

描き順に依存しない手描きクラス図認識のための 近傍オブジェクトを用いたストローク接続法

A Stroke Connection Method Using Neighboring Objects for Stroke Order Free Class Diagram Freehand Drawing Recognition

佐藤 信*

Makoto Satoh

概要

ストロークの描き順に依存せずに UML クラス図を認識するために、近傍オブジェクトを用いたストロークセグメントの接続手法を提案する。この方式では、ある与えられたストロークの近傍を表現する近傍オブジェクトを作成し、この近傍オブジェクトと他のストロークとの接続を判定することにより、ストロークの端点または中間点を接続する。交差するストロークまたは近接するストロークを接続することが可能であり、自由な描き順とストローク数により UML クラス図要素を作成することが可能である。この方式は、クラス図の完成版を作成する場合だけでなく、ソフトウェアを改良する過程でのクラス図作成にも適した方式である。

1 はじめに

本稿では、近傍オブジェクトを用いたストロークセグメントの接続手法を提案する。ソフトウェアの開発過程において、設計を繰り返す段階での思考を支援するための CASE ツールとして UML ツールを用いる場合には、クラス図の入力が容易であることが重要である。提案手法は、自由なストローク入力からクラス図要素の認識を可能とするためのものであり、ストロークの自由度を紙に手描きをする場合により近づけることが目的である。

2 ストローク接続アルゴリズム

2.1 関連研究

ストローク数と描き順が自由な入力からクラス図要素を認識するための研究としては、Hammond 等 [1] があり、各セグメントの端点間の距離を用いて近傍での接続

を判定している。本稿の提案手法では、ストローク数と描き順が自由であり、ストロークの端点または中間の近傍での接続が可能である。

2.2 近傍オブジェクトによるストローク接続

2本のストロークが与えられた場合に、ストロークの接続を判定するためのアルゴリズムを示す。

Step 1 Douglas 等 [2] の手法を用いて、与えられた 2本のストロークを構成する点列から、それぞれ、点列 $p_{1_i} (i = 1, 2, \dots, n)$ と $p_{2_j} (j = 1, 2, \dots, m)$ をサンプリングする。これらのサンプリングした点列は、基の点列の形状特徴を表現する点列である。

Step 2 点列 p_{1_i} と p_{2_j} から、それぞれ、 $l_{1_k} (k = 1, 2, \dots, n-1)$ と $l_{2_l} (l = 1, 2, \dots, m-1)$ を要素とするストローク線分リスト l_1 と l_2 を作成する。ここで、 l_{1_k} は、 p_{1_k} を始点とし $p_{1_{k+1}}$ を終点とする線分である。

Step 3 ストローク線分リスト l_1 を基にして、そのストロークの近傍オブジェクト n_1 を作成する。近傍オブジェクト n_1 は、閉じた折れ線である。

Step 4 近傍オブジェクト n_1 の各線分と、ストローク線分リスト l_2 の各線分 $l_{2_l} (l = 1, 2, \dots, m-1)$ の交差を判定する。交差している線分があれば、与えられた 2本のストロークは接続していると判定する。

この接続判定アルゴリズムの Step3 で用いる近傍オブジェクトを作成するためのアルゴリズムを示す。

Step 1 ストローク線分リスト l_1 の各線分を基にして、オフセット線分リスト $b_{1_k} (k = 1, 2, \dots, n-1)$ を作成する。 b_{1_k} の各要素 b_{1_k} は、それぞれ、 l_{1_k} の線分の両端を長さ d だけ延長した線分を、 l_{1_k} と直交する方向に距離 d だけ移動したものである。このときの移動

*Iwate University

方向は、 l_k の始点と終点をベクトルの始点と終点とした場合に、そのベクトルを反時計回りに 0.5π ラジアン回転した方向である。ここで、 d は予め与えられた値である。

