モバイル環境のユーザインタフェース向上に向けて
~ 携帯情報端末用フォントの高品質化に関する一手法 ~

安本　護[†]　　池田尚志[‡]　　中里酉克[†]　　富田義数[†]

[†]三洋電機株式会社ハイパーメディア研究所
〒 503–0195 岐阜県安八郡安八町大森 180

[‡]岐阜大学工学部
〒 501–1193 岐阜市柳戸 1–1

Email:yasu@gf.hm.rd.sanyo.co.jp
Tel:0584–64–3764　　Fax:0584–64–3127

あらまし　携帯情報端末 (PDA) の普及や移動体通信技術の発展によって，モバイル環境における情報処理の機会が増大している．モバイル環境においても，できる限り高品質の文字表示が望ましいが，PDA 等では低画素のビットマップフォントを搭載するのみである．本論文では，モバイル環境のユーザインタフェース改善を目指して，任意サイズのフォントを高品質に生成する方法を提案する．この手法は，ストロークフォント方式をもとに隣接ストロークの間隔を調整してつぶれを防止する線間補正処理と出力画素数に合わせてストロークの混み具合を調節する複雑度補間処理を適用するものである．本手法を用いれば，画素数の大きな出力で正確な字体を表現できるだけでなく，12 ドット程度の低画素でも判読の容易なフォント生成が可能である．16 名の被験者に対する主観テストの結果，人手によって制作したビットマップフォントに匹敵する品質を得られることが確認できた．

キーワード　ストロークフォント，スケーラブルフォント，判読性，携帯情報端末

For the Purpose of Improving Mobile Computing User Interfaces
— A High Quallity Font for Mobile Terminals —

Mamoru Yasumoto[†], Takashi Ikeda[‡], Yuuki Nakasato[†] and Yoshikazu Tomida[†]

[†]Hypermedia Research Center, Sanyo Electric Co., Ltd.
180 Ohmori, Anpachi-cho, Anpachi–gun, Gifu–ken, 503–0195 Japan

[‡]Faculty of Engineering, Gifu University
1–1 Yanagido, Gifu–shi, 501–1193 Japan

Email:yasu@gf.hm.rd.sanyo.co.jp
Tel:0584–64–3764　　Fax:0584–64–3127

Abstract　This paper proposes a high quality scalable font technology for mobile terminals in order to improve mobile computing user interfaces. This font not only expresses an accurate glyph in a large size but also makes smaller output legible. We use the stroke font technology and the font compensation technique proposed here to accomplish this. The technique consists of two algorithms. The first creates a space between adjacent strokes that have been blurred together by scale-down operations. The second prevents crowding of the strokes by interpolating a high resolution font and a low resolution font. Experiments show that our proposed method can achieve as a good quality font as the man-made bitmap font.

Keywords : stroke font, scalable font, legibility, PDA

1　はじめに

携帯情報端末の普及や移動体通信技術の発展によって，モバイル環境における情報処理の機会が増大している．文字は，もっとも基本的な情報伝達手段であり，その重要性は言うまでもない．モバイル環境においても，できる限り高品質の文字表示が望ましい．JIS Z 8513 規格 [1] は，視覚表示装置を用いるオフィス作業に関して，以下の実現を要求している．

1. 視距離（目と画面との間の距離）が 300mm 以上であること
2. 漢字の最小文字高が視角 25′ であること
3. 漢字の文字構成が 15 画素× 16 画素以上であること

