

人工知能

No.13 線画解釈

工学部 電子情報工学科 3年 03-140413 石見和也

2014年12月3日

1 ソースコードと実行方法

ソースコード（Xcode のプロジェクト）等一式は線画解釈.zip にまとめた。環境は Xcode で C++ をベースに、画像ライブラリとして OpenCV を用いる。今回書いたコードは blackboard.cpp のみで、この中に頂点認識及び辺のラベリングを実装した。実行方法は、Xcode でプロジェクトを開き、blackboard.cpp 内の 242 行目の `constchar * preset_file` に開きたい画像の絶対パスを代入し実行を押す。正しく画像を認識できた場合、ラベリングの結果を画像に重ねて表示する。

2 線画認識の原理

2.1 頂点の取得

以下の流れで画像処理を行うことにより、頂点を正しく取得することが出来た。

- canny 法によるエッジ抽出（図 5）
- cv::findContours 関数による特徴点の抽出（図 6）
- Ramer-Douglas-Peucker algorithm により数ある特徴点の中から頂点にあたる点のみ抽出（図 7）
- 近接する点を 1 つの頂点とみなし、繋がっている頂点の位置からその頂点が A,T,Y,L 型のいずれかを判断（図 8）

頂点のラベリングについては、画像内において A 型:黄色 T 型:緑 Y 型:紫 L 型:白として表示している。

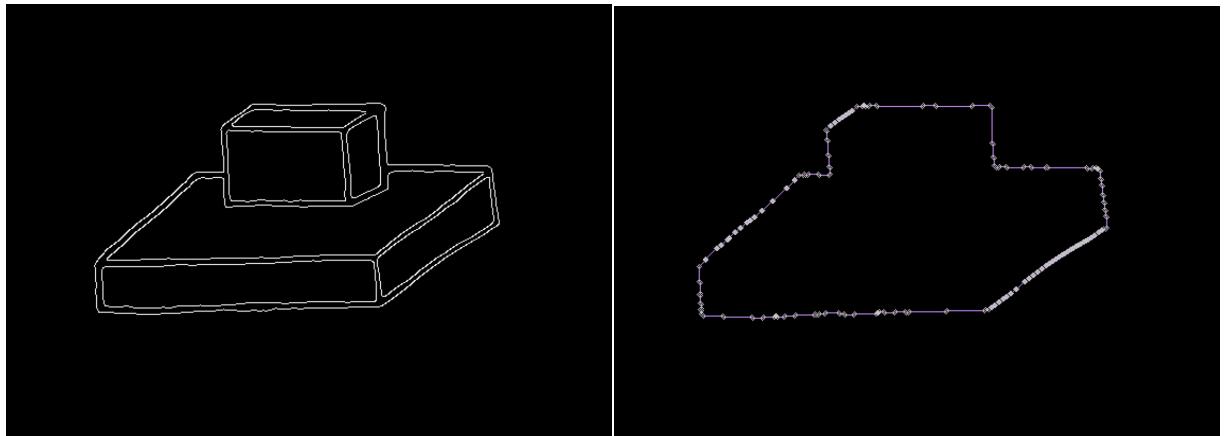


図 1 canny 方によるエッジ検出

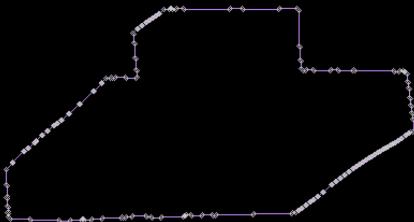


図 2 特徴点の抽出

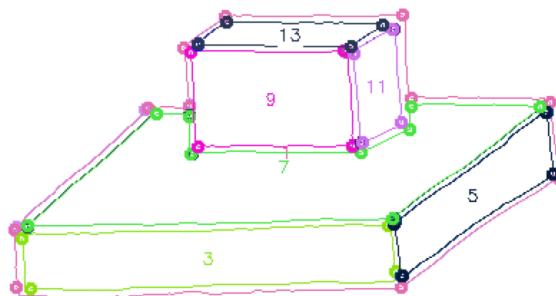


図 3 頂点にあたる点のみ抽出

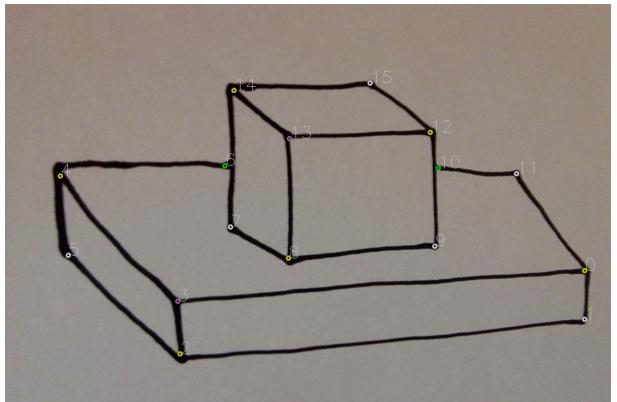


図 4 頂点のラベリング

2.2 辺のラベリングの手法

ラベリング手法については平賀先生の資料 [1] を参考させていただき、制約伝播法により実装した。まず頂点を A,T,Y,L 4 種類に分類する。そして、頂点の型ごとに、それに繋がる辺の取りうるラベルの辞書を登録しておく。各辺のラベル候補の内、両端の頂点の辞書項目にないラベルを削除する。そして、各頂点の辞書項目候補のうち、いずれかの辺にラベル候補がない項目を削除する。このように、辺と頂点に注目して取りうる候補を削っていき、最終的にどの頂点、辺からも削除するラベルがなくなれば探索終了となる。

