

二足歩行ロボットのための全方位画像からの 自己位置推定手法

沼沢 祐一郎^{†1} 神 原 利 彦^{†1} 関 秀 廣^{†1}

ロボットに搭載されたカメラから得られる画像から、ロボットの自己位置や姿勢を推定する研究は、数多く行われている。だが、単眼の画像から位置と姿勢を求めるのは困難とされている。そこで本研究では、ジャイロでロボットの姿勢を計測し、その姿勢情報と画像情報を併用することで高精度な自己位置の推定を目指す。

An Estimation Method of Position from an Omnidirectional Image for a Two-leg Walking Robot

YUUICHIRO NUMASAWA,^{†1} TOSHIHIKO KANBARA^{†1}
and HIEDHIRO SEKI^{†1}

In the intelligent robotics field, many estimation methods of robot position and orientation have been proposed from images which are captured by a camera attached to the robot. Generally, it is difficult to estimate them from a monocular image. Especially, it is too difficult to estimate range data. Thus, stereo vision methods or motion stereo methods are used for the range estimation in many papers. On the other hand, we try to estimate position of the robot from only one monocular omnidirectional image.

1. 序 論

1.1 研究目的

ロボットに搭載されたカメラから得られる画像から、ロボットの自己位置や姿勢を推定す

る研究は、数多く行われている。だが、単眼の画像から位置3次元、姿勢3次元のすべてを求めるのは困難とされている。特に、位置情報では奥行き距離が推定しにくい。そのため、多くの研究では2台以上のカメラで同時に撮影したり、1台のカメラで時間差をつけて撮影することで、視差を発生させて、視差から奥行き距離を推定し、自己位置の推定へと導いている。それに対し本研究では、あくまで単眼の画像から自己位置の推定を行う手法の提案を目指す。

具体例として、図1のような二足歩行ロボットが床に敷かれた太線に沿って歩くタスクを考える。まっすぐ歩いているつもりでも、左や右に逸れることがあるので、線からどれだけ逸れた位置にいるかを計測できなければ、このタスクを実行できない。ロボットの頭部には全方位カメラが搭載され、床に敷かれた二本の太線を画像で計測する。その一方で胸部に取り付けられたジャイロセンサがロボット（カメラ）の傾き角度を計測する。画像計測とジャイロでの計測を統合することで、高精度な自己位置の推定を目指す。

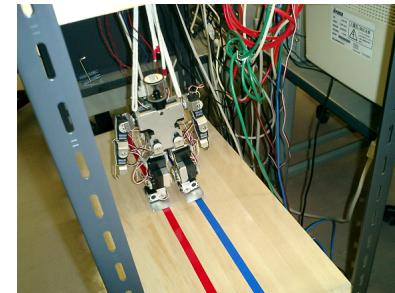


図1 頭部に全方位カメラを搭載した二足歩行ロボットと床に敷かれた二本の太線
Fig. 1 A Two-leg Walking Robot with a Omnidirectional Camera and Two Thick Lines on the Ground

1.2 関連研究

単眼の画像から3次元的な物体計測を行うことは、一般に難しいとされている。その理由は、画像は2次元であり水平方向と垂直方向の情報は記録されるものの、奥行きの情報が失われているからである。そのため、なんらかの仮定や知識を使うことで3次元的な情報を推定している。例えば、平行で幅が既知の道路を単眼のカメラで観測すれば、画像処理で左右両側の境界線が抽出できる。それら境界線の交点が消失点となり、消失点の位置から自己姿勢が推定できる。さらに、画像上で境界線の傾き具合から、境界線に対して垂直な自己位置

†1 八戸工業大学大学院工学研究科

Graduate school of Engineering, Hachinohe Institute of Technology

も推定できる。当然ながら、カメラの焦点距離やカメラ高さなどの固定された情報が既知でなければならぬが、運転者がいなくとも自動的に走行できるシステムを構築するための基礎技術とされている。画像処理技術の発達にともない、頑健に境界線を抽出し精度良く消失点を推定しようという研究も多い。例えば、数井ら¹⁾の手法が挙げられる。その他にも、中心射影カメラの単眼視を用いた研究例は多いが、全方位カメラの単眼視を用いた研究例は少ない。

一方で、2台以上のカメラから撮影した画像から自己位置を推定する手法や、ロボットが動く前と動いた後の2枚の画像を使って自己位置を推定する手法も多数提案されている。だが、これらの手法では1枚目の画像に写っている点が2枚目の画像のどこに写っているかを見つけ出す「対応付け」と呼ばれる処理が不可欠である。この処理では、誤った点に対応付けたり、手前の物体に隠されてしまい対応点が存在しない問題点などが発生する。単眼視であれば、対応付けの必要はない。全方位カメラを用いた自己位置・姿勢推定の研究としては、山澤ら²⁾や畠ら³⁾の研究がよく知られているが、彼らの研究でも複数枚の全方位画像の対応付けを行って、自己位置を推定している。本研究のように、全方位カメラで撮影した画像1枚のみ(単眼)で、対応付けを行わず自己位置を推定している研究例はほとんど存在しない。

