

PixelDoublor

増井俊之[†]

解像度の低い表示装置上でより多くのピクセルを表示するための手法 PixelDoublor を提案する。画面を揺らしたり移動させたりすることにより実際の表示装置の解像度より細かい画像を表示することが可能であるが、本稿では表示装置を動かさなくても同様の効果が得られることを示し、その応用についても述べる。

PixelDoublor

Toshiyuki Masui [†]

We propose a technique *PixelDoublor*, which enables display devices to show more pixels than actual number of pixels. Taking advantage of the visual illusion effect coming from the nature of the human eye, PixelDoublor can display a moving image with upto 100% more pixels than the still image.

1. はじめに

小型計算機ではディスプレイの解像度が常に問題となる。たとえば 160×160 ドットの小型 PDA に 24 ドットフォントの漢字を表示させようとするとき 36 文字しか表示させることができない。このためドット数の少ない簡略化したフォントが用いられることが多いが、読みにくいし文字を区別しにくくなる問題がある。また大型の電光掲示板などでは、なるべく少ないドット数で多くの情報を表示するための手法が重要である。

静止画面では画面のドット数を越える解像度で表示を行なうことは不可能である。たとえば上記の PDA では 80 本しか線を表示することはできない。しかし、ディスプレイを移動させたり揺らさせたりしながら表示を変化させることにより、さらに高い解像度を表現することができる可能性がある。

ディスプレイを移動させながら表示を変化させることにより 6 ドット幅の領域に 4 本の線を表示させることができることを図 1 に示す。ディスプレイを 1 タイムスライス毎に半ドットずつ左に動かしながら、 $t = 0$ において “■■■■□□”， $t = 1$ において “■□■■□□”， $t = 2$ において “□■■■□□”，... を表示すると、 $x = 3$ の位置では黒いピクセルが表示され、 $x = 5$ の位置では点滅するピクセルが表示される。タイムスライスが充分小さい場合はユーザには $x = 5$ の位置には灰色のピクセルが表示されているように見える。この結果図 1 の最下行のようにユーザには 4 本の線が見えることになる。

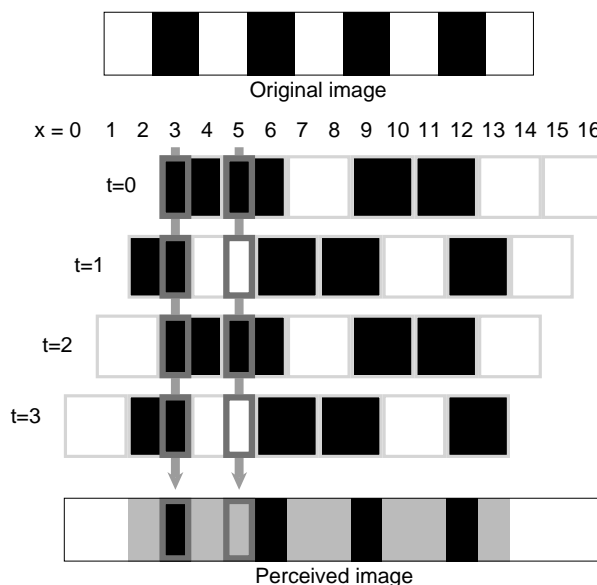


図 1 6 ピクセル幅で 4 本の線を表示

[†] ソニーコンピュータサイエンス研究所
Sony Computer Science Laboratories, Inc.

