自律移動ロボット Smart Dump による実世界環境走行実験 —歩行者環境における人の群れ行動による影響と対策—

伊 達 央* 滝 田 好 宏*

Real World Experiments of Autonomus Mobile Robot Smart Dump —Influence and Countermeasure of Human Crowd Behavior in a Pedestrian Environment—

Hisashi Date* and Yoshihiro Takita*

Smart Dump has been developed to complete the mission of Real World Robot Challenge, or Tsukuba Challenge. The challenge requires robots to travel autonomously in pedestrian environment for 1 [km] or more distance without any help of human operators. It is even prohibited to modify the environment for the robots. This paper reviews three-time participation to the challenge, especially unexpected results due to the real world.

Key Words: Real World Robot Challenge, Autonomous Mobile Robot, Sensor Fusion, Dynamic Environment

1. は じ め に

自律移動ロボットの分野の研究の多くは大学構内あるいは研究施設内における実験によって検証されてきた。屋内のロボットについては、MIT Killian Court や Intel Research Lab 等の壁面データがベンチマークとして公開され [1], DARPA Urban Challenge では、砂漠および市街地を模擬した軍事基地跡等が使用された [2]。自律移動ロボットが直面する多くの課題はそのような環境で想定されたシナリオのもとで解決されてきたが、実世界環境はより複雑で、想定していなかった問題が潜在している。つくばチャレンジは、人が生活する空間の中で自ら考えて行動するロボットの技術の追究を目的として、1 [km] 以上離れた目的地に自律的に到達するという課題を達成するため、公道を含む歩行者環境を使用した公開実験の機会を提供している [3]。つくばチャレンジは、全行程よりも短い距離の到達に目標を定めたトライアルと全行程を走行する本走行（2010 以降はファイナルとよび替え）により構成され、トライアルを通過したロボットのみが本走行を許される。さらに事前に全行程において試験的に走行する機会が複数回設けられる。

筆者らは、これまでに軌道誘導車両の研究に携わってきており、これをより発展させ軌道外で自律走行できる自動車の自動運転化を目指して、前輪操舵式の 4 輪車両 Smart Dump を開発し、2008 年以来、つくばチャレンジに参加している。初回参加であった 2008 年は、防衛大学校内において十分な成果が得

られないまま準備不足の参加となり、トライアルを通過することができなかった。翌 2009 年の参加にあたっては、校内におけるテストを十分に実施して基本的な性能を実現したあとに、実際のコース上における試走に臨んだ。その結果、試走やトライアルにおいて好結果を得た。しかしながら、本走行において多数の観客の作る人垣に予想外の影響を受け、スタート直後で走行不可能となった。これはまさに実世界環境によって顕在化した問題の一つと言える。2010 年においては、前年の失敗を踏まえ、人垣を含む環境の変化に対してよりロバストな自己位置推定アルゴリズムを導入し、課題を達成することができた。

本論文では、筆者らが開発した Smart Dump の設計方針と、これまでの 3 回の参加を通じて得られた教訓と講じた対策について述べる。本論文の構成は以下のとおりである。第 2 章において基本的な設計方針と技術目標について述べ、第 3 章において Smart Dump の概要を述べる。第 4 章において、ロボットの自己位置推定に影響を与える人の群れ行動とその対策について述べ、第 5 章でつくばチャレンジにおける各年の取り組みと結果を示し、考察する。

2. 設 計 方 針

2.1 車体設計

ロボットカーへの応用を目指していることから、車体は乗用車と同様の運動学構造を有する、前輪操舵方式の四輪車とする。つくばチャレンジにおける歩行者保護の必要性和雨天時の部品保護の観点から、車体各部品を可能な限り覆う。

2.2 制御システム

歩道を走行できる大きさの車体には、大きさ・重量・消費電力の観点から演算能力の高い計算機を搭載するのが困難である。

原稿受付 2011 年 7 月 11 日

*防衛大学校

*National Defense Academy

■ 本論文は有用性で評価されました。

実装するアルゴリズムはミニノートとよばれる軽量のパソコンで動作するよう、極力計算負荷が小さくなるようにする。

目的地まで確実にロボットを導くにあたっては、自己位置推定の精度と信頼性を高めた上で正確に経路を追従するアプローチと、環境認識力を高めることで大まかな進路指令で適切な進路に導くアプローチに分けて考えることができるが、つくばチャレンジにおいては目的地までの経路は事前に知らされており、なおかつ事前に走行する機会が複数回設けられており、自己位置推定の性能向上が有利であると判断した。経路上、間隔の狭いボールや狭い通路を通過する必要があることから、少なくとも車体幅より小さな精度が必要であるが、民生用に広く用いられるようになっている人工衛星を用いた Global Positioning System (GPS) では、常に必要な精度を得られるわけではない。歩行者環境は、樹木や建造物など何らかの障害物がレーザスキャナによって検出可能な距離に存在していることが多い。そこで屋内の自律移動ロボットでよく用いられるレーザスキャナによるマップマッチングを組み合わせたこととした。