PDA 等の小型携帯情報端末では，端末サイズをコンパクトにするために大型の表示装置を使用することが難しく，画素数が 240 × 160 ドットから 480 × 240 ドット，ドットピッチが 0.2 ~ 0.3mm/dot の液晶ディスプレイを使用することが多い．また，搭載される日本語フォントは，縦横の画素数が 12 ドットのビットマップフォントが中心となっている．これは，電子メールや WWW 等のアプリケーションにおいて情報の一覧性を確保するには，一行あたり 20 文字以上の漢字表示が必要なためと考えられる．携帯情報端末使用時の視距離は 350mm 程度となるので，12 ドットフォントの文字高は 23′ ~ 35′ である．したがって，携帯情報端末の利用において，視距離と文字高は JIS 規格を満足しているものの，文字の構成画素数は十分と言えない．ユーザビリティの向上や字体の正確な表現を行うには，より画素数の多い文字を表示できることが必要である．また，文字サイズの任意の変更を要求するアプリケーションへの対応も必要である．例えば，WWW の記述に用いる HTML には 7 種類の文字サイズ指定が存在する．しかしながら，大容量記憶装置をもたない携帯情報端末にとって，複数のビットマップフォントやアウトラインフォントを搭載することは困難である．例えば，MS–Windows に標準搭載されている明朝体フォントやゴシック体フォントは，4 ~ 5MB もの記憶容量を必要とする．以下にモバイル環境のユーザインタフェースを向上するために，携帯情報端末用フォントに求められる事項をまとめる．

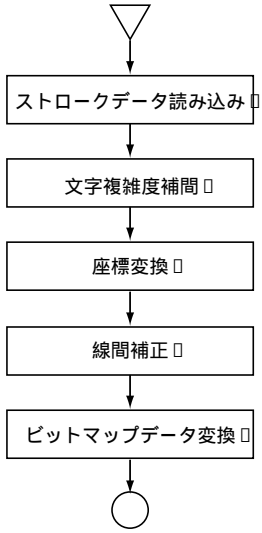


図 1: フォント処理全体のながれ

1. 文字サイズを任意に変更可能であること．
2. 12 ドット程度でも文字がつぶれず，判読が容易なこと．
3. 正確な字体を表現できること．
4. データサイズがコンパクトであること．

本論文では，上記目標を実現するためにストロークフォントに基づく携帯情報端末用フォント処理方式を提案し，試作フォントデータを用いた実験によって有効性を確かめる．

2　フォント処理方式

2.1　フォント処理の概要

図 1にストロークフォントデータから表示用ビットマップデータを生成するまでのフォント処理全体の流れを示す．線間補正処理と文字複雑度補間処理は，低画素フォント生成時に発生する字画のつぶれを防ぐためのフォント形状補正処理である．

2.2　ストロークフォント方式

ストロークフォントデータでは，一辺の大きさ M の正方形枠内にデザインされた文字から骨格

を特徴付ける点を抽出し，その座標 (X,Y) で文字形状を記述する．(X,Y) の取り得る最大値 M はメッシュサイズと呼ばれ，これが大きいほど文字形状を高精度に記述できる．表示・印刷の際は，式 (1) を用いて特徴点データを任意の出力サイズ Sに座標変換し，得られた座標 (X_s,Y_s) を直線または曲線で結んでビットマップデータを得る．

$$\begin{aligned} X_s &= \frac{X}{M}\cdot S \\ Y_s &= \frac{Y}{M}\cdot S \end{aligned} \qquad (1)$$

一方，最近のパーソナルコンピュータやプリンタに広く用いられているアウトラインフォントも文字サイズを任意に変更可能である．アウトラインフォント方式では，文字の輪郭を抽出してその特徴点座標を記憶するので多様なフォント表現が可能である．アウトラインフォントの表示・印刷では，特徴点に囲まれた内部をドットで塗りつぶしてビットマップ化する．

両者を比較すると，大サイズにおける表現力ではアウトラインフォントが優れているが，データサイズや高速処理可能な点ではストロークフォントのほうが有利である．すなわち，主に低画素フォントを使用し，しかも十分なハードウェア資源をもたない PDA への応用にはストロークフォントが適している．

2.3 低画素フォント生成の問題

ストロークフォントを用いると任意画素数のビットマップデータを生成可能であるが，画素数が極端に少ないと字画につぶれを発生する．字画のつぶれとは，隣合うストロークの座標間距離が 1 となって，ビットマップ化した際に間隙が無くなる状態である．字画のつぶれを起こす原因には，出力解像度の不足と座標の丸め誤差がある．

