

タイトル	天井設置カラー標的を利用する自律移動ロボットの 実時間走行
著者	深谷，健一；青塚，卓也；東岱，慎吾
引用	北海学園大学工学部研究報告，34：83-90
発行日	2007-02-20

天井設置カラー標的を利用する自律移動ロボットの実時間走行

深谷 健一*, 青塚 卓也**, 東岱 慎吾***

Real Time Navigation of Autonomous Mobile Robot Utilizing Color Landmark Mounted on Ceiling

Ken-ichi FUKAYA*, Takuya AOTSUKA** and Shingo TODAI***

要 旨

移動ロボットに搭載したカラーCCDカメラ装備の画像処理専用PCを用いて、ロボットを停止させることなく天井設置のカラー標的を読み取ることで自己位置認識をおこなう。認識結果をロボット内蔵のオンボードPCに送信し、2台のPCをプロセス間通信で連携動作させた。20cm/secの平均速度で、障害物のある実験室からドアを通り抜け廊下端まで実時間の連続走行が実現できた。

1. まえがき

ロボットの利用が産業用から民生用へと広がり、環境内を移動して人間との相互作用をするサービスロボットが自動車、テレビ、パソコン、携帯電話の次の生活の一部になると予測されている^{1),2)}。サービスロボットの移動には三つの要素“Where am I?”, “Where am I going?”, “How should I get there?”が不可欠であるが³⁾、第一の「いまどこにいるか」(自己位置認識)は自律移動ロボットにとって不可欠の要素技術であり、多くの研究がなされてきた⁴⁾。我々も廊下天井に設置した単純なカラー標的を用いて自己位置認識し、自律移動するシステムを構築してきた⁵⁾。しかし、標的のカラー画像処理とロボット移動を一つのオンボードPCで処理するため、ロボットが停止して画像処理を実施した後、位置誤差を修正して移動する非実時間の不連続走行であった。実用性を考慮すると、停止することなく自己位置認識をしながら連続走行す

* 北海学園大学工学部電子情報工学科

* Department of Electronics and Information Engineering, Faculty of Engineering, Hokkai-Gakuen University

** テイエステイ

** TST Corporation

*** 株式会社メイテック

*** MEITEC Corporation

ることが必要である。本報告では小型の画像処理専用PCを移動ロボットに搭載することで、実時間走行を試みる。

2. 自律移動ロボットシステム

図1, 2に構築したシステムの構成を示す。自律移動ロボット (Nscout) に画像処理専用ソフト (MVTEC社: HALCON) をインストールした画像処理専用PC (Linux社Fusion) が搭載されており, これにカラーCCDカメラ (SONY: EVI-D100) と無線LAN (WN-G54/C) を接続する。移動ロボット操作にはロボットに内蔵するオンボードPCを用いる。無線LANを接続した2台のPC間でプロセス間通信を行うことにより, 画像処理とロボット移動のデータを両方で送受信し, 停止することなく走行させる環境を構築した。さらに2台のロボット搭載のPCは無線LANを通して別に設置したコンソール型のLinuxPCから遠隔操作が可能である。