同様の手法により 10×10 ピクセルの領域に 14×14

ドットのフォントを表示する手法を図2に示す。ここで、ディスプレイのピクセルは表示するフォントのピクセルより50%大きいかかわらずディスプレイを揺らしながら表示を変化させることによりフォントの細い線も図2(c)のように認識可能となる。

このように、ディスプレイを動かしたり揺らしたりすることにより、装置の解像度以上の表示を行なうことができる。

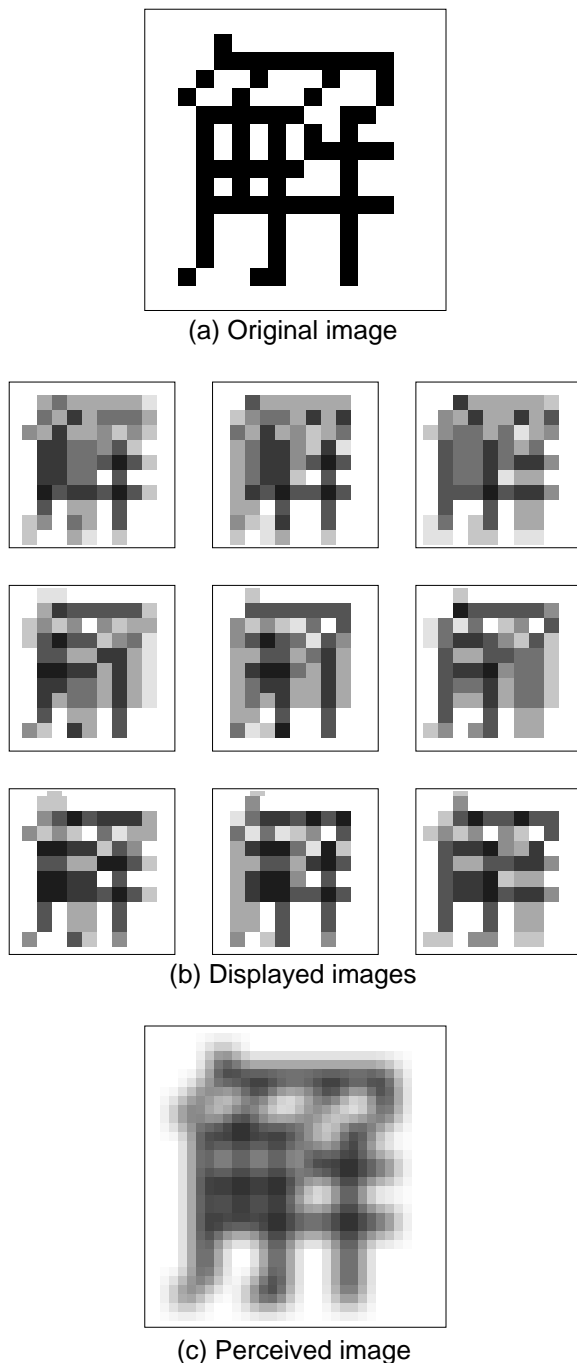


図2 ディスプレイを揺らすことによる高解像度表示

2. PixelDoublor

図1と同様の手法を用いることにより、ディスプレイを動かさずに高解像度の表示を行なうことが可能である。図1においてディスプレイを左に動かさず静止させた場合の表示を図3に示す。ディスプレイが静止している場合はボタンが変化しながら右に移動ことになるが、目が移動に追従するのでユーザの目には図1と同じように4本の線が右に移動するように見える。

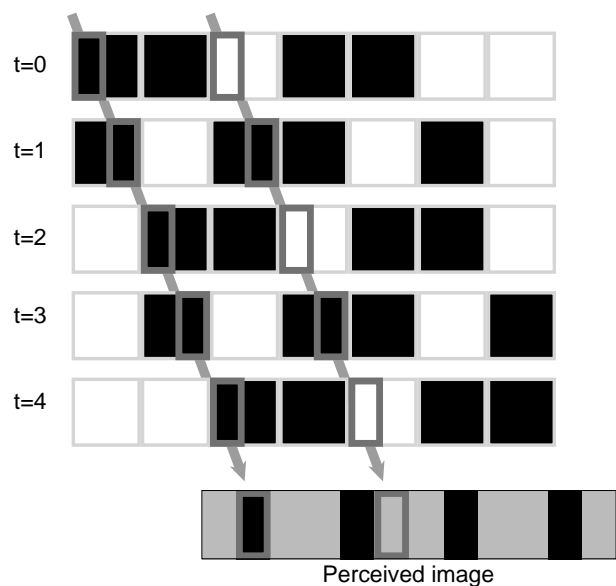


図3 4本の線が右に移動

この効果を利用して画像を移動させながら表示することにより、ディスプレイの解像度以上のピクセルをもつ画像を表示させることができるようになる。この手法をPixelDoublorと呼ぶことにする。

