

地面画像を利用した自己位置センシング

Self-Position Estimation using Video Image of the Surface of the Earth

○学 豊田洋介 (鈴鹿高専)
正 白井達也 (鈴鹿高専)

外 山田泰生 (大阪大学)
正 富岡 巧 (鈴鹿高専)

Yosuke TOYODA, Suzuka National College of Technology, h17d11@ed.cc.suzuka-ct.ac.jp
Yasuo YAMADA, Osaka University
Tatsuya SHIRAI, Suzuka National College of Technology, shirai@mech.suzuka-ct.ac.jp
Takumi TOMIOKA, Suzuka National College of Technology

An autonomous mobile robot needs to always obtain a self-position while it moves towards a target position. Conventional methods based upon recognition of landmarks by using image processing have risk of misrecognition by disturbance. It is difficult that the robot obtain the self-position precisely by using external area sensors such as GPS (Global Positioning System), ultrasonic range sensor, and so on. In this paper, we propose an estimation algorithm which obtains the self-position by using a ground image. We developed an application program based on the proposed algorithm, and verified a feasibility of the proposed method.

Key Words: Mobile robot, Self-position estimation, Ground image

1. 緒言

代表的な遠隔操縦式ロボットとして、原子力発電所内のメンテナンス用ロボットや深海探査ロボットといった極限作業用ロボット、爆発物処理ロボットが既に実用化されている。近年は災害レスキューロボット、地雷撤去ロボットの開発が注目を集めている。古くから工場等で利用されている無人搬送車は床面に貼られた専用のテープや床下に埋設された磁気マーカ等を頼りに通路内を移動するため、あらかじめ整備された環境下でしか使用できないのに対して、GPS (Global Positioning System : 全地球測位システム) や超音波センサ、レーザーセンサ、CCD カメラ等の外界センサを用いて自分自身の位置や周囲の環境を認識し、状況を判断して自律的に移動する自律移動型ロボットも数多く研究・開発され、一部は実用化されている。愛知万博でも活躍した掃除ロボットや案内ロボット、警備会社が開発した警備ロボットなど、単なる輸送目的以外の用途への利用が広がり始めている。

自律移動型ロボットが現在地から目的地へ移動するには、自分自身の現在位置 (以下、自己位置) を常に把握し続ける必要がある。自己位置を把握する手段には前述の GPS、駆動輪の回転数に基づいて移動量を推定するオドメトリ、ジャイロセンサを用いる慣性航法などが用いられてきた。GPS はある絶対座標系における絶対座標、方向を得られるが、1~10[m]程度の位置誤差が生じる上に、GPS 衛星から発信される微弱な電波を受信するのが困難な地下や屋内では利用できない。オドメトリは車輪が滑り易い不整地などでは十分な精度を得ることが難しい。慣性航法は累積誤差の問題がある。標識などを CCD カメラで認識することで自己位置を特定する方法も古くから研究されているが、確実性の低さおよび未知の環境への適用に関して課題が残る。

2. 研究の目的

本研究では、図 1 に示すように、移動ロボット直下の地面の画像 (以下、地面画像) を利用して自己位置を計測する自己位置推定アルゴリズムを提案する。類似の研究として、GPS を利用できない屋内での自己位置を認識するために、天井灯をランドマークとして用いる手法が提案されている[1]。天井の画像は屋内でしか捕捉できないのに対し、地面の画像は屋

内、屋外問わず CCD カメラで捕捉できるためナビゲーションに用いるのに適している。地面画像を用いた自己位置推定方法は GPS より得られた誤差を含む大域的な位置情報を局所的に補正することを想定している。提案する自己位置推定アルゴリズムをプログラムに実装し、実装したプログラムにより推定した自己位置と実際の自己位置を比較し、本アルゴリズムの実現可能性を検証する。