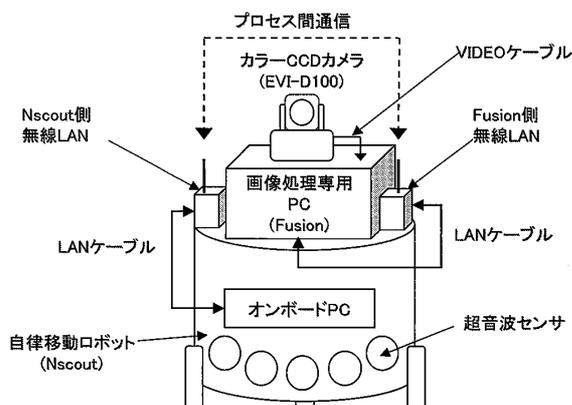


図1 自律移動ロボットシステム構成

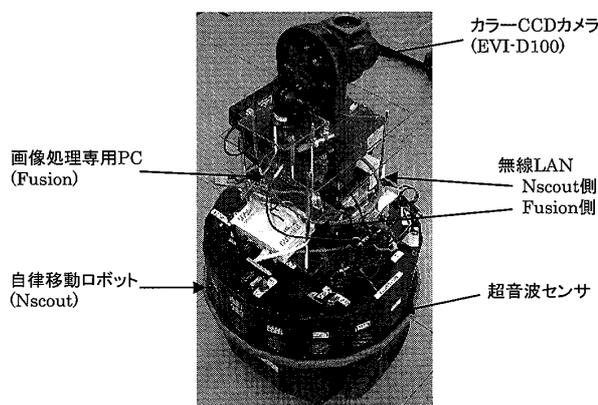


図2 自律移動ロボットシステム

3. 天井設置カラー標的による自己位置認識

3.1 カラー標的

カラー標的は図3に示す赤色, 青色の二つの正方形A (5 cm×5 cm), B (5 cm×5 cm) で構成されており, 実験室と廊下の天井にロボットを進行させる方向に直角に, またそれぞれの標的が一つ手前の標的の中心線延長上になるように設置する。本標的は二種のカラーを識別することで自己位置認識が容易であり, 天井に設置するので実験室・廊下利用の支障にならない。

図4に廊下の天井に設置したカラー標的とそれを用いる自律移動ロボット走行状況を示す。

3.2 カラー画像処理

標的のカラー画像抽出性能を改善するために, カメラから取得したRGB画像 (480×640画素) をI1I2I3形式に変換する⁶⁾。青と赤の正方形はそれぞれI2, I3特徴の諧調が高く, 閾値設定で検出が容易となる。検出領域の重心座標を求め標的の座標とする。図5にカラー標的の設置状況とカラー検出結果を示す。A, B標的の重心を結ぶ直線とカメラ座標軸との傾き角度

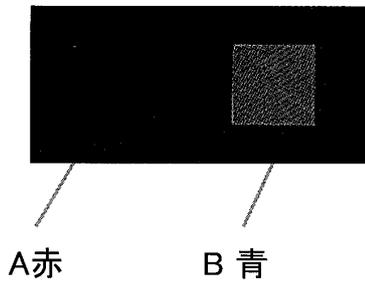


図3 カラー標的



図4 自律移動ロボットの走行状況

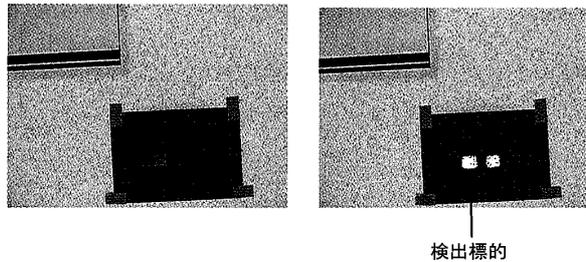


図5 天井に設置したカラー標的の検出結果

表1 カラー標的計測結果

設定値	-180	-135	-90	-45	45	90	135	180
平均値	-179.4	-134.8	-89.2	-43.4	44.2	92.8	136.6	180.2
標準偏差	0.5	0.4	1.0	0.5	0.4	0.4	0.5	0.4

単位 度

を変えて測定した5回の平均値と標準偏差を表1に示すが、設定値に対し数度の誤差であり自己位置認識に使うには十分な精度である。

3.3 自己位置認識

図6にカメラ視野座標系 ($o-x-y$)、標的、ロボットの位置関係を示す。画像処理から点A、Bの重心の座標が求められているので、これらの座標値を用いてロボット進行方向の修正角度 θ_1 と標的座標との前後の偏差 l 、左右の偏差 s が求められる。赤い正方形の重心座標を $A(X_A, Y_A)$ 、青い正方形の重心座標を $B(X_B, Y_B)$ 、この2点の中心点の座標を $C(X_C, Y_C)$ とし、カメラの視野の中心点を $O(X_O, Y_O) = (240, 320)$ とする。このとき点Oはロボットの中心に設定してあるので、点Oがロボットの現在位置になり、点Cが目標点となる。式(1)より修正角度 θ_1 、式(5)より前後の偏差 l と左右の偏差 s のそれぞれの絶対値が求まる。 l の符号はロボ