3. ロボット概要

3.1 車体

Fig. 1 に 2009 年参加の Smart Dump 3 の外観を示す。歩道環境に存在する 20 [mm] 程度までの段差が乗り越えられるよう、四輪が直流モータによって駆動されている。平地における最高速度は最大で 1.1 [m/s] である。レーザスキャナは車体左右両側の 280 [mm] 程度の高さの位置に水平走査するように取り付けられている。それぞれが視野角 270° を持ち、全方位がスキャンできるようになっている。これにより障害物回避とマップマッチングの両方を同時に行う。後輪両外側には従輪がトレーリングアームにより牽引され、1 パルス当たり 0.2 [mm] のオドメトリを実現している。車体上部には GPS コンパスを設置している。諸元については、既報 [4] に記載されている。

3.2 自己位置推定法

前章に示した方針から、自己位置推定を高精度かつロバストに実現することが課題達成の要である。特にレーザスキャナによるマップマッチングは、大部分の行程において主要な自己位置推定手段となる。

Fig. 1 Smart Dump 3

3.2.1 地図生成

地図は、スキャンデータを 1 [m] ごとに保存し、過去に保存した点群に重ね合わせるようにマッチングを行いながら点群を追加して拡大することで生成する。屋内走行のロボットと異なり、GPS データを同時に利用するためには、地図の座標と対応させておく必要がある。そこで 1 [m] ごとに保存されたスキャンデータに GPS 測位結果をひも付けておき、連続するスキャンデータ同士の重ね合わせの誤差と GPS の座標からの誤差が最小となるよう、修正量を文献 [5] の手法で最適化している。最終的に点群の配置が定まったところで格子サイズ 100 [mm] の占有格子 [6] に変換して保存する。その際、データは 8 [bit] 整数型の二次元配列 $m \times n$ に格納され、1 個以上の点が存在する格子は最大の占有度 255 を持たせる。隣接のスキャン点のない格子の値は占有格子からの距離に応じて漸減させることにより分布の広がりを持たせ、マップマッチングにおける引き込み領域を広げる効果を得ている。

3.2.2 マップマッチング

オドメトリと地図データの統合には、パーティクルフィルタを使用したものや Iterative Closest Point (ICP) [7] とカルマンフィルタを用いたものが挙げられる [8]。パーティクルフィルタはベイズ推定を活用し、多峰性のある状況や不正確な初期状態に適用でき、移動ロボットにおける実装のしやすさに優れている反面、多数の点についての評価が必要となる。ICP は地図とスキャンデータの対応する点の誤差距離を最小化するため、誤差距離が正しく評価される限り大きな誤差にも対応できるが、一般に対応点の探索に多くの計算時間を要する。いずれも計算力の乏しい環境における実行には高速化のための工夫が欠かせない。ひとたび地図とスキャンデータが合致すれば、オドメトリに基づく位置推定に大きな誤差が急激には生じないと考え、スキャン点とランドマークの重なりが最も多くなる位置姿勢を限られた近傍において探索する実装とした。オドメトリによる予測と、マッチングの結果を観測出力として用いることで、通常のカルマンフィルタの枠組みに自然に組み入れることができる。

具体的な処理の流れを **Fig. 2** と **Table 1** の疑似コードを用いて示す。まず 1~3 行目において、オドメトリによって得られる微小推進距離 dL 、微小姿勢変化 dTh を用いてロボットの位置姿勢 (X, Y, Th) の予測値 $(X1, Y1, Th1)$ を得る。こ

Fig. 2 Map and coordinates

Table 1 Pseudo code for map matching

```

1 Th1 = toInt(Th + dTh);
2 X1 = X + dL * c[Th1];
3 Y1 = Y + dL * s[Th1];
4 likelihood[N_POINTS] = {0};
5 for (i = 0; i < N_POINTS; i++) {
6   j = 0;
7   for (dx = -1; dx <= 1; dx++){
8     for(dy = -1; dy <= 1; dy++){
9       for(dth = -1; dth <= 1; dth++){
10        for (sum[j] = 0, i = 0; i < N; i++){
11          x = toInt(r[i]*c[i+Th1+dth]+dx+X1);
12          y = toInt(r[i]*s[i+Th1+dth]+dy+Y1);
13          likelihood[j] += m[x][y] * r[i];
14        }
15        j++;
16      }
17    }
18  }
19 }
20 max_j = 27 / 2;
21 max = likelihood[max_j];
22 for (j = 0; j < 27; j++) {
23   if (max < likelihood[j]) {
24     max = likelihood[j];
25     max_j = j;
26   }
27 }
28 Xm = X + max_j / 9;
29 Ym = Y + max_j / 3 % 3;
30 Thm = Th + max_j % 3;