2.3.1 出力解像度の不足

字画の混み具合すなわち文字複雑度が出力解像度を上回るとつぶれを生じる．例えば，「遺」という漢字には x 座標のオーバラップする水平ストロークが 8 本存在する．字体の正確な表現には出力サイズ 16 画素以上が必要であり，これよりも小さいとつぶれが発生して文字品質の著しい劣化が起こる．そこで，人手による低画素ビットマップ

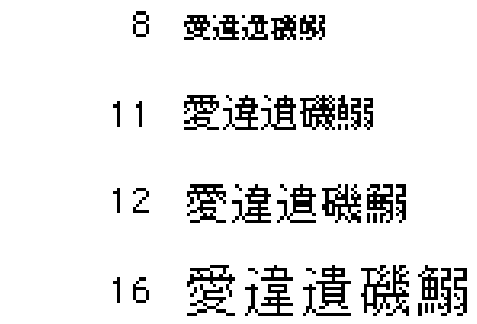
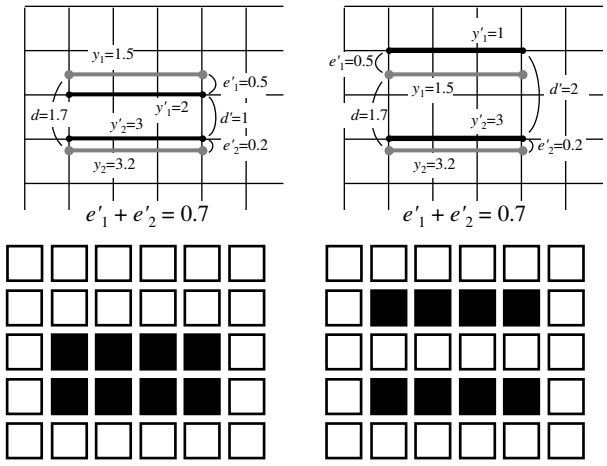


図 2: 低画素ビットマップフォントの例



(a) つぶれ発生 (b) つぶれなし

図 3: 丸め誤差の発生

フォントの制作では，図 2に示すように字画の省略と判読性の確保を両立させるデザインテクニックが使われている．

2.3.2 座標の丸め誤差

ビットマップデータを生成するには，フォントデータを座標変換する際に生じる少数点以下の端数を丸めて整数化しなければならない．このとき発生する丸め誤差は，フォントの生成品質に重大な影響を及ぼすことが知られている [3]．

図 3を用いて丸め誤差によるつぶれの発生を説明する．2 本の水平ストロークを座標変換した結果，y 座標が $y_1 = 1.5$，$y_2 = 3.2$ になったとする．このとき，ストローク間隔 d は 1.7 である．図 3(a) のように両ストロークの y 座標を四捨五入して $y_1' = 2$，$y_2' = 3$ を得るとストローク間隔