3. 地面画像を利用した自己位置推定法

3.1. 提案する自己位置推定法の概要と特徴

以前より、ロボットに搭載したカメラの映像を画像処理し、特定の目標物、目印、標識などの特徴的なマーク (以下、ランドマーク) を認識することで自身の相対的な位置・姿勢を取得する自己位置推定法が広く研究されている。従来法の最大の難点は、ロボットの周辺のどこに存在するのか不明なランドマークをカメラの視野内に捕捉しなくてはならない点である。それに対して地面は移動ロボットの下に常時ほぼ一定距離に存在する。さらに、地面画像から得た地面の色や模様の情報を利用することで、地面の状態 (乾いている/濡れている/穴があいている/砂地である/アスファルトである/草地である等) を獲得できる。地面の状態を把握するこ

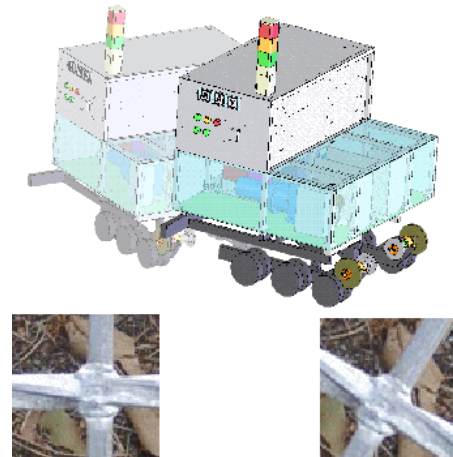


Fig.1 Self-position estimation by ground image

とで、走行に不適な場所を自律的に回避することも可能となる。もし地面にラインが塗装されているならば、ラインを利用した正確な走行も可能である。

本自己位置推定法ではフレーム間の相対的な移動量および角度変化量を時系列に累積加算して自己位置を推定する。CCDカメラから得られた最新の画像（以下、最新画像）と最新画像よりも過去の画像（以下、比較画像）を比較する。最新画像の1フレーム前の画像を以下、直近画像と呼ぶ。自己位置推定のために累積計算を繰り返すことで自己位置情報に累積誤差が蓄積する。これを補正するために本自己位置推定法では、1フレーム前の画像だけではなく数フレーム前の過去の画像も比較画像として用い、最新画像と定期的に比較することで自己位置情報を補正する。この直近画像よりも過去の画像を以下、後方参照画像と呼ぶ。

本研究では二つの画像の変化量を取得する手法として、単純な相互相関を用いる。指定した範囲内で最新画像を比較画像に対して並進させ、各並進位置で最新画像を一定角度単位で回転させて相関度を計算する。この手法は単純に行くと膨大な計算時間を要する。探索範囲内の水平方向および垂直方向の並進量を T_x [pixel], T_y [pixel], それぞれの刻み幅を ΔT_x [pixel], ΔT_y [pixel] とする。回転量を θ_{\max} [deg], 刻み幅を $\Delta \theta_{\max}$ [deg] とすると、二つの画像間の相関度をチェックする回数 N_{scan} [回] は、式(1)のようになる。

$$N_{scan} = \left(\frac{T_x}{\Delta T_x} \right) \left(\frac{T_y}{\Delta T_y} \right) \left(\frac{\theta_{\max}}{\Delta \theta} \right) \quad (1)$$

最新画像と比較画像の時間差が短い、例えば二つの画像間のズレ量が $T_x=50$, $T_y=50$, $\Delta T_x=\Delta T_y=1$, $\theta_{\max}=30(\pm 15)$ と想定しても、 $N_{scan}=7.5 \times 10^4$ [回] の相互相関度の計算が必要である。1回の相互相関度の計算に要する時間は並進/回転する画像の面積に比例する。そこで、図2(a)のように画像全体を比較するのではなく、図2(b)のように最新画像中の比較的小きな領域（以下、テンプレート）を選出してテンプレートと相関の高い領域を比較画像中で探索すれば計算量を大幅に軽減可能である。そこで、比較に用いる小領域を最新画像中から選出するために、図3のように最新画像を横×縦= $n_x \times n_y$ の複数の領域に分割する。左から i 番目、上から j 番目の領域は $R(i, j)$ と表記する。図3において W_p , H_p はそれぞれ分割によって得られた領域（以下、分割領域）の水平方向の幅と垂直方向の高さとする。テンプレートと比較画像領域外との相関度チェックを回避するため、画像中の外周に水平方向、垂直方向にそれぞれ W_M , H_M の枠を設ける。 N_{scan} 回の相関度のチェックに要する処理時間 T_{scan} は図2(a)のような全領域の探索に要する時間を T_{all} とすると、