```

ここで $Th1$ はレーザスキャナの角度分解能に合わせて整数値に離散化する。なお、高速化のため、離散化された角度に対応する正弦と余弦をそれぞれ配列 $c[]$, $s[]$ として事前に計算しておく。次に、4～19 行目において、この点を起点として並進および回転の近傍において、スキャン点とランドマークの重なり具合を評価する。レーザスキャナからは、各スキャン点までの距離が順次出力される (20～30 行目)。これを探索のための微小変位 (dx , dy , dth) を加味した直行座標系 (x , y) にする (11～12 行目)。この点における占有度 $m[x][y]$ に距離を乗じた値を、マッチング尤度として $likelihood[i]$ に積算する。ここで、距離を乗じるのは、放射状にビームを発光するレーザスキャナの特徴から、観測点の密度が距離に反比例するためである。各近傍における尤度を計算したあとに、最も尤度が高い近傍に対応する max_j を探索することで、マッチングによる推定位置 (Xm , Ym , Thm) が出力される。これを観測出力とした拡張カルマンフィルタを用いることで、推定値 (X , Y , Th) と共分散を更新する。なお、観測出力雑音の共分散行列は、尤度

をもとに生成しているが、具体的な実装は年度によって異なり、それぞれ 5.2.4 項と 5.3 節において述べる。

Table 1 に示した例では、各軸 3 自由度ずつ、合計 27 点の隣接する点の尤度を評価していることになる。予備実験の結果から、地図の連続性が保たれている限り、一度正しくマッチングすれば誤ったランドマークにマッチすることはなかった。Intel Atom N270 プロセッサ (1.6 [GHz]) で 2,162 点のスキャナデータを処理した場合に要する時間の実測値は、4 [ms] であった。実際にはこれを 10 [Hz] の頻度で実施している。

探索近傍を絞るすぎると、初期値の誤差の大きさによっては、正しく収束しない。探索範囲を増加させると計算量がその 3 乗で増大するため、格子サイズを 100 [mm]～400 [mm] の範囲で可変としておき、初期化時は探索する近傍点数を増やすと共に、大きな格子を用いて高速化し、高い尤度が得られたら格子を細かくすることとした。

GPS 出力についても、拡張カルマンフィルタによりオドメトリと組み合わせることで自己位置を推定している。GPS コンパスは誤差楕円を出力する仕様であるので、観測出力の共分散行列の算出に用いている。マップマッチングによる自己位置と GPS によるものは、常に同時に利用するのではなく、条件により使い分けている。実装の詳細は参加年度により異なるので、第 5 章で詳述する。

3.3 誘導制御と障害物回避

つくばチャレンジの枠組みでは、ロボットが通過すべき大まかな経路が具体的に指定されているため、ウェイポイントを指定することで誘導する。筆者らが提案している軌道誘導制御のための操舵方式 Sensor Steering Mechanism (SSM) [9] のセンサアームを仮想化し、ウェイポイントを結ぶ直線上に漸近収束させる誘導方式を採用している。これは Line of sight とよばれる誘導方式に類似している。ウェイポイント誘導は回避すべき障害物が存在しない場合にのみ実施され、障害物が存在する場合は回避行動が優先される。障害物の回避は、車体前方に車体よりも大きい矩形領域が存在できる方位を探索する方法 [10] の考え方に基づき、スキャンデータの座標変換と同時に矩形候補を計算する実装により反復処理を減らしている [4]。前方空間の余地に応じて矩形の長さを変えられるようにしており、歩行者が静止障害物の回避を開始するとされる距離 [11] に基づき、矩形の最大長さは 5 [m] とし、なるべく人に驚異を与えないように回避を開始する。回避方向は、ウェイポイント誘導の方向に最も近い方向に回避するものとする。大域的な軌道計画問題を解いていないため、袋小路を事前に回避することができないが、回避解が見つからない場合は一定距離後退し、その地点を仮想的な障害物として回避することで脱出を試みる。

4. 人の挙動と対策

4.1 ロボットの動作に影響を与える歩行者の挙動

歩行者は、ロボットの進路上に存在する場合に行動の妨げとなるほかに、レーザスキャナの視界を遮蔽してマップマッチングに影響を与える可能性を持っている。視界が部分的に遮られた場合、地図に記録されているランドマークが十分な量だけ捉えられていればマッチングへの影響は抑えられる。注意しなければ

