ディスプレイ用高速ページ記述言語インタプリタ

4X-6

垣内 隆志　宮部 義幸　杉田 卓也
松下電器産業㈱ 情報システム研究所

1 はじめに

我々は、テキスト、静止画、音声等を統一的に扱えるマルチメディアデータベース指向電子出版システム[3]を開発している。電子出版 (Electronic Publishing) とは、紙メディア (印刷物) による出版と異なり、光ディスク等の電子メディアによる出版をさすものであり、従来の紙メディアでは実現不可能な動画、音声などの検索／利用が可能となっている。本稿では、マルチメディアデータベースの表示レイアウトシステムにおける静止画表示ツールとして開発中の、ディスプレイ用高速ページ記述言語インタプリタについて述べる。

マルチメディアを扱う電子出版システムの静止画データのフォーマットには、ラスターデータによる表現、ベクトルデータによる表現、両者を同時に使用した表現などが考えられる。しかしながら、いずれの表現を採用しても、静止画データが表示／出力デバイスの解像度などに依存している場合、オーサリングの段階で種々のデバイスを想定してメディアを供給しなければならないために、出版物としての汎用性が失われてしまうことになる。

一方、卓上電子出版 (DTP) の分野では、機器間でのドキュメントデータの高い互換性を保つために、デバイスに依存しない形式で紙面イメージを記述する種々のページ記述言語が既に開発されている。その中でも、PostScript* (PS) 言語は、現在デファクトスタンダードとしての地位を確立しようとしている言語であり、1) デバイスに依存しない形式で図形記述できる (Device Independent) こと、2) オープンアーキテクチャであること、等の特徴を有する。

我々は、これらの特徴に着目し、PS 言語に準拠した言語仕様をマルチメディアデータベースの静止画データフォーマットとして採用している。PS 言語がオープンアーキテクチャであるために、オーサリングツールとして既存のデザインツールを利用でき、出版社などのサードパーティーが静止画データを容易に供給できるようになる。 さらに、Device Independent であるために、ソフトウェアとハードウェアの共通化を図ることができる。

2 設計方針

(1) ビットマップデータの高速表示

表示用のインタプリタは、高応答性が要求される。また、ビットマップデータはスキャナ等を用いて容易に入手できるため、ソースが多数供給されることが予想されること、ディスプレイの解像度がプリンタと比較して低いことを考慮して、ビットマップデータの高速表示を重点的に行う。

(2) X-Window 上でのインプリメント

インタプリタ実現に必要なグラフィックス機能を調査することを兼ねて、標準的なグラフィックス機能を具備した X-Window 上でインプリメントする。

(3) ページメモリを利用しない。

ページメモリを利用せずに、ディスプレイのフレームメモリを最大限活用する手法を採る。

(4) カラー表現

カラーディスプレイを想定しているため、ハーフトーンによるグレースケールの表現は実現しないことにする。

3 PS インタプリタの構成

PS 言語はスタック指向の言語であり、スタック上の要素を操作する種々のオペレータを起動することで、処理を行うようになっている。オペレータは、スタック操作の他に、グラフィックスデータを作成／蓄積し、ページメモリへマッピングする操作なども行う。また、PS 言語はベクトルとビットマップイメージによる図形を扱うことができ、これらをデバイスに依存しない形式で記述できるようにするために、各図形に対してアフィン変換によるデバイスへのマッピングを行うようになっている。さらに、ベクトル図形で指定した任意の閉領域でクリッピングできるようになっている。(言語仕様の詳細については文献[1]を参照)

図1に、本システムの構成を示す。実行部は、入力プログラムを語彙解析し、辞書スタックから組み込み処理を行うオペレータを検索して起動するようになっている。なお、本システムは C 言語でインプリメントされており、フレームメモリとのインターフェースには、X-Window の C 言語サブルーチンライブラリである Xlib を用いている。

