ホームページの広場 第28回

ビットマップフォントとアウトラインフォント

大阪医科大学放射線医学教室　非常勤講師
（関西福祉科学大学 保健医療学部　教授）　上杉康夫

ビットマップフォント

「ビットマップフォント（ bitmap font ）は点の集合体として設計されています※1。

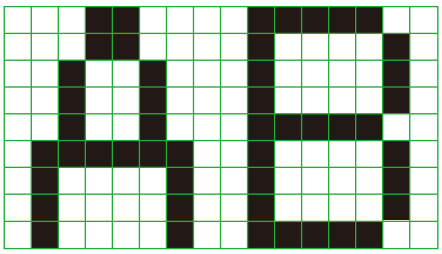

図1：ビットマップフォントの設計※1

したがって、ドットフォントとも呼ばれます。単純な仕組みのためコンピュータが登場した初期から使われているフォント形式です 。数字や文字の形状をドットの有無で表わし、2進数の0と1に対応させています。例えば、アルファベットのAであればドットの有無（白と黒）をそれぞれ0と1に対応させ、下図のように表わします。

図2：ビットマップフォントの2進数化※2

各点は描画されるか否かの1ビットの情報です。一般に8ビットごとにまとめてバイト列で表現します。

上図のAの部分の1行目が□□□■■□□□ですから、白が0、黒が1であるとして 00011000 つまり16進表記で 0x18。次の行も同じで、その次が 00100100 なので 0x24 というようにしていきます。このようにしますと A は

18 18 24 24 24 7E 42 42 42　と表せます。

B は、

7C 42 42 42 7C 42 42 42 7C　となります。

ビットマップフォントは小さい文字だけならばデータ量も少なくて済み、表示のための負担も少ないのでWindowシステムがGUI（Graphical User Interface）を導入する前から使われています。しかし、大きな文字を作るとなるとそれにあわせて文字のデータを作る必要があります。したがって様々な大きさの文字を作っていくとデータ量が莫大になっていきます。

ビットマップフォントのCourierを使って観察してみます（表1）。18ptの「A」のフォントを基準にしてみます。

表の下段はフォントを4倍拡大にした像です。下段で「元々 12PtのA」フォントと「18ptを12相当に縮小」したAフォントの場合を比較してみましょう。18ptを12ptに縮小すると隙間がつぶれたり線がなくなったりと形が崩れます。

ついで下段の4倍拡大した像で、「元々 32PtのA」フォントと「18ptを32pt相当に拡大」したAフォントを比較してみましょう。「18ptを32pt相当に拡大」したAフォントはギザギザが目立ちます。

上記のことを回避するために、何種類かの大きさのフォントを設計（図3）しますので、データ量が大きくなります 。またフォントの拡大・縮小にも制限がある場合があります※1。

	元々 12ptのA	18ptを 12pt相当 に縮小	18ptのA	18ptを 32pt相当 に縮小	元々 32ptのA
本来の大きさ	A	A	A	A	A
4倍に拡大	A	A	A	A	A

表1：ビットマップフォントのCourierの拡大と縮小※1

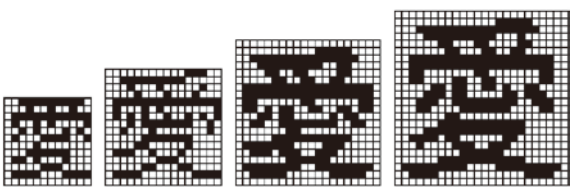

図3：様々な大きさの文字にたいして、文字データを作る必要がある※3

アウトラインフォント

大きく拡大しても形が美しいように、また線の太さの変化や「はね」や「はらい」などのデザインを表現するために文字を構成する線の輪郭（outline）を線のデータとして記憶するようにしました。これがアウトラインフォント（outline font）です。

ディスプレイは点の集まりですから、アウトラインフォントも最終的には点のデータに変換されます。この処理をラスタライズ（rasterize）といいます。まず輪郭線のデータを拡大してから、その中にある点を塗りつぶして文字を作ります。

赤い線が文字の形を表す線です（図4）。緑の線が画面を作っているピクセルを表しています。この赤い枠の中にあれば100%の濃さで黒く塗り、一部がかかっていればその量に応じて薄く塗ってみました。このようにすれば拡大が自由にできます。

アウトラインフォントとビットマップフォントの比較

アウトラインフォントは、フォントのサイズに応じて計算しなおされて出力されるため、文字を拡大縮小しても、なめらかな曲線が表現できるというメリットがあります。このため現在では、アウトラインフォントが主流となっています。

