

情報処理教育のための図形表示システムの実現†

岡田 稔†† 熊谷 毅††
三輪 和久†† 櫻井 桂一††

本論文では多人数情報処理教育を指向したベクトル図形表示システムの実現について述べる。著者らの所属する機関では、情報処理教育のための計算機システムとして汎用大型計算機、ワークステーション、そしてパーソナルコンピュータから構成される分散処理システムを使用している。ここで述べる図形表示インタフェースはパーソナルコンピュータ上に割り込み型の常駐プログラムとして動作するソフトウェアであるため、パーソナルコンピュータを汎用大型計算機やワークステーションの端末エミュレータとして動作させる場合と、スタンドアロンで使用する場合の両方で動作するという特長を持つ。また、最適化された DDA 直線発生、破線発生などにより高速表示が可能である。本システムを利用することにより、専用のグラフィック・ディスプレイを必要とせず、低コストな図形表示システムが実現できた。

1. はじめに

情報処理教育のための計算機環境では、計算機とユーザの会話環境の整備が重要である。特に計算機言語の教育を行う場合などでは、グラフィックス機能は計算結果の可視化はもちろんのこと、受講者の興味をそそり、意欲の向上のために欠かすことのできない機能の一つであろう。

情報処理教育システムとして最近盛んに使用されるようになった UNIX^{*1)} オペレーティングシステムでは、GKS²⁾、plot ライブラリに代表される汎用図形作画インタフェースが提供されている。これらのインタフェースは機種に依存しない図形命令を出力すると共に、既存の汎用グラフィックターミナルをオンライン駆動する図形命令も出力することが可能である。また、GKS は UNIX のみならず多くの計算機システムに装備されている。このようにホスト計算機側の図形作画機能が充実しているにもかかわらず、他方では、それらの計算機システムを使用するための端末は、コスト削減の点から非図形端末を使用しているのが実状である。

本論文では、情報処理教育システムを指向したベクトルグラフィックス図形表示インタフェースの概要とその実装について述べる。本インタフェースは 4014-1 型グラフィックターミナル³⁾ (以下、4014-1) の図形

命令をターゲットとして、ソフトウェアエミュレーションの形でパーソナルコンピュータ上に実装される。

大学等の一般情報処理教育のための計算機システムとして、汎用大型計算機、ワークステーションおよびパーソナルコンピュータ端末を用いた分散処理システムが多く採用されている。ここで述べるインタフェースはパーソナルコンピュータ上でソフトウェア割り込みによる常駐型プログラムとして動作するソフトウェアであるため、不特定の端末エミュレーションソフトウェアの下で動作させることが可能である。また、本インタフェースを実装するパーソナルコンピュータを、汎用大型計算機やワークステーションの端末として動作させる場合と、スタンドアロンで使用する場合のいずれにおいても動作するという特徴がある。さらに、パーソナルコンピュータが本来持っているラスターグラフィックス表示機能を利用するため、専用のグラフィック・ディスプレイやグラフィックターミナルを必要とせず、低コストなパーソナル・グラフィックス環境の実現が可能である。

従来より、多人数を対象とした情報処理教育における問題点はいくつか指摘⁴⁾されているものの、昨今の計算機システムの多様化に伴って新しい問題点^{5),6)}が浮上している。以下、2章ではまず情報処理教育における図形表示に関する問題点を明らかにし、3章では情報処理教育システムの現状について述べる。さらに4章ではパーソナルコンピュータを用いた図形表示システムの実装とその能力評価について報告する。

† Implementation of Personal Graphics System for Computational Lectures by MINORU OKADA, TAKESHI KUMAGAI, KAZUHISA MIWA and KEIICHI SAKURAI (Education Center for Information Processing, Nagoya University).