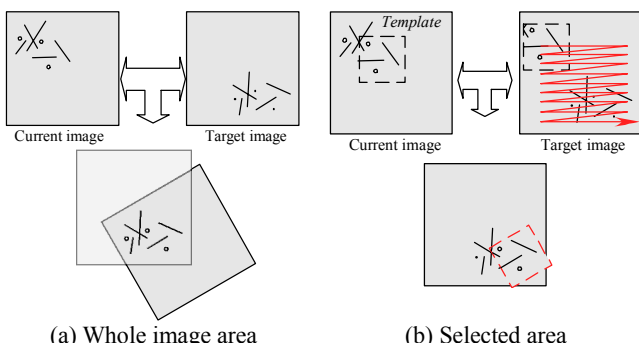


Fig.2 Difference of computational complexity for comparison

$$T_{scan} = \frac{W_p H_p}{W_o H_o} T_{all} \quad (2)$$

となる。 $W_o=720$, $H_o=480$, に対して $W_p=50$, $H_p=40$ とすると、 $T_{scan} \approx 0.006 T_{all}$ に低減される。

3.2. テンプレートの抽出

最新画像中の $n_x \times n_y$ 個の分割領域の中から探索領域に最も適した領域を評価関数に基づいて一つ選出する。図4(a)(b)に示すような無地の領域や微小ノイズからなる画像は原理上、変位量を計測できない。図4(c)(d)のような繰り返し模様からなる領域は誤認識する可能性が高いのでテンプレートとして好ましくない。

各領域の特徴を定量的に数値化するための評価関数を複数個用意し、その総和が最も高い領域をテンプレートとして選出する。各評価関数が返す評価値をそれぞれ E_1, E_2, \dots, E_n とし、定量化に寄与する影響の割合をそれぞれ、 $\alpha_1, \alpha_2, \dots, \alpha_n$ とすると、ある領域の特徴 E は式(3)のように定量化される。

$$E = \alpha_1 E_1 + \alpha_2 E_2 + \dots + \alpha_n E_n \quad (3)$$

移動ロボットが走行中の環境に応じて各評価関数のテンプレート選出に与える影響の割合 $\alpha_i (i=1 \sim n)$ を切替えることで、環境の変化に強い自己位置推定が可能となる。本報告では画像の水平方向にある程度の長さ単位で色調変化を有す画像で最も高い評価値を得られる評価関数を考案した。以下、この評価関数を評価法1Hと呼ぶ。評価法1Hは二値画像を対象に用いる。評価法1Hは、評価対象領域中の水平方向に連続する1のデータ群の数を返す評価関数であり、図5(a)のような領域よりも図5(b)のように、水平方向に十分な大きさを持つパターンが含まれる領域で高い評価値を得られる。

分割領域中に存在する白色値 (=1) の水平方向の基準となる連続量を Cr (任意) とする。評価法1Hの評価関数を式(4)に示す。現状では提案する評価関数は E_{1H} のみなので $E = E_{1H}$ とし、 E_{1H} が最大となる分割領域をテンプレートとして採用する。

