

視覚センサを用いた自律移動ロボットの目標物認識

The Recognition of Target by Autonomous Mobile Robot Using a Vision Sensor

齋藤秀幸*, 大久保重範*

Hideyuki Saito*, Shigenori Okubo*

* 山形大学工学部

*Faculty of Engineering Yamagata University,

キーワード : 画像処理 (Image Processing), 視覚センサ (Vision Sensor), 色検知 (Color Detection)

連絡先 : 〒992-8510 山形県米沢市城南 4-3-16 山形大学工学部 機械システム工学科 大久保研究室 齋藤秀幸,
Tel.: (0238)26-3245, Fax.: (0238)26-3245, E-mail: sokubo@yz.yamagata-u.ac.jp

1. はじめに

本研究室ではカメラを用いた視覚センサによる目標物を発見追従, 把持させたり, ライントレースをする研究を行ってきた^{1, 2)}. これらの研究で使用しているプログラムでは, 色で目標物を認識している. そのため, 画像処理を行う際に予め抽出したい色を設定しておく必要があり, 特定の色しか認識することが出来なかった. 異なる目標物を認識するためには, 抽出したい色を設定しなおす必要があるため, 汎用性の乏しいものであった.

そこで, カメラで目標物を認識するときに, ロボットが抽出したい色を判断させる事で, 異なる色でも認識することは可能になった. しかし, カメラによる目標物の捉え方によっては, 認識することが出来なくなることがあり動作が不安定であった. 今回はこの色を判断させる方法について改善した点について考察する.

2. 画像処理

カメラの映像から目標物の色だけを抽出 (2 値化) し, 抽出した色の画素の一番大きな塊のみを残しそれ以外を消去 (ラベリング) する. さらに, ロボットが目標物の位置を認識するために, 抽出した一番大きな塊の重心を求める処理 (重心計算) を行っている.

これに, 色を判断させるために, Fig.1 の Old から Fig.1 の New のように 2 値化処理の前に色検知という処理を追加することにより, 異なる色の目標物を認識することが出来るようになった.

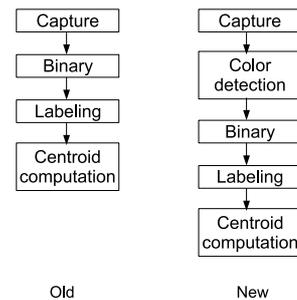


Fig. 1 The flow of image processing

2.1 2 値化

2 値化とは、濃淡画像から 2 値画像を得ることをいう。目標物以外の色を白くすることで、目標物の色と白の 2 値画像が出来る。2 値化する利点は、情報量を削減することにより処理速度が速くなるということである。

2.1.1 HSB による 2 値化

RGB 形式は 1 つの色を表現するのに赤、緑、青の 3 つのパラメータを用いるため、色の判定が煩雑になる。このため本研究では色の識別を 1 つのパラメータで表せる HSB 形式を用いている。HSB とは色の種類を表す色相 (Hue)、色の鮮やかさを表す彩度 (Saturation)、色の明るさを表す明度 (Blightness) の要素に分けた表色系である。各パラメータは色相は $0 \sim 360^\circ$ 、彩度は $0 \sim 100\%$ 、明度は $0 \sim 100\%$ の値である。HSB の利点は、色相のパラメータで色を識別できるため、周囲の明るさの影響が小さい。

2.2 ラベリング

2 値化した画像だけでは、目標物の色以外にノイズが抽出される。このノイズにより、ロボットが目標物の位置を正確に検出できなくなる可能性がある。そこで、ラベリング処理を行いノイズの除去を行う。

ラベリング処理とは、画像内の連結している画素にラベルをつける事で、画素を複数の塊として分類することが出来る。その塊のなかで、最大面積の塊を抜き出すことによって、ボールだけを抽出することが出来る。本研究室ではラベリングの手法に、4 近傍ラベリング、ラインブロックラベリングという手法があるが²⁾、今回は使用したラインブロックラベリング手法について説明する。

ラインブロックラベリング手法とは、横に連結している画素を 1 つのブロックとして捉え、そのブロックの左端を最小値、右端を最大値とする。その最小値と最大値の位置関係によりラベル付けを