ばならないのは、歩行者等の本来地図にない障害物が、地図上に記録されたランドマークに誤ってマッチしてしまうミスマッチである。これにより自己位置推定が破綻すると、現状の局所探索による枠組みでは復帰することが困難である。これは、歩行者が障害物から一定以上の距離を保つという仮定を置けば、マッチングにおける探索範囲を狭めることで対処できる。ロボットに特段の関心を示さない一般の歩行者についてはこの仮定が成り立ち、つくばチャレンジコースにおける試走においても既述の自己位置推定法で問題は生じなかったが、ロボットに関心を寄せる人が多数群がった場合には状況が大きく異なる。つくばチャレンジ2009の本走行のスタート時において生じた現象は次のようなものであった。まずロボットの周囲の視界を完全に覆うような人垣が地図のランドマークに沿うように並び、ロボットが通過するとロボットに追従するように移動を開始した。人垣が地図に記録されているランドマークと類似した陣形を保っていたため、陣形が地図にマッチしてしまった。具体的な挙動は実験データとともに5.2.4項で詳述するが、つくばチャレンジのスタート時という特別な状況に限らず、将来生活空間においてロボットが活躍するようになった場合であっても、ロボットに関心を持つ人が多数いれば再現し得る現象であり、対策が必要である。

4.2 人垣の対策

本質的な改善方法として挙げられるのが、不確定要素の強い人の背丈の高さのスカンデータマップマッチングから除外するというものである。実際、つくばチャレンジ2009において課題達成した5チームのうち、4チームがレーザスカナを傾斜させたり揺動させたりして人の背丈よりも高い位置にあるランドマークを捉える方法を採用していた。一方で、GPSやオドメトリが正常であったことから、適切な取捨選択を行えば問題を回避できたことも明らかである。水平スカンのみでロボスタ性を実現すれば、レーザスカナの改造や追加が不要である。そこで水平スカンデータを使用しつつ、マッチングの確からしさを評価する方法を実装し、ロボスタ性を高めることを目標とした。

カルマンフィルタに基づく融合の問題点の一つは、観測雑音やプロセス雑音の共分散が既知であるという前提である。実際に真値を得ることは難しく、共分散は実用上チューニングパラメータとして扱われることが多い。2009年においては、マッチングの尤度に基づいて共分散を定めていたが、これが現実の確からしさを正しく反映しなかったため、破綻を招いた。

パーティクルフィルタも本質的に類似の問題を抱えており、前節のような現象が発生すると、観測に基づく尤度評価の段階で誤った情報の確率が高められる可能性がある。ミスマッチが発生する状況では、以下の2点が重要であると考えた。

(1) 汚染されていない推定値の保持

何らかの方法により一度ミスマッチの存在が確認された場合、直前の推定値はすでにミスマッチに由来する誤差を含んでいると仮定すべきである。一度推定値が真値から外れてしまうと、さらなるミスマッチを呼び起こす可能性があるため、ミスマッチに由来する誤差（汚染と呼ぶ）を含んでいないと推定される汚染されていない推定値を保持しておき、ミスマッチが判明した時点で推定値を切り替えるも

Table 2 Pseudo code for multiple odometry

```

1 dist = 0; odo_no = 0;
2 while (1) {
3   for (i = 0; i < N; i++) {
4     tho[i] += dTh;
5     xo[i] += cos(tho[i]) * dL;
6     yo[i] += sin(tho[i]) * dL;
7     dist += dL;
8   }
9   if (dist > d) {
10    xo[odo_no % N] = Xm;
11    yo[odo_no % N] = Ym;
12    tho[odo_no % N] = Thm;
13    odo_no++; dist = 0;
14  }
15 }

```

のとする。

(2) ミスマッチは時間的・空間的に、即座に解消しない

ミスマッチは人垣のような動く壁によって引き起こされる場合のほか、静的な障害物の変化によっても引き起こされる。これらは時間的あるいは空間的に即座に解消するとは考えられず、しばらくの間はマップマッチングそのものを中断してしまうほうが好結果が得られる。

そこで通常は3.2節に示す方法により自己位置推定を行い、ミスマッチが検出されたら、汚染されていない推定値に切り替えたあと、一定の距離あるいは時間をオドメトリのみ、あるいはGPSとオドメトリの融合に基づき航行し、やがて状況が改善されていることを期待して再び通常に戻るという方策を採用した。

汚染されていない推定値を維持するため、位置姿勢パラメータ変数 $xo[i]$, $yo[i]$, $tho[i]$ を N 組用意しておき、オドメトリによって得られる微小推進距離 dL , 微小姿勢変化 dTh に基いて積分する。積分誤差を解消するため、一定距離 d 走行するごとに、1組のオドメトリ変数をそのときの推定値 Xm , Ym , Thm によって再初期化する。これにより、各オドメトリ変数は距離 dN ごとに順次最初期化される。以上の処理に対応する擬似コードを **Table 2** に示す。

次に、マップマッチングの破綻の検出について述べる。上記オドメトリ変数とマップマッチングの出力の比較により判定されるが、比較には注意が必要である。オドメトリ変数には、累積された誤差と地図自体の歪みに由来する誤差が含まれており、正しくマッチングしている場合であってもミスマッチと誤検出する恐れがあるためである。例えば、凹凸路面を走行するなどしてオドメトリに乱雑な誤差が含まれる場合、 N 個のオドメトリ変数の間にバラツキが生じる。この場合、オドメトリの正確さは低下していると捉えるべきで、マップマッチングの健全性の判定にはより大きな誤差を許容すべきである。地図に歪みが存在する場合も、マップマッチングによる修正がオドメトリと相対的な誤差を生み、オドメトリ変数のバラツキが増加する。

