

幾何学的模様の歩道のサインパターン抽出

西川 和弘 森 英雄
山梨大学工学部

あらまし カラータイル張りを施した幾何学模様を持つ歩道で視覚を持つ自律移動ロボットが走行する際必要となるサインパターンの抽出を試みた。歩道の模様に応じて最も重要な領域にダイナミックにウィンドウを設定し、そのウィンドウ内でサインパターンを抽出する。この方法を幾何学模様の歩道に適用したところ80%以上の確率でサインパターン抽出に成功した。

SAIN PATTERN DETECTION OF GEOMETRIC PATTERN PAVED SIDEWALK

Kazuhiro Nisikawa Hideo Mori
Yamanashi University
4-3-1, Takeda, Koufu-si, Yamanashi, 400, Japan

Abstract In this paper we introduce a dynamic window setting method to detect a sign pattern paved sidewalk. Window are set up dynamically at the most important region of TV image to restrict the area in which the sign pattern is exist and to shorten sign pattern detection time. This method was implemented on autonomous mobile robot Harunobu-4. Experiment showed that sign pattern detection of method was successful more than 80% of scenes.

1. はじめに

人間の眼球運動の軌跡等を観察すれば、人間は提示されたパターンの部分を継続的に処理し、それらを統合して全体を認知しているかのように思える。逆に全体の印象をつかんでから細部を検討しているという考え方もありたつ[1]。筆者らは後者の考え方を視覚自律移動ロボットのサインパターン（以下SP）抽出にあてはめた。

対象となる画像のほぼ全体を簡単な処理によっておおまかに調べる。そして、その情報から画像内で一部分のSPの情報を持っている領域に狙いをつけてウインドウを開き、そのウインドウの画像データのみを処理する。この方法の利点は

(1) SP抽出ミスの減少

画像の一部分を処理するので、ウインドウ内の画像は比較的単純になり、他のパターンを誤認することが少なくなる。

(2) 画像処理時間の短縮

画像処理の対象になる面積が小さくなるので処理時間は短くなる。

などが考えられる。この処理方法を図1のような模様を持つ歩道にあてはめてみた。



図1 原画像

2. ウインドウの垂直方向の位置決め

画像の全体の構造をおおまかに認識するために以下に述べるような手法をとった。SP抽出の対象となる画像は図1のような模様を持っていることから、図2の様にY軸に平行な数本の直線上の画素のRGB値を微分することによって、茶色のタイルと白色のタイルがなしているエッジ点（図2中の○）を数点求め、その点の情報から画像の構造をおおまかにつかみ、Y軸方向のウイン

ドウの位置決めをおこなう。

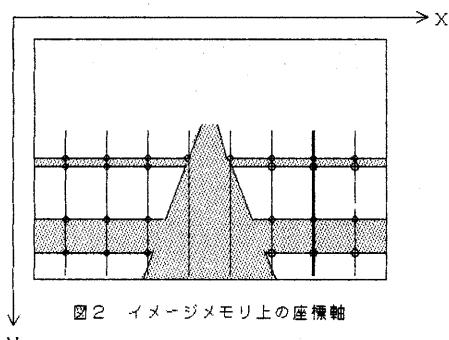


図2 イメージメモリ上の座標軸

2.1 フィルター

図2中の○の部分のエッジの点を求めるために、いくつかの微分フィルターを試してみたが、対象となる歩道がタイル張りであることなどの原因により抽出目的である適切なエッジの点の他に不必要的多数の点（ノイズ）がエッジとして抽出され、それらの情報からでは適切なウインドウの位置決めには至らなかった。

そこで、図2の直線上のRGB値の変化のデータを実際に取り、そのデータを解析することによって、微分フィルターを設計した。

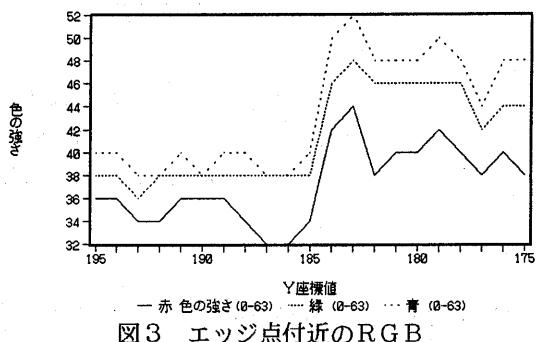


図3 エッジ点付近のRGB

図3のグラフ中央から左が茶色のタイルの領域、右が白色のタイルの領域で、中央の変化の激しいところがエッジである。RGB値はイメージメモリ上のY座標の値を2、3かけて茶色のタイル領域から白色のタイル領域へと変化しているのが解る。この変化に注目し、エッジの部分の微分値を強調するためには、その変化の過程の画素を除いた前後の部分で微分すれば良いと考え、フィルタ