筆者らは⁴⁾にて移動ステレオを使わず全方位カメラの単眼視で1本の太線を観測し、3次元の自己姿勢を推定する手法を提案した。この手法では、自己姿勢は推定できるものの自己位置が推定できなかった。そのため、2本の太線を観測することで、自己位置も推定できるように改良した手法⁵⁾を提案した。だが、いずれの手法でも、シミュレーション実験による推定精度が悪く、とても制御に使えなかった。原因を究明したところ、推定に使用する平面のあてはめ精度が悪いことがわかり、共分散行列を用いた最尤推定を用いる手法⁶⁾を提案した。だが、この手法でも推定精度の向上がわざかだったので、推定に使用する平面あてはめの処理で、平面に拘束条件を加える手法⁷⁾を提案した。この手法に基づいた実験より、平面あてはめという手法では、どんなに処理時間をかけても姿勢の推定精度が向上しないことがわかり、姿勢はジャイロで高精度に計測することにした。さらに、あてはめで膨大な処理時間を費やしていた誤差計算の処理を簡素化し高速化を実現する手法⁸⁾⁹⁾を提案した。本論文では、それらの手法を実装したシミュレーション実験の結果を示し、手法の有効性について述べる。

2. 手法

2.1 問題設定

頭部に全方位カメラを搭載した二足歩行ロボットが、床平面に敷かれた2本の太い平行線を観測し、その観測結果から自己位置を推定する。そして、その位置が太線から外れていれば太線の方へ歩みを戻すように歩行制御する。全方位カメラは双曲面ミラーと上向きのカメラを組み合わせたものを使用し、そのミラーの形状パラメータやカメラの焦点距離を既知とする。また、太線の幅や太線同士の間隔(図2参照)も既知とする。ロボットのボディ内部のジャイロセンサでカメラの傾き角度を計測する。

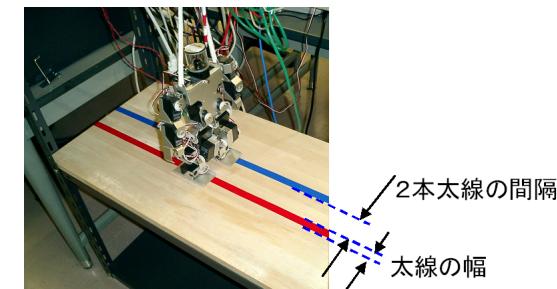


図2 既知の幅と間隔
Fig. 2 Thick lines whose width and interval are known

2.2 全方位カメラの撮像系

全方位カメラの撮像系を図3に示す。ミラー焦点を原点とし、鉛直軸を Z_c とした座標系 $X_c - Y_c - Z_c$ を全方位カメラ基準の座標系とする。このカメラで、 (X_a, Y_a, Z_a) にある物体点を撮影すると、 (X_a, Y_a, Z_a) から放たれた光のうち、原点 $(0, 0, 0)$ に向かう光だけがミラー曲面で反射した後、カメラ焦点 $(0, 0, -2c)$ を通る。原点 $(0, 0, 0)$ に向かわない光線は、反射してもカメラ焦点 $(0, 0, -2c)$ を通りないので撮影されない。焦点距離が f の上向きカメラは、カメラ焦点が正確に $(0, 0, -2c)$ の位置に来るよう設置されている。ミラー曲面上での反射点を (X_m, Y_m, Z_m) とすると、 (X_m, Y_m, Z_m) と $(0, 0, -2c)$ を通る直線が反射光の光線となり、その光線と画像平面との交点が結像点 (u_a, v_a) となる。

双曲面の式は、以下の式で表される。

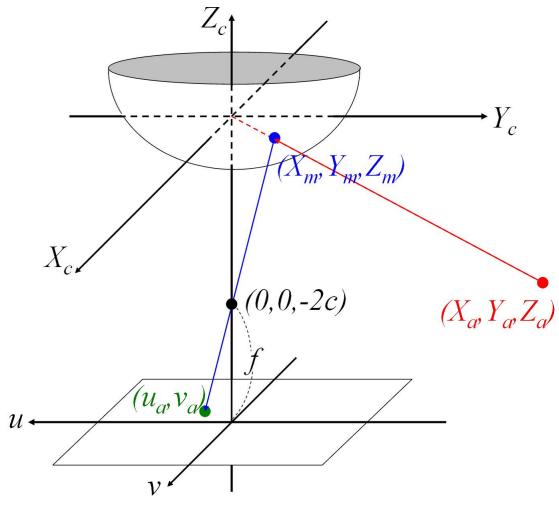


図 3 全方位カメラの撮像系

Fig. 3 Image Capturing System from an Ominidirectional Camera

$$\frac{X_c^2 + Y_c^2}{a^2} - \frac{(Z_c + c)^2}{b^2} = -1 \quad (1)$$

ここで， a と b は双曲面の形状を表すパラメータで $c = \sqrt{a^2 + b^2}$ である．点 (X_a, Y_a, Z_a) から放たれて原点 $(0, 0, 0)$ に向かう光線の式は，

$$\frac{X_c}{X_a} = \frac{Y_c}{Y_a} = \frac{Z_c}{Z_a} = \lambda \quad (2)$$

で表される．式(1)(2)の連立方程式を解いて，媒介変数 λ の解 λ_m は次式で求められる．

$$\lambda_m = \frac{a^2 c Z_a \pm a^2 b \sqrt{X_a^2 + Y_a^2 + Z_a^2}}{b^2 X_a^2 + b^2 Y_a^2 - a^2 Z_a^2} \quad (3)$$

上記の λ_m を使って反射点の座標位置 (X_m, Y_m, Z_m) は $(X_a \lambda_m, Y_a \lambda_m, Z_a \lambda_m)$ と求められる．それと焦点距離 f を用いて，画像平面上での結像点 (u_a, v_a) を求めると

$$u_a = \frac{f X_a \lambda_m}{Z_a \lambda_m + 2c}, \quad v_a = \frac{f Y_a \lambda_m}{Z_a \lambda_m + 2c} \quad (4)$$

