コンピュータビジョンと
イメージメディア 125−17
（2001. 1. 19）

全方位視覚を用いた移動ロボットの高速な障害物回避と自己位置推定

関森 大介* 臼井 智也** 升谷 保博** 宮崎 文夫**
*明石工業高等専門学校 **大阪大学

本稿では，移動ロボットに搭載された全方位視覚で得られる床領域に基づいて，ロボット周囲の障害物を回避する方法と自己位置を簡便に推定する方法について提案する．処理の多くは，画像処理装置 (日立,IP5005) 上で行なうため，高速な実行が可能である．床領域からの特徴量抽出が短時間で実現される．本稿では，障害物回避方法と自己位置推定方法の具体的なアルゴリズムについて提案するとともに，RoboCup 小型機部門のサッカーロボットに適用してその有効性を検証する．

High-Speed Obstacle Avoidance and Self-Localization for Mobile Robot Based on Omni-Directional Vision

Daisuke.SEKIMORI* Tomoya.USUI** Yasuhiro.MASUTANI** Fumio. MIYAZAKI**
*Akashi College of Technology **Osaka University

In this paper, we propose a method of avoiding obstacles near the robot and a method of self-localization simply based on floor information gotten with omni-directional vision. In order to detect features of floor information in short time, all of image processing are executed on the image processor (Hitachi, IP5005). In this paper, we describe concrete algorithms of obstacle avoidance and self-localization. We also inspect the effectiveness of these methods through several experiments with the real robot according to the rule of RoboCup small-size league.

1 はじめに

我々は，RoboCup[1] 小型機部門のルールに準じた全方向移動機構と全方位視覚装置を有する移動ロボットシステムの開発を行なってきた．このロボットシステムは，サッカーロボットだけでなく，一般的な自律分散型ロボットシステムの研究用プラットフォームとして優れている．その理由として，次のようなことが挙げられる: (1) ロボットシステムは全方向移動機構を有しているため，非ホロノミックな拘束を考慮せずに行動計画を生成することができる，(2) ロボットシステムは全方位視覚を搭載しているため，カメラの方向や死角を考慮する必要がない，(3) ロボットシステムが小型であるため，ロボットの台数や実験場所の制約が少ない．本稿では，このようなロボットシステムを対象とする障害物回避方法と自己位置推定方法を取り扱う．これらは RoboCup では重要な機能であり，同時に汎用のプラットホームの基本機能でもある．

RoboCup や多くの室内環境では，床領域と他の物体を色で識別することが可能である．したがって，ロボット近傍の床色を観察することにより，ロボットの移動可能領域や他の物体との位置関係を把握することができる．そこで，全方位視覚から得られる床領域に基づいた障害物回避方法と自己位置推定方法について提案する．本手法では，処理の多くを画像処理装置 (日立,IP5005) 上で行なうため，短時間で結果が得られ，速い動きを要求される RoboCup に適している．

以降，第 2 節では，我々が開発したロボットのシステム構成，全方位移動機構と全方位視覚装置について簡単に説明する．続いて，第 3 節では，全方位視覚に基づいた高速な障害物回避方法，第 4 節では，自己位置推定方法の詳細なアルゴリズムついて説明する．そして最後に，我々が開発したロボットを用いて行なった検証結果について示す．

2 システム構成

本研究で開発したロボットシステムは，実際にフィールドを動き回るロボット部と画像処理や行動決定を行なうメインコンピュータ部から構成されており，両者は無線通信で接続されている (**Fig.1**参照)．また，ロボット同士の通信はそれぞれのメインコンピュータ部が Ethernet を介して行なっている．**Fig.2** に本システムの構成図を示す．