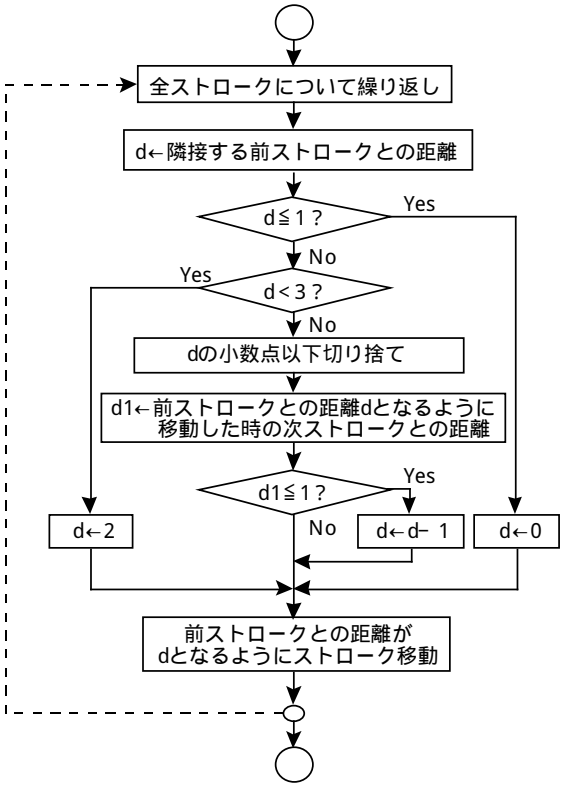


図 4: 線間補正処理アルゴリズム

$d^{'} = 1$ となってつぶれが発生する．ところが，丸め処理前のストローク間隔 $d = 1.7$ の四捨五入結果 $d^{'} = 2$ は字画のつぶれを生じない間隔である．そこで，図 $3(\mathrm{b})$ のように $y_1^{'} = 1$，$y_2^{'} = 3$ とする整数化を用いれば，ストローク間隔は 2 となって字画につぶれは発生しない．丸め誤差の累積は，両者とも 0.7 で等しい．

2.4 フォント形状補正処理

低画素フォント生成時のつぶれ発生を防ぐために 2 種類のフォント形状補正処理を提案する．

2.4.1 線間補正処理

線間補正処理は，隣接する水平 / 垂直ストロークの間隔が 1 とならないように座標を移動し，つぶれの発生を防止する処理である．図 4に線間補正処理のアルゴリズムを示す．隣接するストロークとの間隔が 0 となるように移動する処理は，ストロークの省略に対応する．

2.4.2 複雑度補間処理

線間補正処理は水平 / 垂直ストロークの間隙喪失によるつぶれには有効であるが，斜め線や曲線のつぶれは補正できない．低画素サイズに座標変換してもつぶれないように，字画省略したストロークフォントをデザインすることも可能であるが，このようなフォントは高画素の出力で字画省略が目立ち，十分な品質を得られない．

そこで，線幅の異なる二つのアウトラインフォントデータから任意線幅のアウトラインフォントを生成する手法 [4] を参考にして，式 (2) に示す文字複雑度補間処理を導入する．これは，出力画素数 S_l以上でつぶれないようにデザインした低画素出力用ストロークフォントを D_l (字画省略あり，メッシュサイズ M_l)，出力画素数 S_h以上でつぶれないようにデザインした高画素出力用ストロークフォントを D_h (字画省略なし，メッシュサイズ M_h) とし，D_lと D_hから出力画素数 $S(S_l \leq S \leq S_h)$ に見合った複雑度のストロークフォント D_sを生成するものである．ここで，D_l と D_hは，ストローク数およびストロークを構成する点の数が同一となるようにデザインしておく．すなわち，D_lの字画省略は，複数のストロークを重ねて一本のストロークに見せたものである．

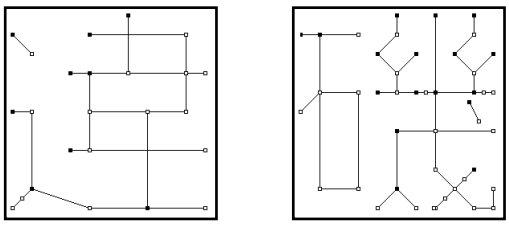
$$
\begin{aligned}
X_s &= \left(R \cdot \frac{X_h}{M_h} + (1-R) \cdot \frac{X_l}{M_l}\right) \cdot S \\
Y_s &= \left(R \cdot \frac{Y_h}{M_h} + (1-R) \cdot \frac{Y_l}{M_l}\right) \cdot S \qquad (2)
\end{aligned}
$$

式 (2) の (X_l, Y_l) , (X_h, Y_h) , (X_s, Y_s) は，それぞれ D_l, D_h, D_sの座標を示す．また，Rは生成結果 D_sに対する D_lと D_hの影響度を決定するパラメータであり，式 (3) を用いて決定する．

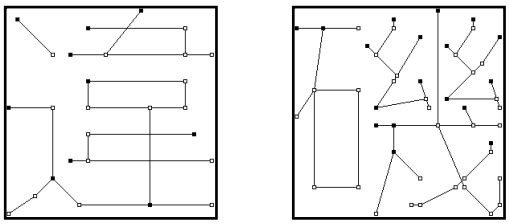
$$
R = \frac{S - S_l}{S_h - S_l} \qquad (3)
$$