3. PixelDoublor 使用例

本節ではビットマップ画像の表示にPixelDoublorを応用した例を示す。2種類の画像を交互にシフトさせながら表示させることにより実際の解像度より細かい画像を表示することができる。

図4(a)に226 × 76ピクセルの原画像I0を示す。これをもとにして、2種類の小さな(170 × 78ピクセル)画像I1(b)とI2(c)を用意する。PixelDoublorは、 $t = 0$ において(0,0)の位置にI1を表示し、 $t = 1$ において(0,0)の位置にI2を表示する。また $t = 2$ において(-1,0)の位置にI1を表示し、 $t = 3$ において(-1,0)の位置にI2を表示する。このように2枚の画像をシフトしながら交互に表示することを繰り返すことにより、ユーザにはI0と同じ解像度もちかつサイズが小さな画像が見えることになる(図4)。I1およびI2もI0より少ない情報しか持っていないが、これらを交互に表示することによりユーザにはI0と同じ画像が認識される。

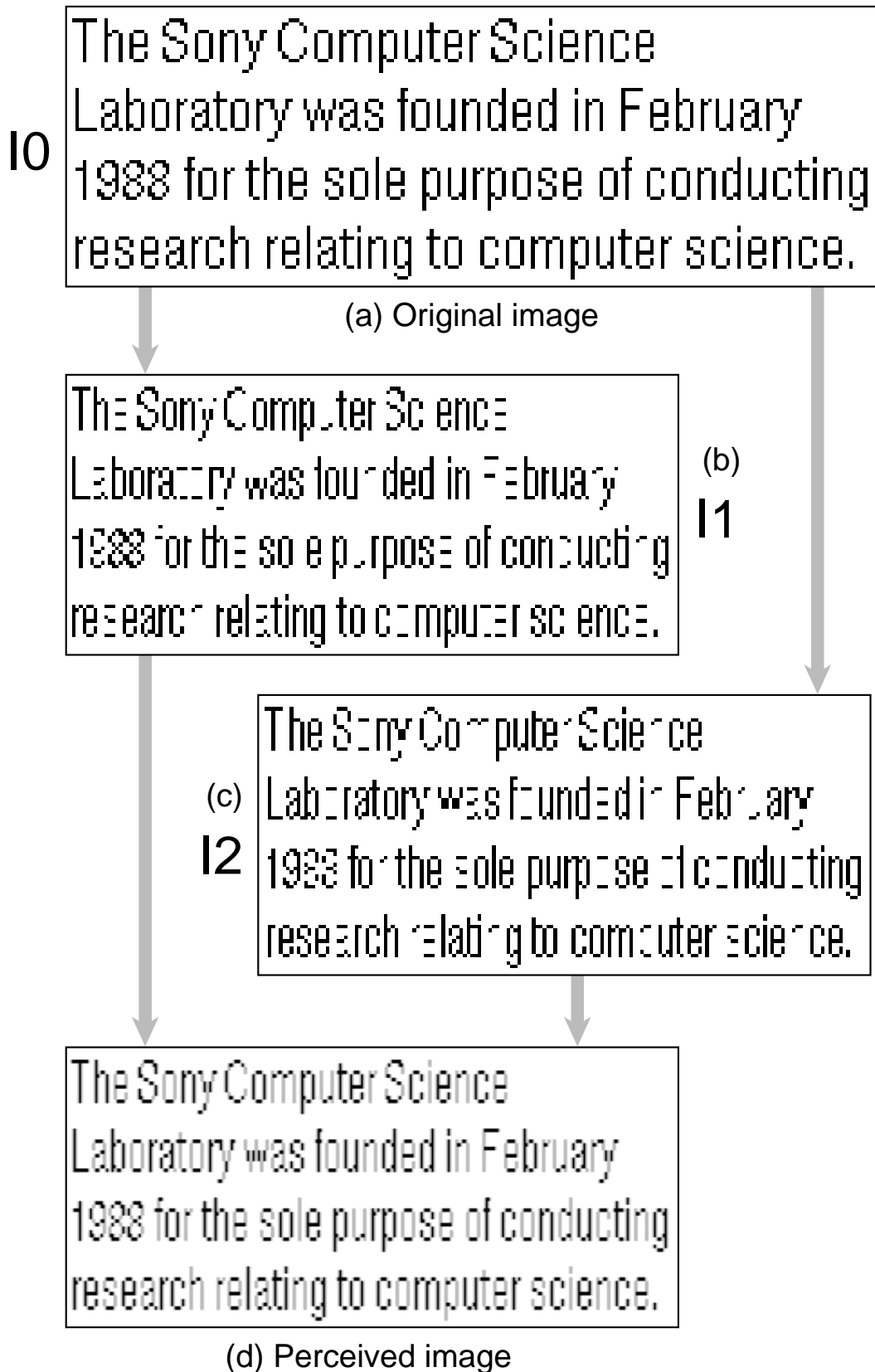


図4 2種類の画像を交互に表示させて高解像度を実現

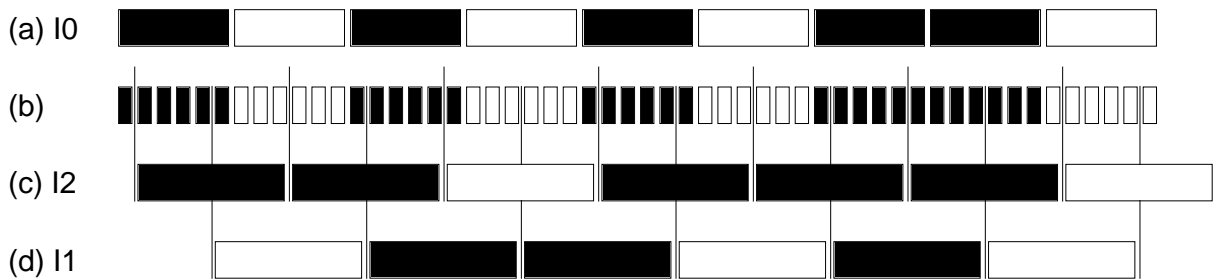


図5 原画像からのI1とI2の生成