現在の RoboCup の小型機部門では，天井カメラの画像を用いるグローバルビジョン方式が主流であ

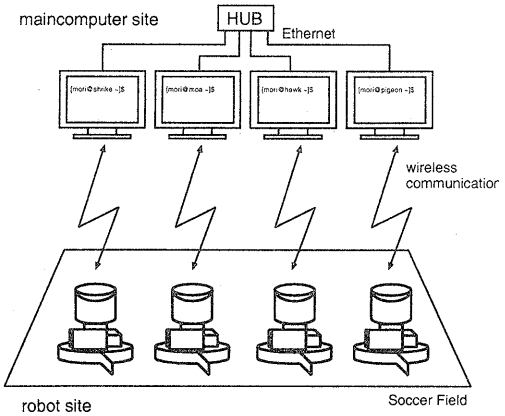


Fig.1 Concept of soccer robot system

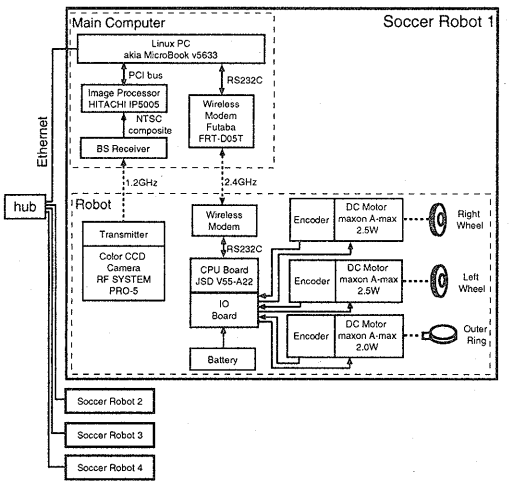


Fig.2 System configuration

るが，我々は自律分散型知能の研究を重視する立場から，ロボットに搭載されたカメラの画像のみを用いるローカルビジョン方式を採用している．

2.1 ロボット部

カバーを外したロボット部の外観を **Fig.3**に示す．ロボット部は円形の車体とその周りを回転できるアウタリングから構成されており，CPU(NEC V55, クロック 16MHz) ボード，I/O ボード，CCD カメラ，全方位ミラー，駆動用モータ 2 個，アウタリング用モータが搭載されている．車体の直径は 140[mm] で高さは 225[mm] である．アウタリングにはフィンが取り付けられており，これを利用してボールの操作

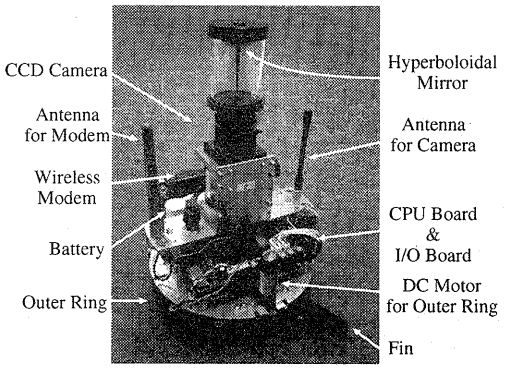


Fig.3 Overview of the robot part

を行なう．

車体上部には無線モデムが搭載されており，メインコンピュータから送信された指令を基に CPU が左右の車輪用モータとアウタリング回転用モータの制御を行なう．また，各モータの回転量を積算してデッドレコニング値の演算も行なっている．

CPU ボードと I/O ボードの上部に CCD カメラが上向きに設置されており，レンズの先にはアコウル製の双曲面ミラーが取り付けられている．これらによって，全方位画像を得ることができる．得られた画像はカメラに内蔵されている送信機でメインコンピュータ部に送られる．

2.2 メインコンピュータ部

メインコンピュータ部には，CPU に PentiumII 333MHz を用いた一般的な PC を使用し，画像処理ボード (日立 IP5005) および無線モデムが接続されている．さらに他のサッカーロボットのメインコンピュータ部と 100Mbps の Ethernet を用いた LAN を構成している．OS には Linux を用いている．

2.3 全方向移動機能

サッカーロボットには時々刻々と変化する動環境に素早く対応するために全方向へのスムーズな移動能力が求められる．そこで，本研究では和田らによって提案されているアクティブ単輪キャスタ機構 [2] を参考に，左右独立駆動方式による全方向移動機構の開発を行なった．全方向移動機構の概観を **Fig.4**に示す．駆動輪は車体中心からオフセットを与えて取りつけられており，それらは 2 つのエンコーダ内蔵のモータによって独立に駆動される．車体側面のア

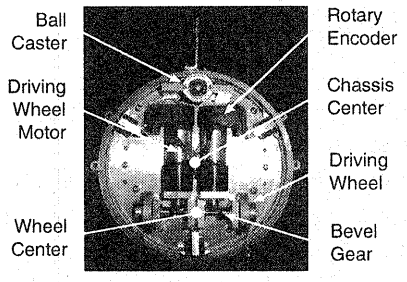


Fig.4 Overview of the omni-directional mobile mech.