3 結果

以下に、プログラムの出力結果を載せる。ラベリング結果は辺上に「+」「-」「」の記号で描いている。複数の選択肢が残った場合、考えられる選択肢すべてを描くこととした。

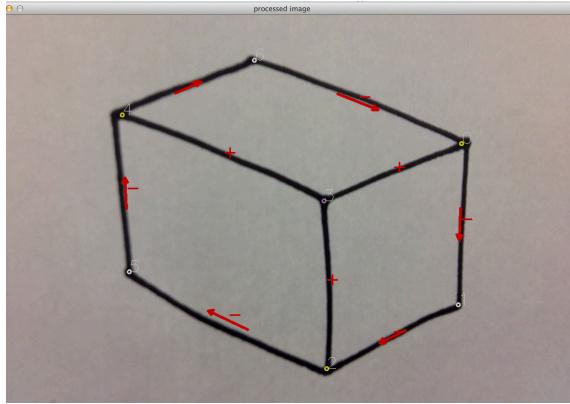


図 5 直方体 (外枠のラベリング無し)

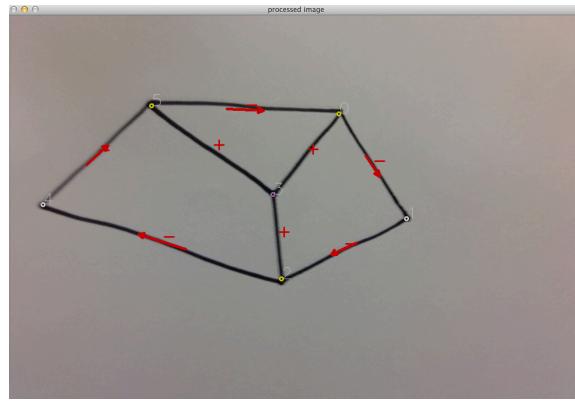


図 6 不可能物体 (外枠のラベリング無し)

図形の外枠はどこにも接していないとし、予め時計回りに矢印のラベルを付加しておくと以下の様な結果となつた。

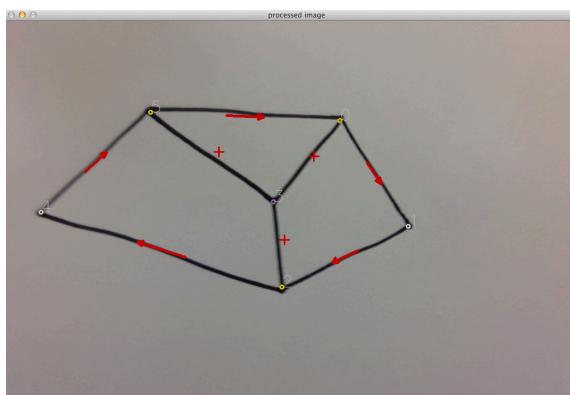


図 7 直方体 (外枠のラベリング有り)

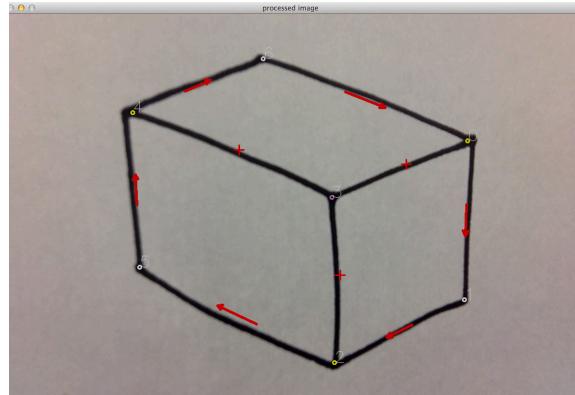


図 8 不可能物体 (外枠のラベリング有り)

複雑な図形についても画像が暗くならないよう気をつけて撮影した画像だと上手く認識でき、以下の様な結果となつた。

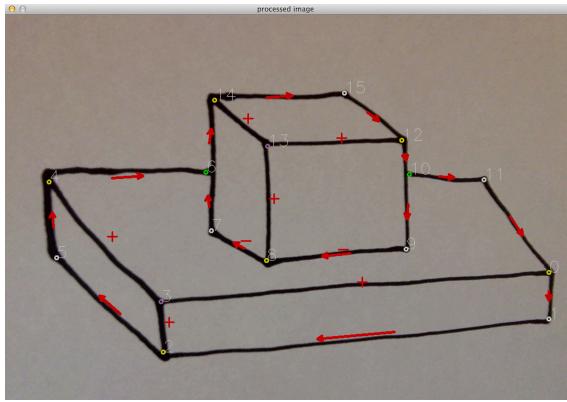


図 9 複雑な立体 1(外枠のラベリング有り)

