

*PostScriptにおけるハーフトーン技術

4X-2

*安藤 弘一 *高木 剛志 **長谷部 幹夫 **堤 義直

* 東芝ソフトウェアエンジニアリング株式会社

** 株式会社 東芝 情報通信システム技術研究所

1. はじめに

現在、米国においてDTP(Desk Top Publishing)は、1つの大きな市場を形成しており、日本においても関心が高まっている。そのDTPの重要な要素のひとつに、PDL(Page Description Language)があり、業界標準となっているのが、PostScriptである。PostScriptは、このDTPのために作られたプログラミング言語であり、強力なグラフィック機能を持っている。PostScriptによる描画はそのグラフィック機能により、指定された領域を塗り潰したり、線を引くことができ、任意のハーフトーン(中間色)も表現することができる。

本稿では、グラフィックにおけるハーフトーンを表すための技術及び、その実現方法について述べる。

2. PostScriptにおけるグラフィック機能

PostScriptのグラフィックオペレータには以下のような種類がある。

- グラフィックステートオペレータ
グラフィックの実行環境を定義する。
- コーディネートシステム&マトリックスオペレータ
座標系をその変換マトリックスを用いて操作する。
- パスコントロールオペレータ
線の形や軌道を定義する。
- ペインティングオペレータ
線の描画や塗り潰しを行う。
- キャラクタ&フォントオペレータ
フォントの作成/変更とキャラクタの描画を行う。
- デバイスセットアップ&アウトプットオペレータ
ラスタメモリと物理出力装置の関係を定める。

これらのオペレータにより線の描画、指定された領域の塗り潰し、文字の描画、イメージの描画などを行ないページ全体を描画した後、出力することができる。

また、描画できる領域を指定するクリップの機能などもある。

3. ハーフトーン

塗り潰しはカラー、黒、白、そしてグレイで行うことができる。

ピクセルが黒または白のみで表われされる出力デバイスで、中間色を表すためにはハーフトーンの技術が必要である。描画するページはハーフトーンセルの格子で定義されており、その中の1つのセルはデバイスピクセルで構成されている。

セルの大きさ、傾き、そしてハーフトーンのパターンはPostScriptのオペレータsetscreen、そしてグレイのレベルはsetgrayにより設定される。

0 setgray 1 setgray
黒 白

4. setscreenオペレータとその実現方法

(1) setscreenオペレータ

setscreenオペレータで指定するのは、セルの大きさ、傾き、ハーフトーンのパターンを決めるための手続きである。このときの手続きとは、PostScriptのオペレータで書かれているものでありセルのそれぞれのピクセルに対して値を与えるための計算式である。セルが黒からグレイになるときにこの値の小さいピクセルから順に白くしていく。一般的な手続きとしては次のようなものがある。

```
{ dup mul exch dup mul add 1 exch sub }
```

セルの大きさを大きくするほど多くの階調を表現することができるが、目が粗くなってしまう。また手続きにより、パターンを自由に変えることができ、図1のような色々なパターンの塗り潰しを行なうことができる。

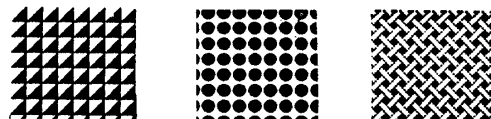


図1

Halftone technique for the PostScript

*Hiroichi ANDOU *Tsuayoshi TAKAGI **Mikio HASEBE **Yoshinao TSUTSUMI

*TOSHIBA SOFTWARE ENGINEERING Corp.

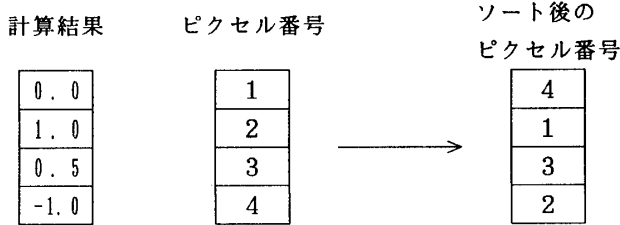
**TOSHIBA Information and Communication Systems Laboratory

* PostScriptは、Adobe Systems の商標です。

(2) 実現方法

setscreenオペレータが実行されると1つのセルの大きさを計算しピクセル数を求める。そして、それぞれのピクセルに与える値を指定された手続きにより計算する。この計算結果は、あらかじめ確保しておいた領域に保存する。これに対応してピクセルに番号を付け計算を行った順にピクセル番号を別の領域に保存する。

幾つかのピクセルを白くするとき値の小さいものから順に白くされるが、そのときの処理を簡単にするために、保存したピクセル番号を計算結果の小さいものから順にソートしておく(例1)。



例1

グラフィックステートを保存して、セルの大きさ、ハーフトーンパターンを再び変更する場合には、以前のセルの状態も保存しておかなければならない。したがって、ピクセル番号を保存するための領域は十分に確保する必要がある。例2に示すような簡単なPostScriptプログラムを実行する場合においても、300dpiのプリンタであるとするならば、1800個のピクセル番号を記憶しておかなければならない。

```
10 0 { dup mul exch dup mul add 1 exch sub }
setscreen
gsave
10 0 { add 2 div } setscreen
```

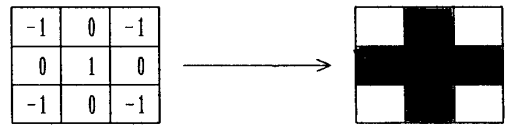
例2

数多くのハーフトーンパターンを描画する際、ピクセル番号を保存せずに、setscreenオペレータを実行すれば何度でもパターンを変えることができるが、その都度計算を行うためスピードが出ない。しかし、保存しておけば瞬時にパターンを変えることができるが、多くのパターンを保存するためには大きな領域が必要になり、これには限界が出てくる。これらの限界は、PostScriptの仕様では規定されていないため、Aのプリンタで出力できるものがBのプリンタでは出力できないということが起ってくるであろう。

5. setgrayオペレータとその実現方法

(1) setgrayオペレータ

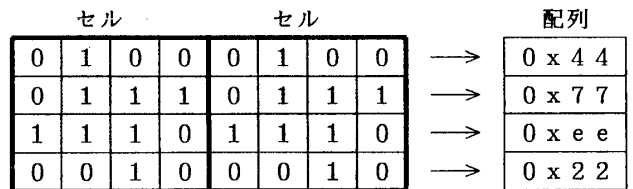
setgrayオペレータは、与えられた値によりセルの中のいくつかのピクセルを白くするかを決定する。setgrayには0.0から1.0までの値が与えられ、白くするピクセル数はfloor(セルのピクセル数×setgrayに与えられた値)により決まる。このとき白くなるピクセルはsetscreenオペレータにより設定された値の小さいものである(例3)。



例3

(2) 実現方法

setgrayオペレータはsetscreenにより昇順に並べ変えられたピクセル番号を参照しその順番でピクセルを白くする。セルのピクセルは、8bitのcharacterの配列で定義されており、最初は全てのbitが1になっている。白くするピクセル番号より、その位置にあたるcharacterのbitを0にする。この処理を繰り返してセルのピクセルに対する配列を設定する(例4)。



例4

6. おわりに

以上、PostScriptにおけるハーフトーン技術とその実現について述べてきた。

今後の課題としては、現在、日本においてもDTPについて関心が高まってきていることから、日本語PostScriptインタプリタを早急に開発することが上げられる。

参考文献

- PostScript Language Reference Manual
- PostScript Language Tutorial and Cookbook
- PostScript Language Program Design

(以上 Adobe Systems Incorporated)