†† 名古屋大学情報処理教育センター

* UNIX は、米国 AT&T ベル研究所の登録商標である。

2. 情報処理教育における図形表示機能の問題点

情報処理教育では、実社会の計算機利用の現場における多種多様な計算機構成、ソフトウェア構成に対応できる環境を提供する必要がある。多くの企業、研究機関等では汎用大型計算機を始めとして、UNIX 環境のワークステーション、MS-DOS* 等を用いた汎用 OS 環境のパーソナルコンピュータが有機的に結合され、OA, FA, 研究開発の環境が整備されている。一方、大学等の情報処理教育センターにおいては、このような多様な実社会の現状に対応するとともに、学生数の増加に伴った適切な情報処理教育環境を整備することが望まれる⁴⁾⁻⁶⁾。

パーソナルコンピュータを端末として通信回線によってセンタホスト計算機を利用する場合には、ホスト計算機が直接、端末のグラフィックス画面を制御することはできない。通常、パーソナルグラフィックス環境を実現するためには高価な専用のグラフィックターミナルを使用するか、図形表示機能を併せ持った端末エミュレータのもとでパーソナルコンピュータを使用する必要がある。

しかし、Ethernet⁷⁾ などのネットワーク環境や、複数のオペレーティングシステムに対応する必要がある教育用の計算機システムでは、特殊化された端末エミュレータの使用を強いられており、その端末エミュレータが図形表示機能を併せ持っているという保証は全くない。したがって、何らかの形で文字表示のための端末エミュレータと、図形表示のためのソフトウェアを分離できると都合がよい。

一方、利用者の立場で考えると、図形処理を行わない場合は通常のテキスト端末、図形処理を行う場合は図形専用端末というように図形処理を行うか否かで使用端末を変更するのをもたずらに利用者を混乱させる恐れがある。図形専用端末や、X Window** などをを用いて図形処理を行うのに対して、ユーザが親しみやすいパーソナルコンピュータで図形処理を行うのは意義のあるものである。

このようなことから、情報処理教育を指向した図形表示機能に対する条件として、次のような項目が挙げられる。

1. 汎用の図形作画ライブラリパッケージの出力に

* MS-DOS は米国 Microsoft 社の登録商標である。

** X Window 標準の仮想端末機能である xterm には、本研究で用いている 4014-1 図形表示機能がある。

対応する。

2. コスト削減のためには、文字端末とグラフィックス専用端末を別個に用意するのではなく、単一端末で双方の機能を実現するのが望ましい。

3. 情報工学科などにおける情報処理専門教育に対して、非専門学科学生を対象とする一般情報処理教育では、親しみやすい端末を使用するのが望ましい。

4. 既存の端末エミュレータプログラムから独立した形態が望ましい。

本論文で述べる図形表示システムはこれらの要求を満たすものである。

3. システム構成

現在の名古屋大学情報処理教育センターにおける計算機システムのハードウェア構成は以下のとおりである (図 1)。

[ハードウェア構成]

ホスト計算機: FACOM M 760/8

端末: FMR-60 HX 20 MB 160 台

端末用レーザプリンタ 40 台

ネットワークプリンタ 4 台

ワークステーション: SUN 3 22 台

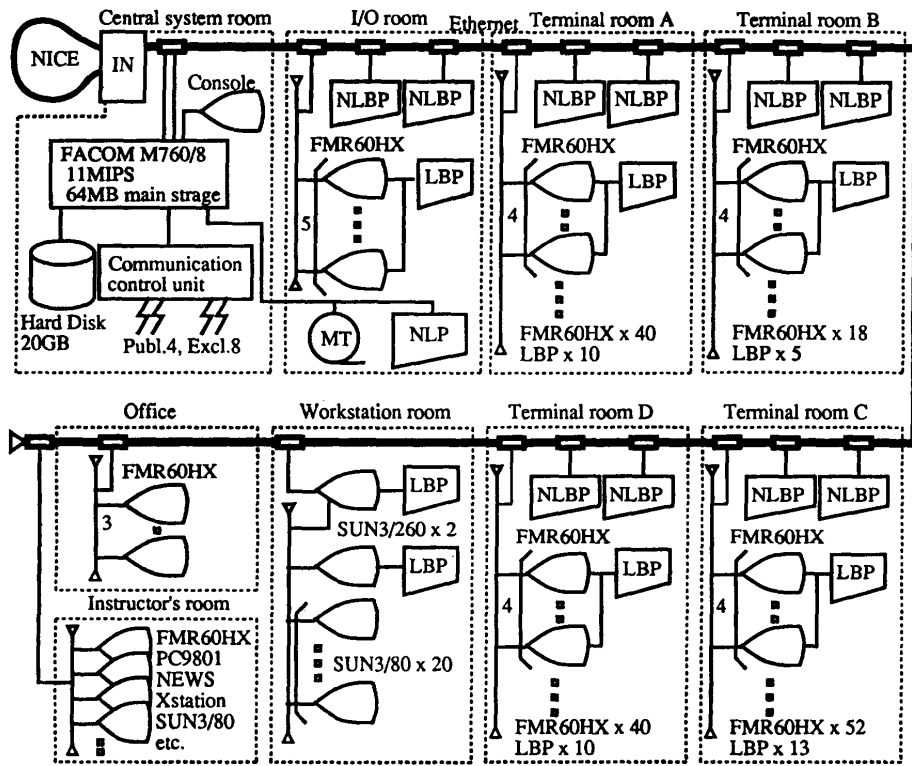
ホスト計算機 M760/8 は、専用 OS (MSP*) と汎用 OS (UTS*/UNIX System V 系) のデュアル OS のもとで動作している。ユーザが使用する端末は専用端末とはせず、Ethernet により LAN 接続されたパーソナルコンピュータを使用している。パーソナルコンピュータは、MS-DOS, UTS 用端末エミュレータプログラムを含む TCP/IP⁸⁾ ソフトウェア群、および MSP 用端末エミュレータのもとで動作している。ここで使用している MSP 用端末エミュレータは、MSP 独自のプロトコルに基づく図形表示機能を有している。また、UTS 用端末エミュレータは、パーソナルコンピュータに内蔵される LAN アダプタイタフェースに固有なソフトウェアであり、図形表示機能は有していない。

