

情景画像中に存在する文字領域の抽出

原島 大輔 諸角 建

拓殖大学工学部

1. はじめに

インターネットやローカルマシン上には多数の画像が存在している。通常、これらを検索する際には文字検索を行うが、これは画像ファイル名や、画像周辺の文章、タグ情報などをキーワードとして検索する。そのため画像中の文字を検索したい場合には期待した結果を得ることはできない。画像中に文字が存在する場合は、この文字を認識し、利用することでより正確な検索が可能となる。文字の認識には一般に OCR ソフトウェアを用いることができる。OCR は文書画像を対象としていて、認識率が 99%以上と非常に高い。しかし、情景中に文字が存在するような画像の場合では、その文字をほとんど認識できない。そこで、本研究では情景画像中から文字領域を抽出し、OCR による認識が可能な画像を生成することを目的とする。画像中の文字を認識することで、検索以外には、文字情報を文書として保存したり、ロボットや視覚障害者向けなどの行き先案内の補助として、また、撮影場所の特定などに活用できる。

2. 情景画像について

情景画像とは風景を撮影して得た画像である。本研究で扱う情景画像は、文字情報を利用するため、画像中に文字が存在している情景画像である。情景中に存在する文字は看板もしくはポスターに書かれていることが多い。そこで、文字が書かれた看板やポスターが含まれている情景画像を本研究での処理対象とする。ただし、看板またはポスターが 1 つのみ存在する情景画像を扱うこととする。また、これらが途切れていたり、文字が読み取れないような画像は除外する。上記の条件に沿った情景画像は、デジタルカメラで撮影して取得、あるいはインターネットから収集を行う。

3. 文字領域について

上述の通り、情景画像中の文字は看板やポスターに書かれているものとする。このことから看板や、ポスターを文字領域と定義して処理を行う。

抽出には文字領域が矩形で構成されていることを利用するため、矩形ではない文字領域を含む情景画像は対象外とする。また、撮影条件については、真正面から撮影するなどの条件は現実的でない

Extraction of a character area in scene image
Daisuke Harashima, Tatsuru Morozumi
Faculty of Engineering, Takushoku University

い場合もあり、また、インターネット上の画像など既に撮影されている場合も条件の指定はできない。よって、撮影条件の指定はしない。この場合、文字領域が傾いていたり、歪んでいたりとすることが多く、この状態では OCR 認識に悪影響を及ぼす。そのため、文字領域を抽出した後に、この歪みを補正し、OCR で認識が可能な画像に加工する。

4. 文字領域の抽出

上記の条件や文字領域の特徴^[1]を利用して、情景画像中に存在する文字領域の抽出を行う。文字領域抽出処理の流れは図 1 に示すように行う。画像を読み込み、文字領域を抽出し、歪みを補正するまでの手順である。これらの処理について、図 2 の画像例を用いて順に説明する。

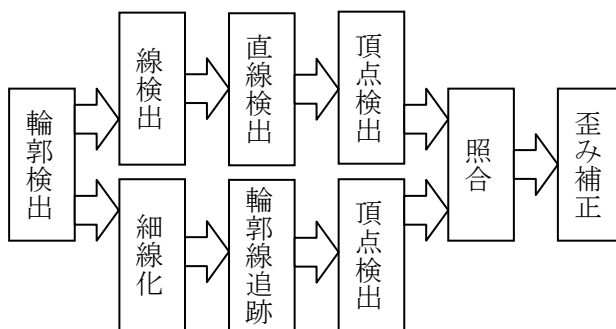


図 1 : 処理の流れ



図 2 : 原画像の例

4.1 輪郭検出

まず、文字領域の輪郭を検出するためにエッジ検出オペレータを用いたエッジ検出を行って、2 値化する。この結果に対して以下の処理を行う。

4.2 直線検出による頂点の抽出

4.2.1 線検出

直線検出を効率的に行うために、線らしい部分を検出する。これには 16 方向のマッチングを行う VanderBrug の線検出オペレータ^[2]を利用する。ここでは上下左右斜めの 4 方向でマッチングを行う。処理結果を図 3 (a) に示す。

4.2.2 直線検出

次に、文字領域の4辺を検出するために、ハフ変換を用いて直線の検出を行う。ハフ変換は画像中の直線を途中で途切れていても精度良く検出することができる方法^[2]である。このハフ変換を行って、直線を表示した結果を図3(b)に示す。