ーの中心部2画素をゼロにする。さらに、ノイズを抑えるために前後3つずつの画素の微分を取り。微分フィルターは図4の様になり、微分値の計算は以下の式になります、エッジの点の識別には、絶対値(微分値)を用いた。

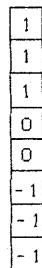


図4 微分フィルター

微分値の計算式

$$R(x,y) : (x,y) の赤成分の値$$

$$G(x,y) : (x,y) の緑成分の値$$

$$B(x,y) : (x,y) の青成分の値$$

$$DR(x,y) : (x,y) の赤成分の微分値$$

$$DG(x,y) : (x,y) の緑成分の微分値$$

$$DB(x,y) : (x,y) の青成分の微分値$$

$$DR(x,y) = R(x,y+3)+R(x,y+2)+R(x,y+1)-R(x,y-2) \\ -R(x,y-3)-R(x,y-4)$$

$$DG(x,y) = G(x,y+3)+G(x,y+2)+G(x,y+1)-G(x,y-2) \\ -G(x,y-3)-G(x,y-4)$$

$$DB(x,y) = B(x,y+3)+B(x,y+2)+B(x,y+1)-B(x,y-2) \\ -B(x,y-3)-B(x,y-4)$$

$$\text{微分値}(x,y) = DR(x,y)+DG(x,y)+DB(x,y)$$

$$ABS(\text{微分値}(x,y)) = \text{abs}(DR(x,y))+\text{abs}(DG(x,y))+\text{abs}(DB(x,y))$$

このフィルターを図2の太直線上の画素に作用させた結果を図5に示す。

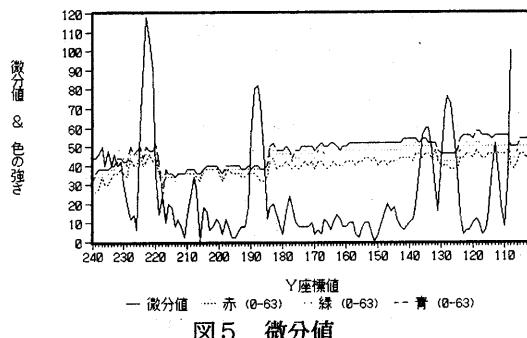


図5 微分値

図6のように適切なBIAS(しきい値)をとることによって茶色のタイルの領域と白色のタイルの領域のエッジ点が検出可能であることがわかる。また図6からY座標(240-237, 191-144, 132-110)の領域が白色のタイルの領域、Y座標(238-190, 145-131)が茶色のタイルの領域と解読できる。