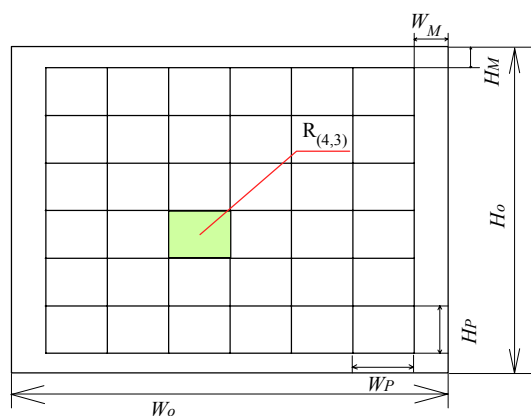


Fig.3 Domain division of image

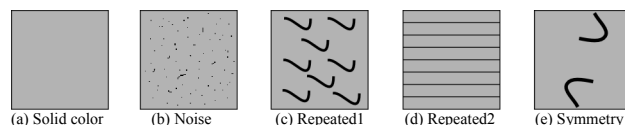


Fig.4 Unsuitable region for template



(a) Noisy pattern (b) Significant pattern
Fig.5 Domain that is provided by evaluation method 1H

$$E_{1H} = \sum_{x_{L0}=0}^{(Wp-1)-(Cr+1)} \sum_{y_{L0}=0}^{(Hp-1)-(Cr+1)} e_{1H}(x_{L0}, y_{L0}) \quad (4)$$

ここで $e_{1H}(x_{L0}, y_{L0})$ は式(5)で表される関数で、ある一つの水平方向のデータ列上の 1 の連続数が Cr 個以上のデータ群の個数を返す。式(5)中の $INT(\sim)$ はカッコ内の数値を小数点以下切り捨てる関数である。

$$e_{1H}(x_{L0}, y_{L0}) = INT \left(\frac{\sum_{i=1}^{Cr} Pix(x_{L0}+i, y_{L0})}{Cr} \right) (1 - Pix(x_{L0}, y_{L0})) \quad (5)$$

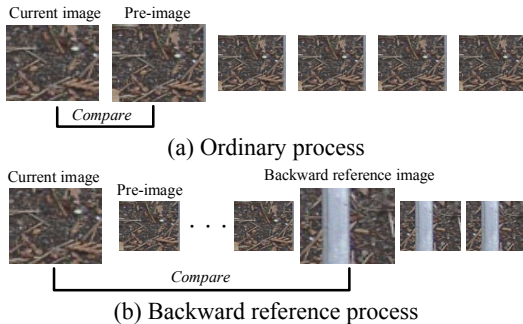
式(5)中の $Pix(X, Y)$ は任意の座標 $(X, Y)^T$ における二値データ('1' または '0')を返す関数である。図 6 は $W_p=25$ の時のある一つの水平線上の二値データの例である。この例では、 $e_{1H}(x_{L0}, y_{L0})=4$ である。

3.3. 後方参照処理

時間の経過とともに増大する累積誤差を軽減するために、定期的に後方参照画像と比較を行い、自己位置を補正する。以下、この処理を後方参照処理と呼ぶ。最新画像のフレーム番号を i とすると、直近画像のフレーム番号は $i-1$ 、 N_r フレーム前の後方参照画像のフレーム番号は $i-N_r$ である。通常は図 7(a)のように直近画像との比較を行なうが、最新画像のフレーム番号が N_r フレーム経過するごとに図 7(b)のように後方参照画像との比較を行う。後方参照で得られたフレーム間の CCD カメラの移動量、回転角は直近画像との比較を繰り返して得られた自己位置よりも累積誤差が少ないため、後方参照処理で得られた情報により現在の自己位置を補正する。



Fig.6 Example of binary data (horizontal line)