3 ストロークフォントデータの試作

情報交換用漢字符号 JIS X 0208 規格の 16 区から比較的字体の複雑な漢字 14 文字 (愛，逢，葵，穐，鰺，綾，慰，萎，謂，違，遺，磯，鰯，胤) を選び，ストロークフォントデータ D_lと D_hを試作した．D_lのデザインには 11 ドットフォント，D_h のデザインには 24 ドットフォントを参照した．すなわち，$S_l = 11$，$S_h = 24$ である．参照に用いたビットマップフォントは，同一ゴシック体字母をもとに人手によって制作したものである．また，



(a) 低画素出力用



(b) 高画素出力用

図 5: 試作ストロークフォント

ストロークフォントのメッシュサイズは，複雑度補間処理を考慮して 11 と 24 の公倍数 264 に選んだ．このように十分大きなメッシュサイズを使用した結果，D_hは字画省略がなく，24 ドット以上の出力にも耐え得るように制作できた．試作したフォントの例を図 5に示す．

次に，試作結果から JIS X 0208 全文字 (6789 文字) のストロークフォントを制作した場合の大まかなデータサイズ S_sを式 (4) を用いて見積る．

$$\begin{aligned} S_s \quad = \quad & \text{文字あたり平均画数} \\ & \times \text{ ストロークあたり平均特徴点数} \\ & \times \text{ 特徴点あたりデータサイズ} \\ & \times \text{ 文字数} \end{aligned} \tag{4}$$

JIS X 0208 に含まれる漢字 (6355 文字) の平均画数は，12.341 本 / 文字である．1 ストロークあたり平均特徴点数には，試作フォントから得た 2.7114 点 / 本を用いる．また，特徴点のデータは，複雑度補間を行わないときで 1 点につき 3 バイト (x 座標× 1，y 座標× 1 および制御情報)，複雑度補間を行うときは 5 バイト (x 座標× 2，y 座標× 2 および制御情報) で表現できる．以上より，ストロークフォントのデータサイズ見積り結果は，複雑度補間なしのとき 682K バイト，複雑度補間ありのとき 1136K バイトとなった．これは，一般

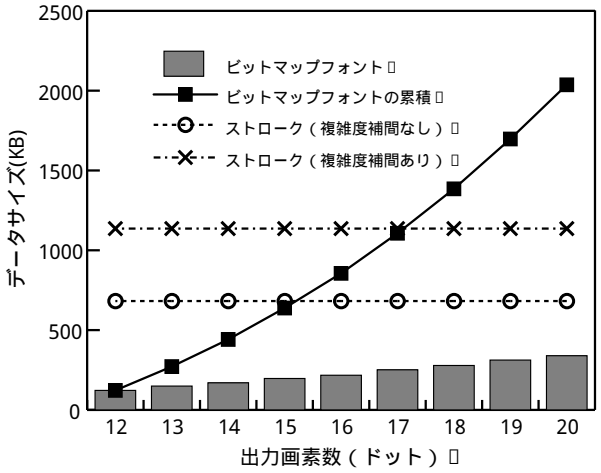


図 6: フォントデータのサイズ

的なアウトラインフォントのサイズが 4 ~ 5M バイトであるのに比べると，かなりコンパクトである．また，ストロークフォント方式では，部首等の組合せによる漢字生成 [5],[6] が可能なので，さらにデータ圧縮することも期待できる．

図 6にビットマップフォントの画素数ごとのデータサイズ，12 ドットから 1 ドット刻みに複数のビットマップフォントデータを持つ場合の累積データサイズおよび複雑度補間処理の有無による二つのストロークフォントのデータサイズを示す．ここで，画素数 n ドットのビットマップフォントのデータサイズ S_bは，式 (6) で与えられる．

$$S_b = \left\lceil \frac{n^2}{8} \right\rceil \times \text{ 文字数} \tag{5}$$

但し ,$\lceil\ \rceil$ はシーリング関数

複雑度補間なしのストロークフォント処理に必要なデータのサイズは，画素数 12 ~ 15 ドットの 4 種類のビットマップフォントデータの合計に相当する．同様に複雑度補間付きのストロークフォントデータは，12 ~ 17 ドットのビットマップフォントのデータ合計に相当する．このようにストロークフォント方式によれば，ビットマップフォントを複数種類もつよりも，少量のデータで多様なサイズのフォントを生成可能である．