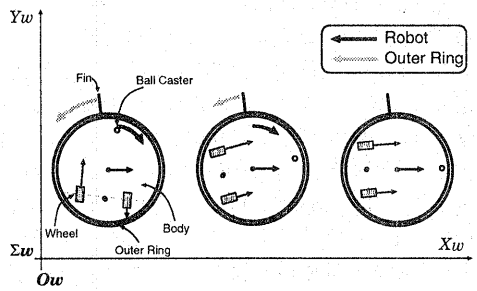


Fig.5 Principle of omni-directional mobility

ウタリングはボールベアリングで支持されており，アウタリング用モータによって回転される．

全方向移動は，**Fig.5**のように，左右の駆動輪の速度比を逐次変えることで実現する．また，姿勢の制御はアウタリングの指令速度に車体の回転を打ち消す回転速度を加えることで実現する．本研究では，全方向移動の速度生成法に加えて，駆動輪のスリップを抑制するための車体の加速度の最大値を制限する速度生成法や，高速旋回中の車体の揺動現象を抑制するためのフィードフォーワード補償を提案している．詳しくは文献 [3] を参照されたい．

2.4 全方位視覚

本研究では広範囲の視野を獲得するために，ロボットの視覚に双曲面ミラーを用いた全方位視覚を採用している．**Fig.6**に得られた画像の一例を示す．

画像処理装置の機能を利用して，全方位視覚によって得られた画像から特定の色領域を抽出し，画像座標系の座標 X, Y を得る．さらに画面の中心から重心座標までの距離 R と方向 Φ を求める．

$$R = \sqrt{X^2 + Y^2} \tag{1}$$

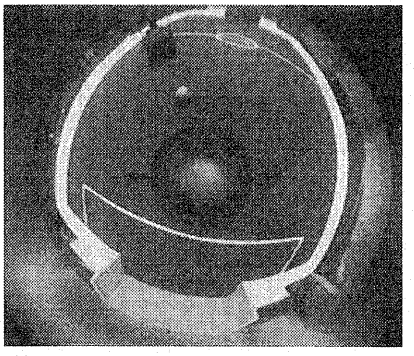


Fig.6 Omni-directional image

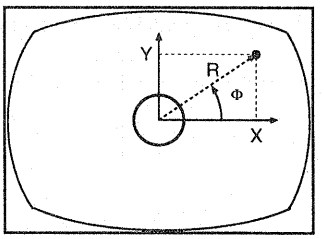


Image plane

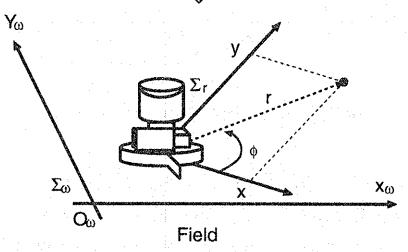


Fig.7 Distance and direction of object

$$\Phi = \text{atan2}(Y, X) \tag{2}$$

なお，画像メモリの解像度は 256×220[pixel] としている．このとき **Fig.7**のようにロボット座標系 Σ_r における床面上点の方向 ϕ は全方位画像上の方向 Φ と一致すると見なす．

$$\phi = \Phi \tag{3}$$

一方，画像上の距離 R[pixel] から床面上の実距離 r[mm] への変換には **Fig.8**のモデル [5] を用いると，以下の変換式が成立する．

$$r = \frac{R}{A - B\sqrt{C + R^2}}$$

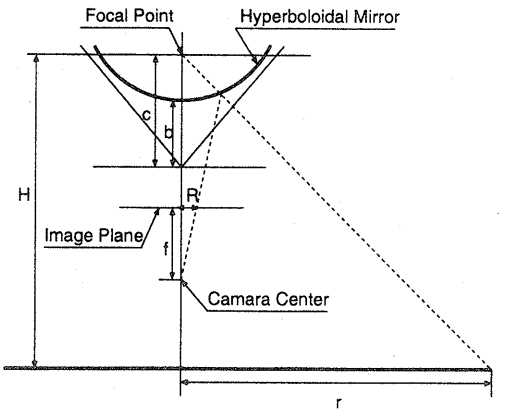


Fig.8 Geometric model of shperical mirror

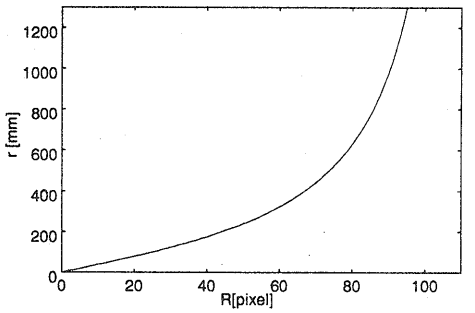


Fig.9 Graphic chart of Eq.(4)