(b) Backward reference process
Fig.7 Reduction of cumulative error

4. 評価実験

4.1. 実験環境

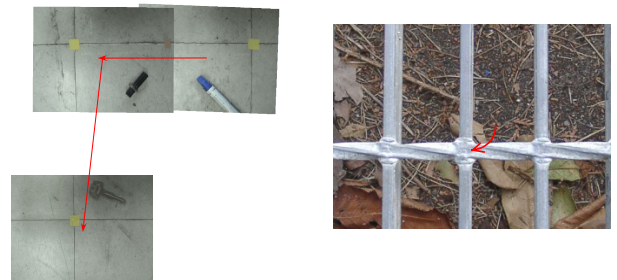
二種類の地面画像データを用いて自己位置推定アルゴリズムの実現可能性を検証した。図8(a)に示す映像Aは並進量を検証するための画像である。屋内の蛍光灯の下でデジタルビデオカメラにより撮影された動画を利用する。SonyDVコーデックでエンコードされた動画(AVI形式)を秒間30コマのビットマップ画像データとして出力した計151フレームの静止画像データである。撮影時のデジタルビデオカメラの移動経路は図11(a)の通りである。カメラ並進移動時はカメラの姿勢が変化しないように注意した。

図8(b)に示す映像Bは姿勢角度の変位量を検証するための画像である。屋外の自然光の下でデジタルカメラにより撮影した一枚の静止画像をレタッチソフトウェアで画像中央を回転中心として、1[deg]刻みで時計方向に90[deg]まで回転し、各姿勢角度ごとに出力した計90フレームの静止画像データである。

本アルゴリズムの実装の為に作成したプログラムの開発環境はBorland C++ Builder Professional Version 6.0 (ボーランド(株)), 本プログラム実行環境は, Dell DIMENSION 8400 (CPU: Pentium 4/3.20[GHz], Memory: 1.00 [GB]) である。画像サイズは横幅×縦幅=320×240[pixel]とする。表1に今回の実験のパラメータを示す。

4.2. 実験結果

テンプレートのサイズおよび後方参照処理に用いる N_r の二つのパラメータを表1のように変化させて、並進移動・回転への追従性の精度および処理速度についてどのように変化するかを観察した。なお、後方参照時に追従すべき領域を誤認させないために、 N_r に応じて T_x , T_y も変化させる。プログラムにより出力された自己位置データを図9の(a), (b)および図10の(a),



(a) Video image A (b) Video image B

Fig.8 Ground image for verification

Table.1 Experimental parameters

No.	A/B	θ_{max} [deg]	$\Delta\theta$ [deg]	N_r	T_x [pixel]	T_y [pixel]	n_x	n_y
E1	B	6	1	1	15	15	8	8
E2	B	6	1	1	15	15	6	6
E3	B	6	1	1	15	15	4	4
E4	A	6	1	1	15	15	8	8
E5	A	6	1	1	25	25	6	6
E6	A	6	1	1	35	35	4	4
E7	B	6	1	2	15	15	6	6
E8	B	6	1	3	15	15	6	6
E9	A	4	1	1	50	50	4	4
E10	A	4	1	2	50	50	4	4
E11	A	4	1	3	50	50	4	4

(b)に示す。

回転への追従性は、図9(a)に示すようにテンプレートのサイズが小さい場合、テンプレートマッチング時に追従すべき領域を誤認することが確認された。テンプレートのサイズを大きくした場合、回転への追従性能が向上することを確認した。後方参照処理なしでは低速度での回転に追従できないが、図9(b)に示すように、後方参照処理を行うことにより回転への追従性を向上できることが確認された。

並進への追従性は、図10(a)に示すようにテンプレートのサイズを小さくすると、回転時のデータと同様にテンプレートマッチング時に追従すべき領域を誤認識することが確認された。本実験において後方参照処理を行った場合、図10(b)に示すように並進への追従性能の低下が確認された。これはロボットの移動速度およびテンプレートマッチング時のテンプレート探索領域のサイズ ($T_x \times T_y$) に起因すると考えられる。

テンプレートサイズの大きさによって処理時間がどのように変化したかを計測した結果を表2に示す。テンプレートのサイズ以外のパラメータが同じであれば単位フレームあたりの処理時間はテンプレートサイズに比例することが表2中の v に示すように確認された。