4 実験と評価

本論文で提案するストロークフォント方式の効果を評価するために，試作ストロークフォントから次の 4 種類のビットマップデータ生成した．

G1 : D_l (字画省略あり) から生成，複雑度補間なし，線間補正なし

G2 : D_h (字画省略なし) から生成，複雑度補間なし，線間補正なし

G3 : D_h (字画省略なし) から生成，複雑度補間なし，線間補正あり

G4 : D_l (字画省略あり) と D_h (字画省略なし) から複雑度補間を用いて生成，線間補正あり

4.1 生成結果

図 7は，G1 ~ G4 の生成結果である．G1 は，字画を省略しているので画素数の小さなところではすっきりとして読みやすいが，大きくなるにつれて字画省略が目立っている．G2 は，24 ドットでは正確な字体を再現できているが，低画素になるにつれ，つぶれが目立ち，判読が困難になっている．G3 は，水平ストロークの多い「違」や「遺」の低画素展開でもつぶれの発生がなく，線間補正処理の効果が現れている．しかし「磯」の低画素におけるつぶれや「違」の 12，13 ドット生成時に文字構成要素の「口」がつぶれて「一」となる問題が発生している．G4 は，G3 に見られる問題も解消し，低画素から高画素まで判読性の高い結果となっている．

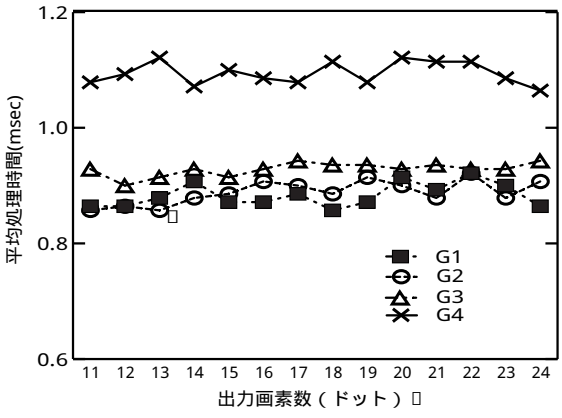
図 7: ビットマップデータ生成結果

4.2 フォント処理時間

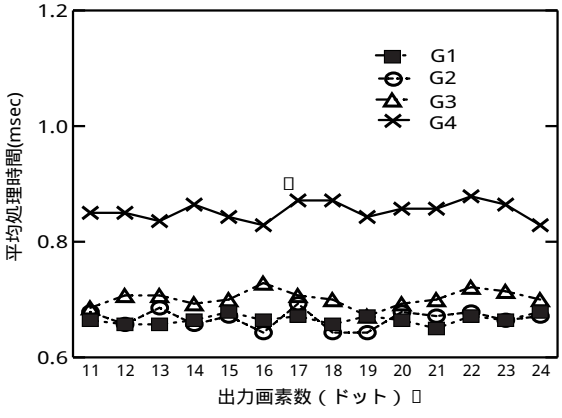
ストロークフォント処理の内容を機能で分割すると座標計算と描画処理になる．図 8は，試作ストロークデータからビットマップデータを生成するまでに要した平均処理時間を描画処理を含む場合と座標計算のみで描画処理を含まない場合に分けて示したものである．ストロークフォントデータの特徴点数は出力画素数に関わらず一定なので，座標計算に要する処理時間は出力画素数に依存しないはずである．図 8(b) の結果は，これを裏付けるものである．一方，描画処理時間は出力画素数が大きくなるにつれて増加すると考えられるが，図 8(a) では，出力画素数に依らず，ほぼ一定となった．これは，出力画素数 11 ~ 24 ドットの範囲では描画処理時間の差が小さく，計算処理のオーバーヘッドに埋もれてしまったためであろう．実際に，G1 の平均処理時間を計測したところ，11 ドットのときが 0.86msec であるのに対し，256 ドットでは 1.2msec に増加した．