4. 図形表示機能の実現

4.1 図形作画ライブラリ

図形作画ライブラリとは、グラフィックターミナルを駆動する図形命令を出力するためのソフトウェア群である。ユーザは文字列として構成されている図形命令をグラフィックターミナルに送ることにより、CRT

* MSP, UTS は富士通のオペレーティングシステムの名称である。



NICE : Campus network of Nagoya University.
 IN : Optical network node.
 NLBP : Network laser beam printer.
 NLP : Japanese-printable line printer.
 LBP : Laser beam printer.

図 1 名古屋大学情報処理教育センター計算機システム構成図
 Fig. 1 A computer system at Education Center for Information Processing, Nagoya University.

上に図形を表示することができる (図 2)。

UNIX オペレーティングシステムを始めとして、GKS 系の汎用図形作画ライブラリが充実してきており、また、GKS 系の図形命令セットを既存のグラフィックターミナル用図形命令セットに変換するフィルタ群も充実している。既存のグラフィックターミナルの中でも、4014-1 は古典的でありながら、事実上現在の図形作画ライブラリのための標準端末とされている。そこで今回作成する図形表示システムの命令体系として 4014-1 の命令体系を採用した。これにより、GKS, plot を始めとする多くの汎用図形作画ライブラリに対応できる。

4.2 図形表示機能の実装

本図形表示システムは MS-DOS パーソナルコンピュータとその上で動作する制御プログラムによって構成されている。以下、パーソナルコンピュータで動作する制御プログラムを図形表示インタフェースとい

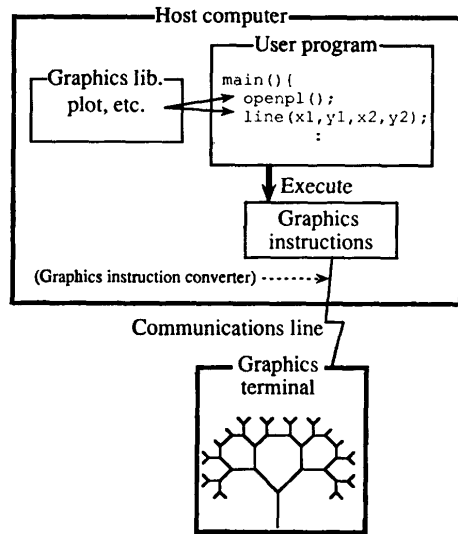


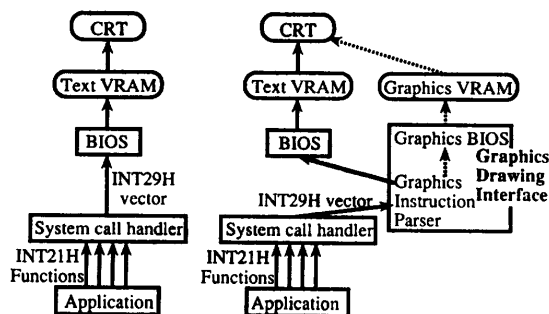
図 2 図形作画ライブラリと図形表示
 Fig. 2 Graphics libraries and a method of drawing figures.

