

図面解析のコードレビュー、図面認識と図面上の推論のアルゴリズム設計をしています。プログラム開発と同じで、段階的にきっちり問題解決がいかないものです。手戻りがある。図面認識を考えていって、この辺の解析は、もう一度、別の視点で解析していった方がよいという問題がありました。人間でも、風景を見ていて、ある解釈に納得がいかないと曖昧性のもととなる視覚状況を改変して再解釈して、納得いくようにします。このように、知識の枠組みから解析の下となる視覚データを改変して・・・解析方法を変えて、再図面認識すること、オンデマンド部分再解析は人工知能にとって、必須の能力になるようです。

そんな、オンデマンド再解析が必要な例は「ピストン」図面にありました。

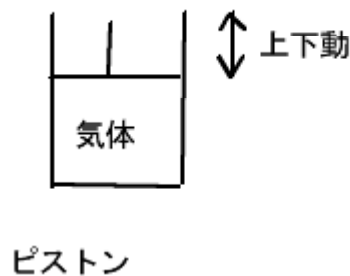


図1 ピストンの図面

今の図面解析では長い線を抽出しています。これだと、枝分かれを越えて直線（シリンダー面）が強調されません。

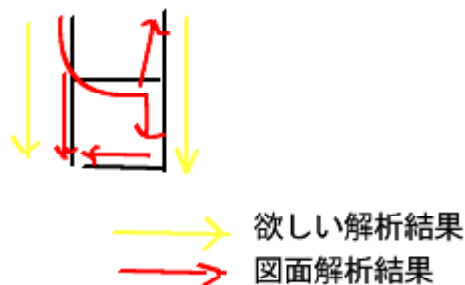


図2 現在の図面解析結果と認識から本来求めたい結果の齟齬

最初の図面解釈・・・もしくは注釈の「ピストン」から連想をもって知識を得てその知識に本当にマッチングしているかを再度解釈し直すのです。

ピストンの知識はこんなものでしょうか。

(stay,agent_case,line1,left_case,line2)^(be,agent_case,line1,attribute,vertical)
^(be,agent_case,line2,attribute,vertical)

(stay,agent_case,line3,between_case,line1,and,line2)^(be,agent_case,line3,attribute,horizontal)

(stay,agent_case,line3,attach_case1,line1)^(be,agent_case,attach_case1,attribute,not_bottom)
^(be,agent_case,attach_case1,same,attach)

(stay,agent_case,line3,attach_case1,line2)^(be,agent_case,attach_case1,attribute,not_bottom)
^(be,agent_case,attach_case1,same,attach)

ピストンの線画の知識は、こんなものですが、こうゆうものが無数にあるわけです。全部の知識とマッチングさせるのは難しいでしょう。インデックス体系が必要です。部分的なコマンドをもって、インデックスに登録して、段々にインデックスは詳しくなっていく。部分的に、間違いやすい雑音もいれて、インデックスは正しい解釈知識に導いていくようにもします。

ピストンの例では、2つの平行線の中に水平線分が入るという知識ですから、そのようにして、図面の線分の交差関係を再解釈していくのです。オンデマンドな再解析が必要な良い例になっています。

おわり