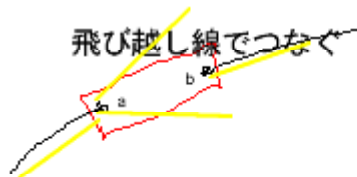


図面解析で、もう一つ機能を作り込む必要を感じました。オンデマンドでなくて、自動的に解析するプロセスです。それは、飛び越し線（かすれ線）の検出です。分岐点を越えて、直線として判断したり、ギャップを越えて2つの線分を結ぶ線を想定する処理ですね、これ重要なのです。



2つの線分の接線方向が同じで、端点が迎角5度くらいに収まるものどうしを結ぶということで良いのでは無いでしょうか。厳密には心理学で実験してみないと分からないことですが。ギャップの線の長さも最大はシンボルの大きさくらいでないかと。こうすると簡単に飛び越し線がもとまるのです。キャップ長の向かいの端点を探し、・・・迎角の2本の直線の傾きから、端点が右側（左の迎角直線からみて）でかつ左側（右の迎角直線からみて）にあるものを探す。そうして、向かい合う線分の接線が迎角くらいのもをもって、飛び越し線を引く・・・でよろしいのではないのでしょうか。

そこで、図面認識を考えてみました。どんな知識が必要でしょうか。線画解析から得たデータからパターンマッチングして、オブジェクト表現を得る過程、・・・そこから数式群を設定する過程です。

基本的に知識はつぎのものになります。

(1) 解析知識パターン

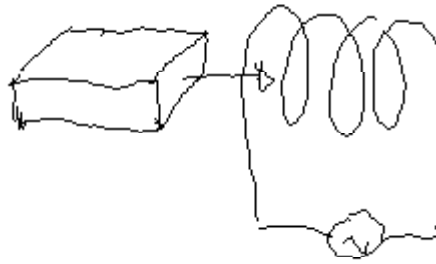
線とか領域、閉じ線とかの配置を解析し、どんな図面かを把握する為のコマンド群です。

(2) 解析結果の意味を捉えるパターン

領域とか閉じ線とか直線とかの意味を解析するためのコマンド群です。抵抗とかコンデンサとかのシンボルを得ること、容器の状況を得ることなどを行うものです。

(3) オブジェクト構成、数式連想を行うパターン

シンボル間の関係を得ること、3次元的な推論をおこなうこと・・・そうしてそれに関する数式を連想で得ることを行うコマンド群です。



( 1 ) 磁石の認識パターン

```
(stay,agent_case,square1,up_case,square2)^(stay,agent_case,square1,up_case,square3);  
(stay,agent_case,square2,left,square3);  
-->(associate,object_case,magnet_stick);
```

( 2 ) コイルの認識パターン

```
(repeat,agent_case,C);  
c:-(stay,agent_case,top_curve,left_case,branch)^(stay,agent_case,branch,up_case,close_line);  
-->(associate,object_case,coil);
```

( 3 ) 磁石とコイルの配置から数式を連想するパターン

```
( stay , agent _ case , magnet _ stick , left _ case , coil ) ^  
(move,agent_case,magnet_stick,toward_case,coil)  
-->(associate,formura)  
formura:(=,agent_case,a,and,b);  
a:-(rot,object_case,E)  
b:-(partial_differenciation,by_case,time,object_case,B);
```

おわり