ここで，$A = \frac{f(b^2+c^2)}{H(c^2-b^2)}$, $B = \frac{2bc}{H(c^2-b^2)}$, $C = f^2$ を表す．以上の関係式には測定しにくいパラメータも含まれているため，R[pixel] と r[mm] の実測値から最小二乗法を用いて，決定すると以下の式になる．

$$r = \frac{R}{9.762 - 2.008 \times 10^{-2}\sqrt{2.238 \times 10^5 + R^2}} \quad (4)$$

また，この式をグラフ上に表すと **Fig.9**のようになる．以上から，ロボット座標系における床面上の座標 x, y は以下の式を用いて表すことができる．

$$x = r\cos\phi \quad (5)$$

$$y = r\sin\phi \quad (6)$$

この全方位視覚システムを用いて，位置推定の実験を行なったところ，床面上の点に関しては，フィールドのほぼ全域で同定することができた．

3 高速な障害物回避方法

RoboCup では，敵ロボットに衝突することはルールで禁じられており，また，フィールドの壁際での巧みな処理が必要である．したがって，ロボットには敵ロボットや壁との衝突を回避する機能が必要不可欠である．また，ロボット周囲の環境は激しく変化するため，高速な障害物回避が求められている．幸いなことに，RoboCup の環境は比較的簡単であり，色で物体を識別することが可能である．したがって，ロボット近傍の床色の部分は移動可能領域であり，それ以外の部分は他のロボットや壁のような障害物であるという仮定を置くことができる．

本節では，全方位視覚画像を用いた高速な障害物回避の具体的なアルゴリズムについて提案する．全方位画像から抽出された床領域をいくつかのセルに分割し，画像処理装置 IP5005 のラベリング機能を利用して，1/30 秒周期でロボット全周囲の移動可能領域の導出を行なう．そして，この移動可能領域を利用した障害物回避の一例として，ロボットの移動速度を考慮した障害物回避方法について述べる．

3.1 移動可能領域の導出

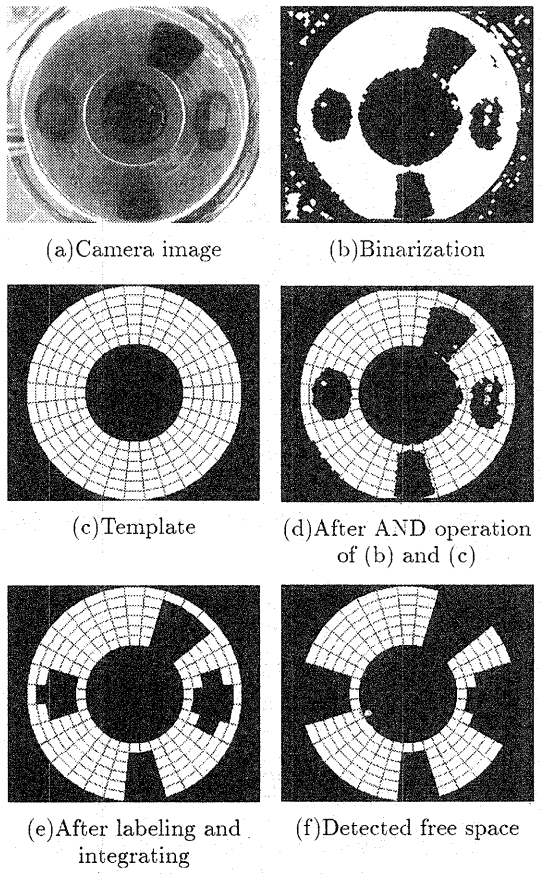
前提条件:

1. カラーの全方位画像を使用する．
2. 床領域だけが緑色とし，他は別の色とする．
3. 床領域にはある幅以下の細い白線が存在する．

以上の前提条件の下で，全方位視覚画像に基づいた移動可能領域の導出アルゴリズムの手順は以下のようになる．

Step1 (画像の 2 値化)
画像処理装置の先行取得機能を用いて全方位画像を取り込み (**Fig.10**(a) 参照)，床色である緑色を抽出して 2 値画像を得る．次に，数回の膨張と収縮処理を施して白線の除去を行なう (**Fig.10**(b) 参照)．なお，画像の解像度は最小の 256×220[pixel^2] とする．