う。図形表示インタフェースは主記憶空間に常駐⁹⁾してソフトウェア割り込みによって駆動されるため、端末エミュレータなどの主アプリケーションソフトウェアの動作にはほとんど影響を与えず、共存が可能である。

4.3 MS-DOS とのインタフェース

MS-DOS では、アプリケーションソフトウェアと OS のインタフェースとしてシステムコール¹⁰⁾と呼ばれるソフトウェア割り込みを使用している。画面に対するテキスト文字出力においても標準化のためにシステムコールを利用している端末エミュレータが多い。この文字出力のためのシステムコールは複数存在するが、それらはすべて INT 29 と呼ばれる非公開な内部的ソフトウェア割り込みを経由している。これは筆者が調査した限り、すべての MS-DOS パーソナルコンピュータにおいて同様であった。INT 29 割り込みが生じると、通常（図形表示インタフェースが常駐していない状態）は文字描画のための BIOS に制御が移る。この動作を、割り込みベクトルが BIOS に向いている、という。

一方、本インタフェースを常駐させると、先の INT 29 割り込みベクトルは書き換えられ、インタフェース内の処理部に向けられる。処理部の後部には元の BIOS に制御を移すための命令が置かれている。常駐後、INT 29 割り込みが生じるとインタフェース内部の処理部を経由して BIOS が動作することになる。この処理部では後に述べる図形命令の字句解析、構文解析、図形命令の実行が行われ、図形命令でない文字は BIOS に送られる。



(a) フッキングなし (b) フッキングあり
(a) Without hooking. (b) With hooking.

図 3 割り込みのフッキングによるインタフェースの実現
Fig. 3 Implementation of graphics interface by hooking of interrupt.

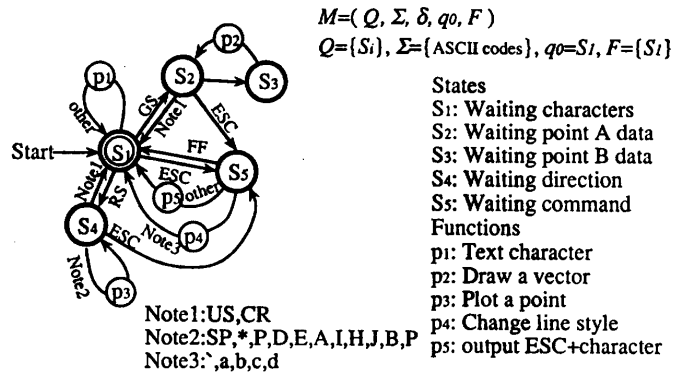


図 4 図形命令パーザの状態遷移図
Fig. 4 State transition diagram in the graphics instruction parser.

$$M = (Q, \Sigma, \delta, q_0, F)$$

$$Q = \{S_i\}, \Sigma = \{\text{ASCII codes}\}, q_0 = S_1, F = \{S_i\}$$

States

- S1: Waiting characters
- S2: Waiting point A data
- S3: Waiting point B data
- S4: Waiting direction
- S5: Waiting command

Functions

- p1: Text character
- p2: Draw a vector
- p3: Plot a point
- p4: Change line style
- p5: output ESC+character

これらの処理過程の様子を図 3 に示す。なお、常駐時に元の INT 29 割り込みベクトルを退避しておき、開放時に復元させれば図形表示インタフェースの影響を残すことなく終了させることができる。

4.4 図形命令の解析

図形命令は、1 文字または複数文字から構成される文字列であり、通常のテキスト表示文字に混入されてホスト計算機から送出される。このため、端末側ではこれらの図形命令文字と非図形命令文字の分離を行いながら、図形命令の解析および実行を行う必要がある。4014-1 図形命令の仕様に基づいて作成した状態遷移図を図 4 に示す。本図形表示インタフェースでは、この状態遷移に基づいて図形命令の構文解析、字句解析を行っている。

