatt, Prakash Ambwani, Swagat Kumar, and Rajesh Sinha. An actor based architecture for multi-robot system with application to warehouse. In *Proceedings of the 1st International Workshop on Internet of People, Assistive Robots and Things*, pp. 13–18, 2018.
- [11] Jur van den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics research*, pp. 3–19. Springer, 2011.
- [12] Thierry Fraichard and Hajime Asama. Inevitable collision states—a step towards safer robots? *Advanced Robotics*, Vol. 18, No. 10, pp. 1001–1024, 2004.
- [13] Jacob Higgins and Nicola Bezzo. Negotiating visibility for safe autonomous navigation in occluding and uncertain environments. *IEEE Robotics and Automation Letters*, Vol. 6, No. 3, pp. 4409–4416, 2021.
- [14] Kenny Schlegel, Peter Weissig, and Peter Protzel. A blind-spot-aware optimization-based planner for safe robot navigation. In *2021 European Conference on Mobile Robots (ECMR)*, pp. 1–8. IEEE, 2021.
- [15] Naoki Akai, Luis Yoichi Morales, Eijiro Takeuchi, Yuki Yoshihara, and Yoshiki Ninomiya. Robust localization

using 3d ndt scan matching with experimentally determined uncertainty and road marker matching. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1356–1363. IEEE, 2017.

- [16] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [17] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [18] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, Vol. 32, No. 1, pp. 7–25, 1999.
- [19] Chongjian Yuan, Xiyuan Liu, Xiaoping Hong, and Fu Zhang. Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments. *arXiv preprint arXiv:2103.01627*, 2021.