

レスキュー支援のための移動ロボットシステムに関する研究

Research on Mobile Robot System for Rescure Support

1 0 6 1 0 1 0

研究者代表 香川大学工学部知能機械システム工学科 准教授 高橋 悟

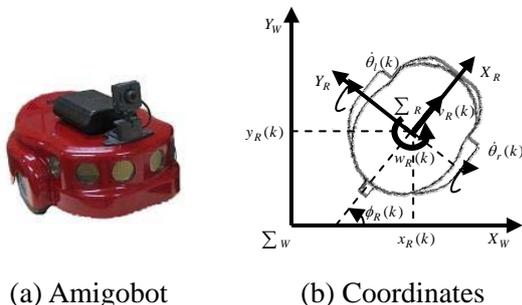
[研究の目的]

未知の環境下において行動する移動ロボットのレスキュー支援を目指し、障害物との衝突を回避しながら自律的に行動する手法を確立することを目的とする。

本研究では、カメラとソナーセンサを持つ移動ロボットが障害物との衝突なく目的地へと到達するための手法を新たに提案する。このとき、カメラとソナーセンサの情報を協調し、障害物の三次元位置を検出する。さらに、得られた情報に基づき自律的に障害物を回避しながら目的地に到達するための移動ロボットの軌道生成を実施する。

[研究の内容、成果]

本研究では、図 1 (a)に示す Active Media Robotics 社製の移動ロボット Amigobot を用いる。本移動ロボットは 1 台のカメラ、8 つのソナー、2 つのエンコーダ付独立駆動輪を持つ。カメラは光軸が移動ロボットの前進方向と平行に設置し、移動ロボットの位置 $(x_R(k), y_R(k))$ と姿勢 $\phi_R(k)$ をエンコーダ情報に基づき算出する。また、図 1 (b)に示すように移動ロボットを基準としたロボット座標系 Σ_R とワールド座標系 Σ_W を定める。



(a) Amigobot

(b) Coordinates

図 1 移動ロボットと座標系

1 障害物の検出手法

カメラとソナーに基づく障害物の位置検出手法を本章では述べる。このとき、カメラ画角内の情報を取得可能な前方二つのソナーを用いる。ここで、画像情報に基づく画像上での障害物検出法、

障害物を示すソナーの計測点位置の算出法、さらに各情報を統合して得られる障害物の位置決定手法について記す。

1-1 画像処理による障害物領域の検出

まず、障害物は移動体ではないと仮定する。ここで、移動ロボットが行動することにより生じる画像変化、すなわちオプティカルフローに着目する。オプティカルフローは連続して撮影された画像間に生じる速度ベクトルである。算出手法として、本研究ではブロックマッチング法を適用する。これはテンプレートマッチングを活用し、画像中の小領域の移動量を求めることで得ることができ、計算時間が長くかかる問題があり、ここではテンプレートマッチング時の画像情報を間引く等の工夫を施し対応する。テンプレートマッチングに正規化相関を適用し、二枚の入力画像を $f_{k-1}(i, j)$, $f_k(i, j)$ とおくとその相関値 $C(u, v)$ を次式により算出する。

$$C(u, v) = \frac{\iint_S (f_{k-1}(i, j) - \bar{f}_{k-1})(f_k(i+u, j+v) - \bar{f}_k) didj}{\sqrt{\iint_S (f_{k-1}(i, j) - \bar{f}_{k-1})^2 dx dy} \sqrt{\iint_S (f_k(i+u, j+v) - \bar{f}_k)^2 didj}}$$

ここで、最も相関値が大きい (u, v) をオプティカルフローベクトル (OFV) (u_{ij}, v_{ij}) と定める。取得したオプティカルフローの算出結果は図 2 (a)に示す。さらに、OFV を用いて画像上の障害物領域の検出を行う。各画素における OFV の大きさ r_{ij} と方向角度 ψ_{ij} ($0 \leq \psi_{ij} < 360$) を次式より導く。

$$r_{ij} = \sqrt{u_{ij}^2 + v_{ij}^2}$$

$$\psi_{ij} = a \tan 2(v_{ij}, u_{ij})$$

さらに、方向角を 30 度刻みに 0 から 11 の数字に量子化し、ラベリング処理を施すことで、同じ方向成分の領域分割を行う (図 2 (b)参照)。一般のラベリング処理では同じ色の画素をひとまとまりの領域とするが、ここでは近傍画素の色情報が隣り合うものであれば、それらの画素はひとまとまりの領域とみなす。本研究では、領域を分割した