なお、本図形表示インタフェースでは 4014-1 図形命令を使用したことによって、一部の文字、文字列は図形命令として優先的に解釈され、テキスト画面に対して反映させることはできない。これらは状態遷移図中の S1 から出る GS*, RS* および ESC* で始まる文字列である。ただし、ESC で始まる文字列のうち、S1 に終端されない文字列は図形命令ではないと判断され、テキスト画面に反映される。GS, RS は多くの端末制御シーケンスでカーソル移動命令とされており、UNIX 等の端末として使用する場合に不都合が起きるが、GS, RS を使用せずともカーソル移動が実現できるため、この不都合は容易に回避可能である（付録 1 参照）。

4.5 DDA による線分発生と破線発生

ラスタ型ディスプレイを用いて線図形を表示させるときに基本となる処理は、1 画素の描画とそれを用い

* GS, RS, ESC は ASCII コード制御文字の名称である。

たデジタル線分発生である。ここではデジタル微分解析 (Digital Differential Analyser) に基づく Bresenham の方法¹¹⁾によるベクトル・ラスタ変換を用いてデジタル線分を発生している。

ところで、単純な直線分や鎖線を始めとして多くの種類の破線を選択的に発生させるためには、複雑な処理を必要とするためにコストが大きくなりがちである。本図形表示インタフェースでは以下に述べるビットパターン法によって5種類の破線を発生させている。

まず、描画位置に基づき整数値を決定する。実際にはこの値として DDA 内で使用されるカウンタの下位4ビットを用いている。2進数表現された整数値のビットパターンの複数の特定ビットの論理和が真のときに当該画素を表示し、偽のときは非表示とする。論理和の対象となるビットを変更すると、それに応じて破線パターンが変化する。この処理を DDA での逐次的な画素描画と併用することによってさまざまな線種の破線が得られる (図 5)。

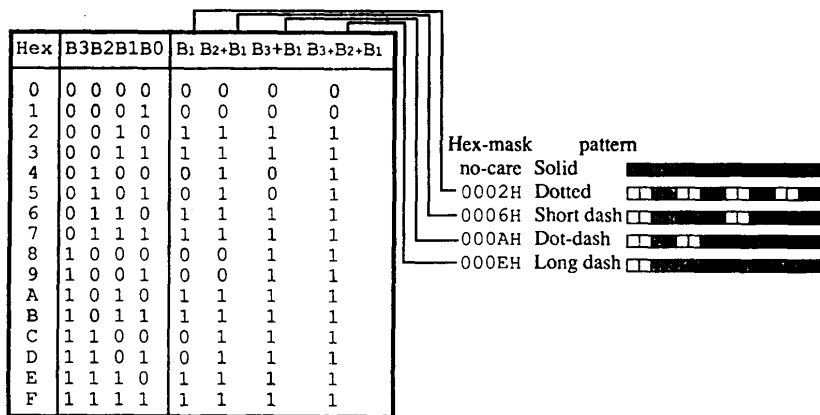
この破線発生方法は先に述べた DDA による線分発生方法と親和性が良く、DDA アルゴリズム中に容易に挿入可能である。また、実際の処理プログラムにおける破線判定では、ワード単位の並列論理演算を用いて行われるために低コストとなっている。なお、以上の DDA 直線発生、破線発生およびウィンドウクリッピング¹²⁾はすべて整数演算に基づくものである。これにより高級言語で実装したにもかかわらず、高速表示

および、浮動小数点演算ルーチンの排除によるオブジェクトの小型化が可能となっている。

4.6 図形表示実験と描画能力評価

本図形表示インタフェースを用いて描画した図形を図6に示す。図6(a), (b)はよく知られている再帰曲線である。図6(c)は本図形表示インタフェースで生成可能な破線の線種をすべて含む図形である。表1にはそれぞれの図形における表示経過時間を示す。表中、左欄 (Gross と表記) は Ethernet (TCP/IP, 10 Mbps) で接続されたホスト計算機 M 760/8 上で図形命令を出力するプログラムによりオンラインでリモート表示したときの経過時間である。右欄 (Net と表記) は同プログラムにより出力された図形命令をファイルとしてパーソナルコンピュータのハードディスク上に置き、MS-DOS の type コマンドにて最も理想的な状態でローカル表示したときの経過時間である。Gross に対して Net では概ね2倍の描画速度であると考えられる。