Step2 (領域の分割)
半径方向に I 個，円周方向に J 個のセルに等分割され，各セルの閉領域，面積がそれぞれ $S_{ij}, A_{ij} (i = 1, \cdots, I; j = 1, \cdots, J)$ であるテンプレート (**Fig.10**(c) 参照) と AND 演算を行な



(a)Camera image (b)Binarization

(c)Template (d)After AND operation of (b) and (c)

(e)After labeling and integrating (f)Detected free space

Fig.10 Example of method of detecting free space

い，床領域を極座標における細かなセルに分割する (**Fig.10**(d) 参照).

Step3 (ラベリング処理と障害物の有無の判定)
画像処理装置のラベリングの機能を利用し，ラベル付けされた N 個の各領域の面積 a_k およびその重心位置 $\boldsymbol{p}_k = (x_k, y_k)^T, (k = 1, \cdots, N)$ を求める．そして，式 (7) を用いてセル毎の床色の比率 r_{ij} を算出する.

$$r_{ij} = \sum_{\boldsymbol{p}_k \in S_{ij}^{k}} a_k / A_{ij} \qquad (7)$$

さらに，式 (8) を用いて，セル毎の障害物の有無を判定する (**Fig.10**(e) 参照).

$$q_{ij} = \begin{cases} 0 & r_{ij} < r_{th} \\ 1 & r_{ij} \geq r_{th} \end{cases} \qquad (8)$$

ここで，q_{ij}=0/1 は障害物の有/無を表わし，r_{th} は障害物の有無を判定する閾値を示す.

Step4 (移動可能領域の検出)
障害物より遠い部分 (半径方向) を障害物無しの候補から外す (**Fig.10**(f) 参照). 式 (9) により，ロボットの周囲の各方向の最も近い障害物までの距離が離散化された 1 次元データ a_j として得られる．なお，計算の便宜上，$q_{(I+1)j} = 0$ としている.

$$a_j = \min_{q_{nj}=0}(n-1) \qquad n = 1, \cdots, I+1 \qquad (9)$$

3.2 移動速度を考慮した障害物回避方法

ロボット全周囲の移動可能領域が得られれば，目的に応じて様々な障害物回避のアルゴリズムを用いることができる．我々は，移動可能領域を利用した障害物回避方法として，目標方向のみが与えられた場合の障害物回避のアルゴリズムについて提案を行ってきた [4]．本節で提案する手法は，さらにロボットの移動速度も考慮したものである.

我々のロボットでは，駆動輪の急な加減速により，ロボットのデッドレコニング値が実際の軌道に対して大きく外れることを防ぐために，駆動輪の加速度の最大値に制限を与えている．よって，ロボットの移動速度が大きくなると，急な方向転換が困難になり，障害物に衝突する危険性が増大する．そこで，ロボットの現在の移動速度を基にロボット自身の軌道を予測し，未然に障害物との衝突を避けるようにロボットへ速度指令を行なう方法について提案する．以下にその具体的なアルゴリズムを示す.

Step1 (障害物情報の抽出)
テンプレートを用いた移動可能領域の算出結果を用いて，ロボット座標系 (床面) における各方向 ϕ に対する最近接障害物までの距離 $r_o(\phi)$ を求める (実際には，全方向を N 分割した数表として計算する．). ただし，ロボットの大きさを考慮し，ロボットの半径分 (r_{robot}) だけ障害物を膨張させる (**Fig.11**参照).

Step2 (ロボットの現在速度の取得)
ロボット部から現在の速度を取得し，それを現在速度 $\boldsymbol{v}_a$ とする.

