J-025

ベクター変換における曲線最適化アルゴリズムの一検討
A Study on Curve Optimization Algorithm in Vectorization

河村 圭 †　　**渡辺 裕** †
Kei Kawamura　　Hiroshi Watanabe

1. はじめに

e-インクや電子ペーパーなど，新しい表示ディバイスの登場やペーパレスの観点から，文書をディスプレイ上で閲覧する機会が増大すると考えられる．また，効率的な配信，蓄積，閲覧等を考慮した場合，ベクター形式の利用が挙げられる [1]．しかし，現在利用可能な文書の多くはラスター形式であるため，Potrace[2] をはじめとするベクトル化ツールの適用が必須であり，その性質が大きな影響を及ぼす．Potrace によるベクター形式では，細部の調整やコーナーとカーブの切り替えなどの編集に適さず，操作性が低いという問題がある．これは，ベジエ曲線の通過点（アンカーポイント [3]）の取り方に原因がある．本稿では，操作性の高い通過点の取り方について検討し，これに適した曲線最適化アルゴリズムを提案する．提案手法により，操作性の高いベクター形式が得られる．

2. Potrace

Potrace の処理手順について述べる．まず，入力された 2 値画像を輪郭パスに分解し，それぞれのパスごとに輪郭データを多角形で近似する．その際，それぞれの辺は 1/2 画素以下の誤差となるようにする．次に，辺の数が少ないほど，同じ辺数の場合は累積誤差が小さいほど良い近似であるとして最適多角形を得る．そして，多角形の頂点ごとにコーナーかカーブかを判断し，カーブである場合には 1 本のベジエ曲線で置き換え，辺の中点にベジエ曲線の通過点（接続点）を置く．最後に曲線最適化を行う．最適化とは，連続する複数の曲線を 1 つの曲線に統合する処理である．この処理により，近似精度を変えることなく通過点を減らすことが可能となる．

曲線を統合する方針は以下の通りである．まず，連続する曲線のみを統合し，直線は含めない．次に，凸方向が一致する曲線のみを統合する．そして，統合する曲線群の方向変化は 180 度未満とする．図 1 に具体例を示す．a_i は多角形の頂点を示し，b_i は辺の中点を示す．b_0 から b_n まで統合し，点 O を新しい頂点と見なした 1 本のベジエ曲線を得る．太線は統合前のベジエ曲線，細線は統合後のベジエ曲線を示す．

3. ベクター形式の操作性

ベジエ曲線によるベクター形式の操作性について述べる．ここでは，オーサリングツールとしての使い勝手ではなく，ベジエ曲線の本質的な操作性を考える．

多角形を曲線で置き換えるには，2 つの手法が考えられる．頂点を 1 本のベジエ曲線で置き換え，辺の中点で 2 つのベジエ曲線を滑らかに接続する手法（Potrace，図 2 左）と，辺を 1 本のベジエ曲線で置き換え，頂点付近で 2 本のベジエ曲線を滑らかに接続する手法（図 2 右）である．黒丸は通過点（接続点）を示し，白丸は制御点（コントロールポイント [3]）を示す．ただし，図 2 右では，端点において通過点と制御点が重なっている．

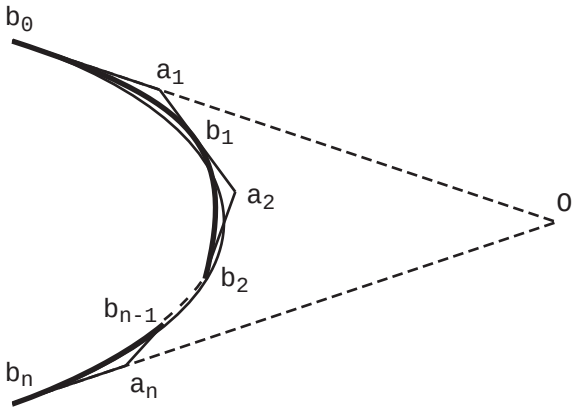

図 1: A curve optimization in the Potrace

図 2: Two drawing methods of curves

一般にユーザーが曲線を描くときは後者の手法をとる．この手法で描くと以下に挙げるような操作性が向上する．まず，滑らかな接続を維持するという条件により，2 本のベジエ曲線の制御点を同時に移動させることで曲線の膨らみを容易に変更できる．次に，制御点を通過点に重ねるかどうかによって，頂点のコーナーとカーブの切り替えが容易に行える．これらの操作性は，全自動ベクトル化において一様に決定されてしまう結果を，ユーザーの意図により人手で修正する要求に応えるために重要である．

前者の手法で描くと，曲線の膨らみを変更するには滑らかな接続を破棄して制御点を移動する手法，滑らかな接続を維持するために隣り合う曲線にも影響を及ぼして制御点を移動する手法，曲線上の頂点部分に新たに通過点を追加する手法のいずれかを取らざるを得ない．いずれの手法も最初から頂点に通過点がある後者の手法に比べて操作性が悪い．

4. 曲線最適化アルゴリズム

4.1 ベクトル化

Potrace によって得られた最適多角形の辺を，図 3 のように 1 本のベジエ曲線で置き換え，頂点付近で滑らか

†早稲田大学大学院 国際情報通信研究科，
Graduate School of GITS, Waseda University.