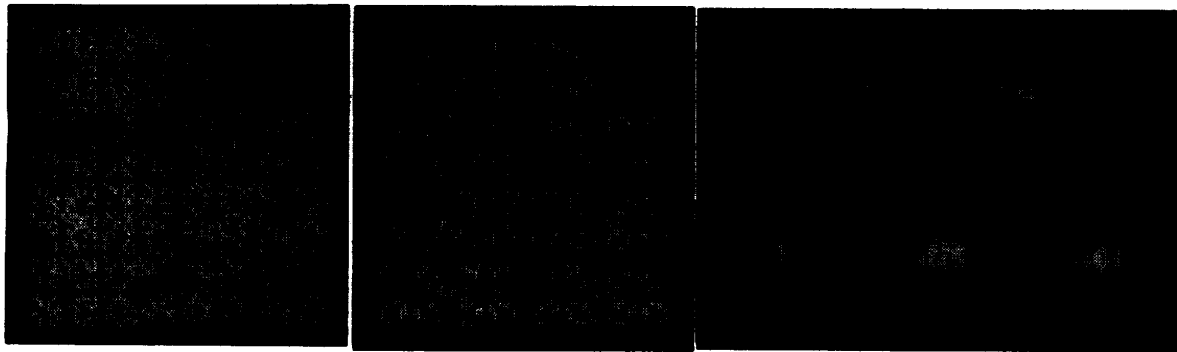
つぎに、(1)再帰レベル数を変化させた Sierpinski 曲線、(2)一様乱数によって発生された線分、のそれぞれの Gross 条件における描画ベクトル数と所要時間の関係を図7に示す。(1)の Sierpinski 曲線では1再帰ごとにベクトルの長さが短くなるが、ベクトル数が4倍になる。このために個々の画素の描画処理と比較して、図形命令の解析とクリッピングが支配的となり、図中の破線 (ベクトル数に対して比例) に漸近する。したがって、この破線は本図形表示インタフェー



(a) DDA カウンタの LSB4 ビットに対する論理演算の真理値表
(a) Truth table of Boolean operations for 4-LSBs of DDA counter.

(b) DDA カウンタに対するマスクと、生成される破線
(b) The masks for DDA-counter and produced line patterns.

図 5 ビット演算による高速破線発生
Fig. 5 Fast broken line generation by bit operations.



(a) 5レベルのヒルベルト曲線 (b) 4レベルのシェルピンスキ曲線 (c) 破線
 (a) Hilbert curves consist of 5 recursive levels. (b) Sierpinski curves consist of 4 recursive levels. (c) Broken lines.

図6 描画実験例
 Fig. 6 Examples for drawing.

表1 表示所要時間
 Table 1 Drawing time.

TIME [sec]	PC9801 VX21		FMR 60 HX	
	Gross	Net	Gross	Net
Hilbert Fig. 6(a)	23	11	11	5
Sierpinski Fig. 6(b)	25	11	11	5
Broken lines Fig. 6(c)	3.5	2	2	1

スの能力限界 ($R_1=1240[\text{vector/sec}]$) を示しており、また、(2)の一樣乱数による線分の結果は平均能力 ($R_2=50[\text{vector/sec}]$) を示していると考えられる。

さて、4014-1 図形命令でベクトルを描画する場合、座標値の差分情報により最適化され、図形命令は可変長となるが、最悪の場合でも1ベクトル当たり5バイトである。ベクトル描画速度 $R[\text{vector/sec}]$ から、これに必要な通信速度 $S[\text{bps}]$ を逆算すると、

$$S=5 \cdot 8 \cdot R[\text{bps}] \quad (1)$$

である。先に得られたベクトル描画速度 R_1, R_2 に対する必要通信速度 S_1, S_2 は、

$$S_1=49,600[\text{bps}]$$

$$S_2=2,000[\text{bps}]$$

となる。すなわち、Ethernet (10 Mbps) や、シリアル回線 (例えば 9600 bps) にて使用する場合は、平均的な (ベクトル長の分布が片寄っていないという意味での) 図形にはボトルネックとなる可能性がある。しかしながら、パーソナルコンピュータに付属の Ethernet インタフェースでのオーバーヘッド (実際には無視できない) や、ホスト側プログラムでの図形命令送出速度とを考え合わせると、Ethernet やシリアル回線利用での実通信速度は大幅に下回る。実際、表1におい