Step3 (方向を仮定したロボットの軌道予測)

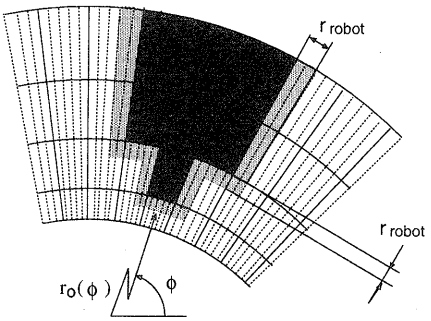


Fig.11 Growing obstacle

各方向毎に目標速度$\boldsymbol{v}_{bj}(j=1,\cdots,J)$を仮定する．

$$\boldsymbol{v}_{bj} = v_c \begin{pmatrix} \cos 2\pi\frac{j-1}{J} \\ \sin 2\pi\frac{j-1}{J} \end{pmatrix} \tag{10}$$

ここで，v_cは，目標速度の大きさである．次に，式(11)を用いて，微小時間毎のロボットの軌道$\boldsymbol{x}_j(t)$を予測する．

$$\boldsymbol{x}_j(t) = \begin{cases} \boldsymbol{v}_a(t+\delta t) + \frac{1}{2}\frac{\Delta\boldsymbol{v}_j}{|\Delta\boldsymbol{v}_j|}\alpha t^2 & t < \frac{|\Delta\boldsymbol{v}_j|}{\alpha} \\ \boldsymbol{v}_a\delta t + \frac{\Delta\boldsymbol{v}_j|\Delta\boldsymbol{v}_j|}{2\alpha} + \boldsymbol{v}_{bj}t & t \geq \frac{|\Delta\boldsymbol{v}_j|}{\alpha} \end{cases} \tag{11}$$

ここで，$\Delta\boldsymbol{v}_j(=\boldsymbol{v}_{bj}-\boldsymbol{v}_a)$は仮定した目標速度と現在速度の差，$\alpha$は加速度の上限値(設定パラメータ)，$\delta t$はロボットがコマンドを受け取るまでの空走時間を表わす．予測軌道の一例として，$\boldsymbol{v}_a=[400,0]^T$,$v_c=400$mm/s, α=300mm/s^2,δt=0.03sにおける各方向($J=16$)の軌道を**Fig.12**に示す．

Step4 (移動可能時間の導出)
各方向毎に，軌道を予測して障害物に衝突するまでの時間を求める．まず，方向jを固定し，$t=0$とする．そして，式(11)を用いて，Δt毎の軌道の座標値$\boldsymbol{x}_j(t)=[x_j(t),y_j(t)]^T$を求め，ロボット中心の距離と方向を算出する．

$$r_j(t) = \sqrt{x_j(t)^2+y_j(t)^2} \tag{12}$$

$$\phi_j(t) = \mathrm{atan2}(y_i(t),x_j(t)) \tag{13}$$

そして，以下により移動可能時間T_j(最初に障害物に衝突すると予測されるまでの時間)を導出する．

$$T_j = \min_{r_j(t)\geq r_o(\phi_j(t))} t \tag{14}$$

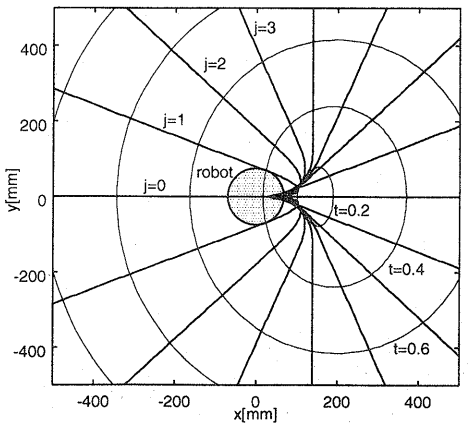


Fig.12 Estimation of robot's trajectory

方向jを1～32に変化させ，全方向の移動可能時間を求める．

Step5 (移動方向の決定)
移動可能時間$T_j(j=1,\cdots,J)$の値をある閾値T_{th}で移動可能/不可能の2値に判別し，移動可能の中で目標方向j_gに最も近い方向jを選ぶ．

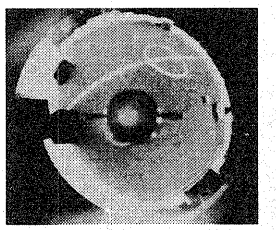
4 自己位置推定方法

複数台の移動ロボットが協調して作業を行なう場合，ロボット自身が自己位置を把握することは重要である．他のロボットと情報交換することで，お互いの相対関係が明確になり，正確な協調作業の実現が可能となる．特に，RoboCupのような複数台での高度な協調作業が要求される環境では，自己位置推定機能が必要不可欠である．

そこで，本研究では，RoboCup小型リーグ用のフィールド(2740mm×1525mm)を対象とし，全方位画像から得られる床領域から，簡便に自己位置推定を行なう方法について提案する．以下にその具体的なアルゴリズムについて説明する．