となる．以上で，太線の境界線 4 本の上にある点 (X_a, Y_a, Z_a) は画像上の点 (u_a, v_a) で撮影されることが分かるが，画像上では小数点以下の座標値が丸められ，整数となる．例え

ば，2.34 という座標値は 2 に量子化され，3.72 は 4 に量子化される．そのため，カメラでの撮影時には，最大で ± 0.5 [pixel] の量子化誤差が発生する．図 4 に全方位画像の例を示す．この画像を処理して，赤い線や青い線の境界線を構成する画素を検出して (u_{ij}, v_{ij}) という整数座標の系列が得られる．ただし， $i = 1, 2, 3, 4$ であり， $j = 1, 2, \dots, N_i$ とする．

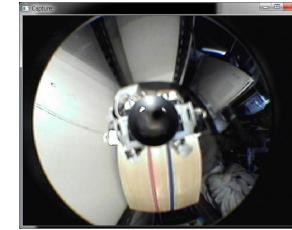


図 4 全方位画像の例

Fig. 4 Example of an Omnidirectional Image

2.3 世界座標系と全方位カメラ座標系との関係

世界座標系を $X_w - Y_w - Z_w$ とし，全方位カメラ基準の座標系を $X_c - Y_c - Z_c$ とする．これらの 2 つの座標系の幾何学的な関係を図 5 に示す．世界座標系から，水平方向に x_0 ，奥行方向に y_0 ，垂直方向に z_0 だけ並進させた場所に座標系 $X_c - Y_c - Z_c$ の原点があり，世界座標系からオイラー角 α, β, γ を使って座標軸を回転させたものがカメラ基準の座標系 $X_c - Y_c - Z_c$ である． Z 軸を中心に α 回し， X 軸を中心に β 回し， Y 軸を中心に γ 回したものである．この場合の $X_w - Y_w - Z_w$ 座標系と $X_c - Y_c - Z_c$ 座標系との関係は次式で表される．

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \mathbf{R} \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

ただし， \mathbf{R} は 3×3 の回転行列であり，オイラー角 α, β, γ を用いて次式で表される．

$$\mathbf{R} = \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta \\ 0 & \sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

本手法では，カメラ座標系を基準として推定を行うのでカメラ座標系を基準とした形に変換すると，

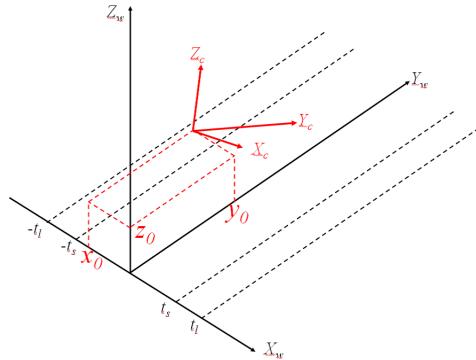


図 5 世界座標系を基準とした場合のカメラ座標系

Fig. 5 The camera coordinate system which is based on the world coordinate system

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R^{-1} \cdot \begin{bmatrix} X_w - x_0 \\ Y_w - y_0 \\ Z_w - z_0 \end{bmatrix} \quad (5)$$

となる。ここで、 R^{-1} は回転行列の逆行列であり、次式で表される。

$$R^{-1} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & \sin \beta \\ 0 & -\sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \gamma & 0 & \sin \gamma \\ 0 & 1 & 0 \\ -\sin \gamma & 0 & \cos \gamma \end{bmatrix}$$

床面に敷かれた 2 本の大線の境界線は、媒介変数 s を用いて世界座標系において $(-t_l, s, 0)$, $(-t_s, s, 0)$, $(t_s, s, 0)$, $(t_l, s, 0)$ で表される。これを式(5)に代入すれば、カメラ座標系を基準とした床面境界線の位置が算出できる。算出した例を図 6 に示す。

2.4 平面を用いた推定手法概要

前述のように、上向きのカメラで撮影されるのはミラー焦点に向かってくる光線のみである。1 本の境界線上の点群から放たれてミラー焦点に向かう光線の集合は平面となる。そこで、本手法では境界線とミラー焦点とで張る平面について考慮する。図 7 に平面の概念を示す。太線の左右 2 本の境界線とミラー焦点で張られる平面が 2 枚生成され、その 2 枚の平面の交線は、境界線に対し平行である。

床面に敷かれた 4 本の境界線は世界座標系上で互いに平行な直線であり、焦点とそれぞ

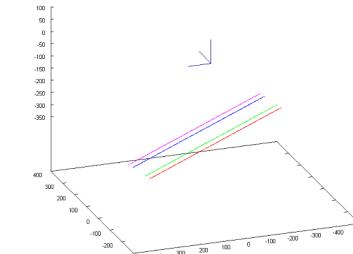


図 6 カメラ座標系を基準とした場合の床面上境界線
Fig. 6 Four boundary lines on the ground which are based on the camera coordinates

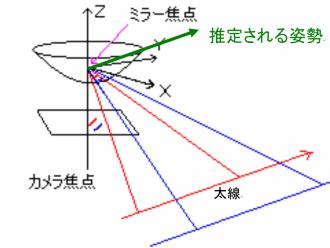


図 7 ミラー焦点と境界線が張る平面
Fig. 7 A Plane which is Spanned by a Mirror Focal Point and a Boundary Line

れの境界線が張る 4 枚の平面に含まれている。これらの条件から 4 枚の平面の交線は 1 本であり、その線の方向は境界線の方向と等しいと言える。任意の平面であれば、4 枚の平面には 6 本の交線が存在するはずだが、4 枚の平面すべてが 4 本の平行線を含むという条件と、4 枚の平面がすべてミラー焦点という一点を通るという条件から、交線が唯一となり、ミラー焦点もその交線上に含まれる。このような拘束条件の下にある平面を 4 枚使うことで、自己位置を推定する。手法概要を以下に示す。