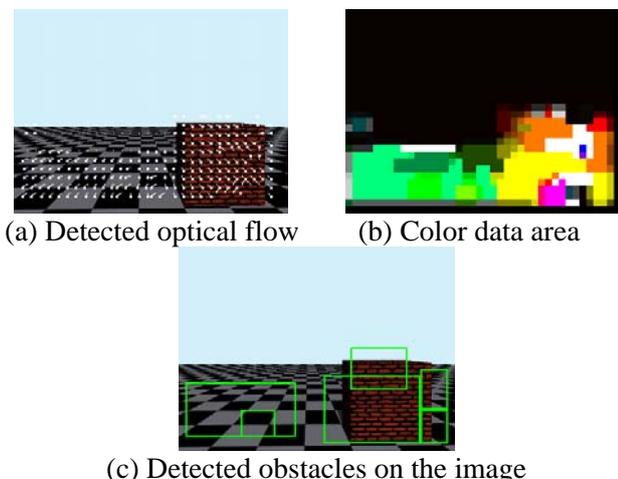


図2 画像処理による障害物領域の検出

際に、その領域を構成する画素の数が 100 画素以上であればその領域を障害物候補領域として検出する (図 2 (c)参照)。

1-2 ソナー計測点の算出

ソナーが示す計測点の位置計算法を述べる。障害物検出で用いるソナーは移動ロボットの前方向二つであり、1m 以内の情報を示す情報のみを扱う。

このとき、ロボット座標系 \sum_R での計測点位置 $({}^R x_n^p(k), {}^R y_n^p(k))$ は次式で得られる。

$$\begin{bmatrix} {}^R x_n^p(k) \\ {}^R y_n^p(k) \end{bmatrix} = \begin{bmatrix} x_n^s \\ y_n^s \end{bmatrix} + l_n(k) \begin{bmatrix} \cos \phi_n^s \\ \sin \phi_n^s \end{bmatrix} \quad (1)$$

ここで、 (x_n^s, y_n^s) 、 ϕ_n^s はロボット座標系 \sum_R におけるソナーの位置、姿勢角を表す。また、添え字 n はソナーの番号を表し、ここでは $n=1,2$ である。さらに、ワールド座標系 \sum_W でのソナー計測点は次式で与えられる。

$$\begin{bmatrix} {}^W x_n^p(k) \\ {}^W y_n^p(k) \end{bmatrix} = \begin{bmatrix} x_R(k) \\ y_R(k) \end{bmatrix} + \begin{bmatrix} \cos \phi_R(k) & -\sin \phi_R(k) \\ \sin \phi_R(k) & \cos \phi_R(k) \end{bmatrix} \begin{bmatrix} {}^R x_n^p(k) \\ {}^R y_n^p(k) \end{bmatrix}$$

1-3 センサ情報の統合による障害物位置検出

前節まで、画像上に障害物候補領域とソナー計測点位置情報が導出しているが、これら二つの情報のみでは障害物情報として不十分である。画像情報については、得られた領域情報が本当に障害物であるのか、またその障害物の三次元位置情報は未知である。一方、ソナーより得られる情報ではその障害物の大きさを知ることは困難である。従って、これらの情報を統合し、三次元位置、大

きさ情報を持った障害物の検出手法を次のように導く。二つの情報を画像上で対応させるため、ソナー計測位置を画像上に表示する。このとき、カメラから見たソナー計測点位置、すなわち式(1)で得られた値を次式に代入し、次の値を算出する。

$$\begin{bmatrix} {}^C x_n^p(k) \\ {}^C y_n^p(k) \\ {}^C z_n^p(k) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & c_x \\ -1 & 0 & 0 & c_y \\ 0 & -1 & 0 & c_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} {}^R x_n^p(k) \\ {}^R y_n^p(k) \\ {}^R z_n^p(k) \\ 1 \end{bmatrix}$$

