

自律移動ロボットに対するヒューマンインターフェースの開発

Development of Human Interface for an Autonomous Mobile Robot

佐藤忠大*, 及川一美*, 大久保重範*, 高橋達也*

Tadahiro Sato*, Kazumi Oikawa*, Sigenori Okubo*, Tatsuya Takahashi*

*山形大学

*Yamagata University

キーワード： 自律移動ロボット(Autonomous Mobile Robot), グラフマップ(Graph Map), 画像処理(Image Processing)

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
大久保重範, Tel.: (0238)26-3245, E-mail: sokubo@yz.yamagata-u.ac.jp

1. 緒言

グラフマップを用いた自律移動ロボットのナビゲーションにおいて、グラフマップを作成するために手書き地図や既存の地図を利用した手法がある。それらの地図からグラフマップを作成するためにさまざまな画像処理を繰り返す必要があるが、画像処理の多くは細かい設定や使う人間の主観が必要な場合がある。

本研究では誰でも容易に扱えるインターフェースを目指すため、細かい設定や操作を必要としないインターフェースの開発を目的とする。地図の画像の中からロボットのナビゲーションの作業環境である通路部分を抽出するために、ユーザーの指示が必要であったという問題に対して、本研究では輪郭線追跡という画像処理を用いることにより解決を図った。ここでは、グラフマップ作成までに使用する画像処理における問題及び、輪郭線追跡を用いた通路抽出法について述べる。

2. グラフマップ

本研究では実際に存在する既存の地図をロボットに与えてナビゲーションを実現させる。既存の地図を利用してナビゲーションを行うためには、既存の地図をグラフマップ(Fig.1)と呼ばれるグラフ構造の地図に変換する必要がある。

グラフマップはA, B, Cなどのノードオブジェクトと、それらをつなぐアークによって構成される。ノードを地図内の通路の端点や交差点、アークを通路とする。

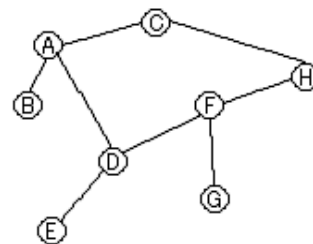


Fig.1 Graph Map

3. ラスタベクトル変換

グラフマップを作成するために既存の地図にラスタベクトル変換(Fig.2)という処理を施す。処理の流れは次のようになる[1]。

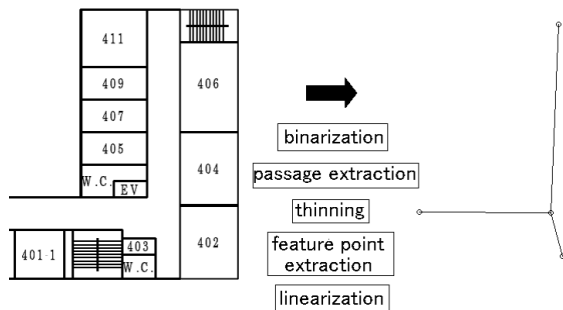


Fig.2 Raster to vector

このラスタベクトル変換のうち、特に問題点となる設定や操作を抱えている画像処理について述べる。

4. 画像処理

4.1 通路抽出

ナビゲーションを確実に行うためには、既存の地図からグラフマップに変換する際に、地図内の通路を抽出する必要がある。

従来の方法は、ユーザーが直接画像内からマウス等で通路部分を指示し抽出する必要があった。そのため、ユーザーの操作に頼らなければならずさらにマウス等の周辺機器が必要であるため改善が必要であった。

まず、手動による通路抽出の方法について述べる。

4.1.1 領域選択

地図の画像の中からユーザーが通路を選択し、その領域だけを抽出する(Fig.3)。ユーザーが直接選ぶため、確実に通路が抽出される。

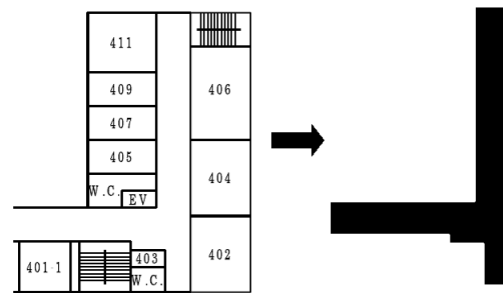


Fig.3 Manual passage extraction

次に、画像内から通路領域のみをプログラムで自動的に抽出する方法について述べる。

4.1.2 輪郭線追跡

地図の画像から通路を自動的に抽出する方法として輪郭線追跡を利用する[2]。地図の枠内の部分を対象とし輪郭線追跡を行い輪郭線を抜き出す(Fig.4)。

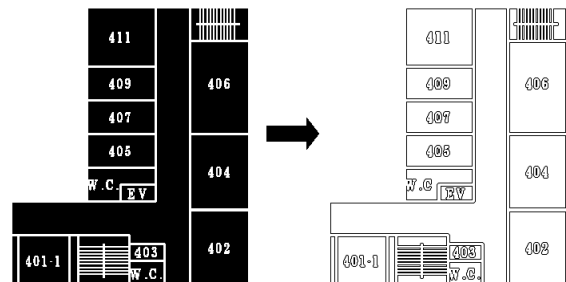


Fig.4 Border following

抽出された輪郭線の中から、最も輪郭線が長い部分を抜き出し、それを通路と判断する(Fig.5)。

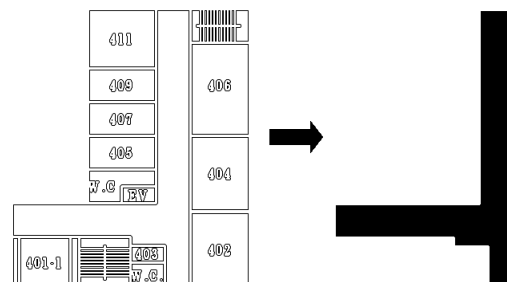


Fig.5 Automatic passage extraction

4.2 細線化

二値化された画像から線幅 1 の中心線を抽出する処理を細線化という。この画像処理により、抽出された通路の画像から線幅 1 の線図形を得ることができ、端点や交差点、角点を認識することが可能となる。