- (1) ロボット本体に搭載したジャイロで、世界座標を基準とした全方位カメラ座標系の傾き角度 (α, β, γ) を計測する。座標変換の式より、全方位カメラ座標系を基準とした境界線の単位方向ベクトル e を算出する。
- (2) その単位方向ベクトル e に直交するという条件で、4 枚の平面の法線ベクトル $n_1 \sim n_4$ を探索により推定する。平面とミラー曲面との交線（曲線）を画像平面上に投影し

た曲線と、画像処理によって検出された点の系列 (u_{ij}, v_{ij}) との位置ずれが最小となるように 4 枚の平面の法線ベクトルを探索で求める。

(3) 4 枚の平面のなす角度から、円周角一定の法則を用いて、 x_0 の推定値と z_0 の推定値を求める。

2.5 カメラ座標を基準にした境界線の方向の推定手法

2.4 節の手順 (1) に示したように、ジャイロで、世界座標を基準とした全方位カメラ座標系の傾き角度 (α, β, γ) を計測する。座標変換の式 (5) より、全方位カメラ座標系を基準とした座標系に変換すると、境界線と平行な単位方向ベクトル e は次式で表される。

$$e = \begin{bmatrix} \sin \alpha \cos \beta \\ \cos \alpha \cos \beta \\ -\sin \beta \end{bmatrix} \quad (6)$$

2.6 境界線の方向に直交する法線ベクトルの設定

2.4 節の手順 (2) に示したように、単位方向ベクトル e に必ず直交するという拘束条件の下で、原点を通る平面の法線ベクトル n を推定する。式 (6) の方向ベクトルに直交する法線ベクトルは、回転角度 θ を用いて、次式で表される。

$$n(\theta) = \begin{bmatrix} \cos \alpha \sin \theta + \sin \alpha \sin \beta \cos \theta \\ -\sin \alpha \sin \theta + \cos \alpha \sin \beta \cos \theta \\ \cos \beta \cos \theta \end{bmatrix} \quad (7)$$

法線ベクトルは 3 次元だが、変数は θ の 1 次元のみである。したがって、境界線を含む平面を求める探索は 1 次元探索となる。

2.7 平面から画像上に投影される連続曲線を求める

法線ベクトル $n = (n_x, n_y, n_z)$ とした場合に、原点(ミラー焦点)を通る平面の式は、次式で表される。

$$n_x X_c + n_y Y_c + n_z Z_c = 0 \quad (8)$$

式 (1) と式 (8) の連立方程式でミラー曲面に含まれる空間曲線を求められる。途中の式は省略するが、媒介変数 \overline{X}_c を使って曲線の Y_c 座標 \overline{Y}_c は以下の式で計算される。

$$\overline{Y}_c = \frac{-a^2 n_y (n_z c - n_x \overline{X}_c)}{(b^2 n_z^2 - a^2 n_y^2)}$$

$$\pm \frac{bn_z \sqrt{(a^2 n_y^2 + a^2 n_x^2 - b^2 n_z^2) \overline{X}_c^2 - 2a^2 n_x n_z c \overline{X}_c + a^4 n_z^2}}{(b^2 n_z^2 - a^2 n_y^2)} \quad (9)$$

\overline{X}_c と式 (9) で求められた \overline{Y}_c を使って曲線の Z_c 座標 \overline{Z}_c は以下の式で計算される。

$$\overline{Z}_c = -\frac{n_x \overline{X}_c + n_y \overline{Y}_c}{n_z} \quad (10)$$

以上で、ミラー曲面に含まれる曲線上の一点 $(\overline{X}_c, \overline{Y}_c, \overline{Z}_c)$ が求められる。カメラの焦点は $(0, 0, -2c)$ にあるので、この点 $(\overline{X}_c, \overline{Y}_c, \overline{Z}_c)$ を焦点距離 f のカメラで撮影すると、画像上の座標値 (\bar{u}, \bar{v}) は以下の式で求められる。

$$\bar{u} = f \frac{\overline{X}_c}{\overline{Z}_c + 2c} \quad (11)$$

$$\bar{v} = f \frac{\overline{Y}_c}{\overline{Z}_c + 2c} \quad (12)$$

ここで、画像上の点 (\bar{u}, \bar{v}) は角度 θ と媒介変数 \overline{X}_c の 2 变数で決まるので、次式のような 2 变数関数のような形で表記できる。これらは連続曲線の垂直座標と水平座標を表す。

$$\bar{u} = g_u(\theta, \overline{X}_c), \quad \bar{v} = g_v(\theta, \overline{X}_c) \quad (13)$$

2.2 節で述べたように、画像処理で i 番目の境界線を構成する画素の系列 $(u_{i1}, v_{i1}), (u_{i2}, v_{i2}), \dots, (u_{iN_i}, v_{iN_i})$ があり、この系列に最も適合するように角度 θ を決めなければならない。

2.8 曲線と点との最小距離の算出手法

一般に、連続曲線と点の最小距離を求ることは難しい。例えば、画像から求められる一点 (u_{ij}, v_{ij}) と式 (13) で表される連続曲線との最小距離を求めるためには、まず次式で表される距離関数 $D_{ij}(\theta, \overline{X}_c)$ を定義する。

$$D_{ij}(\theta, \overline{X}_c) = \sqrt{(u_{ij} - g_u(\theta, \overline{X}_c))^2 + (v_{ij} - g_v(\theta, \overline{X}_c))^2} \quad (14)$$