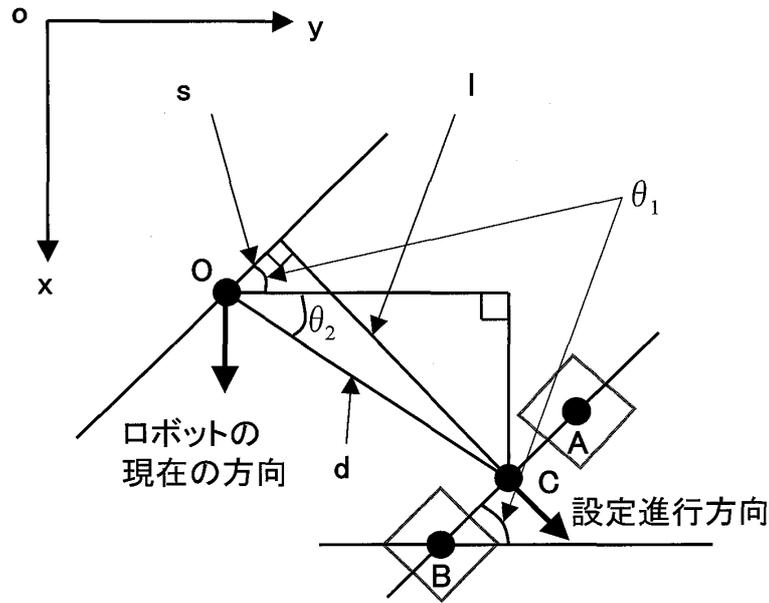


図6 カメラ座標系の標的とロボットの関係

ットの現在位置よりも目標点が前にあるとき+, 後ろにあるとき-となり, s の符号はロボットの現在位置よりも目標点が右にあるとき+, 左にあるとき-となる.

$$\theta_1 = \tan^{-1} \frac{X_A - X_B}{Y_A - Y_B} \quad (1)$$

$$X_C = \frac{X_A + X_B}{2}, Y_C = \frac{Y_A + Y_B}{2} \quad (2)$$

$$\theta_2 = \tan^{-1} \frac{X_O - X_C}{Y_O - Y_C} \quad (3)$$

$$d = \sqrt{(X_O - X_C)^2 + (Y_O - Y_C)^2} \quad (4)$$

$$s = |d \sin(\theta_2 \pm \theta_1)|, l = |d \cos(\theta_2 \pm \theta_1)| \quad (5)$$

画像処理専用PCの画像読み込み時間は平均0.02secなので, 移動ロボットの走行スピードが22.9cm/secのときには読み取り誤差が4.6mmとなる. これはカメラ視野座標に変換すると約2画素となり走行時の読み取り誤差としては小さく計算では考慮しなくてもよい.

3.4 コンピュータ間通信⁷⁾

2台の計算機は図7に示すコネクション型プロセス間通信によりデータ交換をおこなう. 画像処理専用PCからオンボードPCへは標的画像全体がカメラ視野に入るまでは“no”の文字を送り, 正常に標的画像が検出できた時点で図8に示す修正角度 θ_1 , 前後の偏差 l , 左右の偏差 s を12桁の数字列の形で送る. 数字列の前4桁が角度, 中央4桁が前後の距離, 後4桁が左右