Fig. 3 Multiple odometry periodically reset to the latest measurement

この場合もオドメトリの正確さが劣っているものとすべきである。そこでオドメトリ変数の分散共分散行列

$$R_o = \begin{bmatrix} \text{var}(x_o) & \text{cov}(x_o, y_o) & \text{cov}(x_o, t_h o) \\ \text{cov}(x_o, y_o) & \text{var}(y_o) & \text{cov}(y_o, t_h o) \\ \text{cov}(x_o, t_h o) & \text{cov}(y_o, t_h o) & \text{var}(t_h o) \end{bmatrix} \quad (1)$$

を求め、オドメトリ変数の平均値ベクトル ($\bar{x}_o$, $\bar{y}_o$, $\bar{t}_h o$) と観測出力ベクトル ($\bar{x}_m$, $\bar{y}_m$, $\bar{t}_h m$) のマハラノビス距離によって評価する。すなわち、楕円体の不等式

$$x^T R_o^{-1} x < n, \quad x = [\bar{x}_m - \bar{x}_o, \bar{y}_m - \bar{y}_o, \bar{t}_h m - \bar{t}_h o]^T \quad (2)$$

が成り立つ場合に観測が正常であると判定する。ここで var は 1 変数の N サンプルの分散を、 cov は 2 変数間の N サンプルの共分散を、 n は標準偏差の何倍の誤差を許容するかを表している。すなわち $n = 2$ の場合、 2σ までの誤差を許容する。 R_o は、予測に誤差がないと仮定した場合にマップマッチングによる観測出力雑音の共分散を実際のサンプルから近似的に計算したものと捉えることができ、マッチングの確からしさを他の手段により評価したものと言える。一般には、これをチューニングパラメータとして扱うことが多いのに対し、現状に即した適応的な対応が可能となる。Fig. 3 に概略を示す。2 本のオドメトリの軌跡が描かれており、丸印 (白色および黒色) は観測出力を表している。等確率楕円 (灰色) の内側に観測されていれば、マッチングは正常であると判定される。

本手法の共分散を推定する部分については、観測および予測における雑音の共分散を直接推定する適応カルマンフィルタ [12] において、残差の履歴に基づいて共分散の推定を行う手法に着想を得ている。複数センサを使用したエラー検出については、異なる予測モデルや観測を用いた複数のカルマンフィルタを同時並行的に走らせ、それぞれの共分散を比較する方法 [13] の実装例と見ることができる。

4.3 予備実験

防衛大学校内において、ミスマッチ検出アルゴリズムの有効性を検証した。Smart Dump のホイールオドメトリは、少な

Fig. 4 Scanned points (black) plotted over the map (gray)

くとも 10 [m] 程度であれば、単独航行が可能である。そこで d を 1 [m]、 N を 10 個とした。これにより各オドメトリ変数は 10 [m] ごとに再初期化される。また、許容誤差の閾値 n は 3 とした。試験走行の結果、この値であれば地図や路面の誤差に起因する誤検出は見られなかった。また、オドメトリのみあるいは GPS とオドメトリの融合推定値による走行は、10 [m] 走行するか 20 [s] 経過するかいずれかの条件により停止した。

Fig. 4 は、起伏や傾斜のある路面を走行したときの地図データ (灰色) とスキャンデータ (黒色) を示している。破線は推定された軌跡を現しており、実際の軌跡とは異なることに注意する。地図のランドマークとして記録されているのは主に樹木や灌木である。当該地点は、ミスマッチのために自己位置推定が破綻し、通過が困難な場所であった。Scene 1 では、スキャンデータがよく合致しているように見えるが、実際には車体の傾斜によりスキャンデータが図の上方に移動しており、マッチングの結果、推定値が図の左下方に引き寄せられた状態を示している。Scene 2 においてミスマッチが検出され、それまでの汚染されていない推定値への切り替えが発生した結果、ロボットの軌道が修正され、ミスマッチの原因となった輪郭が地図上のランドマークから離れた。以後、10 [m] の間オドメトリのみで走行する。Scene 3 では、上り傾斜に差し掛かったため、後方 (図の上部) には地表が検出されている。また、オドメトリの誤差のため、スキャンデータが地図と完全に合致していない (図中楕円部)。Scene 4 において一時停止をしてマッチングを再開し、正常な状態に復帰している。