行っていく。Fig.2, Fig.3 のように $Max1 \geq Min2$ かつ $Min1 \leq Max2$ の時、それぞれの 2 つのブロックは連結していると判断する。ブロックの最小値と最大値の位置関係で画素の連結を判断することが出来るため、従来使われていた隣接する 4 つの画素を調べるといった方法より調べる画素が削減できる。よって、画像処理速度が向上できるため、今回用いることにした。

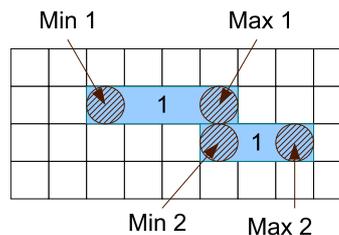


Fig. 2 lineblock labeling1

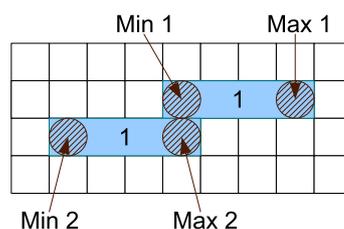


Fig. 3 lineblock labeling2

2.3 色検知

色検知という処理は、まず、Fig.4 の黒枠内部の色相に注目する。黒枠内部を簡略的に表したのが Fig.5 である。これら色相の平均値を求めて、2 値化する際の判別条件に設定することで、予め抽出したい色を設定する必要がなくなった。

例えば、Fig.4 の左図のように枠内に表示される色相が Fig.5 のように単色であれば、平均値は 40 である。しかし、Fig.4 の右図のように黒枠内に絵柄などの、認識したい色と異なる色が含まれてしまう場合は Fig.5 の右図のように複数の色が含まれていると、平均値を求めると 89.25 という値であり、目標物の色相

はほとんどが 40 付近の色なので、目標物を認識することができない。

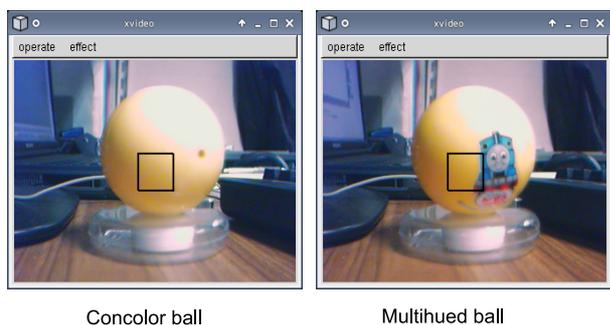


Fig. 4 The Recognition of Target Thing

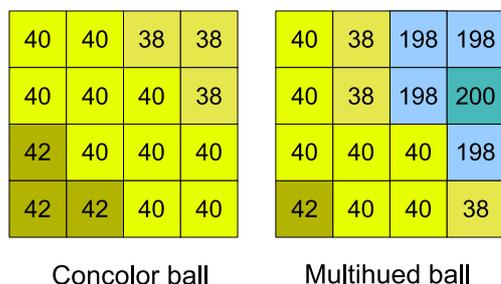


Fig. 5 Attention Pixels

そこで、平均値を求めるのではなく、中央値を使用することで、この状況から抜け出そうと考えた。中央値の計算は、 x_1, x_2, \dots, x_n を実数とする。これを小さい順に並べたものを x'_1, x'_2, \dots, x'_n とするとき、 $x = (x_1, x_2, \dots, x_n)$ の中央値 $Q_{\frac{1}{2}}(x)$ は

$$Q_{\frac{1}{2}}(x) = \begin{cases} x'_{\frac{n+1}{2}} & \text{if } n \text{ is odd} \\ \frac{1}{2}(x'_{\frac{n}{2}} + x'_{\frac{n}{2}+1}) & \text{if } n \text{ is even} \end{cases}$$

を用いて計算した。その結果中央値は 40 となり、目標物を認識することができる。

3. 実験

Fig.4 の両方の場合をビジュアルトラッキングを行った。

3.1 実験方法

視覚的にわかり易いように抽出する色以外を白とすることで抽出した色がわかるように設定した。Fig.4 に示すように画面の黒枠に抽出させたい色をセットしてから画像処理を行う。

3.2 実験結果

Fig.6, Fig.7 に黒枠内が単色のととき、複数色が含まれているときの結果を示す。また、そのとき求めた平均値と中央値、さらに抽出した面積を Table 1, Table 2 に示す。

