

# 三次元測域センサデータと航空画像のマッチングによる 移動ロボットの自己位置認識

○ 横田 恵助      伊藤 恒平      正 大矢 晃久 (筑波大)

## Localization of mobile robot by matching three-dimensional data of SOKUIKI sensor and aerial imagery

Keisuke YOKOTA, Kouhei ITO, Akihisa OHYA, University of Tsukuba

**Abstract**— In recent years, the demand for autonomous mobile robot in outdoor environment has been increasing. In order to guide the robot, it is essential to get the current position. In this study, we propose a method to correct the self-position of the robot, which is estimated by odometry, by aerial imagery and SOKUIKI sensor. First, we explore how to superimpose the reflection point cloud on top of the aerial imagery. Next, matching the edge data of aerial imagery and the upper edge of the building from the inclined reflection point cloud corrects the self-position of the robot, by which the robot was able to run on the straight road about 60m autonomously.

**Keywords:** Self-localization, SOKUIKI sensor, Aerial imagery

### 1. 緒言

人間にとって困難・煩雑な仕事を代行してくれる屋外自律移動ロボットの需要が近年高まってきている。それに伴い、屋外環境におけるロボットの自律移動に関する研究が盛んに行われてきている。

自律走行時に課せられるタスクの1つとして、ロボットの現在自己位置認識がある。屋内・屋外に限らず、移動ロボットで最も利用されている自己位置認識手法としてまず挙がるのがオドメトリであるが、タイヤの空回りやドリフトによる横滑りを検知できないため誤差が生じ、また時間にほぼ比例して誤差が大きくなるため、路面の状態が安定しない屋外環境においては大きなずれが生じやすい。オドメトリを使って推定した自己位置を修正する方法の1つとして、測域センサで得られた周辺の物体形状と地図とを照合する手法がある[1]。ロボットに測域センサを取り付けて走行させ、測域センサで取得される建物の反射点群を、仮想的な平面あるいは空間上に蓄積させることにより地図が生成される[2][3]。これらの手法では、地図生成のために予めロボットを1度走行させなければならない、未知の屋外領域でナビゲーションが行えないことを意味する。

本研究では航空画像と測域センサを用いて屋外領域で自己位置推定を行いながら、指定された経路上を移動ロボットに自律走行させることを試みる。ロボットの出発地点は予め指定され、航空画像上に引いた経路を辿り、ロボットが目的地まで自律的に到達することが本研究の達成目標である。

これまでも航空画像や電子地図を用いて移動ロボットをナビゲーションする研究はあったが[4][5]、いずれも2次元の測域センサデータを利用したものであり、斜めから撮影された画像に適用することはできなかった。これに対して、本研究では、ロボットが走行しながら蓄積させた三次元のセンサデータを用いることで、斜めから撮影された画像でも正しくマッチングを行えるようにしている点が異なる。

### 2. ロボットの構成

#### 2.1 ロボット

本研究では、筆者らの研究室で開発してきた独立二輪駆動型移動ロボットを用いる[図1]。大きさは、幅500mm×高さ600mm×奥行き1100mmである。



Fig. 1 PWS mobile robot

このロボットには、SH2をプロセッサとしたシングルボードの走行制御コントローラが搭載されており、制御用PCからシリアル通信により与えられた車輪左右それぞれの目標回転速度と、エンコーダからの回転量フィードバックより、車輪に与えるトルクを計算してPWM制御を行う。走行制御は、筆者らの研究室で開発された走行制御ソフトウェアを用いて、直線や円弧などの軌跡追従によりロボットを走行させる。

#### 2.2 測域センサ

測域センサとは、レーザ光を走査して周囲に照射し、反射点までの距離と受光角度を取得するセンサである。カメラや超音波センサに比べ、日照条件に影響されにくく、測距精度も高いという特徴を持つ。本研究では、北陽電機社が開発し

**Table 1** Specification of UXM-30LX-E (Tough-URG)

インターフェース	Ethernet
電源電圧	DC10~30V
光源	半導体レーザー (波長 = 905nm)
検出保障値	0.1~30m
最大検出距離	60m
測距精度	3,000lx 以下 → 0.1~10m : ±50mm 100,000lx 以下 → 0.1~30m : ±100mm
走査角度	190deg
角度分解能	約 0.25deg
走査時間	50ms
質量	約 800g (ケーブル除く)

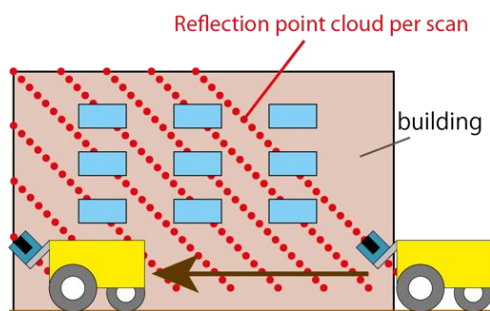
た UXM-30LX-E(Tough-URG)[図 2, 表 1] を使用する。