4.2.3 頂点検出

得られた直線の交点から文字領域の頂点の検出を行う。文字領域の4辺の特徴として、頂点が2つずつ存在すること、1辺の頂点から頂点までエッジが存在することを利用して、条件に当てはまらない直線を消去する。これらの処理により、残った直線の交点を検出された頂点とする。得られた頂点が4つでない場合は文字領域を抽出できないため、次の処理を行う。

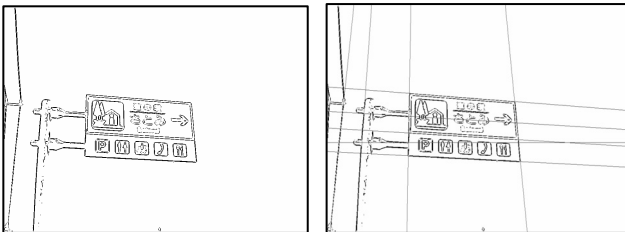


図3：直線検出結果

4.3 輪郭線追跡による頂点の抽出

4.3.1 細線化

輪郭検出結果に対して理想的な線の位置である骨格線を求める細線化^[2]を行う。細線化の結果、得られた画像を図4(a)に示す。

4.3.2 輪郭線追跡

次に、輪郭線追跡を行い、文字領域の4辺を検出する。輪郭線が閉じている場合を文字領域候補として、それ以外の画素を除外する。結果を図4(b)に示す。

4.3.3 頂点検出

得られた輪郭線から文字領域の頂点を検出する。検出には文字領域が矩形であることを利用して、頂点の検出を行う。

4.4 照合

直線検出、および輪郭線追跡によって得られた文字領域候補から照合を行い、文字領域を割り出す。それぞれの処理で得られた頂点検出結果に対してマッチングを行う。

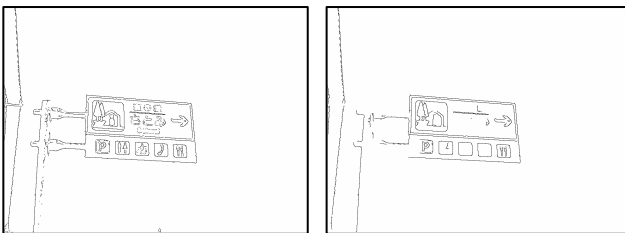


図4：輪郭線追跡結果

4.5 歪み補正

選択された4頂点を用いて原画像について文字領域を特定する。その文字領域に対して射影変換^[2]を適用することで歪み補正を行う。抽出した文字領域の歪みを補正した結果を図5に示す。

5. 文字領域の加工

抽出して歪みを補正した文字領域をOCRで文字の認識が可能な画像である文書画像に近くなるように加工する。抽出した文字領域に関して、閾値を自動的に設定できる判別分析2値化法^[2]を用いて2値化を行う。一般に文書画像は背景が白、文字が黒の場合が多いので、このような2値化を行う。この結果を図6に示す。



図5：歪み補正結果

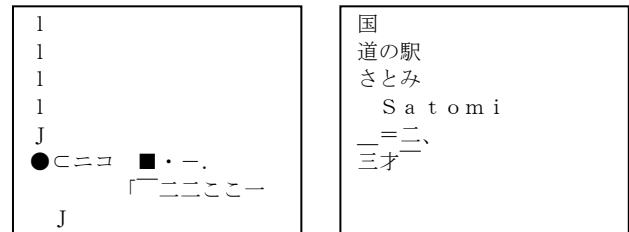


図6：2値化結果

6. まとめ

生成された文字領域に対して文字認識を行う。OCRは市販の汎用品を使用する。原画像および抽出画像を認識した結果を図7に示す。

図のように原画像の認識結果に対して抽出画像の認識結果は良好である。OCRは文字列領域を判別して認識を行うため、文字領域に文字以外の図などが含まれていると、文字列領域の認識に影響が出て文字の誤認識が起こる。そのため、文字領域を文字列毎に分割した場合は認識率が高くなる。しかし、一般にOCRは高い解像度を必要としているため、デジタルカメラで撮影するような解像度では、文字が潰れている場合や雑音があることも考えられ、認識ができないことも多い。認識率を高めるためには、分割の精度を良くして文字列ごとに分割したり、文字領域の補間などを行うことが必要と考える。



(a) 原画像

(b) 抽出画像

図7：認識結果

参考文献

- [1] 葛西浩一、高橋祐樹、中嶋正之、“情景画像からの文字領域抽出”、2000年電子情報通信学会総合大会、D-12-37、P207
- [2] 下田陽久、“画像処理標準テキストブック”、財団法人 画像情報教育振興協会