5. 実験走行

本章では各年度におけるつくばチャレンジの取り組みと、参加によって得られた知見を時系列順で示す。

5.1 つくばチャレンジ 2008

防衛大学校内における試験走行およびつくばにおける試走会のいずれにおいても、トライアル区間を完走できる水準に達しておらず、トライアル不通過に終わった。具体的には、障害物を回避しようとして通常とは異なる方向に操舵した結果、植え込みの 30 [mm] 程度の段差に乗りあげてしまった。歩行者にとつ

てはわずかな段差であったが、ロボットには乗り越えられないものであった。路面状況を監視して自律的に回避行動を取るか、安全が確認できているルートから大きく逸脱してはならないという教訓を得た。これらは校内における試験によりあぶり出すことのできる不具合であり、事前の試験を十分に重ねておくべきであることが最大の教訓となった。

5.2 つくばチャレンジ 2009

5.2.1 事前準備

2008 年の反省のもと、事前に校内でのテスト走行を徹底することとした。マップマッチングを利用した基本的な走行機能は、つくばチャレンジ 2008 終了後の 12 月には確認できたが、以降、歩行者をはじめとする様々な移動障害物、駐車車両を回避する問題、緩やかに傾斜した路面、雑草の植生変化の影響などを多岐に渡って確認した。延べ走行距離は 148 [km] に及んだ。また、つくばチャレンジコースにおける試走会においては、延べ 51.4 [km] の走行を行った。その結果、信頼性を高めることができた。

カルマンフィルタによって推定値を統合する際に用いられる観測出力雑音の共分散行列は、正規化された尤度

$$\frac{\text{ランドマーク上の点数}}{\text{ランドマーク上および近傍上の点数}} \quad (3)$$

に基づき生成した。これにより、地図のランドマーク付近に存在しない点は評価から除外され、スキャン点そのものの多寡の影響が軽減される。正規化尤度に重み係数を乗じた値と GPS の誤差楕円を比較し、正規化尤度が高い間はマップマッチングを優先し、尤度が低下した場合に GPS コンパスとオドメトリを使用したカルマンフィルタの推定値を初期値として、より粗い格子を使用して近傍領域を拡大してマッチングの初期化を繰り返すこととした。

路面状況を監視するため、三次元レーザスキャナ（日本信号製）を搭載した。視野が十分に広くなく、凹凸検出の分解能が 20~30 [mm] 程度であったため、回避が必要な障害物をすべて検出することはできなかったが、レーザの反射強度を活用することで、芝生と舗装路の識別が可能であることを確認した。これにより、距離情報だけでは判別の難しい舗装路と芝生境界を明瞭に捉えることができるようになった。使用したセンサの路面反射強度と距離の関係は既報 [4] において報告している。

5.2.2 試走会

8 月 1 日、9 月 11 日、10 月 23 日、11 月 7 日、11 月 18 日、11 月 19 日の 6 回にわたり参加した。10 月 23 日までの試走会においては、地図データの生成とその動作検証を繰り返した。11 月 7 日の試走会において、10 月 23 日に採取したスキャンデータに基づいて生成した地図を使用して初めての完走を果たした。その後は 4 回連続して完走することができた。障害物の少ない芝生のある広場においてマップマッチングの正規化尤度が低下し、他のロボットや歩行者などの障害物の存在によってはコースを逸脱することがあったため、以後の試走会においては、広場における通過成功率向上のため、GPS と地図の遷移に関する重みパラメータの調整を中心に実施した。

5.2.3 トライアル

試走会に比べて、スタート地点のあるつくばエキスポセンター敷地内（以後、スタートエリアと呼称）内に多数の観客が集まっ

Fig. 5 Spectators inside the start area (Trial, 2009)

Fig. 6 Spectators outside the start area (Trial, 2009)

Fig. 7 Spectators inside the start area (Final, 2009)

Fig. 8 Spectators outside the start area (Final, 2009)

ていたため (Fig. 5)、スタート直後のマップマッチングの正規化尤度が低下したが、スタート地点を抜けると樹木等の既知のランドマークが見えるようになり (Fig. 6) マッチングの正規化尤度が試走会と同レベルにまで回復し、トライアルは達成し、そのまま完走することができた。

5.2.4 本走行

本走行ではトライアルよりもさらに多くの観客が人垣をなし、スタート地点の先が見通せない状態であった (Fig. 7)。スタート地点を抜けてもなおレーザスキャナの視野を遮るほどの人垣が続いており (Fig. 8)、大きな問題が発生し始めた。このとき

Fig. 9 Scanned points overlaid on the map based on wrong estimate (Map matching)