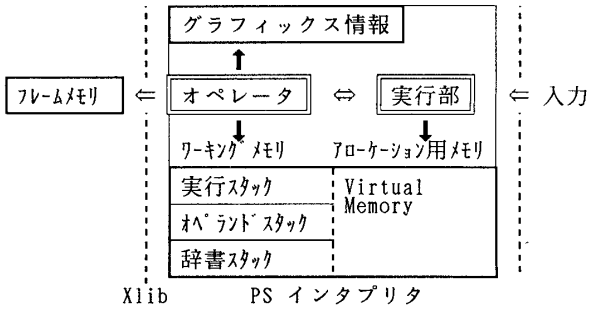

図1　システム構成

A Page Description Language Interpreter for a Color Graphics Display
Takashi KAKIUCHI, Yoshiyuki MIYABE and Takuya SUGITA
Information Systems Research Laboratory, Matsushita Electric Industrial Co., Ltd.

*PostScript は、Adobe Systems 社の商標である。

表１　辞書検索時の平均比較回数

ｵﾍﾟﾚｰﾀの順序	辞書順序		頻度順	
ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ	ﾘﾆｱｻｰﾁ	ﾊｯｼﾝｸﾞ	ﾘﾆｱｻｰﾁ	ﾊｯｼﾝｸﾞ
#1(ｲﾒｰｼﾞ)	57.590	0.7875	27.461	0.7771
#2(複合文書)	56.071	1.0525	31.676	0.9389
#3(ﾍﾞｸﾄﾙ)	52.728	1.4447	50.073	1.4119
#4(ﾍﾞｸﾄﾙ)	47.401	1.0621	34.069	1.1066
#5(ﾃｷｽﾄ)	25.118	2.1117	20.963	2.0745
#6(ﾍﾞｸﾄﾙ+ﾃｷｽﾄ)	30.736	1.4383	24.212	1.3534
#7(ｲﾒｰｼﾞ+ﾃｷｽﾄ)	58.319	0.7641	22.769	0.6345
#8(ﾃｷｽﾄ)	19.039	1.2877	15.278	1.2387
#9(ﾃｷｽﾄ)	38.821	0.9934	17.435	0.8823
#10(ﾃｷｽﾄ+ﾍﾞｸﾄﾙ)	50.441	1.6549	44.844	1.5677

4高速化処理

4．1ビットマップデータの高速表示

PS 言語は、ビットマップデータをデバイスに依存しない形式で扱うために、原画像データの全画素に対して、アフィン変換によるマッピングを行うようになっている。しかしながら、原画像の全画素について四隅の座標をアフィン変換し、その内部を塗りつぶすという方法を採用すると、(原画像の画素数×４）回の変換演算を行う必要があり、非効率的である。この問題を解決するために、まず原画像の一画素をアフィン変換し、対応するディスプレイ上のビットマップデータを得る。次に、このビットマップデータをフレームメモリ上に配置することでマッピングを行うようにする。この方法を採用すると、アフィン変換の回数は、(原画像の画素数＋２）回となる。

また、原画像データの解像度（b1）がディスプレイに表示したときの解像度（b2）と比較して高い場合、アフィン変換の結果フレームメモリ上の同一画素への書き込みが起こり、無駄な変換演算を多数行うことになる。これを避けるために、ディスプレイ上の同一画素にマッピングされるデータをアフィン変換することのないように、実行時に間引き処理を行っている。間引き処理とは、不要なビットを除去する方法で原画像の縮小を行う処理である。通常の処理コストが O(b1) であるのに対し、間引き処理を行うことで処理コストは、O(min(b1,b2)) となる。したがって、スキャナなどで読み込んだ高解像度の原画像を、ディスプレイのような低解像度のデバイスに表示する際（b1>b2）には、間引き処理の効果が大きくなる。

