計測自動制御学会東北支部 第260回研究集会 (2010.10.29)
資料番号 260-2

視覚センサを用いた移動ロボット

Mobile Robot Using Visual Sensor

○三谷大輔*，大久保重範**,高橋達也***

○Daisuke Mitani*, Shigenori Okubo**,Tatsuya Takahashi***

*山形大学

*Yamagata University.

キーワード： 視覚センサ (visual sensor)，画像処理(image processing)，2値化 (binarization)，膨張収縮 (dilation and erosion)，差分処理（subtraction processing）

連絡先： 〒992-8510　山形大学米沢市城南4-3-16 山形大学 工学部 機械システム工学専攻 大久保研究室
三谷大輔，Tel.: (0283)26-3245，Fax.: (0238)26-3245，E-mail: sokubo@yamagata-u.ac.jp

1. はじめに

近年世界では様々なロボットが活躍している．産業用に限らず家庭での利用を目的としたロボットも多く開発されてきている．その中でもロボット自ら判断して人間同様の働きをこなすような自律型の移動ロボットの開発が期待されている．自律行動をする上で視覚情報は周囲の情報を読み取るのに長けておりコンピュータの発達と共に視覚情報を利用したシステムが増えてきている．

本研究ではカメラから画像を取り込み，その画像を処理することによって目標物を認識させロボットに追従させることを目標としている．カメラから取得した画像は多くの情報を含んでいるが，必要な情報の選定，処理時間などが問題となってくる．

2. 開発環境

使用したパソコンとカメラの概略をTable 1に示す.，ソフトはMicrosoftでフリーで配布されているVisual C++にDirectXを使用する．カメラはCCDよりも消費電力が少なく，高速に処理できるCMOSカメラを用いた．

Table 1　Development environment

Host Computer	
OS	Windows XP SP2
CPU	Pentium M 1.70GHz
software	Visual C++ 2005
Camera	
performance	320×240 CMOS camera

研究で使用したロボットの全体図をFig.1に示す．ロボットにはH8-3069Fマイコンボードを搭載し，RS-232Cでホストコンピュータとシリアル通信を行う．ホストコンピュータでは，

カメラより得られた画像の画像処理，およびロボット側のCPUに対するプログラムのクロスコンパイルを行う．

Fig. 1　Robot

3.　画像処理

3.1　２値化

２値化とはある基準となる値Tをしきい値(threshold)とし，それより大きい値を１，小さい値を０として表す方法である．

$$g[i,j] = \begin{cases} 1 & (f[i,j] \geqq T\ ) \\ 0 & (f[i,j] < T\ ) \end{cases} \qquad (1)$$

コンピュータ内部の色表現は通常RGB形式で記憶されるが，RGB形式は色情報や明るさ情報が混ざっており色の判別が難しい．そのためカメラのフラッシュや照明などで光による明るさの変化がある場合，色の抜き出しが困難になる．そこでHSV形式を使用することにする．

HSV形式は色の空間を色相（Hue,0〜360），彩度（Saturation,0〜100），明度（Value,0〜100）の3種類に分ける手法である．HSV形式の利点は, 明るさや鮮やかさの調整ができるので中間色の判別が容易となる．RGBからHSVへの変換は次式のように表すことができる．

$$V_{max} = max\{R,G,B\}$$

$$V_{min} = min\{R,G,B\}$$

$$V = V_{max} \qquad (2)$$

$$S = \frac{V_{max} - V_{min}}{V_{max}}$$

$$H = \begin{cases} \dfrac{\pi}{3}\left(\dfrac{G-B}{V_{max}-V_{min}}\right) & (R = V_{max}\text{のとき}\ ) \\ \dfrac{\pi}{3}\left(2 + \dfrac{B-R}{V_{max}-V_{min}}\right) & (G = V_{max}\text{のとき}\ ) \\ \dfrac{\pi}{3}\left(4 + \dfrac{R-G}{V_{max}-V_{min}}\right) & (B = V_{max}\text{のとき}\ ) \end{cases}$$