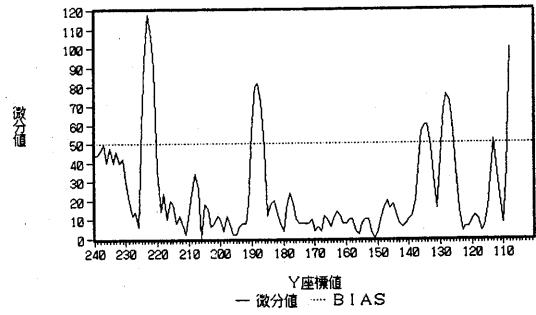


図6 微分値とBIAS

2.2 位置決めの過程

それらのエッジ点が抽出されることにより、ウインドウを画面中の有効な領域に配置することができる。具体的なアルゴリズムは以下である。

ウインドウの垂直方向の位置決めのために用いた情報は、抽出された茶色のタイルと白色のタイルのエッジ点の

(1) イメージメモリ上のXY座標

(2) エッジ点の微分値の正負

である。(2)の微分値の正負の情報によって図7のようにエッジの点の上方、下方のタイルの色が識別できる。

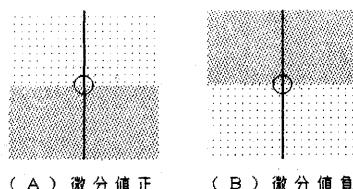


図7 微分値の正負とタイルの色

まず、求められたそれぞれのエッジ点をグループごとに分ける。ここで述べるグループの定義とは、白色のタイルと茶色のタイルの横方向の一つのエッジの直線上にある複数のエッジ点を要素とするグループである。図9に各エッジ点に識別のために番号を付けると図9の場合には {1, 2, 3, 4, 5} {6, 7, 8, 9, 10, 11, 12} {13, 14, 15, 16, 17, 18, 19, 20, } {21, 22, 23, 24, 25, 26, 27, 28} の四つのグループに分けられる。

次にグループの要素のエッジ点の微分値が正ならば、グループ内の要素の最大のY座標値を選択

し、そのグループのY座標値とし、微分値が負ならば、グループ内の要素の最小のY座標値をグループのY座標値とする。図8の場合にはそれぞれ、各グループからエッジ点 1, 12, 13, 28 が選択される。最終的にはこの選択された点の座標からウィンドウの画面に対して垂直方向の位置が決定される。この処理により歩道が画面に対して傾いているときにも適切な位置にウィンドウを開ける。

この一連の処理によって認識されるSP抽出に有効な領域は図9を例に取ると図10の網掛けの領域になる。

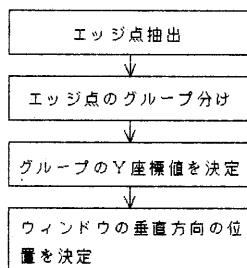


図8 位置決めの段階

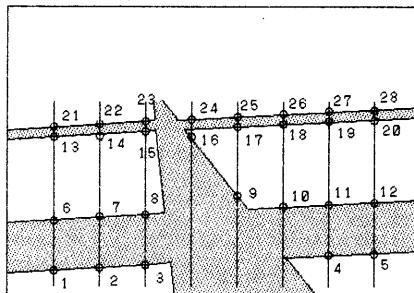


図9 エッジ点

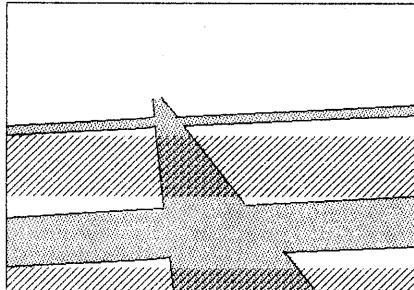


図10 認識された領域

3. ウィンドウの水平方向の位置決め

効率良くSPを抽出するために

- ・FINDING MODE : SP発見モード
- ・FOLLOWING MODE : SP追跡モード

の二つのモードを設定する[2]。

FINDING MODE、FOLLOWING MODE の二つのモードを設定する目的は限られる資源の中での、より効率的なSP抽出である。ロボットが屋外環境を認識して自律走行を行う場合、一回のSP抽出にかかる時間が少ないほど、一定時間内に多くの情報を得ることができ、正確な環境認識や、安定した走行が可能となる。FINDING MODE、FOLLOWING MODE のそれぞれの特徴を考慮したうえで、有効的に組合せることにより効率的なSP抽出を目指した。