Fig. 10 Scanned points overlaid on the map based on GPS reading

のスキャンデータを地図データの上に重ねたものを **Fig. 9** に示す。地図データを灰色で、スキャンデータを黒色の点で表す。 $t = 18[s]$ が **Fig. 8** に対応している。この時点では、図中楕円で囲んだ既知のランドマークが合致しており、自己位置推定が正確であると判断できる。 $t = 22[s]$ においても、わずかながら地図のランドマークが捉えられており、図から判断する限り大きなずれは生じていない。これだけの視野が遮られているながらこの時点で自己位置推定が破綻していないのは、マッチングにおいて地図のランドマークから離れた点を評価対象としていないためであり、実装された手法が部分的な遮蔽に対してロバスト性を有していることを示している。しかし、この直後にロボットの後方を観客が取り囲み、人垣により視界のほとんどが遮蔽されてしまった。地図のランドマーク近傍に立っていた人によりマッチングの正規化尤度が低く評価され、GPS の測位結果を初期値とする、より粗い格子サイズで探索範囲を広げたマッチングが開始された。その結果、人垣の $\cap$ 型の陣形を地図のランドマークに合致させるように図の上方に引き寄せる効果が作用した。 $t = 26[s]$ の時点の推定位置は、 $t = 22[s]$ に比べて後退している。以後、後方の視野を遮っている観客が、ロボットとともに移動するたびに、地図上のランドマークに合致させようとする作用が働き、結果としていつまでもスタートエリアの出口付近から移動していないものと推定された。**Fig. 10** は **Fig. 9** と同じ時刻のスキャンデータを GPS 測位データに基づいて地図に重ねたものである。ここでは空が開けており、測位状況は

Fig. 11 Actual trajectory (GPS) vs. wrong estimate (Map matching)

誤差楕円が $0.5[m]$ 程度と良好であったため、実際の位置を表しているとみなせる。ロボットは、ウェイポイントを結ぶ線分に漸近収束するように誘導されるが、誤った推定自己位置に基づき、**Fig. 9** の $t = 26 \sim 35[s]$ に Direction として示した矢印の方角に進もうとする。結果として図中下方に直進を続け、コースから逸脱した。**Fig. 11** は、マップマッチングと GPS の測

位結果の軌跡を示したものである。GPS の軌跡が図中下方で途絶えているが、先方に池があったため走行不能となった。

これだけの多数の人がロボットを取り囲むように特定の陣形を保って追従してくる状況は、ありふれた日常とはいえないが、将来自律移動ロボットが人の生活空間の中で使われるようになった場合に起こりえる現象を浮き彫りにしており、回避手段を用意しておく必要があるという教訓を与えており、第4章で提案するアルゴリズムを導入する動機となった。2009年のログデータに対し提案アルゴリズム適用した結果を **Fig. 12** に示す。ミスマッチの検出は、推定自己位置が大きく乱される以前に検出できている。これは、**Fig. 9** における $t = 18$ [s] の図に対応する。これより、大きく逸脱する前に検出できることが確かめられた。

5.3 つくばチャレンジ 2010

前年の反省を踏まえ、第4章に示したアルゴリズムを導入した。また、オドメトリとマップマッチングの融合において、マッ

Fig. 12 Mismatch detection applied to 2009 data: Scanned points (black) plotted over the map (gray)

プマッチングの共分散行列は、正規化尤度ではなく、マップマッチングの正規化前の尤度そのものの分散を直接用いた。

5.3.1 実験走行・トライアル走行・ファイナル走行

実験走行は、10月15日、11月6日、11月16日、11月17日の4回実施した。10月15日は地図作成のためのデータ収集を実施し、11月6日には作成した地図データによる自律走行を試みた。初回はデータケーブルの接触不良のため中断したものの、以後4回の往復の完走に成功した。いずれもミスマッチの検出は認められなかった。11月16日においては4回の試み、11月17日は雨天のため2回の試みであったがいずれも完走した。以上の実験走行の中で、片方のレーザスキャナが停止したままの状態や、両方のレーザスキャナのデータ更新を一時停止してマップマッチングの破綻を発生させた場合についても完走を確認した。トライアル走行も実験走行と同様に、制限速度に近い速度で走行し、3分58秒の所要時間で課題を達成した。

ファイナル走行は21分20秒の所要時間で完走することができた。一時停止を含め、実走行距離 1,176 [m] を平均時速 3.3 [km]

Fig. 13 Scanned points in final run of 2010 challenge

Fig. 14 Standard deviation of the odometry variables (Top) and scanned data at 70 [m] (Bottom left) and 730 [m] (Bottom right) where the deviation has a large peak

で走行したことになり、歩行者空間において活動するのに実用的な速度が維持されていたといえる。人垣の対策を導入しているものの、スタート後 30 [m] はオドメトリのみによる走行とした。**Fig. 13** は、スタート直後のロボット周囲の状況をスキャンデータにより示したもので、2009 年における Fig. 10 の $t = 35$ [s] とほぼ同じ地点におけるものである。人垣は疎らであり、実際にはマップマッチングを有効としてもミスマッチが発生しなかった可能性が高い。**Fig. 14** はファイナル走行の全行程における、走行距離に対するオドメトリ変数の東西方向、南北方向の標準偏差の変化と、偏差が局所的に大きくなった地点におけるスキャンデータを示したものである。ほとんどの箇所において標準偏差は 0.1 [m] 以下であるが、2 箇所においてピークが見られる。いずれも、進行方向に平行で平坦な壁面が存在し、そのほかにわずかに樹木等の面積の小さいランドマークが存在する空間であることが共通している。このような状況では、マップマッチングが進行方向に不正確となり、わずかな外乱によって大きく変化する。複数回の試験走行においても同様の傾向が再現されており、Fig. 14 は地図の品質評価に用いることができる。

