

アウトラインフォントの変換方法

4V-4

北村隆* 笹山和宏* 藤原範治**

*(株) 沖テクノシステムズ ラボラトリ

**沖電気工業 (株)

1 はじめに

近年、アウトラインフォントが多く、の計算機で利用可能になり、日本語フォントの書体数の増加や外字のカスタマイズが望まれている。

新しいフォントデザインを作成するには、紙の上のデザインをスキャナでデジタル化する方法や、画面上で直接作成する方法がある。しかし、日本語の全文字のデザインを作成するのは容易ではない。そこで、印刷メーカ等より供給される既存のデザインを使用することが考えられる。

この場合、印刷メーカ等より供給された字母のデータを計算機で使用可能なフォーマットへと変換する必要がある。本稿では変換前と変換後のフォントをそれぞれデジタル字母、実装フォントと呼ぶ。

このデジタル字母と実装フォントのフォーマットは多数存在し、変換処理系はその組合せの数だけ必要となる。そのため新たなフォーマットの追加は大きな作業量になり、フォーマットに依存しないフォントの変換が望まれている。

そこで本稿では、共通フォーマットの導入やヒント情報の共通化によって、フォーマット依存を小さくしたフォント変換方法について述べる。

2 共通フォーマットへの変換

変換処理のフォーマット依存を小さくするためには、デジタル字母の依存部と実装フォントの依存部を切り離して共通部分を設ける必要がある。

そこで、まずデジタル字母を共通フォーマットへ変換する。共通フォーマットへの変換はアウトラインのデータや識別子を変更しない物理フォーマット変換である。

この共通フォーマットを用いることにより、デジタル字母のフォーマットに依存しないヒント情報の抽出と保存、文字コードの管理を行なう。また、GUIのフォント作成ツールやヒント調整ツールの

開発も、この共通フォーマットを使用すれば容易に行なえるようになる。

共通フォーマットを用いたフォントの変換処理の概要を図1に示す。

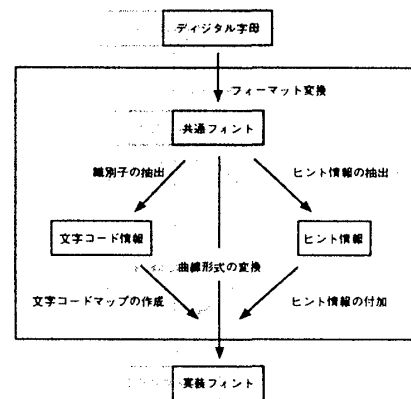


図1：フォント変換処理

3 ヒント情報の抽出

ヒント情報の抽出と付加はデジタル字母の曲線形式と、実装フォントの仕様に大きく依存するが、次の方法により依存を小さくする。

- ・曲線形式の依存部を切り離す。
- ・ヒント情報の種類を限定する。

デジタル字母で使用されている曲線形式に依存する部分を小さくするために、図2の手順でヒント情報の抽出を行なう。

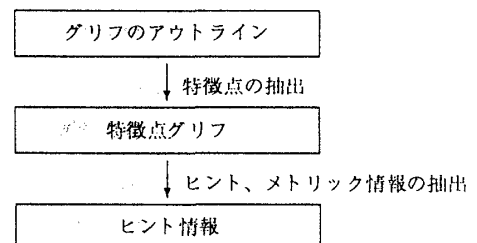


図2：ヒント情報の抽出

まず1つのグリフのデータから、以下の点のリス

トである特徴点グリフを作成する。

- ・ 直線と直線の交点
- ・ 直線と曲線の交点
- ・ 曲線上の極点
- ・ 曲線上の変曲点

グリフの特徴点の抽出部において曲線形式に依存するのは、極点と変曲点の抽出処理のみで、それ以外の処理は依存しない。

次に、抽出した特徴点グリフよりヒント情報を抽出する。ヒント情報の種類は実装フォントによって異なるが、以下の項目に限定したヒント情報を共通フォントから抽出する。

- (1) 個々の文字のヒント情報
(ステム位置と幅)
- (2) メトリック情報
(Bounding Box)
- (3) フォント全体のヒント情報
(標準ステム幅、アラインメント位置)

ただし、(3) のフォント全体のヒント情報は、(1) の個々の文字のヒント情報を統計することにより自動的に求まるが、実装フォントごとに補正が必要となる。

以上のようにヒント情報の種類を制限することにより、デジタル字母のフォーマット依存を小さくした。

4 曲線形式の変換

デジタル字母の曲線形式がBスプラインなどのパラメトリック曲線の場合、実装フォントのパラメトリック曲線への変換は難しい。実装フォントで使用する曲線は、極点で曲線を分割するといった制限が付加されることがあるため、デジタル字母と実装フォントの曲線形式がまったく同じであっても変換が必要となる。

パラメトリック曲線からの変換は、曲線上のサンプル点列を作成した後で行なう。パラメトリック曲線以外の場合も同様である。これによって、デジタル字母の曲線形式に依存する部分はサンプリング処理のみとなり、実装フォントの依存部と切り離すことができる。

図3に曲線形式の変換の概要を示す。サンプル点列からの曲線あてはめは、最小二乗法を用いたあてはめ方法^[1]を用いる。

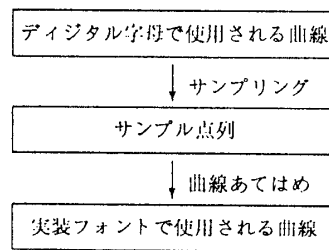


図3：曲線形式の変換

5 おわりに

共通フォーマットの導入やヒント情報の共通化によって、デジタル字母のフォーマットの依存を少なくしたフォント変換ツールを作成した。このツールによって日本規格協会のデジタルマスタ字母^[2]と独 URW 社の I K フォーマット^[3]のフォントを自動的に Apple TrueType フォント^[4]へと変換した。変換した TrueType フォントの出力例を図4に示す。

今後の課題としては、(1) 対応フォーマットの追加 (2) フォーマット追加を容易にするためのフォーマット依存部の独立モジュール化 (3) アウトラインやヒントの目視調整を行なう GUI エディタの開発が考えられる。

参考文献

- [1] M.Plass and M.Stone "Curve-Fitting with Piecewise Parametric Cubics" Computer Graphics, Vol 17, No.3 (1983)
- [2] "平成元年度 高品質フォントの標準化調査研究報告書" 日本規格協会 文字フォント開発・普及センター (1990)
- [3] Peter Karow "Digital Formats for Typeface" URW Verlag (1987)
- [4] "TrueType™ Font Format Specification Version 1.0" Apple Programmers and Developers Association (1990)

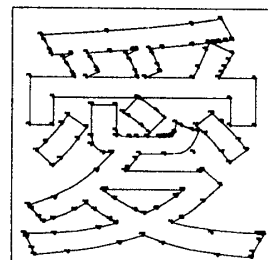


図4：変換結果の例

(左：TrueType フォントのアウトライン出力
右：同ビットマップ出力 (12pt/300dpi を拡大))