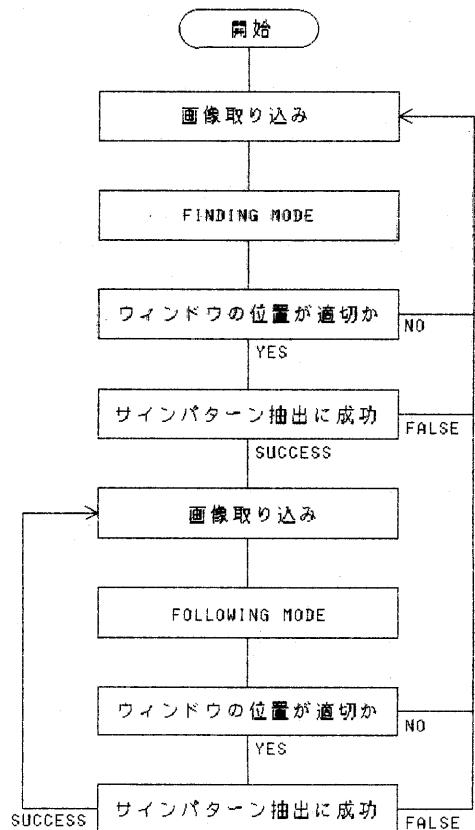


図11 流れ図

4. FINDING MODE

FINDING MODE とは、SPを一画像が持っている情報のみからウィンドウを設定するモードである。SPが画像中のどこに位置するかは予測できないので、ウィンドウは横方向に広く取り、逃さないようにする。ウィンドウの水平方向の位置と長さは固定とし、垂直方向の位置は2節で述べたアルゴリズムによって決定され、長さはウィンドウ1枚時、2枚時の場合にそれぞれの値が固定されている。

歩道の模様とウィンドウの関係を考えると図12のようにウィンドウの開き方は(A)、(B)、(C)、(D)の四つのパターンに絞れることが解る。

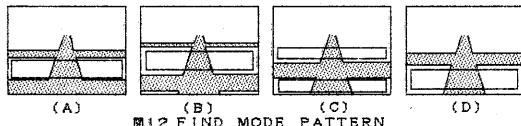


図12 FIND MODE PATTERN

5. FOLLOWING MODE

FOLLOWING MODE とは、FINDING MODE または前回の FOLLOWING MODE で画像処理して得られたSPの情報を使うことにより、SPの情報を持つている領域が次場面にくる位置を水平方向に予測することができる。この予測と2節で述べた垂直方向の位置決めにより、FOLLOWING MODE では FINDING MODE よりウィンドウの横方向の長さを短縮することが可能となる。この結果、画像処理を加える画像の面積が少なくなり処理時間が高速になる。

(図13参照)

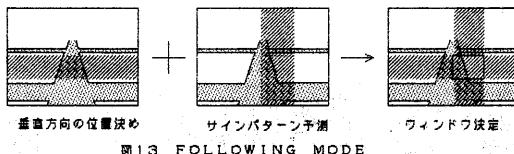


図13 FOLLOWING MODE

表1 色のデータ

		R/(R+G+B)	G/(R+G+B)	B/(R+G+B)
茶色のタイルの領域 SAMPLE 103	平均	0.2792	0.3043	0.4163
	標準偏差	0.0167	0.0090	0.0124
その他の領域 SAMPLE 206	平均	0.2514	0.3170	0.4315
	標準偏差	0.0176	0.0096	0.0115

歩道の模様とウィンドウの関係は図14に示したように FINDING MODE 同様に (A)、(B)、(C)、(D) の四つのパターンに絞られる。

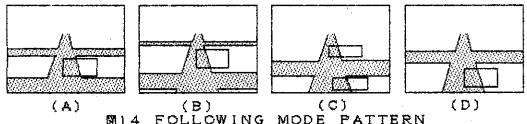


図14 FOLLOWING MODE PATTERN

6. ウィンドウ内の画像処理

FINDING, FOLLOWING MODE のウィンドウ内の画像は、SP抽出に有効な領域を狙ってウィンドウを開くので障害物等はウィンドウ内に入りにくく、基本的には茶色のタイルの領域と白色のタイルの領域がくる。SPを得るためにロボットと同じ進行方向のエッジが必要となるので、画面中の茶色のタイルの領域を選択し、その領域の持っている情報（縦方向のエッジ）から、SPを得るようにする。

領域分割後、ウィンドウ内の領域の中から、茶色のタイルの領域を抽出するKeyとしては領域の属性値の面積と色が考えられる。面積については、ある一定の面積以上のものを選び、色については以下のようなデータ解析をおこない表1の結果を得た。茶色のタイル領域とその他の領域でデータを比較すると $R/(R+G+B)$ に最も格差があることが解る。そこで $R/(R+G+B)$ に抽出条件を設定する。標準偏 = σ とすると、まず茶色のタイルの領域の平均値から σ の範囲内にある領域をピックアップする。複数あったならば、 $R+G+B$ の値が小さい方を目的の領域とし、もし該当するものが無いならば、さらに 2σ の範囲にある領域をピックアップし、該当する領域がなかったならばそのシーンでの領域選択を諦め、画像を再入力する。

7. 実験結果

使用機材：

MPU MC68020-VME133KIT(16.7MHz)