本研究では航空画像を地図の代わりに使用するため、地面付近から建物の上縁まで計測しなければならない。そのため、測域センサを上向きにしながら水平に移動させることにより、反射点群の三次元データを取得する必要がある。本研究では、ロボット正面にセンサを上向きに取り付けて走行させることでこれを実現させる [図 3]。

測域センサの取り付け位置は、ロボットの前方、地面から 40cm の高さに、45 度上向きに取り付けた。理由としては、真上に向けて取り付ける場合に比べ、建物を斜めに測ることでセンサが検出できる反射点の数を増やせると考えたためである。



**Fig. 2** UXM-30LX-E (Tough-URG)



**Fig. 3** How to get the point cloud of reflection



**Fig. 4** Third Gymnasium, University of Tsukuba (Digital-Globe, 2011)

### 3. 航空写真と三次元測域センサデータの重ね合わせ

本実験は、ロボットを走行させながら測域センサの点群を取得・蓄積させたデータを、航空画像の画像データと照合できるかどうかを確認するために行った。

#### 3.1 実験環境

オドメトリの誤差をできる限り小さくするために、路面が十分に舗装され、段差はなく、地面は傾いていない場所で測定する。今回の実験場所は、筑波大学内の第3体育館周辺 [図 4] である。建物の壁などをより確実に測定するために、人や自転車がほとんど通らない深夜の時間帯を選び、体育館の東側を秒速 0.3m で走行させ、Tough-URG から得られた反射点群を三次元空間上に蓄積させた。

#### 3.2 実験結果

三次元空間上の反射点群を真上からつぶして平面化したデータと、航空写真の画像データとを、建物と地面との境目を基準に手で合わせた結果が図 6 である。なお、航空画像はエッジをわかりやすく示すために、OpenCV[6] で作成した同画像の Canny エッジデータを重ね合わせている。

図 6 を見ると、建物の上縁が重なっていない箇所があることが分かる。また、左側に写る第3体育館に検出された点群については、航空画像上で地面との境が写っていないため、その場所に地面とのエッジがあるだろうという推測しかできない。そのため、この重ね合わせ手法で自己位置修正を実現するのは困難である。

前述のとおり、航空写真は必ずしも真上から撮影されているわけではない。そこで筆者らは、反射点群を傾けることにより上から見た時の建物の下縁と上縁の位置をずらすことで、それぞれをエッジデータと重ね合わせることができないか考えた。具体的には、地図の情報より傾ける方向を  $\theta$ 、傾きを  $\phi$  とし、三次元空間の反射点  $(x_i, y_i, z_i)$  に対して以下の計算を施すことで点の位置を  $(x'_i, y'_i, z'_i)$  に移動する。なお、今回は角度パラメータを航空画像より手で読み取り、 $\theta = -0.8243, \phi = 0.3592$  とした。

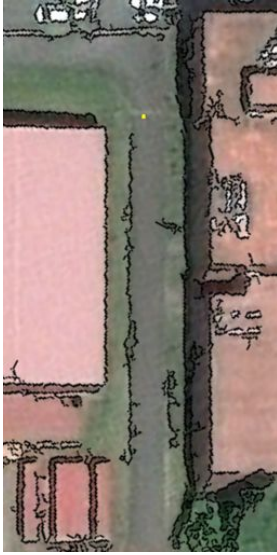


Fig. 5 Original Aerial imagery (it contains Canny edge data)

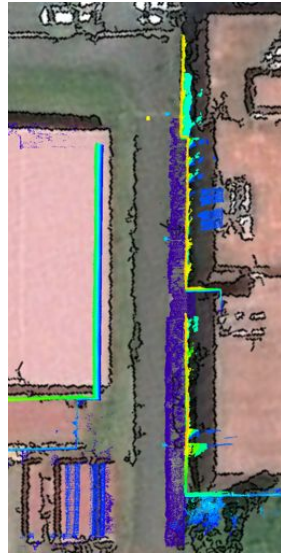


Fig. 6 Result of superposition of the reflection point cloud

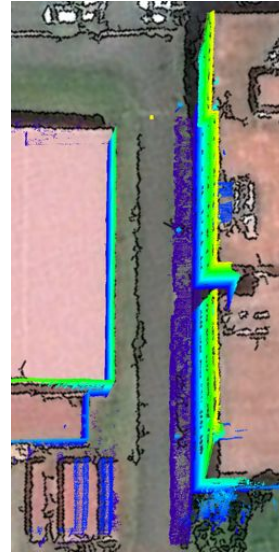


Fig. 7 Result of superposition of the reflection point cloud, which is inclined



Fig. 8 Result of extracting the highest points for each scan data from from Fig.7

$$\begin{pmatrix} x'_i \\ y'_i \\ z'_i \end{pmatrix} = \begin{pmatrix} x_i + z_i \cos \theta \sin \phi \\ y_i + z_i \sin \theta \sin \phi \\ z_i \cos \phi \end{pmatrix} \quad (1)$$