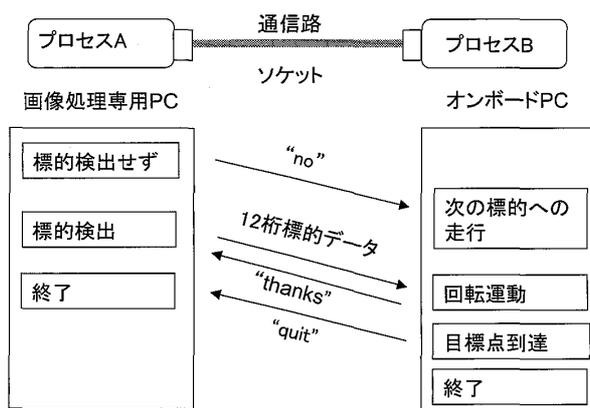


図7 PC間の通信手順

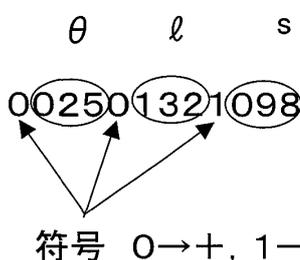


図8 計測データ送信フォーマット

の距離となっており、それぞれが一番手前の数字が正負の判別（0→+，1→-）に使われている。オンボードPCは標的データ受信時に“thanks”，目標点到達時に“quit”の文字列を画像処理専用PCに送る。

4. 移動ロボットの走行

4.1 リアルタイム走行

実験に用いた移動ロボットNscoutでは走行中の割り込みができない。オンボードPCは走行命令後、画像処理専用PCから文字列を受け取り、その後、再度走行命令を出すので、走行命令→読み取り命令→走行命令→読み取り命令、を繰り返し正確には間欠走行であるが、完全に停止する前に次の走行命令を与えることで、停止することなく連続走行が可能となる。

4.2 前進・回転運動

Nscoutは正確に曲がる事が出来ない為、前進しながら角度を修正できるように、画像処理専用PCから送られてきたデータ値に修正を加えて回転角度を設定する。内界センサ（エンコーダ）によって得られる角度を回転角度と同じになるように左右の車輪の速度を変えながら前進させる。本ロボットは差動駆動方式を採用している為、向きを変えるにはその場で回転をしなくてはならないが、滑らかな運動とはならないため、カラー標的を読み取った後、標的の真下まで移動するのではなく、次の標的に向けて直接移動するように設定した。このため図9に示すように、ロボットの補正を加えた修正回転角度 φ は検出標的と次の標的との距離 L 、

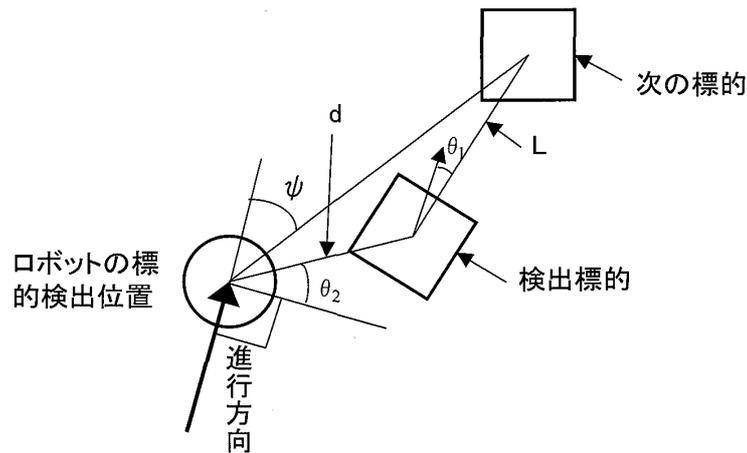


図9 ロボット修正回転方向

ロボットの標的検出位置と標的中心との距離 d ，角度 θ_2 およびロボットの進行方向修正角度 θ_1 から次式で求められる。

$$\psi = \frac{1}{2} \left(\frac{\pi}{2} + \theta_1 - \theta_2 \right) + \tan^{-1} \left[\frac{d-L}{d+L} \cot \left(\frac{\pi}{4} + \frac{\theta_1 + \theta_2}{2} \right) \right] \quad (6)$$

4.3 走行の流れ

- 1) プログラムを作動すると，画像処理専用PCの準備が出来るまで待機する。
- 2) 画像処理専用PCから“no”が送られている間，前進し続ける。前進では，左右の車輪の速度を変えることで自己位置を修正しながら設定してある角度に沿って進んでいく。
- 3) 画像処理専用PCから12桁の文字列が送られてくると，“thanks”の文字を送り，ロボット自身の設定角度を更新して旋回を行う。
- 4) 2)と3)を繰り返す，画像処理専用PCが最後の標的を読み込んで目標点に達したなら，画像処理専用PCに“quit”を送り，走行の終了を伝えてオンボードPCも終了し，ロボットが停止する。