2010 年の挑戦では、動く人垣によるミスマッチへの対策を講じ、結果として二次元のマップマッチングを使用した自己位置推定の信頼性を上げることに成功した。ファイナル走行ではその機能が活かされる場面に遭遇しなかったが、マップマッチングの確からしさを事後評価することができるようになった。

6. ま と め

本論文では、つくばチャレンジへの 3 回の参加を通じて得られた教訓と講じた対策について述べた。特に、2009 年のファイナル走行において遭遇した動く人垣によるミスマッチ現象は事前に経験することが難しく、実世界環境の事象として貴重な体験であった。動的障害物による影響はレーザスキャナの二次元地図に基づく自律移動では不可避な問題であり、それに対する具体的な解決策を示した。

多数の参加により問題の共有ができるつくばチャレンジの枠組みが果たす役割は大きく、今後もこのような機会が継続的に

得られることを期待したい。特に、屋内移動ロボット分野で行われているようなデータ共有 [1] が、動的障害物を含む歩行者環境に関して進めば、この分野の発展がより促進されるものと考えられる。

謝 辞 本研究の一部は、科学研究費補助金（基盤研究（C）No. 22560237）の助成を受けて遂行された。

参 考 文 献

- [1] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss and A. Kleiner: Slam benchmarking webpage, <http://ais.informatik.uni-freiburg.de/slamevaluation>, 2009.
- [2] M. Buehler, K. Iagnemma and S. Singh (Eds.): The DARPA Urban Challenge: Autonomous Vehicles in City Traffic. Springer Verlag, 2010.
- [3] 油田信一, 水川真, 橋本秀紀: “つくばチャレンジのねらいと成果”, 計測と制御, vol.49, no.9, pp.572–577, 2010.
- [4] 滝田好宏, 伊達央: “Smart Dump 3 で実現した自律機能”, 計測と制御, vol.49, no.9, pp.636–639, 2010.
- [5] C. Estrada, J. Neira and J.D. Tardos: “Hierarchical SLAM: real-time accurate mapping of large environments,” IEEE Trans. Robotics, vol.21, no.4, pp.588–596, 2005.
- [6] A. Elfes: “Using occupancy grid map for mobile robot perception and navigation,” IEEE Trans. Computer, vol.22, no.6, pp.46–57, 1989.
- [7] S. Rusinkiewicz and M. Levoy: “Efficient variants of the ICP algorithm,” Proceedings of 3-D Digital Imaging and Modeling, pp.145–152, 2001.
- [8] S. Thrun: “Probabilistic robotics,” Communications of the ACM, vol.45, no.3, pp.52–57, 2002.
- [9] 滝田好宏: “SSM を用いた軌道誘導車両の高速化”, 日本機械学会論文 C 編, vol.65, no.630, pp.620–629, 1999.
- [10] 大島章: “移動ロボットによる測域センサを用いたセンサベースナビゲーション”, つくばチャレンジ 2007 参加レポート, no.3, 2007.
- [11] 建部謙治, 辻本誠, 志田弘二: “回避行動開始点の判定と前方回避距離”, 日本建築学会計画系論文集, no.465, pp.95–104, 1994.
- [12] R. Mehra: “On the identification of variances and adaptive Kalman filtering,” IEEE Trans. Automatic Control, vol.15, no.2, pp.175–184, 1970.
- [13] P. Goel, G. Dedeoglu, S.I. Roumeliotis and G.S. Sukhatme: “Fault detection and identification in a mobile robot using multiple model estimation and neural network,” ICRA 2000, vol.3, pp.2302–2309, 2000.

伊達 央 (Hisashi Date)

2003 年東京工業大学大学院理工学研究科機械制御システム専攻博士課程修了。同年防衛大学校情報工学科助手、2007 年同助教となり現在に至る。蛇ロボットおよび移動ロボット等のロボットの制御の研究に従事。博士（工学）、IEEE、計測自動制御学会、日本機械学会の会員。（日本ロボット学会正会員）

滝田好宏 (Yoshihiro Takita)

1979 年東京都立大学機械工学科卒業、1988 年東京工業大学より博士（工学）号授与。同年防衛大学校機械工学科教室助手。1990 年同講師。1996 年同助教授。2002 年同情報工学科教授となり現在に至る。振動制御および移動ロボットの研究に従事。日本機械学会、自動車技術会の会員。

（日本ロボット学会正会員）