また，$H < 0$のときは2πを加える．$V_{max} = 0$のときは，$S = 0$，H =不定となる．

3.2　ラベリング

ラベリングとは，2値化された画像上において連結された画素に番号をつけることで画素を集合ごとに分類する処理である．今回は隣接する8方向に同じ番号をつける8近傍のラベリングを用いた．この処理で連結成分が最大面積のものを目標物として認識させる．

3.3　膨張と収縮

ラベリング処理を施した画像上で,最大面積の画素にラベル1, それ以外の画素にラベル0をつける．ラベル1の画素のまわりにラベル0が接している場合, 0を1として画素をひとまわり加える処理を膨張という．逆に1を0とし画素をひとまわりはぎとる処理を収縮という．これはノイズや光の影響で画像をうまく抜き出せない場合に有効である．抜き出した画素の形状をなめらかにするために8近接で膨張処理を行った後，4近接の収縮処理を行うことにする．膨張と収縮を行った例をFig.2に示す．また，n回膨張処理を行った後，n回収縮処理を行う処理をクロージングといい，逆の処理をオープニングという．今回は2回クロージング作業を行った．

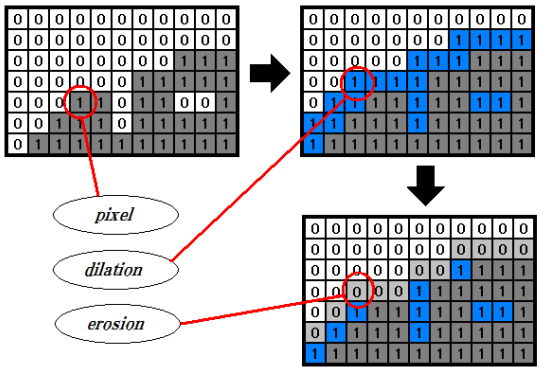

Fig. 2　Dilation and erosion

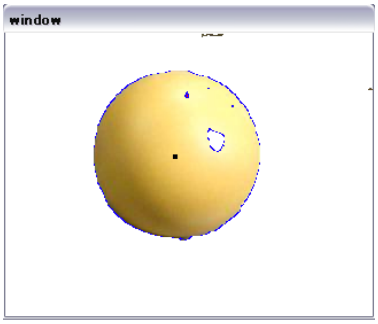

Fig. 3　Closing image

Fig.3にクロージング処理を施した画像を示す．画像上の青色画素が1度目の膨張処理，赤色画素が2度目の膨張処理を表している．処理時に抽出しきれなかったボールの欠落箇所を補正しているが，欠落部分が大きい場所では補正しきれていないことがわかる．クロージングの実行回数を増やすことで，さらに補正することが可能となるが，画像の情報を著しく損なう可能性があるので充分考慮して補正処理を行う必要がある．

3.4　形状認識

ラベリングで抜き出された画像が必ずしも認識したい目標物であるとは限らない．今回の目標物はボールと定めているため，画像上の表示は円形になる．そこで抜き出した画像の重心を求め，重心から上下左右に斜め4方向を加えた8方向の距離を出し,その平均を求める．その平均と8方向の距離の差が少なければ目標物として認識する．また，平均距離から面積を求め，面積とラベリングで抜き出した画素数とを比較し，両者の差が少ないようなら目標物として認識する．

4.　移動物体の検出

移動物体の画像処理は，監視カメラの不審者の発見や一般道路の車両の取締り，スポーツの運動解析などの目的で幅広く使われている．

動画像の処理は，静止画一枚毎の集合体であるため，コンピュータ上では画像に映し出された物体が移動しているかどうかを判断することができない．実際に動作している物体がわかり，事前に予測してカメラを目標物に向けることができれば画像処理をする上でも効率がよくなる．今回は異なる時間の2枚の画像を比較することで，変化の生じた領域を抽出する差分処理で動物体の検出を行う．ボールを手に持ちながら動かした時の差分処理結果をFig.4に示す．動きのある物体を捉えることができたが，変化の違いを色相で比較しているため動きがあった場所でも色相の変化がみられない場合は検出することができなかった．