上式をもとに反射点群を傾け、上から見た図が図7になる。また、ロボットの位置から点群を左右に分け、それぞれの領域の1スキャンごとの最高点を図9のように取り出し蓄積させたものが図8になる。

図7を見ると、建物の壁面が航空画像とほぼ重なっていることが分かる。図8を見ると、建物の屋根の縁のエッジデータと反射点群がほぼ重なっているのが分かる。

#### 4. 航空画像のエッジデータと反射点群蓄積データによる自己位置修正

航空画像のエッジデータから抽出した線分と、反射点群蓄積データから抽出する線分との位置の比較により、オドメトリの自己位置誤差を修正するアルゴリズムを実装し、屋外実験を行った。

##### 4.1 自己位置修正アルゴリズムの実装

###### 4.1.1 航空画像からの建物上縁選出

図4から、建物の上縁を1本選出する。上縁を選出する理由としては、地面と天井の色の差が明確であり、Cannyオペレータでエッジがはっきり現れるためである。このエッジに沿って手動で直線  $l_{edge}$  を引き、走行中に計測される反射点群データと比較するためのエッジ直線とした。

###### 4.1.2 上縁の直線検出

反射点群のうち、[図9]に示すように1スキャンごとの最高点を蓄積させた点群  $P$  から1本の直線を抽出する。まず候補として取り出す点群  $P'$  の条件は、直線  $l_{edge}$  からの距離が1m以内のものとする [図10(1)]。その後、得られた  $P'$  から

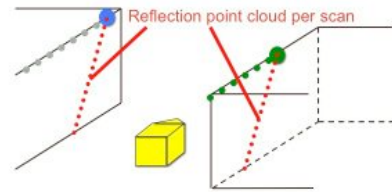


Fig. 9 How to extract the highest points of the reflection point cloud

最小二乗法により直線  $l_{LSM}$  を求める [図10(2)]。

###### 4.1.3 自己位置修正

検出された直線  $l_{LSM}$  に直線  $l_{edge}$  が重なるような座標変換行列を求める。まず  $P'$  のうち、最も古い反射点  $(x_0, y_0)$  を取り出し、 $l_{LSM}$  に向かって垂線を引き、その交点を  $(x_c, y_c)$  とする [図10(2)]。この点  $(x_c, y_c)$  が  $l_{edge}$  上に重なるように、 $l_{LSM}$  を  $y$  軸方向に平行移動させる。これを実現するアフィン変換行列を  $A$  とする [図10(3)]。

次に、点  $(x_c, y_c)$  がアフィン変換行列  $A$  によって移動された点  $(x'_c, y'_c)$  を中心に、 $l_{LSM}$  から  $l_{edge}$  へのなす角  $\theta$  分回転させるアフィン変換行列  $B$  を求める [図10(4)]。

最後に、変換行列の積  $AB$  を用いて、反射点群とロボットの座標を変換することにより、ロボットの位置を修正する。自己位置修正後、再び1m以上走行したら、上縁の直線検出の処理に戻る。

##### 4.2 走行実験

図5に示される直線道路上を、上記のアルゴリズムを用いた場合と、用いずにオドメトリのみを利用した場合の2つの走行実験を行った。ロボットが約60m走行した時点での、2つの場合の反射点群の蓄積結果を図11に示す。