ここで、 (c_x, c_y, c_z) はロボット座標系 \sum_R におけるカメラの位置を示す。次に、ソナー計測点上の画像撮影範囲を算出する。画像撮影幅 $x_n^c(k)$ と撮影高さ $y_n^c(k)$ は次式で表される。

$$x_n^c(k) = z_n^p(k) \cdot \frac{x_{CAM}}{f}$$

$$y_n^c(k) = z_n^p(k) \cdot \frac{y_{CAM}}{f}$$

ここで、 x_{CAM} 、 y_{CAM} はカメラの撮像素子サイズ、 f は焦点距離である。これらを用いて、画像上のソナー計測点位置 $u_n^p(k)$ 、 $v_n^p(k)$ は次のように計算される。

$$u_n^p(k) = X_{SIZE} \cdot \left(\frac{x_n^p(k)}{x_n^c(k)} + \frac{1}{2} \right)$$

$$v_n^p(k) = Y_{SIZE} \cdot \left(\frac{y_n^p(k)}{y_n^c(k)} + \frac{1}{2} \right)$$

ここで、 X_{SIZE} 、 Y_{SIZE} は画像サイズを表し、本研究では $X_{SIZE}=320$ 、 $Y_{SIZE}=240$ とする。さらに、ソナーは指向角 θ_{sd} を持ち、本研究では指向角を考慮したソナー計測範囲を画像に対応させる。ソナー計測点位置での指向角による広がり $W_n^p(k)$ は次式で算出する。

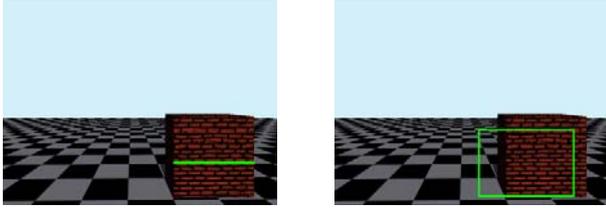
$$W_n^p(k) = \frac{X_{SIZE}}{x_n^c(k)} \cdot l_n(k) \cdot \tan \theta_{sd}$$

得られた情報を基に画像上のソナー計測範囲を図 3 (a)のように表示し、1-1 節で得られた障害物候補領域画像 (図 2 (c)参照) と重ね、候補領域とソナー計測範囲が一部分だけでも重なっていればその領域を障害物とみなし、候補領域に距離情報を与える。このとき、障害物検出画像を図 3 (b)に示す。さらに、得られた障害物の上下左右端を囲み、その中心座標 (u_{ob}, v_{ob}) 、幅 im_{ob} 情報を用いてワールド座標系 \sum_W での障害物位置 (x_{ob}, y_{ob}) と大きさ d_{ob} を次のように求める。

$$x_{ob} = (u_{ob} - \frac{X_{SIZE}}{2}) \cdot \frac{x_n^C(k)}{X_{SIZE}}$$

$$y_{ob} = {}^W y_n^p(k)$$

$$d_{ob} = im_{ob} \cdot \frac{x_n^C(k)}{X_{SIZE}}$$



(a) Sonar range data (b) Detected obstacle

図3 情報統合による障害物の検出

2 障害物回避手法

前章で得られた障害物の情報に基づき障害物を回避する手法について述べる。

2-1 障害物回避軌道の生成

図4, 図5に示す障害物を回避するための軌道生成する手法について説明する。

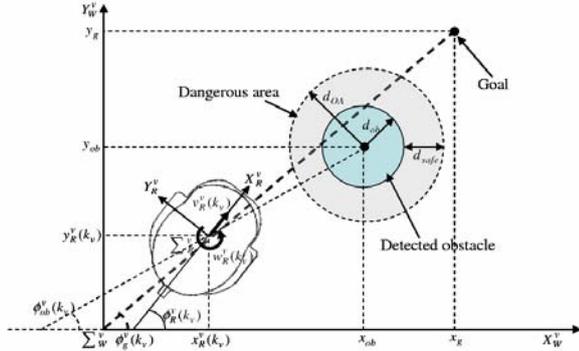


図4 障害物回避軌道の生成

本研究では、仮想空間を定義し、その中で仮想移動ロボットが障害物を回避する軌道を作成する。すなわち、地障害物が検出された瞬間の移動ロボット、障害物、目的地の位置を仮想空間に配置し、さらに障害物の大きさを考慮した衝突危険領域 d_{OA} ($= d_{safe} + d_{ob}$) を設ける (図4参照)。この仮想空間上で仮想ロボットに並進速度 $v_v(k_v)$, 回転速度 $w_v(k_v)$ を与え、目的地に到達させる。ここで、並進速度 $v_v(k_v)$ は一定とし、回転速度 $w_v(k_v)$ を次式で与える。