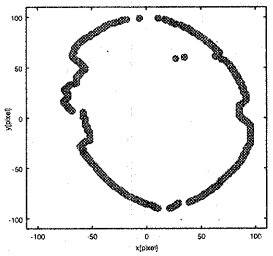
Step1 (画像の2値化)
画像処理装置により全方位画像を取り込み(**Fig.13**(a)参照)，床色である緑色を抽出して2値画像を得る(**Fig.13**(b)参照)．なお，画像の解像度は最小の256×220[pixel^2]とする．

Step2 (床領域の境界の抽出)
画像処理装置の領域座標抽出機能を使って，各Y座標$\{Y_1,\cdots,Y_N\}$に対応する領域のX座標の

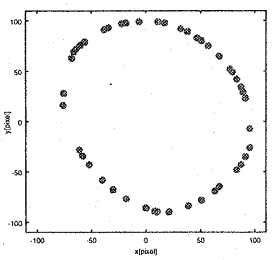


(a)Camera image

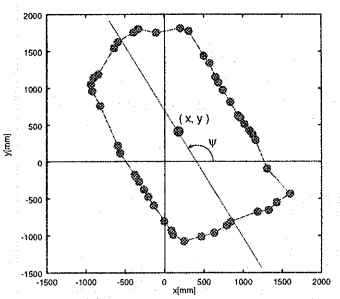
(b)Binarization



(c)Extract Boundary



(d)After convex hull operation



(e)Projection to floor

Fig.13 Example of method of self-localization

最小値 $\{X_{min1}, \cdots, X_{minN}\}$ と最大値 $\{X_{max1}, \cdots, X_{maxN}\}$ を求める (**Fig.13**(c) 参照).

Step3 (凸包処理)
凸包アルゴリズム [6] を用いて，得られた点列 $\{(X_{min1}, Y_1), \cdots, (X_{minN}, Y_N), (X_{max1}, Y_1), \cdots, (X_{maxN}, Y_N)\}$ から，凸包列 $\{(X_{c1}, Y_{c1}), \cdots, (X_{cM}, Y_{cM})\}$ に変換することにより床領域境界を復元する (**Fig.13**(d) 参照).

Step4 (床面上への投影と自己位置推定)
凸包処理後に残った頂点を式 (2)〜(6) を用いて，画像座標系からロボット座標系に変換し，点列 $\{(x_{c1}, y_{c1}), \cdots, (x_{cM}, y_{cM})\}$ を得る．この点列で得られる閉領域 (**Fig.13**(e)) の面積 S，重心位置 x, y，2次モーメント I，2次モーメントを最小にする主軸の傾き方向 ψ を求める．面積と主軸回りの2次モーメント値 (最大値，最小値) をチェックし，あらかじめ定められた範囲内にあれば処理が正常に行われたと判断し，以下の式で，ワールド座標系におけるロボットの位置姿勢 ${}^w x, {}^w y, \theta$ を求める．

$$\theta = -\psi, -\psi + \pi \quad (15)$$

$${}^w x = -x\cos\theta + y\sin\theta \quad (16)$$

$${}^w y = -x\sin\theta - y\cos\theta \quad (17)$$

なお，θ の曖昧さは，両ゴールとの位置関係や時系列の値から判断する．

5 検証実験

提案する障害物回避と自己位置同定方法について，我々のロボットシステムを用いて検証実験を行なった．以下に，その結果について報告する．

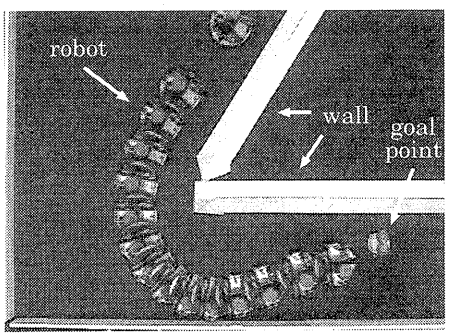
5.1 障害物回避

障害物回避の一例として，ロボットが約55度の鋭角のコーナを曲がり，ゴールポイントを目指す移動実験を考える．テンプレートのセルの分割数を $I = 6$, $J = 32$，ロボットの最高速度を $v = 400$mm/s，最大加速度を $\alpha = 300$mm/s^2，空走時間 δt=0.03s，移動可能領域の閾値 T_{th}=2.75s の条件の下で，移動速度を考慮しない回避法を用いた場合と考慮した回避法を用いた場合の2通りについて移動実験を行なった．**Fig.14**に 1/3s 毎の両者の軌跡を示す．