次に、それを最小とする解 \hat{X}_c を求めなければならない。

$$\hat{X}_c = \underset{\overline{X}_c}{\operatorname{argmin}} \{ D_{ij}(\theta, \overline{X}_c) \} \quad (15)$$

そして、求めた解 \hat{X}_c を式 (14) に代入して最小距離を求める。

$$\hat{D}_{ij}(\theta) = \sqrt{(u_{ij} - g_u(\theta, \hat{X}_c))^2 + (v_{ij} - g_v(\theta, \hat{X}_c))^2} \quad (16)$$

解 \hat{X}_c を求める手法としては、 $D_{ij}(\theta, \overline{X}_c)$ を \overline{X}_c で偏微分した式が零となる式を導出し、

その式を解いて解析解 \hat{X}_c を求める方法が一般的である。だが、式(9)(10)(11)(12)からも明らかなように、式の中に二次式の平方根が含まれており、なおかつ、それが分母にも表れている。このように非常に複雑な式であるため、偏微分を計算して、それが零に等しいという等式が次元的に5次を超える。したがって、解析解を求めるることは不可能で、文献⁷⁾では \hat{X}_c を繰り返し計算によって数値解として求め、それで最小距離 $\hat{D}_{ij}(\theta)$ を導出した。だが、たった1点 (u_{ij}, v_{ij}) の最小距離であっても繰り返し計算による \hat{X}_c 算出の計算コストが非常に高いため、 N_i 個の点すべてに対し最小距離を求め、次式で誤差 $E_i(\theta)$ を評価すると積算して数十秒の処理時間を要することとなった。

$$E_i(\theta) = \frac{1}{N_i} \sum_{i=1}^{N_i} \hat{D}_{ij}(\theta) \quad (17)$$

さらに、式(17)の $E_i(\theta)$ を最小とする θ を求めるための探索(繰り返し計算)に加え、 i を変えて境界線4本分の計算をしなければならないので、処理時間が数分もかかり、とても制御には使えないことが文献⁷⁾で明らかになった。

2.9 極座標による最小近似距離の算出手法

本手法の主目的は、式(14)の最小距離を求ることではなく、角度 θ を高精度で求ることである。そこで、厳密な最小距離とはならないが、擬似的に最小距離に近い「最小近似距離」を提案する。詳細は、文献⁸⁾⁹⁾で述べているが、極座標を使って、この最小近似距離 $A_{ij}(\theta)$ を求める。その概念を図8に示す。

まず、画像計測で得られた境界線画素の点 (u_{ij}, v_{ij}) を偏角 ϕ_{ij} と長さ r_{ij} の記述に変換する。図8左では緑色の丸が境界線画素を表す。変換式は次式で表される。

$$\phi_{ij} = \tan^{-1} \left(\frac{u_{ij}}{v_{ij}} \right) \quad r_{ij} = \sqrt{u_{ij}^2 + v_{ij}^2}$$

ここで、 $Y_c = X_c \tan \phi_{ij}$ という平面の式を考える。この平面と式(8)で表される境界線を含む平面との交線を算出する。交線の式は次式となる。

$$-\frac{X_c}{n_z} = -\frac{Y_c}{n_z \tan \phi_{ij}} = \frac{Z_c}{n_x + n_y \tan \phi_{ij}} = \lambda$$

さらに、その交線の式と式(1)で表されるミラー曲面の式との連立方程式から交点2つが求められる。その交点2つを画像平面上に投影すると、図8左の赤丸で示した位置となる。 $Y_c = X_c \tan \phi_{ij}$ という平面の式を満たしているので、画像上では $v = u \tan \phi_{ij}$ という直線上に乗っているはずである。境界線を含む平面とミラー曲面の交点から算出しているので、画像上の曲線上に乗っているはずである。原点からこの点までの長さ R_{ij} を求めるには、

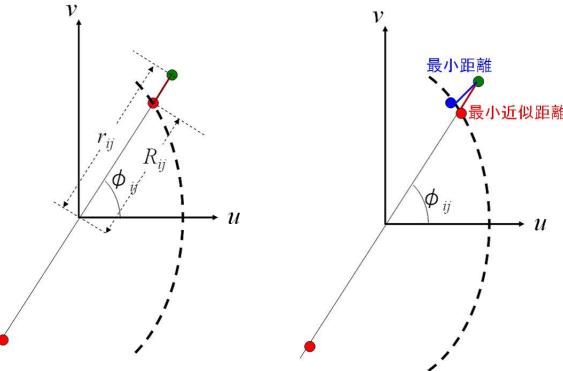


図8 最小近似距離

Fig. 8 A Minimum Approximate Distance

媒介変数 λ を次式で求めて、

$$\lambda = \frac{-a^2 c(n_x + n_y \tan \phi_{ij})}{b^2(n_z^2 + n_z^2 \tan^2 \phi_{ij}) - a^2(n_x + n_y \tan \phi_{ij})^2} \pm \frac{a^2 b \sqrt{(n_x + n_y \tan \phi_{ij})^2 + (n_z^2 + n_z^2 \tan^2 \phi_{ij})}}{b^2(n_z^2 + n_z^2 \tan^2 \phi_{ij}) - a^2(n_x + n_y \tan \phi_{ij})^2}$$

となる。その λ を使って原点から曲線までの距離を次式で算出する。

$$R_{ij} = \frac{fn_z \lambda}{[-(n_x + n_y \tan \phi_{ij})\lambda + c] \cos \phi_{ij}}$$