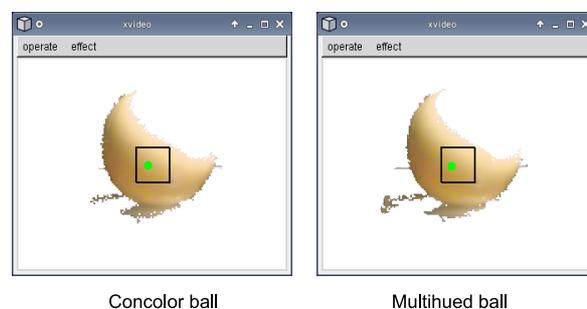


Fig. 6 The extraction of concolor ball

Table 1 Concolor ball

	Average		Median	
	Hue	Area	Hue	Area
Average	38.29	1.503×10^4	39.34	1.408×10^4
Standard Deviation	0.5199	105.7	1.301	237.1

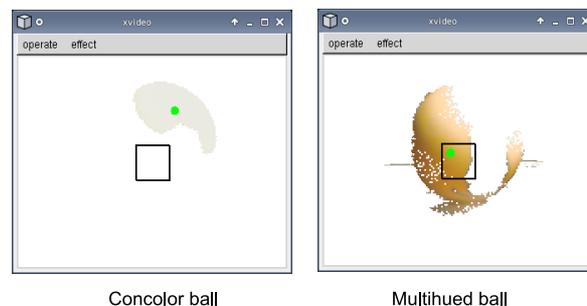


Fig. 7 The extraction of mutihued ball

Table 2 Multihued ball

	Average		Median	
	Hue	Area	Hue	Area
Average	75.54	188.7	35.06	1.016×10^4
Standard Deviation	1.876	87.77	0.06222	100.8

3.3 考察

Fig.4 の左図では、画像処理実行後、Fig.6 のように画像処理を変更しても見た目上は変化が見られない。また、Table 1 では、黒枠内が単色のときは、平均値も中央値も色相の値はほぼ変わらない値がでた。しかし、中央値の抽出した面積は、平均値で抽出できた面積より少なく表示されている。これは、実験環境は変わらないように設定しているのだが、光の影響などで多少見えにくくなったことが要因であると考えられる。

Fig.5 の右図では、画像処理実行後、Fig.7 の左図のように抽出すること出来ないが、中央値では、Fig.7 の右図のように、抽出することに成功する。また、Table 2 を見てわかるように抽出したい色の色相が求められていない。

これにより、多少目標物に別の色が含まれていても目標物を認識できるようになった。しかし、未だに黒枠内から目標物が外れてしまうと認識できなくなってしまうという現象は解消されていない。そこで、次にその解決案として別のアプローチを考えたのですこし説明する。

4. 解決案

色を記録するという動作を人間がすることで解決しようと試みている。画像を描画したウィンドウの画面をクリックすると、Fig.8 のような、画像処理を選択するウィンドウが表示される。このツールウィンドウで画像処理の選択をすると、Fig.9 のようなウィンドウが表示される。Manual を選ぶと Fig.10

のようなウィンドウが表示され、色相、彩度、明度、色の範囲を手動で設定できるウィンドウが表示される。これらのツールを使い、色検知によって求められた色相の値を設定することで、色の記憶という工程の代わりをおこなう。



Fig. 8
Tool window



Fig. 9
Method tool
window

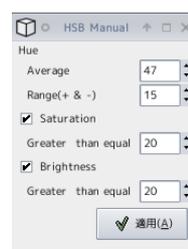


Fig. 10 Manual tool window

5. おわりに

今回は中央値を用いた色検知の不安定さの改善点と色の記憶させるための解決案について述べた。今後は、色を記憶させることと、実機に実装したときに適応できるかを検証していきたい。

参考文献

- 1) 蔵前智映 他：視覚センサを用いた自律移動ロボットの開発，計測自動制御学会東北支部，第 224 回研究集会 (2005.10.14)，資料番号 224-3
- 2) 山口智充 他：視覚センサを用いた自律移動ロボットによるライントレース，計測自動制御学会東北支部，第 238 回研究集会 (2007.10.18)，資料番号 238-6