ただし、かなり小さい文字を表現する場合には、アウトラインフォントでは輪郭線がぼやけるので、ビットマップフォントの方がきれいに表現できる場合が多いです。

このあたりを見てみましょう。

文字を4倍に拡大表示でアウトラインフォント（表2）とビットマップフォント（表3）を観察します。アウトラインフォントは32ptの文字でもビットマップフォントに比べ滑らかに見えます。ところが一方8ptの文字ではアウトラインフォントでは輪郭線がぼやけ、むしろビットマップフォントの方がきれいに見えます。

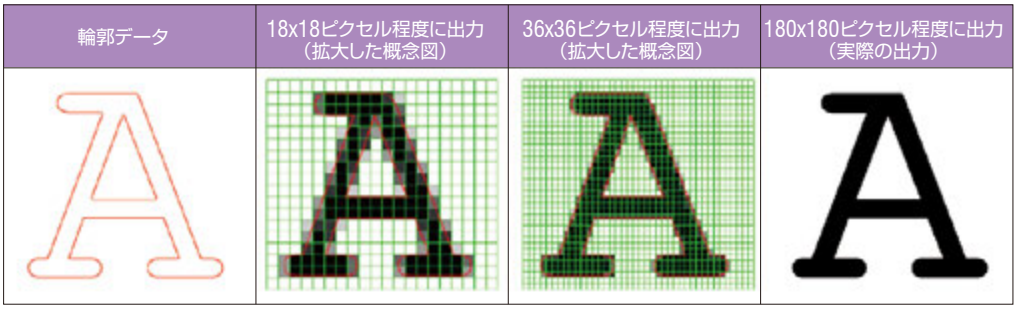

図4：アウトラインフォントの処理※1

pt	8	10	12	14	18	24	32
本来の大きさ							
4倍に拡大							

表2:アウトラインフォントの Courier 10 Pitch、下段は文字を4 倍に拡大表示したもの※1

pt	8	10	12	14	18	24	32
本来の大きさ							
4倍に拡大							

表3:ビットマップフォントの Courier、下段は文字を4 倍に拡大表示したもの※1

アウトラインフォントの間違い

次の認識は間違いとされています※1。

間違い1 データを線で持っているのでギザギザが無くきれい

最終的に表示される時はビットマップと同じにピクセルで表されますからギザギザが無くなるわけではありません。ビットマップでは大きなフォントが用意されていないために無理に拡大するとギザギザが拡大されて目立つということなので、大きなフォントをビットマップで用意すれば同等のきれいさを得ることができる。特に小さな文字はアウトラインフォントでは返って汚くなりがちです。アウトラインフォントの中には小さい字のために内部にビットマップデータを隠し持っている場合があります。

間違い2 ビットマップでは線の太さの変化を表現できない

大きなフォントをビットマップで用意すれば同等に表現できます。ビットマップは線の太さの違いなどを表現できないような小さな文字に使われることが多いので細かな変化を付けられないだけです。

間違い3 アウトラインフォントは色の濃さも使う

上記のアウトラインフォントの Courier 10 Pitch の例では黒の他に灰色の部分がありギザギザの解消に一役買っています。小さい文字ではこれがないとつぶれて読めないことがありますが、色の濃さを使わないアウトラインフォントもあり得ますし、逆に灰色の部分を持つビットマップを作ることもできます。

間違い4 アウトラインフォントはデータ量が多い

小さなフォントだけならビットマップの方がデータ量を少なくできます。しかし品質を保ちながら大きなフォントを作るとなるとアウトラインの方が少なくできます。ビットマップは大きさが2倍になるとデータ量は4倍になります。しかも拡大・縮小をすると品質が落ちるのでいろいろな大きさのフォントをたくさん作る必要がありますのでアウトラインフォントと同じ品質で大きな文字を自由な大きさで作れるようにするにはビットマップの方がデータ量は多くなります。確かに実際のビットマップフォントのファイル容量がアウトラインフォントより小さいですが、これは大きなサイズの文字を入れていないことと、灰色の部分がなく白か黒かであることによります。

間違い5 アウトラインフォントは表示に時間がかかる

確かにアウトラインフォントは輪郭データと大きさから計算してビットマップにあたるデータを作り出してから表示しますから、ビットマップの方が表示については速そうです。小さな文字ではそれは揺るぎません。しかし大きな文字になるとビットマップでは文字のデータが大きくなり全部をあらかじめメモリ上に用意することができなくなる可能性があります。この場合ハードディスクなどに読みとりに行くことになりますがこれはかなり時間がかかる作業になります。こういう状況では一概にアウトラインフォントは表示に時間がかかると断言できません。

今回は、ビットマップフォントとアウトラインフォントの概要について述べました。

参考文献

※1：ビットマップフォントとアウトラインフォント
http://www.seiai.ed.jp/sys/text/cs/chp02/c02a140.html

※2：フォントとは｜アウトラインフォント・ビットマップフォント
http://yamanjo.net/knowledge/others/others_09.html

※3：ビットマップフォント｜フォント｜NEC
http://jpn.nec.com/font/bitmap/