誤差関数は次式となり、この A_{ij} を最小近似距離と呼ぶ。

$$A_{ij}(\theta) = |R_{ij} - r_{ij}|$$

図8右に示したように、最小距離は青い線で示した距離となるが、算出する計算コストが非常に高い。そこで、必ずしも最小ではないが、ほぼ最小と近似できる距離という意味で、赤い線で示した最小近似距離を使って誤差を評価する。

この最小近似距離を、式(17)の代わりに用いて

$$E_i(\theta) = \frac{1}{N_i} \sum_{i=1}^{N_i} A_{ij}(\theta) \quad (18)$$

のように、誤差を評価する。最小近似距離を使うことで、数値解を求めるための繰り返し計算が不要になったため、高速で、式(18)の $E_i(\theta)$ が算出できる。この誤差関数を用いて、次式で推定角度 $\hat{\theta}_i$ を算出する。

$$\hat{\theta}_i = \operatorname{argmin}_{\theta} \{E_i(\theta)\} \quad (19)$$

2.10 円周角一定の定理による位置推定手法

系列 (u_{ij}, v_{ij}) に対して, i 番目の平面が最も適合するように探索した結果を $\hat{\theta}_i$ とする。この $\hat{\theta}_i$ を用いて位置を推定する。前述のように、境界線を含む平面 2 枚の交線はミラー焦点通り、その交線の方向は境界線の方向と一致する。その関係を図 9 に示す。青い境界線 l_1 と赤い境界線 l_2 がそれぞれ平面 1 と平面 2 に含まれる。そして、それらの平面 2 枚の交線は l_1 と l_2 という平行な線の方向と等しい。前節の手法で、平面 1 と平面 2 がなす角度 $|\theta_1 - \theta_2|$ が計測され、 l_1 と l_2 の間の距離も「既知の幅」としてわかっているが、 l_1 と l_2 を基準として、ミラー焦点がどこにあるのかはわかっていない。それを推定するために、境

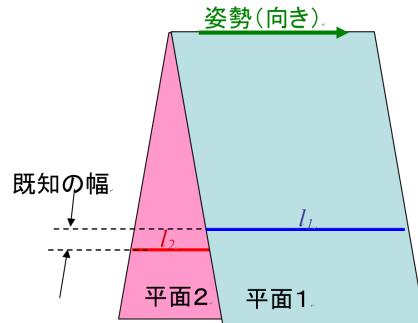


図 9 平面の交線と 2 本の境界線との関係

Fig. 9 Relation Between a Cross Line and Two Boundary Lines

界線の方向に対して垂直な断面で切斷した断面図を考える。図 10 に、その断面図を示す。断面図においては、2 本の境界線が 2 つの点で表される。そして、境界線に含まれる平面は、それぞれの点を通る直線で表される。それら 2 つの直線のなす角度が $|\theta_1 - \theta_2|$ となるような「2 直線の交点」の位置が推定位置となる。図 10 左に示したように、「円周角一定の法則」に基づきある固定された 2 点のそれぞれを通る 2 直線の交点のうち、その 2 直線のなす角が常に等しいのであれば、その交点位置はある円周上に拘束される。この位置を拘束する円周のことを、本手法では「拘束円」と呼ぶ。図 10 右に示したように、2 つの境界線が座標系 $X_w - Z_w$ 上で、 $(t_s, 0)$ と $(t_l, 0)$ で表される場合、その中点 $(\frac{t_s+t_l}{2}, 0)$ を通って

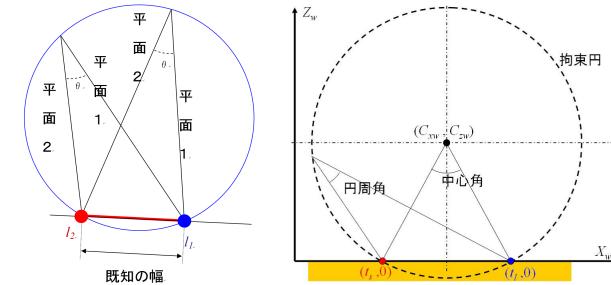


図 10 断面上で位置を拘束する円弧
Fig. 10 Circle Curve which Constrains Robot Position on a Profile Plane

Z_w に垂直な軸上に拘束円の中心はある。中心の位置は、次式で求められる。

$$C_{wx} = \frac{t_s + t_l}{2}, \quad C_{wz} = \frac{t_l - t_s}{2 \tan |\theta_1 - \theta_2|}$$

拘束円の半径は $\frac{t_l - t_s}{2 \sin |\theta_1 - \theta_2|}$ で求められる。以上の計算から、1 つの拘束円を描くことが可能になる。同様に、2 本目の太線の境界線 l_3 を含む平面 3 と l_4 を含む平面 4 とのなす角度 $|\theta_3 - \theta_4|$ が求められ、境界線 l_3 と l_4 の位置が既知であるから、2 つ目の拘束円を描くことができる。図 11 に、左側の赤い太線の両端の境界線 2 本から描いた左側の拘束円と、右側の青い太線の両端の境界線 2 本から描いた右側の拘束円を示す。これら 2 つの拘束円の交点が推定位置となる。

交点は 2 つ存在するが、推定された位置から、平面 2 と平面 3 のなす角度 $|\theta_2 - \theta_3|$ を計算すれば、2 つのうち片方が、推定された $|\theta_2 - \theta_3|$ よりも大きく異なる角度となるので、真の推定値を選ぶことができる。

