

7Q-2

図形編集ツールに関する考察

山口 文雄 山上 宣彦

*(株)東芝 情報通信システム技術研究所

1.はじめに

現在、文書処理システムでは、テキスト、図形、イメージ等、様々なデータを統合的に管理する必要性が高まっている。その中で、図形処理は重要なものの1つとなっている。特に最近ではプレゼンテーションをはじめとする各種資料作成の機会が増加していることや、PostScriptプリンタの普及などにより、高度な図形処理機能への要求が高まってきた。そこで、一般のユーザが短時間で高品位な図形を作成できるシステムをワークステーション上で開発(プロトタイピング)したのでここで報告する。

2.図形編集ツールの機能

現在の文書処理システムでは、プリントアウト機器としてPostScriptプリンタへの対応が増えていることに着目し、本ツールではPostScriptでサポートしている機能を生かせるようなシステムとした。主な特徴として次の4点が挙げられる。

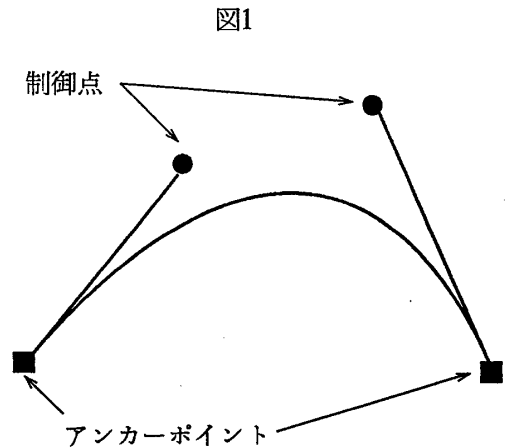
- (1) PostScriptで採用されている Bezier 曲線が描けること。
- (2) グラデーション、ラインキャップ、ラインジョインといった飾り付けが可能であること。
- (3) デジタル2値画像の輪郭をベクトル化し、Bezier 曲線として表示できること。
- (4) プリンタ出力前にプリントアウトで得られるものにより近いイメージを画面上で確認できること。

以下にそれぞれについて詳しく述べる。

(1)Bezier曲線のハンドリング

Bezier曲線は図1のように2つのアンカーポイント(必ず通る点)と2つの制御点(曲線の接線ベクトルの方向を決める点)で形が決まる曲線である。Bezier曲線の描画に当たっては、これらアンカーポイントと制御点を指定しなければならない。アンカーポイントは曲線の通過点であるためユーザは画面を見ながら簡単に指定することができる。一方、制御点に関しては必ず通過する点ではないため、ある程度の慣れが必要になって来る。そこで、本ツールではBezier

曲線を描く手順として以下のようにした。



- [1] 曲線の始点を指定する。これは第1のアンカーポイントとなる。
 - [2] 始点に対する曲線の接線ベクトルを指定する。これは第1の制御点となる。この時、第1のアンカーポイントと制御点を結ぶ接線ベクトルを直線で表示する。(図1-1参照)
 - [3] 曲線の終点を指定する。これは第2のアンカーポイントになる。
ここで、第1のアンカーポイントと[2]で指定した制御点、第2のアンカーポイントを結ぶ曲線を表示させる。ただし、ここではまだ第2の制御点は指定されていないので第2のアンカーポイントと同じ場所を制御点として曲線を表示させる。(図1-2参照)
 - [4] 終点に対する接線ベクトルを指定する。これにより第2の制御点を最終的に決定する。
ここでは、制御点の移動にあわせてリアルタイムで曲線、接線ベクトルの表示を行なう。(図1-3参照)
- [1]~[4]までのステップで曲線の描画が終了するが、引き続き各制御点、アンカーポイントの移動を可能とし細かい整形を行なうようにした。
すでに描かれている曲線を整形する場合を想定して、変形のためのコマンドを設けた。コマンド選択と同時に曲線に対するアンカーポイント、制御点を表示して整形可能とする。整形中も指定した点の動きを

わせてリアルタイムに曲線の形を変えていく。

図1-1

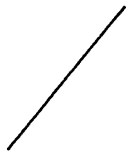
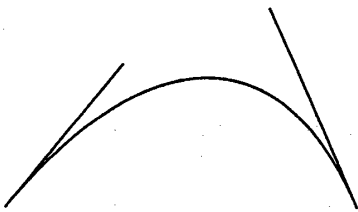


図1-2



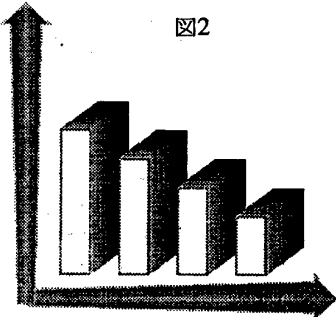
図1-3



(2) 図形の飾り付け

2次元図形を立体的にみせるために図形に影をつけ、影と本体間を段階的に変化させて表示することができる。(図2参照) 本体図形を指定し、影をつけるコマンドを選ぶことにより処理を始める。このコマンドでは、本体に対する影の位置、影の塗りつぶしパターンを選択することができる。本体と影の補間は、本体の塗りつぶしパターン、影の塗りつぶしパターン、本体と影の相対距離により指定できる。

図2



(3) 2値画像の輪郭線ベクトル化

複雑な図形を書くには描き手のセンスが必要になってくる。そこで、誰でも手軽に描画ができるよう2値画像をスキャナーで読み込みそのデータを編集できるようにした。しかし、2値のイメージデータでは、拡大縮小、回転といった編集を施した際ギャザが目立ってしまう。そこでイメージデータに編集を施すには、輪郭線を1度ベクトルデータにし、それを編集する。ベクトルデータに変換する際、イメージデータの品質によりベクトル化手法を変えなければならない。つまり、粗いデータをそのままベクトル化するのではなく、誤差の許容範囲を大きめにとるなどの工夫が必要になる。

(4) プリントアウトイメージの確認

ディスプレイ・ポストスクリプトの普及が遅れている現状では、PostScriptプリンタに出力される結果を画面上で確認することは難しい。しかも図形編集中にプリントイメージを表示することは応答速度が遅くなりマン・マシンインタフェースを考えたとき非常に使いづらいものとなる。そこで、図形編集中は実際のプリントアウトイメージとは合致しない応答速度を重視した表示とし、必要に応じて別ウィンドウにプリントアウトイメージを表示する方法を取る。この時のプリントアウトイメージは極力プリンタ出力に近いものの表示とし、この表示上では図形の整形、編集は一切行なわないものとする。プリントアウト前にプリンタ出力イメージを画面上で確認することにより、プリンタ出力回数を減らすことができる。

3. まとめ

Bezier曲線の描画、整形をサポートすることにより、より複雑な図形、滑らかさの要求される図形を描けることが確認できた。また、PostScriptで可能な修飾を図形に施すことにより従来よりも豊富な出力が得られるようになった。普通のユーザにとっては既存のイメージデータを取り込んで再利用するケースが多い。そのためには、イメージデータからベクトルデータへの変換機能が非常に有効になる。ただし、このときにシステム側である程度自動的に整形する機能が重要となる。

4. 参考文献

PostScriptリファレンス・マニュアル
Adobe Systems著 ASCII出版局

注 PostScriptはAdobe Systems社の登録商標です。