図1に示したように、画像I1、I2は原画像I0を低い解像度の穴から覗いて得られる画像に対応している。I0からI1、I2を生成するアルゴリズムを図6に示す。

- 画像I0の各行について以下を実行
- (1) ピクセルを6倍に拡大(図5(b))
 - (2) ひとつめのギャップを始点としてピクセルを8ピクセル毎のブロックに分割する(図5(c))
 - (3) 各ブロックに対し、■の数と□の数を比較してその結果によりI2のピクセルの色を決定する
 - (4) 5番目のギャップを始点としてピクセルを8ピクセル毎のブロックに分割する(図5(d))
 - (5) 各ブロックに対し、■の数と□の数を比較してその結果によりI1のピクセルの色を決定する

図6 I1, I2作成アルゴリズム

4. 議 論

4.1 応 用

ディスプレイを動かしたり揺らしたりすることにより解像度を上げる手法は、ScrollDisplay¹⁾のようなディスプレイとマウスが融合した装置に対して直接適用可能である。ScrollDisplayではディスプレイが1ピクセル以上移動するまで表示は変わらないが、ディスプレイの位置に応じて表示を変更することにより図2のような効果を容易に得ることができる。このような拡張ScrollDisplayでは、解像度の高い表示が要求されるときにユーザが装置を揺らすという使いかたができる。