3. 実験

3.1 実験機器

実験に用いた機器を以下に示す。二足歩行ロボットには、近藤科学製 KHR-1 を使用した。全方位カメラには、ヴィエストン製 VS-14N を使用した。ジャイロには、NEC トーキン製 MDP-A3U9S-DK を使用した。画像入力には、恵安製画像キャプチャケーブル VS-USB2800D/JP1 を使用した。式 (1) の形状パラメータには、VS-14N の a の値 10.9407 と、 b の値 12.1500 を使用した。VS-14N の焦点距離 f を予備実験で調べた結果、328.75[pixel]

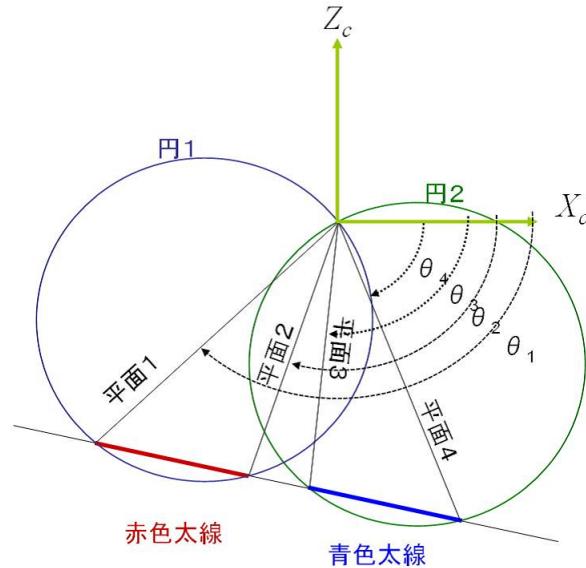


図 11 2 つの拘束円の交点で求められる推定位置

Fig. 11 Two Positions which are Estimated from Two Crossing Points of Two Constraint Circles

とした。計算機には、CPU に Core-i7(1.4GHz) で RAM4GB の PC を使用した。画像処理には、ライブラリ OpenCV 2.1 を使用した。

3.2 シミュレーション実験

図 5 に示した座標系で、 t_s を 60[mm]、 t_l を 100[mm] として、シミュレーション実験を行った。まず、ロボットの姿勢 α, β, γ と位置 x_0, y_0, z_0 を既知として、4 本の境界線の画像点列 (u_{ij}, v_{ij}) を生成した。そして、 α, β, γ はジャイロで計測できるものとして、 ϕ と ψ を算出した。本手法における誤差を最小とする探索で、 $\hat{\theta}_1 \sim \hat{\theta}_4$ を推定した。さらにその推定値から円周角一定の法則に従い、拘束円 2 つを求めてその交点から推定位置 (\hat{x}_0, \hat{z}_0) を導いた。

角度の探索では、まず第一段階として -90° から 90° まで 1° 刻みで変化させて最小値を選び出した。次に第二段階として、その選び出された角度を中心に $\pm 1^\circ$ の範囲で、 0.001° 刻みで変化させて最小値を選び出した。図 12 にその変化の例を示す。図の左側に、第一段階の探索例を示す。横軸が角度で縦軸が誤差を表す。図の例では、 77° が最小値と導出され

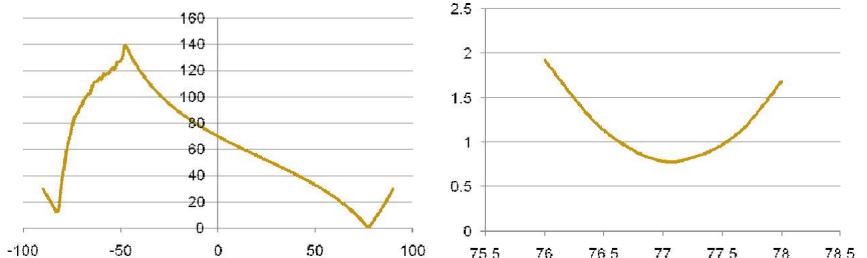


図 12 誤差の変化の例
Fig. 12 Example of Changing Errors

た。図の右側に、第二段階の探索例を示す。横軸が角度で縦軸が誤差を表す。図の例では、 77.080° が最小値と導出された。

次に、様々な位置・姿勢における推定誤差の計測実験を行った。姿勢 α, β, γ は、 $-20^\circ, -10^\circ, 0^\circ, 10^\circ, 20^\circ$ の 5 通りについて、水平方向位置 x_0 は、 $-100[\text{mm}], -50[\text{mm}], 0[\text{mm}], 50[\text{mm}], 100[\text{mm}]$ の 5 通りについて、垂直方向位置 z_0 は、 $250[\text{mm}], 275[\text{mm}], 300[\text{mm}], 325[\text{mm}], 350[\text{mm}]$ の 5 通りについて、推定のシミュレーションを行った。その結果、 $5^5 = 3125$ 通りの推定結果が導かれた。その推定結果から、正解の x_0 と z_0 をそれぞれ引き算したものを誤差として評価した。その誤差の分布を図 13 に示す。横軸が x_0 の誤差、縦軸が z_0 の誤差をそれぞれ表す。

この誤差の分布中心は、 $(\bar{x}_0, \bar{z}_0) = (0.0103, 0.10551)$ であった。分布の共分散行列 Σ は次式となった。

$$\Sigma = \begin{bmatrix} 55.46417 & 2.133949 \\ 2.133949 & 25.41783 \end{bmatrix} \quad (20)$$

この行列の固有値は、55.615 と 25.267 で、それぞれの平方根は 7.4575 と 5.0266 であるから、水平方向の偏差が約 7.5[mm]、垂直方向の偏差が約 5[mm] の推定手法と評価される。