オドメトリのみで走行させた場合、ロボットが少しずつ右

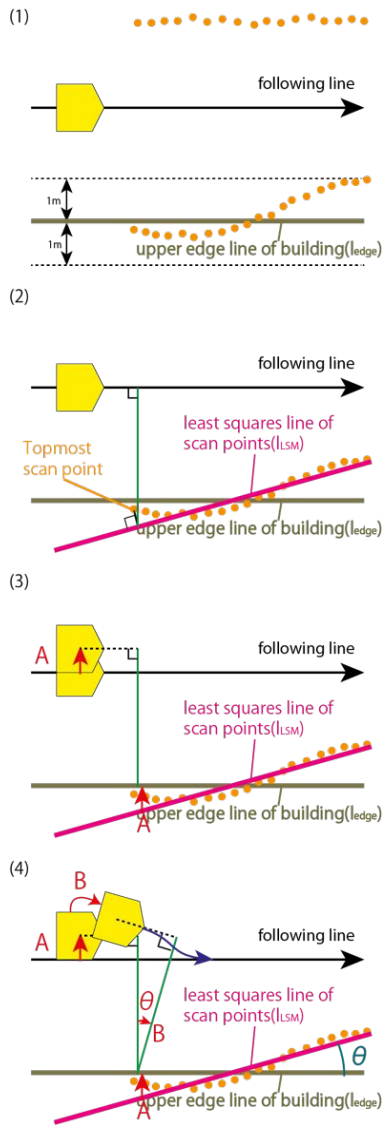


Fig. 10 How to correct the self-position of the robot

側に傾き、約 45m 進んだところで右側の縁石と衝突した。自己位置修正アルゴリズムを加えて走行した場合、多少の姿勢の修正はあったものの、道路のほぼ中央を常に進み、縁石とぶつからずに 60m 走行できた。

図 11 のうち、赤色の点が反射点群、緑色の線がエッジ直線である。オドメトリのみで点群を蓄積させた場合、角度修正が行われず、オドメトリの誤差によりエッジ直線から少しずつ反射点が離れていってしまっている。誤差修正しながら点群を蓄積させると、エッジ直線と反射点群がほぼ重なり、ロボットの自己位置が修正されていることが分かる。

## 5. 結論

本稿では、航空画像と測域センサを用いることにより、未知の屋外領域での自己位置推定を行う手法を提案した。独立二輪駆動型の移動ロボットを筑波大学構内で自律走行させることを第一目標として実験を行った結果、建物の間の直線道

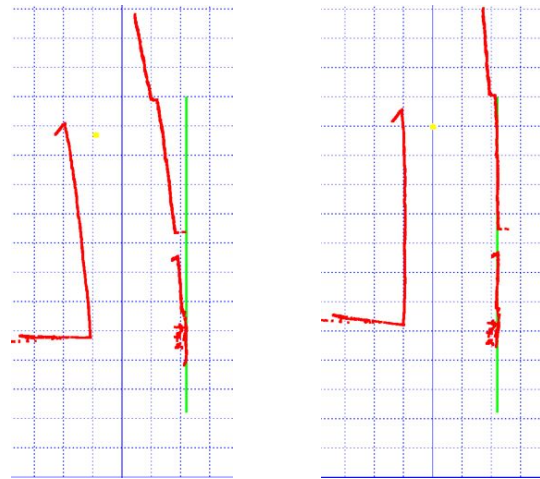


Fig. 11 Result of accumulating the reflection point cloud (left: self-localization by odometry, right: self-position correction)

路を、建物の上縁データのマッチングにより縁石にぶつからずに約 60m 自律走行できた。

本稿では直線走行のみを考えたが、ロボットが建物の角を曲がる際に生じるオドメトリ誤差の修正方法についても検討し、建物の周りを 1 周して自律走行させることが今後の課題である。また、今回は測域センサの反射点群を傾ける際のパラメータを手動で指定した。しかし実際には、傾ける方向や量は航空画像の撮影角度によって異なるので、ロバスト性を高めるためにはこれらのパラメータを航空画像から自動的に求める手法を検討する必要がある。また、反射点群からのエッジ抽出法として 1 スキャン分の最高点を蓄積させたが、建物の角検出を明示的にしているわけではない。反射点群データから建物の角を見つけることができれば、マッチングの材料が増え、より高い精度での自己位置推定が行えることが期待できる。

## 参考文献

- [1] Sebastian Thrun, Wolfram Burgard, and Dieter Fox : "Probabilistic Robotics" , The MIT Press, 2005.
- [2] 岩井 純一, 村松 聡, 富沢 哲雄, 末廣 尚士, 工藤 俊亮 : "物体の光反射特性を考慮した自由空間モデルに基づく環境地図生成", 日本機械学会ロボティクス・メカトロニクス講演会 2011, 2A1-O03, 2011.
- [3] 古川 陽介, 池澤 良介, 尾崎 雄介, 安 弘 : "車輪型移動ロボットによる環境地図情報取得のための自律走行法", 日本機械学会ロボティクス・メカトロニクス講演会 2011, 2A1-O16, 2011.
- [4] 森 武俊, 佐藤 崇浩, 黒田 藍子, 栗原 誠, 田中 雅行, 下坂 正倫, 福井 類, 佐藤 和正, 野口 博史 : "航空写真・電子地図に基づく複数レーザ測域スキャナーデータを用いた屋外マップ構築", 第 11 回システムインテグレーション部門講演会, 203-1, 2010.
- [5] 中澤 文彬, 安藤 吉伸, 吉見 卓, 水川 真 : "Google マップから作成したランドマーク地図による屋外自律移動ロボットの自己位置修正に関する研究", 第 11 回システムインテグレーション部門講演会, 203-5, 2010.
- [6] OpenCV, <http://opencv.jp/>, 2012.