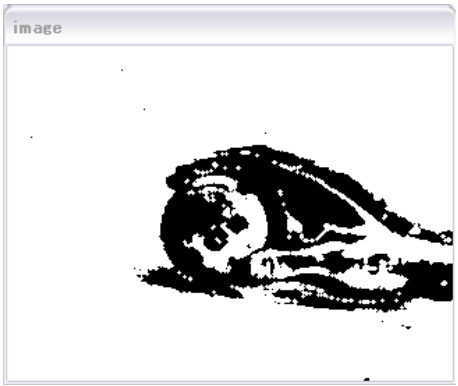

Fig. 4　Subtraction processing

5. 結果

同じボールが画面上に存在する時，移動しているボールだけを認識することができるか確かめる実験を行った．処理結果をFig.5に示す．差分処理に色の指定を加えることで動きのあるボールだけを抽出している．

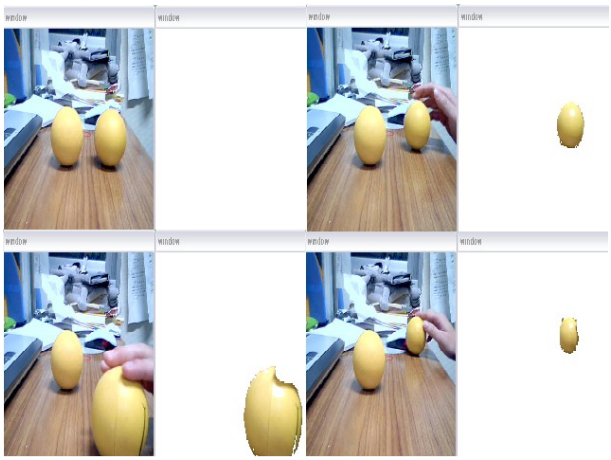

Fig. 5　Processing image

6. 考察

今回差分処理を用いることで移動するボールを捉えるができたが，静止した状態の物体が抽出されることがあった．これはカメラのノイズや画像取得時の微小な変化によるものだと思われる．この影響を抑えるために変化の度合いが顕著になった時に認識するように改善したが，完全には抑えることはできなかった．またボールが2つ存在し両方に変化がみられる場合，現在では片方しか検出することができない．そこで条件に適した物体ならそれら全てを捉えたり，指定条件を増やすことでより条件に適した物体を抽出できるように改善を図っていく必要がある．

7. Windowsアプリケーション

これまではコンソールアプリケーションを用いてプログラムを実行してきた．しかしこの方法だと別のプログラムを実行したい時，その都度コンパイルしなくてはならなく状況に合わせた画像処理ができない．そこでツールバーやスクロールバーによってプログラムを変更することができるアプリケーションを開発することで同一のツールで複数の処理ができるようにする．アプリケーションをFig.6に示す．

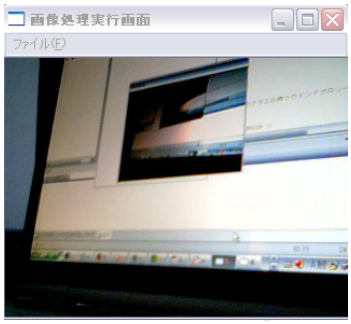

Fig. 6　Windows application

これを使用することで，コンパイルの手間を省けウィンドウ上で状況に応じたプログラムを実行することができる．

8. おわりに

物体の抽出処理を行うにあたり，より正確に対象物を捕捉できるように努めてきた．しかし安価なカメラではノイズが発生しやすく光の影響も受けやすい．画像処理の種類を増やすことができれば様々な問題にも対応できるようになるが，必然的に処理が重くなってしまう．そこで今回作成したアプリケーションと組み合わせることで状況に応じたプログラムを実行し，PCに負担の掛からないような画像処理の開発を行っていく．

参考文献

1) 蔵前智映:視覚センサを用いた自立移動ロボットの開発，修士論文(2006)

2) 土井滋貴：はじめての動画処理プログラミング，CQ出版社(2007)

3) 奥富正敏（編集委員長） ディジタル画像処理編集委員会:ディジタル画像処理，財団法人画像情報教育振興協会（CG-ART協会）(2004)