ただし、図 13 の右下には、水平方向誤差が-60[mm]、垂直方向誤差が-105[mm] という極めて大きな「はずれ値 (outlier)」となる場合が計測されている。

3.3 実機実験

画像処理用ライブラリ OpenCV を用いて、全方位動画像をキャプチャしてから、太線の色情報を基に、二値化、ラベリングをしてから、輪郭追跡によって境界線を抽出した。だ

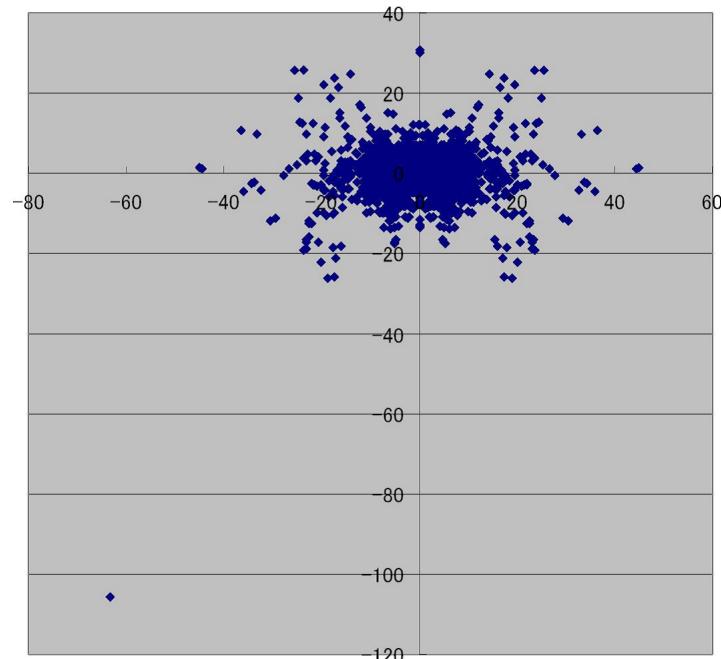


図 13 3125 回の推定誤差の分布
Fig. 13 Distribution of Estimation Errors under 3125 situations

が、太線の端のエッジで境界線でない部分も抽出される問題点がある。手作業で、それらの境界線画素を除去した。

オフラインで抽出した (u_{ij}, v_{ij}) とジャイロの計測値 (α, β, γ) を用いて本手法でロボットの位置を推定した。実験に用いたジャイロの計測誤差が、真値から $\pm 10^\circ$ と非常に大きいため、角度の推定に使う誤差を最小にするものを選んでも、その誤差自体が大きくて、画像点列に適合する曲線が選ばれず、結果的に位置の推定値が安定しなかった。

シミュレーション実験では、ジャイロで誤差なく姿勢が計測できることを前提にして、量子化誤差に対して高精度な推定ができる事を示した。だが、実機実験ではジャイロの大きな誤差が推定に悪影響を与えることがわかった。

4. 結論

二足歩行ロボットが床面に敷かれた線に沿って歩けるようにするために、単眼の全方位画像計測と、ジャイロによる姿勢計測からロボットの位置を高精度で推定する手法を提案した。シミュレーション実験で、水平方向の誤差の偏差が 7.5[mm]、垂直方向の誤差の偏差が 5.0[mm] という精度で計測できることが証明できた。実機実験では、リアルタイムでの画像処理がうまくいかず、オフラインで画像処理してジャイロによる姿勢計測結果を用いて推定を行ったがシミュレーションほど誤差が小さくならず、推定された位置も安定しなかった。

今後は、シミュレーション結果における「はずれ値」が誤って推定された原因を究明する。シミュレーションでジャイロによる姿勢推定の誤差がどの程度であれば位置推定への影響が少ないのかを調べ、実機実験ではその許容誤差範囲内のジャイロを使用する。実機実験での画像処理手法を工夫し、リアルタイムで境界線画素が抽出できるようにする。

参考文献

- 1) 数井誠人、長谷山美紀、北島秀夫：複比を用いた自動走行システムのための消失点推定、信学論 D-II, Vol.J84-D-II, No.7, pp.1319-1327, (2001).
- 2) 山澤一誠、八木康史、谷内田正彦：移動ロボットのナビゲーションのための全方位視覚センサ、信学論 D-II, Vol.J79-D-II, No.5, pp.698-707, (1996).
- 3) 畑幸一、青木利道、栄藤稔：全方位画像を用いた移動ロボットの運動パラメータ推定、信学論 D-II, Vol.J84-D-II, No.3, pp.448-458, (2001).
- 4) 神原利彦、末松伸朗、林朗：二足歩行ロボットのための自己位置・姿勢推定手法、電子情報通信学会 2006 年総合大会講演論文集, p231(2006).
- 5) 神原利彦：二足歩行ロボットのための自己位置・姿勢推定手法、電子情報通信学会 2007 年総合大会講演論文集, p221(2007).
- 6) 神原利彦、谷川翼：二足歩行ロボットのための自己位置・姿勢推定手法、計測自動制御学会東北支部第 236 回研究集会資料 (2007).
- 7) 神原利彦、沼沢祐一郎：拘束された平面による二足歩行ロボットの自己位置推定手法、電子情報通信学会 2009 年総合大会講演論文集, p(2009).
- 8) 神原利彦、沼沢祐一郎、関秀廣：二足歩行ロボットのための全方位カメラからの自己位置・姿勢推定手法、計測自動制御学会東北支部第 250 回研究集会資料 (2009).
- 9) 沼沢祐一郎、神原利彦、関秀廣：二足歩行ロボットのための全方位カメラからの自己位置推定手法、電子情報通信学会 2010 年総合大会講演論文集, p.194(2010).