$$w_v(k_v) = \begin{cases} A(d)(\phi_{ob} - \phi_R(k_v) \pm \frac{\pi}{2}) & \text{if } d \leq d_{OA} \\ \phi_g(k_v) - \phi_R(k_v) & \text{otherwise} \end{cases} \quad (2)$$

$$A(d) = \frac{1}{\exp(d^2/\sigma^2)}$$

$$\sigma = d_{OA} / 2$$

$$d = \sqrt{(x_R(k_v) - x_{ob})^2 + (y_R(k_v) - y_{ob})^2}$$

$$\phi_{ob}(k_v) = \text{atan2}(y_{ob} - y_R(k_v), x_{ob} - x_R(k_v))$$

$$\phi_g(k_v) = \text{atan2}(y_g - y_R(k_v), x_g - x_R(k_v))$$

このように設定した仮想移動ロボットが逐次通過する点を移動ロボットの軌道と定める。本研究における軌道情報はこの点情報の集合として次のように $((x_R^v(0), y_R^v(0)), (x_R^v(1), y_R^v(1)) \dots (x_R^v(k_v), y_R^v(k_v)) \dots (x_R^v(k_{end}), y_R^v(k_{end})))$ と表す。また、式(2)中の符号±の選択により、障害物の左側、または右側を通る場合の二種類の軌道が得られる。ただし、どちらの軌道が移動ロボットに対して付加が少ないかが軌道選択の判断基準とする。本研究では次式を最小にする軌道を常に選択する。

$$J_n = \sum_{k_v=0}^{k_{end}} \text{atan2}(\Delta y_R^v(k_v), \Delta x_R^v(k_v))$$

例えば、図5においては障害物の左側を通過する軌道が選択されている。

2-2 生成軌道への追従制御

前節で導出した障害物回避軌道に移動ロボットを追従させる。まず、移動ロボットの座標 $(x_R(k), y_R(k))$ と最も近くに位置する生成軌道の座標点 $(x_R^v(k_v), y_R^v(k_v))$ を探索する。探索された軌道上の座標点から $step$ 個先の軌道上の座標点 $(x_R^v(k_v + step), y_R^v(k_v + step))$ を移動ロボットが目指すべき点、すなわち参照点 $(x_d(k), y_d(k))$ として定義する。ここで、 $step$ は実験的に適用に定め、値が小さいほど軌道に正確に追従するが、移動ロボットの動作が滑らかでない。また、 $step$ 値が大きいほど移動ロボットの動きは滑らかになるが軌道上を正確に走行することはできない。実験的に $step=10$ とする。

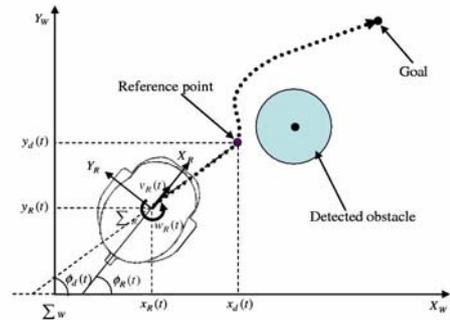


図5 生成軌道への追従

次に、設定された参照点 $(x_d(k), y_d(k))$ に向うための移動ロボットの並進速度 $v_R^{cmd}(k)$ と旋回速度 $w_R^{cmd}(k)$ を次式に従い算出する。

$$v_R^{cmd}(k) = \alpha \cdot \sqrt{(x_R(k) - x_d(k))^2 + (y_R(k) - y_d(k))^2}$$

$$w_R^{cmd}(k) = \beta \cdot (\phi_d(k) - \phi_R(k)) + \dot{\phi}_d(k)$$

$$\phi_d(k) = a \tan 2(y_R(k) - y_d(k), x_R(k) - x_d(k))$$

$$\dot{\phi}_d(k) = \phi_d(k) - \phi_d(k-1)$$

導出された並進速度 $v_R^{cmd}(k)$ 、旋回速度 $w_R^{cmd}(k)$ の具現化ため、各車輪の回転速度を次のように求める。

$$\dot{\theta}_r^{cmd}(k) = \frac{1}{2a} \cdot (b \cdot w_R^{cmd}(k) + 2 \cdot v_R^{cmd}(k))$$

$$\dot{\theta}_l^{cmd}(k) = \frac{1}{2a} \cdot v_R^{cmd}(k) - \dot{\theta}_r^{cmd}(k)$$