5. 動作実験

図10に走行実験経路を示す。障害物のある実験室からドアを通り抜け廊下を走行する。この経路を移動ロボットが走行したときの標的による回転角度の測定結果を表2に示す。6回の実験で標的1から4で最大4度，標的5では7度の誤差で再現性がよい。最後の標的では誤差が大きいが，そこで目標点に到達している。1820cmの距離を88秒で，すなわち平均速度20.7cm/secで連続移動している。オンボードPCのみで画像処理とロボット移動を間欠的におこなったときには⁹⁾各標的で30秒停止しており，大きく改善された。

カラー標的検出は自律走行に十分な精度をえているが，蛍光灯の近くに標的を設置すると閾値処理をする際のI1I2I3の値が変化してしまい，標的をうまく検知できないときがあった。

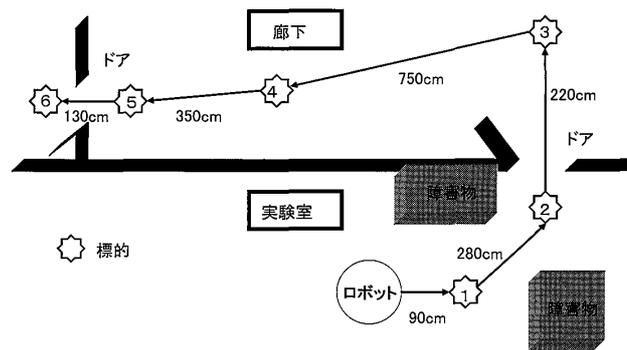


図10 走行実験経路

表2 標的毎のロボット回転角度

実験番号 標的	1	2	3	4	5	6	平均	最大 誤差
1	26	25	22	22	26	22	24	± 2
2	54	53	54	54	52	55	54	- 2
3	84	82	85	86	88	84	85	± 3
4	1	- 1	- 1	- 3	- 6	- 2	- 2	- 4
5	-0.3	3	- 1	- 1	- 9	- 2	- 2	- 7
6	-53	29	112	-82	-69	-84	-25	87

単位 度

5. あとがき

走行実験では廊下だけでなく、障害物のある実験室内でもカラー標的によるスムーズな連続移動ができた。蛍光灯による標的不検出問題に関しては閾値の値を環境に合わせて適応的に変化させる、あるいは蛍光灯の影響を受けないようにストロボなどの能動照明を用いる方法が考えられるが、今後の検討課題としたい。

本研究は北海学園大学ハイテク・リサーチ・センター研究費の支援を受けて行われた。

参考文献

- 1) R. D. Schraft, G. Schmierer : Service Robots, Products Scenarios Visions, A K Peters, 2000.
- 2) 経済産業省：技術戦略マップ（2-4. 製造産業分野, ロボット分野の技術戦略マップ）, www.meti.go.jp/press/20050330012/20050330012.html, 2005.
- 3) J. J. Leonard, H. F. Durrant-Whyte : Mobile Robot Localization by Tracking Geometric Beacons, IEEE Transactions on Robots and Automation, Vol. 7, No3, p. 376, 1991.
- 4) 深谷健一：自律移動ロボットの自己位置認識, 工学研究（北海学園大学大学院工学研究科紀要）, 第5号,

pp. 3 - 9, 2005.

- 5) 深谷, 尾倉, 武田: 天井設置カラー標的を用いた自律移動ロボット用自己位置計測法, 工学部研究報告, No.28, pp.61-67, 2001.
- 6) MVTec: HALCON ver6.1マニュアル
 $I1 = (R+G+B) / 3, I2 = R-B, I3 = (-R + 2G - B) / 2$
- 7) 雪田修一: UNIXネットワークプログラミング入門, 初版, pp. 1 -54, 2003, 技術評論社.