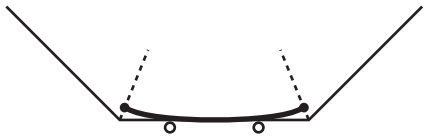

図 3: A novel side replace method

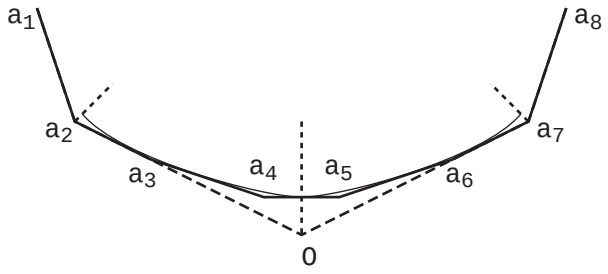

図 4: A novel curve optimization 1

に接続する手法について検討する．

まず，頂点がコーナーである場合を考える．通過点は頂点に置き，制御点も通過点に重ねる．

次に，頂点がカーブである場合を考える．通過点は頂点の 2 等分線上で，頂点からの誤差が 1/2 画素以下の所に置く．制御点は，頂点を含む三角形の底辺に平行で，接続点を通る直線上の適当なところに置く．この条件により，2 つのベジエ曲線は滑らかに接続する．

最後に，ベジエ曲線の定義式と，ベジエ曲線が媒介変数 $t = 1/2$ で辺に接する，または通過する条件式を示す．ただし，$y_2, y_3 = 0$ として座標系を単純化している．

$$
\begin{aligned}
x(t) &= x_1(1-t)^3 + x_2(1-t)^2 t + x_3(1-t)t^2 + x_4 t^3 \\
y(t) &= y_1(1-t)^3 + y_2(1-t)^2 t + y_3(1-t)t^2 + y_4 t^3 \\
y(1/2) &= 0 \\
\frac{\mathrm{d}y(t)}{\mathrm{d}x(t)} &= \frac{\mathrm{d}y(t)}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}x(t)} = 0 \quad (t = \frac{1}{2})
\end{aligned}
$$

それぞれの辺に対して両端の条件を設定することで，曲線の制御点の位置は一意に決定される．そして，操作性の高いベジエ曲線で置き換えることが出来る．

4.2 統合条件

4.1 で提案したベクトル化における曲線最適化手法について検討する．最適化により統合した曲線にも 4.1 のベクトル化と同じアナロジーが使えるようにする．具体的には図 4 に示すように，a_2 から a_7 までの辺が統合され，辺 (a_2, O)，辺 (O, a_7) の 2 本になる．そして，それぞれの辺をベジエ曲線で置き換える．

曲線の統合条件は次のようになる．まず，連続する曲線のみを統合し，直線は含めない．また，端点以外に角を含めない．次に，開始点と終了点の頂点における通過点の位置を，統合前後で保存する．また，変曲点を含む辺も保存する．そして，凸方向が一致する曲線群の方向変化は 180 度未満にする．最後に，元の曲線との誤差を計算する．

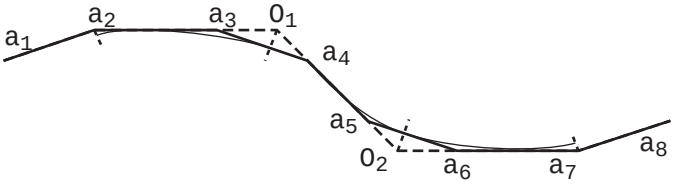

図 5: A novel curve optimization 2

4.3 通過点と変曲点

曲線を統合することによって，隣接する曲線への影響は少ない方が望ましい．しかし，曲線を一意に決定する際，通過点か制御点のいずれかに影響が出る．そこで，再計算が容易な制御点のみに影響が出るようにし，開始点と終了点の頂点の角度を統合前後で保存することで 2 つの通過点の位置を保存する．よって，統合後の頂点 O は，辺 (a_2, a_3) と辺 (a_6, a_7) を延長した交点とする．さらに，図 4 のように交点が生じるためには，曲線の方向変化は 180 度未満である必要がある．

また，図 5 のように変曲点を含む場合は，変曲点を含む辺を保存する．例えば，a_2 から a_7 までの辺を統合すると，辺 (a_2, O_1)，辺 (O_1, O_2)，辺 (O_2, a_7) の 3 本になる．そして，それぞれの辺をベジエ曲線で置き換える．通過点の位置と変曲点を含む辺を保存するために，頂点 O_1 は辺 (a_2, a_3) と辺 (a_4, a_5) を延長した交点，頂点 O_2 は辺 (a_4, a_5) と辺 (a_6, a_7) を延長した交点となる．さらに，交点が生じるためには，凸方向が一致する曲線の変化方向は 180 度未満である必要がある．

4.4 最適化手法

統合可能な曲線群を得た後に累積誤差を計算し，グラフ理論における最短経路問題に帰着させる．そして，辺の数が少ないほど，同じ辺数の場合は累積誤差が小さいほど最適であるとして，パス全体の曲線最適化を行う．

5. まとめ

本稿ではベクター形式の操作性に着目し，Potrace の改良を行った．まず，操作性の高い通過点の取り方について検討し，連続した曲線を統合する最適化アルゴリズムを提案した．そして，操作性の高いベクター形式においても曲線最適化が可能であることを明らかにした．

参考文献

[1] K. Kawamura, H. Watanabe, H. Tominaga, “A study on Vector Representation of Binary Images Containing Halftone Dots,” Picture Coding Symposium of Japan (PCSJ) 2003, P-2.13, Nov. 2003.

[2] Peter Selinger, “Potrace: a polygon-based tracing algorithm,” http://potrace.sourceforge.net/potrace.pdf, 2003.

[3] “Illustrator,” http://www.adobe.co.jp/