G1 ~ G4 の処理時間を比較すると，形状補正を行わない G1，G2 が最も短く，続いて G3，G4 となった．G1 と G2 の処理に用いるストロークフォントデータは異なるものであるが，特徴点数が同一であるため処理時間にほとんど差がないのだと考えられる．G3 の結果からは，線間補正を行うと処理時間が G1，G2 から約 5%増加することがわかる．線間補正に加えて複雑度補間も行う G4 は，同じく約 24%の処理時間増となる．

実験に使用した計算機 (CPU：SuperSparcII 75MHz，メモリ：128M バイト，Sun OS 4.1.4，120MIPS) では，最も計算量の多い G4 でも約 900 文字 / 秒の生成が可能であった．最近では，PDA にも数十 MIPS 程度の RISC CPU が搭載されているので，ストロークフォントの利用は十分に可能と考えられる．



(a) 描画処理を含むとき



(b) 描画処理を含まないとき

図 8: ストロークフォント処理時間

4.3 生成品質の評価

G1 ~ G4 をもとに生成したビットマップデータの品質を評価するために正常な視力を有する被験者 (20 代 7 名 , 30 代 9 名) に対して , 以下の主観テストを行った .

1. 評価対象フォント G1 ~ G4 は , 人手で制作した基準フォント (評価対象フォントと同一画素数) および正確な字体を提示するためのフォント (字画省略なし , 48 ドット) と同時に提示する .

2. 評価指標は「判読性 (正しく読めるか)」「読みやすい」「美しい」「目がつかれにくい」,「好きな」の 5 つとする . これらは , 可読性に関する印象語である [7] .

3. 字種ごとに G1 ~ G4 を基準フォントと比較して , 優れている (+2) , やや優れている (+1) , 同等 (0) , やや劣る (−1) , 劣る (−2) の 5 段階で評価する .

図 9: 主観テストの画面例

4. G1 ~ G4 の提示順は , 字種ごとにランダムに変更する .

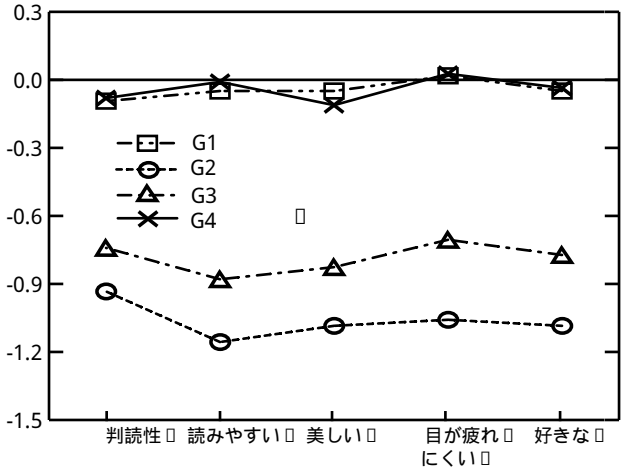
5. 評価対象フォントの画素数は 11 ドットおよび 16 ドットとする .

6. フォントの提示には , WindowsCE ベースのハンドヘルド PC (480 × 240 画素モノクロ液晶 , ドットピッチ 0.25mm/dot) とその上で動作する WWW ブラウザ (Pocket Internet Exploror) を用いる (図 9) .

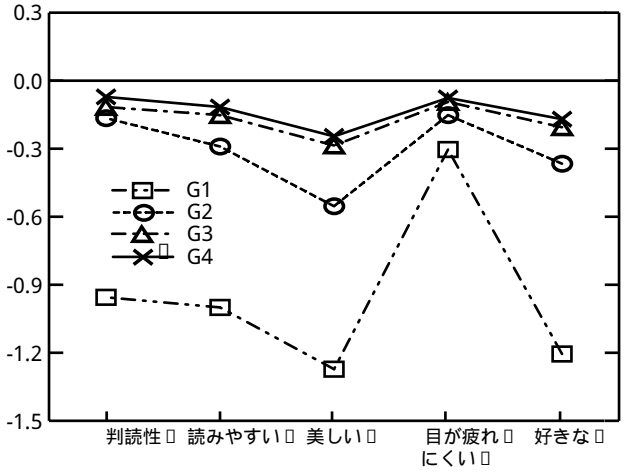
被験者から得られた得点を指標ごとに平均した結果を図 10に示す . 画素数 11 ドットのときも 16 ドットのときも指標によって G1 ~ G4 の得点順位が大きく逆転することはなかった . すなわち , 実験に用いた指標には相関があり , フォントの可読性評価に有効であると言える .