4．2システム辞書の順序の並べ替えとハッシング処理

PS 言語には約２５０個の組み込みオペレータがあり、キーワードとオペレータの対がシステム辞書に蓄えられている。インタプリタは、与えられたキーワードに対する組み込み処理を行う際、対応するオペレータを得るために、システム辞書の検索を行っている。オペレータの検索処理は、組み込み処理起動時に常に実行されるため、検索処理を高速化することで実行速度を大幅に向上させることができる。

本システムでは、検索処理を高速化するために、1）辞書構造の並べ替えと、2）辞書キーのハッシングを行っている。PS 言語の組み込みオペレータのうち、実際に使用されるものは全体の約３０％にすぎず[4]、システム辞書の構造を頻繁に使用されるオペレータから順に並べ替えるだけで、検索に要するキーワードの比較回数は、大幅に減少する。表１に辞書検索時の平均比較回数を示す。

表２　実行時間の比較（単位：秒）

ｻﾝﾌﾟﾙﾌﾟﾛｸﾞﾗﾑ	本ｼｽﾃﾑ	LaserWriter
#1(ｲﾒｰｼﾞ)	17	77
#2(複合文書)	22	114
#3(ﾍﾞｸﾄﾙ)	29	160
#4(ﾍﾞｸﾄﾙ)	60	425
#5(ﾃｷｽﾄ)	18	51
#6(ﾍﾞｸﾄﾙ+ﾃｷｽﾄ)	9	40
#7(ｲﾒｰｼﾞ+ﾃｷｽﾄ)	13	98
#8(ﾃｷｽﾄ)	31	222
#9(ﾃｷｽﾄ)	14	63
#10(ﾃｷｽﾄ+ﾍﾞｸﾄﾙ)	68	125

5評価

5．1実行時間の比較

本システムのパフォーマンスを評価するために、１０個のサンプルプログラムについて、Sun3/280 の X-Window 上でインプリメントした本システムと LaserWriter で実行した場合の実行時間の比較を表２に示す。

ビットマップイメージやベクトル図形が主体となったドキュメントについては、５～６倍のパフォーマンスが得られたのに対し、テキスト主体のドキュメントについては、３～４倍のパフォーマンスしか得られていない。これは、LaserWriter がシステム起動時に予めフォントキャッシュへのフォントのローディングを行っていること等の特別なフォント処理機構を持っていることに起因していると考えられる。

5．2移植性

本システムをインプリメントするのに使用したX-Windowのグラフィックス機能は、1）多角形領域内部の塗りつぶし機能、2）直線描画機能、3）ビットマップデータをフレームメモリ上へ転送する機能、の３つの基本的な機能のみである。したがって、これらの機能が利用できる環境への移植は、容易に行うことができる。

6まとめ

本稿で述べた、PS 言語の Device Independent でオープンアークテクチャな考え方は、マルチメディア記述言語に必須の要素になると考えられる。また、将来ディスプレイ用インタプリタが専用ハードウェアの下で高速に実行できるようになれば、画像を扱える電子メールや、プレゼンテーションシステム等の多くの応用分野が開拓されていくと予想される。このような観点に立って、今後は、ディスプレイ用ページ記述言語インタプリタに必要な機能を言語仕様の拡張も含めて検討し、実現していく予定である。

謝辞
本システムの原型となった、プリンタ用 PS インタプリタ[2]を開発し提供して頂いた、大阪大学基礎工学部情報工学科宮原研究室の下條真司助手、嶋谷朗氏に感謝致します。

参考文献

1) Adobe Systems Inc.: PostScript Language Reference Manual, Addison-Wesley (1987).
2) 嶋谷 他: UNIX 上で動く PostScript ドライバの開発, 第32回情処全大, 3E-7 (1987).
3) 橋野 他: 電子出版指向データベースシステム－基本構想－, 第37回情処全大 (1988).
4) 手塚 他: Desk Top Publishing と Page Description Language について, 第36回情処全大, 2S-4 (1988).