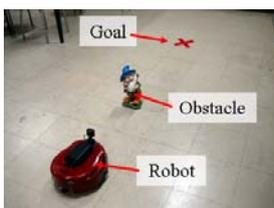
ただし、上述した手法にて生成される軌道は障害物の奥行きについて十分な考慮は成されていないため、カメラ視野外、すなわち移動ロボットの側面方向に存在する障害物の回避に困難を生じる場合がある。

3 実験結果

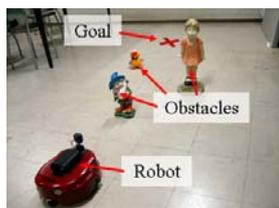
前章において、提案した各種手法の有効性を実験により検証する。

第一に、図 6 (a) のように障害物が一つだけ存在する場合における移動ロボットの挙動を確認する。このとき、目的地は移動ロボットの初期位置から前方 2m の位置に設定する。実験結果を図 7 に示す。実験結果より移動ロボットは障害物を回避し、目的地に到達していることがわかる。しかし、移動ロボットが検出した障害物とソナーが計測した障害物の位置誤差が生じている。また、生成軌道に対して追従遅れやオーバーシュートがあることがわかる。

第二に、図 6 (b) に示すように複数の障害物を配置した場合を検討する。目的地位置は移動ロボットの前方 3m とする。図 8 の実験結果より障害物は回避であることが言える。



(a) Experiment 1



(b) Experiment 2

図 6 実験構成

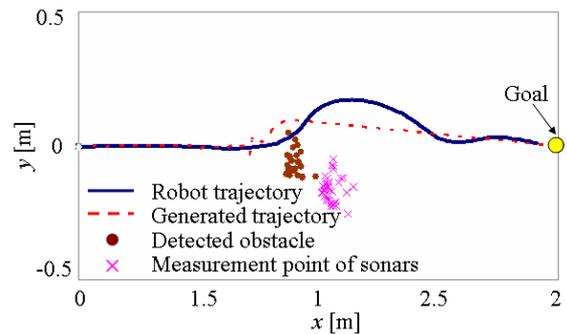


図 7 実験結果 1

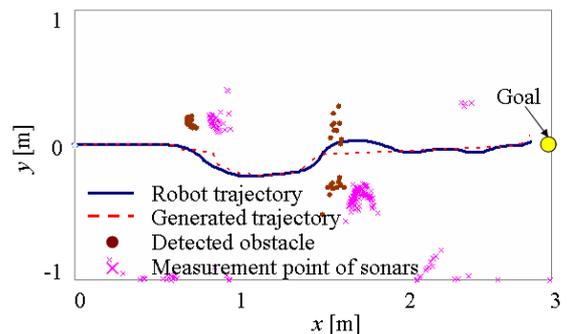


図 8 実験結果 2

[今後の研究の方向, 課題]

今後、移動ロボットの側面方向の情報を組み込み最適な衝突回避軌道生成を実施する。さらに、リアルタイム性を向上させるために、プログラムの高速化をはかる。現在、屋内実験のみ実施しているため、レスキュー支援実践型を目指し、本手法の本格的な屋外実験を行い、さらなる課題の抽出を行い、手法改善を行う予定である。

[成果の発表, 論文等]

- 1) 奈良俊介, 高橋悟, 日高浩一: センサーフュージョンによる移動ロボットのナビゲーション, 第 49 回自動制御連合講演会論文集, SA-9-3-4, CD-ROM (2006)
- 2) 奈良俊介, 高橋悟: 視覚センサとソナーセンサを適用した移動ロボットのハイブリッド制御, 第 7 回計測自動制御学会 SI 部門講演会論文集, pp. 918-919 (2006)
- 3) 奈良俊介, 長原大輔, 高橋悟: 移動ロボットナビゲーションに基づく障害物回避制御, 電気学会産業計測制御研究会資料, pp. 13-16 (2007)