Step 2 点列 p_{1_i} から、 $r_{1_k} (k = 1, 2, \dots, n-1)$ を要素とする反転線分リスト r_1 を作成する。ここで、 r_{1_k} は、 $p_{1_{n-(k-1)}}$ を始点とし $p_{1_{n-k}}$ を終点とする線分である。反転線分リスト r_1 の各要素の両端を長さ d だけ延長した線分を l_k と直交する方向に距離 d だけ移動する。このときの移動方向は、 r_{1_k} の始点と終点をベクトルの始点と終点とした場合に、そのベクトルを反時計回りに 0.5π ラジアン回転した方向である。この移動した線分を、オフセット線分リスト b_1 に追加する。ここで、オフセット線分リストの要素数は $(n-1)*2$ となり、その添え字は $k = 1, 2, \dots, (n-1)*2$ となる。

Step 3 要素順にオフセット線分の交差判定をおこなう。はじめに、オフセット線分リスト b_1 の隣り合う要素 b_{1_k} と $b_{1_{k+1}}$ について交差判定をおこない、その後、オフセット線分リスト b_1 の最後の要素と最初の要素について交差判定をおこなう。線分が交差している場合には、その交点を近傍オブジェクト節点リスト a_1 に追加する。線分が交差せず、線分 b_{1_k} を基準とした $b_{1_{k+1}}$ の回転角 $-\pi \leq \theta \leq \pi$ が $\theta \leq \epsilon$ の場合には、 b_{1_k} の終点と $b_{1_{k+1}}$ の始点を近傍オブジェクト節点リスト a_1 に追加する。線分が交差せず、 $\theta > \epsilon$ の場合には、近傍オブジェクト節点リスト a_1 に何も追加しない。ここで、 ϵ は予め与えられた値である。

Step 4 近傍オブジェクト節点リスト a_1 の隣り合う節点を結ぶ線分、そして、近傍オブジェクト節点リスト a_1 の最後の節点と最初の節点を結ぶ線分を要素として、近傍オブジェクト n_1 を作成する。

3 実装と結果の検討

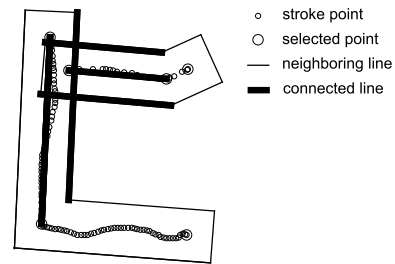
提案手法を、Java 言語を使用して実装した。実行結果の図は、実装したプログラムで SVG 形式ファイルを直接出力し、それを EPS 形式に変換したものである。

Example 3.1.(図 1 参照)

この例では、与えられた 2 本のストロークを接続する場合について、アルゴリズムの各段階を説明している。図 1(a) は、2.2 節で説明したストロークの接続判定アルゴリズムの **Step 1** での特徴点選択の結果である。ここで、小円と大円は、それぞれ、ストロークを構成する点列と選択した特徴点である。図 1(b) は、**Step 3** で作成した近傍オブジェクトを表現する線分(細線)と **Step 4** で接続



(a) stroke points and selected points.



(b) neighboring lines and connections.

図 1: An example of stroke connection.

を検出した線分(太線)である。L字型のストロークを表現する線分のうちの 1 本の線分と他方のストロークの近傍オブジェクトのうちの 2 本の線分の交差と、L字型のストロークの近傍オブジェクトのうちの 1 本の線分と他方のストロークを表現する線分のうちの 1 本の線分の交差を検出している。ストロークが近接している場合に、ストロークの接続を検出できていることが分かる。

4 おわりに

近傍オブジェクトを用いたストロークセグメントの接続手法を提案した。今後の課題としては、接続したストロークからのクラス図の認識に関する研究がある。

参考文献

- [1] Hammond, T. and Davis, R.: Tahuti: a geometrical sketch recognition system for UML class diagrams, in *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA (2006), ACM.
- [2] DOUGLAS, D. H. and PEUCKER, T. K.: ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE, *Cartographica: The International Journal for Geographic Information and Geovisualization*, Vol. 10, No. 2, pp. 112–122 (1973).