PixelDoublersでは画像は等速度で一定方向に移動する必要がある。この制限が最も大きな問題であるが、長いテキストをスクロールさせながら表示させる場合や電光掲示板のようにテキストが等速度で一定方向に動くことが好ましい場合に適用が有効であろう。

4.2 解像度の向上

PixelDoublersにおいて、画像を1タイムスライス毎に必ずしも半ドットずつ移動させる必要はなく、さらに低速に

動かすことによりさらに高解像度の画像を表示させることが可能である。1/4ドットずつ画像をシフトさせた場合を図7に示す。この場合ユーザは何段階かの輝度レベルを認識することになり半ドットずつ動かす場合よりも高い解像度が認識可能である。しかしタイムスライスが充分小さくない場合は画面がちらつくことになるため、CRTモニターなどのように表示サイクルが1/60秒程度の装置では適用がむずかしい。LEDを並べた場合のように表示サイクルを高速にできる場合に有効であろう。

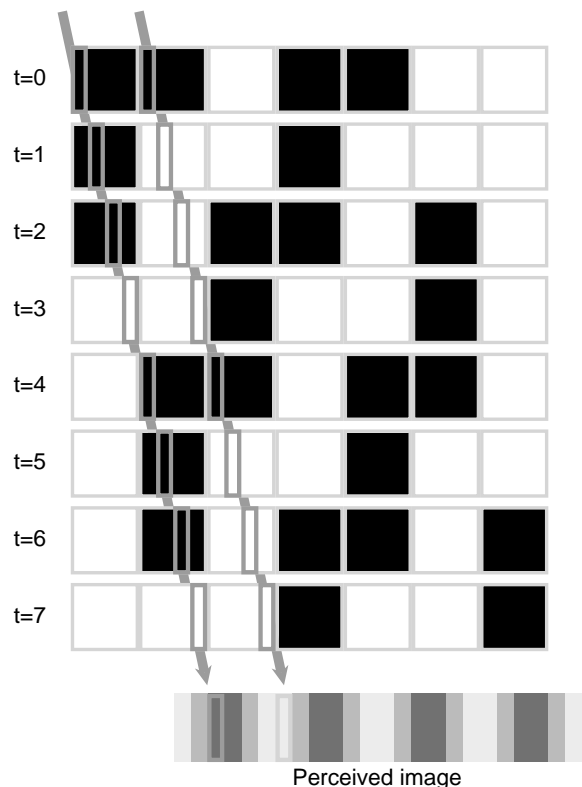


図7 画像を1/4ドットずつシフトさせた場合

4.3 関連研究

Salient Stills²⁾は通常のビデオ画像から画像のクラスを抽出して再利用するシステムであり、動画から高解像度の静止画像を抽出する点でPixelDoublersに類似している。高速に動く物体をビデオのような時間解像度の低い表示

装置で自然に表現するために、動く物体をぼやけた画像で表現するモーションブラーの手法がよく使われる。これは人間の視覚特性を利用している点で PixelDoubler に似ているが、解像度を上げる効果は無い。

5. 結 論

低解像度の表示装置を移動させたり揺らしたりすることにより高い解像度を得るシステム PixelDoubler を提案した。PixelDoubler は人間の視覚特性を利用することにより実際よりも多くのピクセルをディスプレイ上に表示することができる。この手法は PDA のような小型端末や電光掲示板など広い範囲に適用可能である。

参 考 文 献

- 1) Siio, I.: Scroll Display: Pointing Device for Palmtop Computers, *Asia Pacific Computer Human Interaction 1998 (APCHI98)*, IEEE Computer Society, pp.243–248 (1998).
- 2) Teodosio, L. and Bender, W.: Salient Video Stills: Content and Context Preserved, *ACM Multimedia'93 Proceedings*, pp. 39–46 (1993).