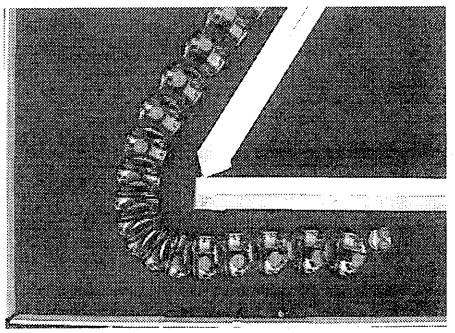
結果より，両者とも壁に接触することなくコーナを曲がり，ゴールポイントまで達しているが，移動速度を考慮した方が，早い時刻から方向転換を開始しているため，コーナーを無理なく曲がっていることが観察できる．

5.2 自己位置推定

自己位置推定の一例として，RoboCup 小型リーグ用のフィールド内で，ロボットが障害物回避を実行しているときの位置を推定する実験を考える．前節の実験に比べて処理が増えたため，空走時間を δt=0.08s とし，他の実験条件は前節の実験と同様とした．また，移動可能領域については移動速度を考慮したものを利用する．**Fig.15**に 1/3s 毎の実際の軌跡と自己位置推定値を示す．

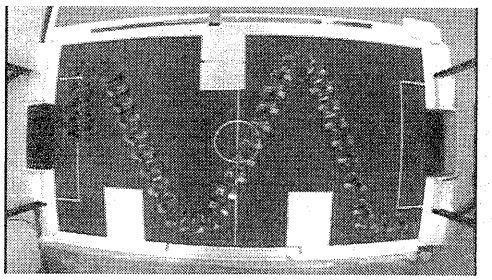


(a)Taking account of robot's velocity

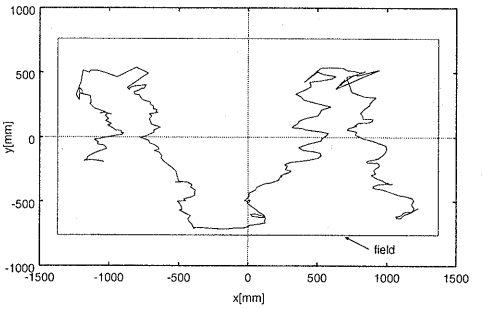


(b)Taking no account of robot's velocity

Fig.14 Experimental results of obstacle avoidance (top view)



(a)Real trajectory (Top view)



(b)Self-localization

Fig.15 Experimental result of self-localization

走行時の振動が全方位視覚に伝わっているため，推定位置のばらつきが大きいが，フィルタ処理等である程度の精度で自己位置推定が可能であると考えられる．また，4節の **Step4** の後に床面の矩形図形に対して，データの分散を考慮した最小2乗法を適用することも検討している．

6 おわりに

本論文では，ロボットに搭載された全方位視覚から得られる床領域に基づいた障害物回避方法と自己位置推定方法について提案した．また，床領域の処理に画像処理装置の機能を有効に用いることにより，処理の高速化を実現した．今後，我々は本手法を発展させ，ロボットと障害物がともに移動速度を持っている場合の障害物回避や自己位置推定の信頼性の向上を目指す予定である．

参考文献

[1] 浅田稔 他: "研究活動とロボットコンテスト(RoboCup)", 日本ロボット学会誌, Vol.15, No.1, pp.13-16, 1997

[2] M.Wada and S.Mori: "Holonomic and Omnidirectional Vehicle with Conventional Tires", 1996 IEEE Int. Conf. on Robotics and Automation, pp.3671-3676, 1996

[3] 関森大介 他: "左右独立駆動型全方向移動機構の制御アルゴリズム", ROBOMEC'00 講演会講演論文集, 2PI-41-056, 2000

[4] 関森大介 他: "全方位視覚画像のラベリングに基づく移動ロボットの障害物回避", 第 18 回日本ロボット学会学術講演会予稿集,Vol.3,pp.995-996,2000

[5] 八木康史: "実時間全方位視覚センサ", 日本ロボット学会誌, Vol.13, No.3, pp.347-350, 1995

[6] M.ドバーグ 他, 浅野哲夫訳: "コンピュータ・ジオメトリー計算幾何学: アルゴリズムと応用－", 近代科学社, 2000