Table.2 Relationship between size of template and Processing time

W_p [pixel]	H_p [pixel]	Area S [pixel ²]	Processing time per frame v [s/frame]	Ratio v/S
29	19	551	1.15	0.00208
39	25	975	1.94	0.00199
59	39	2301	4.57	0.00199

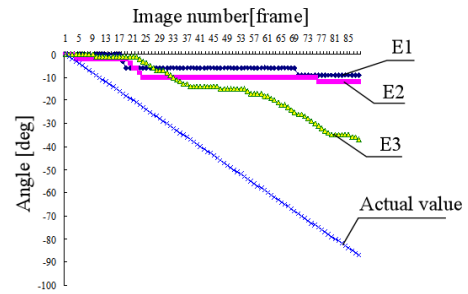
5. まとめと今後の課題

本研究では地面画像を用いた自己位置推定アルゴリズムを提案した。本アルゴリズムでテンプレート選出に使用する評価関数の一つである評価関数1Hを提案した。本アルゴリズムを検証するためのプログラムを実装し、アルゴリズムが所望の働きをすることを確認した。現時点ではリアルタイム処理は実現できていない。また、本実験において追跡に要する各パラメータを人為的に適当に操作しなければ、十分な追従精度を得られない。

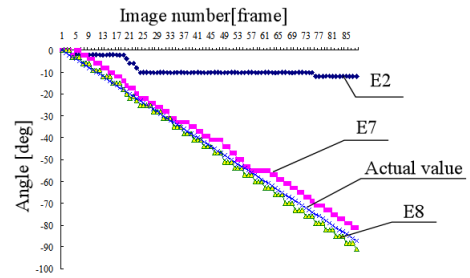
今後の課題はプログラムのアルゴリズムを工夫し、リアルタイム処理を可能とする。地面の種類や環境光、ロボットの移動速度などに応じて適切なパラメータを自動的に決定するアルゴリズムを開発する必要もある。また、地面の種類や状態を判別する手法を開発し、自己位置推定に加えて地面画像をナビゲーションに利用する価値を高める。

7.参考文献

- [1] 谷, 松本, 小笠原: “天井画像を用いた屋内ナビゲーション”
第19回日本ロボット学会学術講演会, 2001

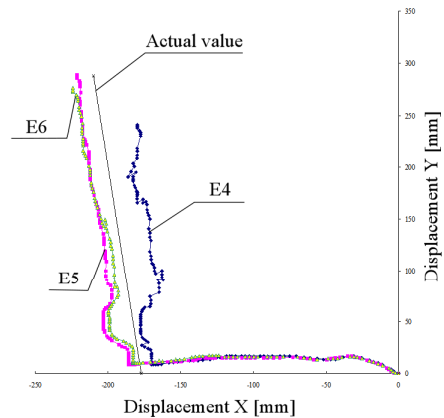


(a) Relationship between tracking error and size of template

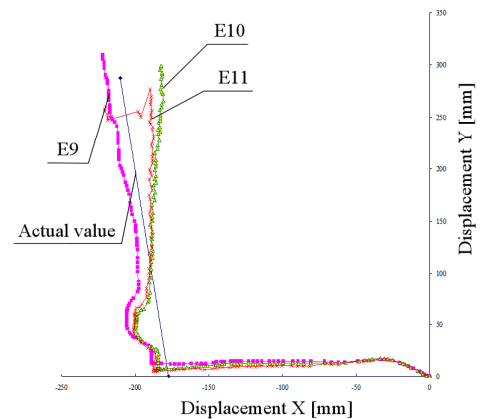


(b) Relationship between tracking error and backward reference number N_r

Fig.9 Experimental results (Rotational motion)



(a) Relationship between tracking error and size of template



(b) Relationship between tracking error and backward reference number N_r

Fig.10 Experimental results (Translation motion)