この処理にはひげ削除回数というパラメータがあり、設定された回数まで端点を削除し、余分な端点や交差点の発生を防ぐことが出来る。しかし、ひげ削除回数を誤った数値に設定してしまうと、端点や交差点が増減しナビゲーションに不備が生じてしまう。また、ある画像で成功した設定でも、大きさや形が変わるだけで失敗してしまうこともある。以下に、ひげ削除回数や対象を変更した場合の細線化処理の結果を示す(Fig.6, Fig.7, Fig.8, Fig.9).

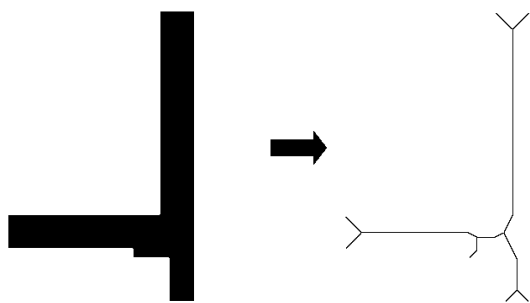


Fig.6 Deletion frequency:0 times

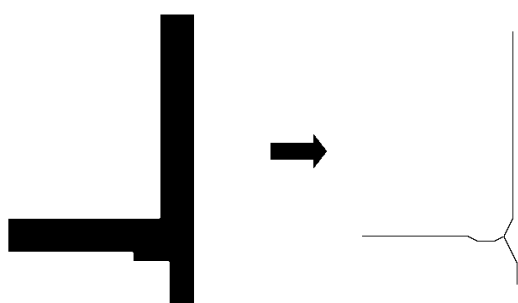


Fig.7 Deletion frequency:50 times

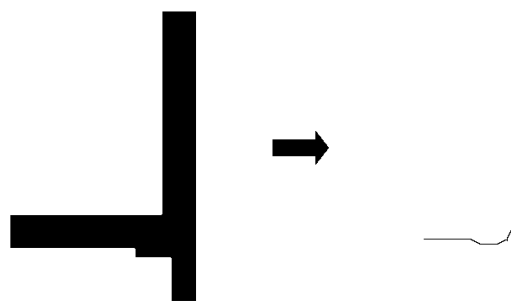


Fig.8 Deletion frequency:120 times

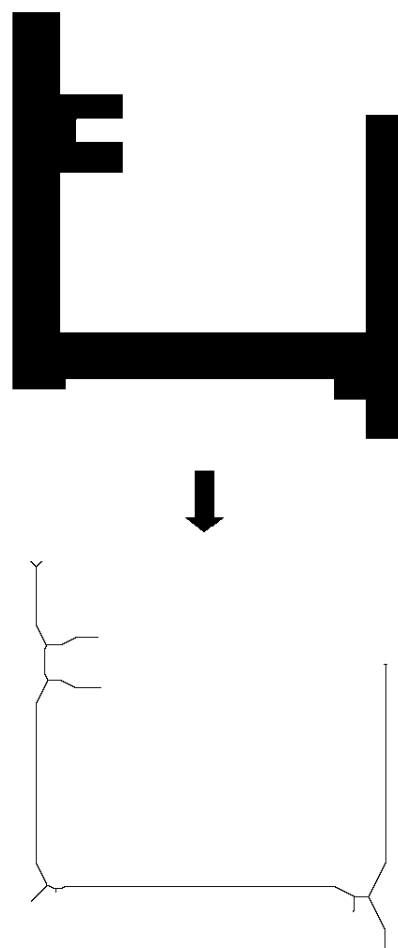


Fig.9 Deletion frequency:50 times(Change of size and shape)

Fig.7のようにひげ削除回数を50回と設定した場合、余分なひげは発生せず、細線化は成功した。しかし、ひげ削除回数を0回、120回とした場合、Fig.6、Fig.8のように余分な端点が出現したり、必要な端点が削除されてしまい細線化は失敗した。

Fig.7の成功を元に、同じようにひげ削除回数を50回としてFig.7で用いた画像より大きく形状の違う画像を細線化した。その結果、Fig.9のように端点が余分に発生してしまい、大きさや形状の変化によって最適なひげ削除回数は変化することがわかった。

5. 結言

輪郭線追跡を用いた通路抽出法は自動で通路領域を判断するため、ユーザーが手動で通路を選択する必要がなく容易に通路抽出を行うことができる。しかし、対象を輪郭線の長さのみで判別しているため、通路ではない複雑な部屋などを誤って抽出してしまう可能性もある。よって今後は、確実に通路を抽出できる判断基準を増やし検証する。

また、細線化においてひげ削除回数に誤った数値を設定してしまうと端点や交差点が増減してしまう可能性があるため、グラフマップに対して最適な細線化の方法の導入も必要である。

参考文献

- 1) 酒井幸市:「デジタル画像処理入門」CQ出版,(2002)
- 2) 画像に関するソフトウェアコンポーネントの作成と評価：<http://homepage2.nifty.com/tsugu/>