出力画素数 11 ドットのとき , G1 と G4 は , ほとんど同じ得点で基準ビットマップフォント並の品質となった . これは , 11 ドットに最適化されたストロークフォントデータが , 人手で制作したビットマップフォントと同等の品質を達成できたことを示すものである . また , G3 は G2 よりも良い評価を得ており , 線間補正処理の効果が現われている .

出力画素数 16 ドットでは , G4 と G3 が , ほぼ同じ得点で基準フォントよりわずかに劣り , G2 はそれらよりもやや劣る結果となった . これは , 図 7からわかるように今回の実験では線間補正を行わない G2 でも出力画素数が 16 ドットになると字画のつぶれがほとんど発生しなかったからであろう . G1 は「目が疲れにくい」という指標だけが比較的高い得点を得ている . このことから , たとえ字画が省略されていても , すっきりとしたつぶれの無いフォントであれば , 目への負担は少ないと考えられる .



(a) 出力画素数11ドット



(b) 出力画素数16ドット

図 10: 主観テストの結果

以上より，線間補正と複雑度補間を併用するG4を用いれば，11ドット，16ドットのいずれにおいても品質の高いフォントを生成できることがわかった．出力画素数12～15ドットに対する主観テストは実施しなかったが，図7の生成結果から11ドットと16ドットの中間的な結果が得られるものと予想できる．すなわち，出力画素数が11ドットから16ドットに近付くにつれて，生成品質は，G1下降，G2上昇，G3上昇となり，G4は安定して良い品質を得られるであろう．

5 まとめ

本論文では，文字サイズを自由に変更することが可能で，大きく拡大したときに正確な字体を表現できるだけでなく，小型携帯情報端末で多く用いられる低画素サイズにおいても字画につぶれの無い判読が容易なフォントの生成を目指して，線間補正処理と文字複雑度補間処理を用いたストロークフォント方式を提案した．被験者による主観テストによって，本論文で提案するストロークフォント方式を用いれば，ビットマップフォントに匹敵するフォント品質を得られることを確認した．また，本方式のデータサイズはパソコン等で使用されているアウトラインフォント方式よりも小さくなり，大容量の記憶装置を搭載できない携帯端末でも使用可能な見通しを得た．

このようなフォント方式は，今後ますます普及するであろう携帯情報端末のユーザインターフェイスの改善に寄与するものと考える．

参考文献

[1] 人間工学− 視覚表示装置を用いるオフィス作業− 視覚表示装置の要求事項，日本工業規格，JIS Z 8513 (1994).

[2] 上原徹三：フォント関連技術の現状と課題，情報処理，Vol.31, No.11, pp.1570–1579 (1990).

[3] 本門慎一郎，西村幹男，平野文昭：パーツセパレート方式の漢字アウトラインフォント技術，FUJITSU.46, 6, pp.598–604 (1995).

[4] 山崎一生，入山徳夫:高精細ディジタル文字図形の変倍の一方法，信学論(D)，Vol.J70–D, No.2, pp.387–396 (1987).

[5] 陳和明，小沢慎治:多様な明朝体文字の規則的な生成，信学論(D–II)，Vol.J72–D–II, No.9, pp.1423–1431 (1989).

[6] 田中哲朗，岩崎英哉，長橋賢児，和田英一:部品合成による漢字スケルトンフォントの作成，情報処理学会論文誌，Vol.36, No.9, pp2122–2131 (1995).

[7] 宮崎紀郎：新聞紙面における可読性 –読みやすさへの実験を通して–，印刷雑誌，Vol.76, 12, pp17–23 (1993).