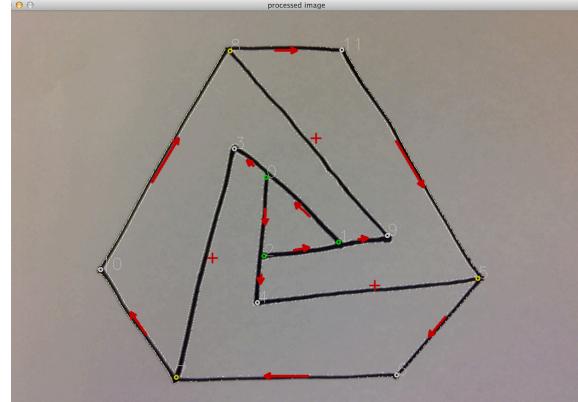


図 10 複雑な立体 2 (外枠のラベリング有り)

以下のように 2 辺が重なって表示されているネッカーキューブのような立体はプログラムの仕様上正しく認識できずエラーとなった。

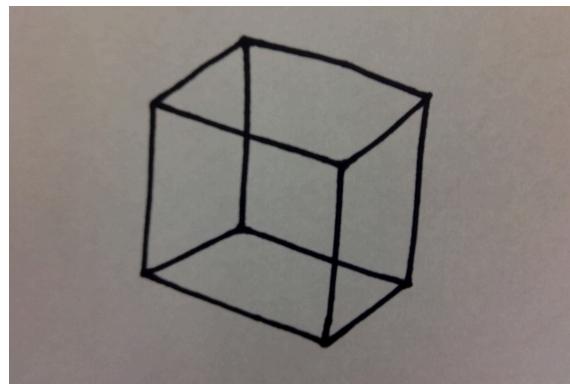


図 11 ネッカーキューブ

4 考察

4.1 画像認識についての考察

辺上の特徴点のみ抽出するのには苦労したが、エッジ抽出の段階で Blur や二値化処理を取り入れることで、線の太さの歪みやマジックのインキのシミに左右されず正しく閉じた領域の枠に沿った特徴点を抽出出来た。また、Ramer-Douglas-Peucker algorithm のパラメータを調整することで、線の僅かなブレを無視し、線が曲がっている部分のみの特徴点を選び出すことに成功した。

頂点の型の特定には初めはそれぞれの頂点型に固有の方法でなんとか分類しようと試みたが、取得漏れがどうしても生じてしまったので、最終的には外積を用いて頂点に繋がる辺を資料 [1] に載っている順に並び替え、そして内積から各辺の角度を求め判断することにした。それにより複雑な图形であっても定義された 4 つの頂点からなる图形であればすべて正しく読み込むことが出来た。

4.2 ラベリングについての考察

例えば図10は不可能物体であるが正しく一意のラベリングが行われている。線画で描かれた物体が現実世界ではどのような形をしているのかを調べるために行っているラベリングであるので、このような物体に正しくラベリングが行えてしまってはいけないため、これはラベリングミスということになる。これは、今回用いたラベル辞書が、頂点の型に対して、それに繋がる辺がどのようなラベルを持ちうるかという、2頂点間の関係のみしか注目していないためと思われる。間違いのない方法は、全部の頂点や辺を一塊とみてラベリングする方法であるが、それは登録されている物体の認識を行っているに過ぎず、探索を用いて未知の線画に対してラベリングをするという本来の目的が失われてしまっている。ただし、この問題についてよい解決案は浮かばなかった。

ネッカーキューブのラベリングについては、このまま扱うと現在のラベリング規則に合わない2辺が重なった点が出来てしまうため、何かしらの画像処理を行いA,T,Y,L型の頂点のみにすれば認識できうるのではないかと考えた。2辺が重なっているということは、本来見えるはずのない裏側の線、頂点まで描かれているということである。例えば図11のネッカーキューブの場合、外枠の内部に位置する頂点の内のどちらかと、それに繋がる辺をないものとして扱えば、直方体として正しくラベリング出来る。この方法は複雑な立体にはそのまま適応出来ないが、適切なアルゴリズムを用いて隠れている可能性のある点と辺を消してラベリングを行うことで、今回使用したのと同じラベリング規則で線画のラベリングが可能であると考えられる。

5 参考文献

- [1] 平賀先生『制約伝播:線画のラベリング』資料 <http://www.rsch.tuis.ac.jp/mizutani/online/latex/biblio.html>
- [2] 川瀬 喜理 漢字の形に見える不可能物体の制作 <http://www.math.ryukoku.ac.jp/tsutomo/graduate/2007/07kawase.pdf>