CAMERA SONY Handycam CCD-V30

IMAGE MEMORY MEC-1809, MEC-1732

512*512*8bit*3planes

画像処理対象の歩道のビデオ撮影は、影の問題を考慮して曇の日におこなった。

2節で述べたウインドウの位置決めにかかる時間は0.4(sec)。ウインドウの位置決めにかかる時間も含めたSP抽出にかかる FINDING MODE、FOLLOWING MODE の1ループの処理時間を表2に示す。SPの抽出成功確率は表3に示すように80%以上の成果をあげ、動的ウインドウの有効性が実証されている。画像処理の様子を図15に示す。図15.Aから図15.Cは連続的なシーンを処理した過程を示し、図15.GはSP抽出の失敗例である。図中の白色の領域が選択された茶色のタイルの領域である。

図15の補足

A : FINDING MODE で SP を抽出

B : FOLLOWING MODE で SP を抽出

C : FOLLOWING MODE で SP を抽出

D : ウィンドウの位置決めに失敗したので正しいSPを得られなかった

E : DでSP抽出に失敗したので FINDING MODE に戻り SP を抽出

F : FOLLOWING MODE で SP を抽出

G : FOLLOWING MODE で SP を抽出

H : 領域選択に失敗し、正しいSPを得られなかつた

表4にはSPの抽出に失敗したときの各エラーの割合を示す。表中のエラー項目は

位置決め：ウインドウの位置決めに失敗

画像処理：ウインドウ中の画像処理に失敗

領域選択：茶色のタイルの領域の選択に失敗である。

表2 処理時間

MODE	ONE WINDOW	TWO WINDOWS
FIND	1.4 (100*300)	1.7 (60*300)*2
FOLLOW	0.80 (100*42)	0.95 (60*42)*2

単位 秒 (画素数)

表3 サインパターン抽出成功確率

MODE	抽出成功確率
FIND	82%(SAMPLE 238)
FIND, FOLLOW	84%(SAMPLE 204)

表4 エラーの割合

ERROR			
位置決め	画像処理	領域選択	その他
68%	26%	3%	3%

図15 画像処理

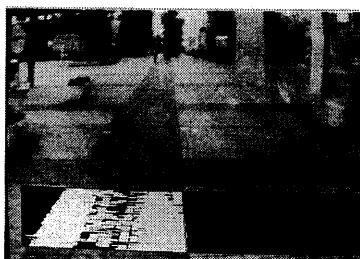


図15.A

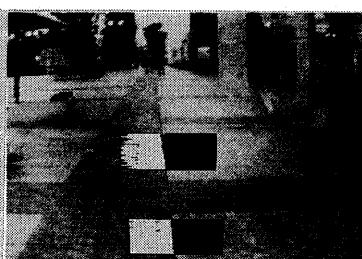


図15.B



図15.C



図15.D

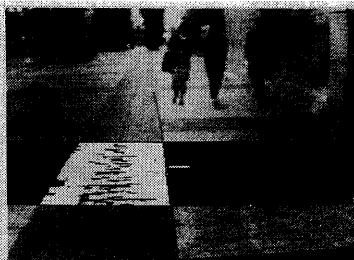


図15.E



図15.F



図15.G

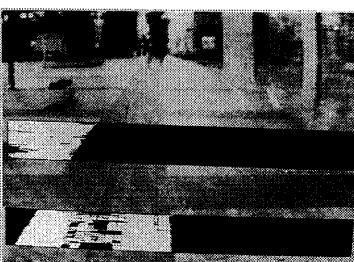


図15.H

8 おわりに

この手法は対象を、模様を持つ歩道に絞ってアルゴリズムを設定したが、ウィンドウをダイナミックに設定する戦略はアクティブ・センシングの一つとして自律移動ロボットにとって有効であると思われる。Graefeらはリアルタイムで処理する動的ウィンドウの開発に成功している[3]。今後の課題としては、晴信4号機を使用しての走行実験、それにともなった領域の選択条件のアルゴリズムの改善などがある。

参考文献

- [1] 乾、川入、森川：「Mental scanningについて」応用物理学会生理光学研究会 1976.9.1
- [2] 中井、森：「自律移動ロボットの沿目標移動について」第60回コンピュータビジョン研究会、1989
- [3] V.Graefe : Dynamic Vision Systems for Autonomous Mobile Robot, Proc. of Int'l Workshop on Intelligent Robots and Systems, 1989, pp.12-23