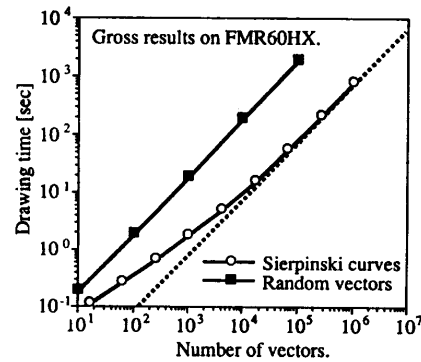


図7 描画ベクトル数と描画時間の関係
 Fig. 7 Relationship between a number of vectors and drawing time.

て、本来同程度の処理速度を持つ PC9801 VX と FMR 60HX で2倍ほどの差が出ているのは、描画処理速度以外の要因が大きいことを示している。実際には、初心者教育用としては十分に満足できる描画速度であるとの評価を得ている。

一方、[付録2] に示すように本図形表示インタフェースのプログラムサイズは非常に小さなものである。主記憶空間の小さいパーソナルコンピュータにあっても、本インタフェースの常駐下における主アプリケーションとの共存はほとんど問題ない。

5. おわりに

本稿では、情報処理教育システムを指向したベクトル図形表示システムの概要と、そのパーソナルコンピュータ上への実装について述べた。著者らの所属する機関では、情報処理教育のための計算機システムとして汎用大型計算機、ワークステーション、およびパー

ソナルコンピュータから構成される分散処理システムを使用している。このような分散処理環境においては、分散処理を指向したグラフィックス環境が望まれる。パーソナルコンピュータ上に実装された本図形表示インタフェースは次のような特徴を持っている。

(1) 専用のグラフィック・ディスプレイやグラフィックターミナルを必要とせず、低コストなベクトル・グラフィックス環境を実現できる。

(2) 割り込み型の常駐プログラムであるため、端末エミュレータなど、不特定の主アプリケーションの起動中でも動作する。

(3) 4014-1 型グラフィックスターミナルの作画命令をターゲットとしたため、多くの図形作画ライブラリに対して汎用性がある。

(4) 整数演算に基づく DDA による直線発生、最適化された破線発生、ウィンドウクリッピングにより、高速な図形表示が可能である。

現在、図形表示インタフェースとしての基礎的機能が完成している。カラー図形の表示およびシステムコールを使用していない端末エミュレータへの対応が課題となっている。

謝辞 本研究に当たり、ご指導いただいた名古屋大学情報処理教育センター長・鬼頭幸生教授、ならびに、ご助言を賜った同工学部・鳥脇純一郎教授に深謝する。なお、本研究の一部は文部省科研費（一般研究 (B) No. 01450270) による。

参 考 文 献

- 1) Ritchie, D. M. and Thompson, K.: The UNIX Time-Sharing System, *CACM*, Vol. 17, No. 7, pp. 365-375 (1974).
- 2) International Standard Organization: Computer Graphics—Graphical Kernel System (GKS) Functional Description, International Standard number 7942, ISO, Switzerland (1985).
- 3) 4014-1 型グラフィック・コンピュータ・ターミナルインストラクション・マニュアル, ソニー・テクトロニクス.
- 4) 米沢宣義, 志村 武, 南 敏: 多人数初心者向きプログラミング教育システムの基本設計について, *情報処理学会論文誌*, Vol. 27, No. 1, pp. 96-102 (1986).
- 5) 江口三代一, 猪飼秀隆, 坂田彰一郎, 朱雀保正: パーソナルコンピュータによる情報処理教育と CAI のためのシステム, *情報処理学会論文誌*, Vol. 31, No. 11, pp. 1556-1564 (1990).
- 6) 情報処理教育に関する研究会各論文, 国立大学情報処理教育センター協議会 (1990).
- 7) Metcalfe, R. M. and Boggs, D. R.: ETHER-NET: Distributed Packet Switching for Local Computer Networks, *CACM*, Vol. 19, No. 7, pp. 395-404 (1976).
- 8) Postel, J. B.: Internetwork Protocol Approaches, *IEEE Trans. Commun.*, COM-28, pp. 604-611 (1980).
- 9) 星野 操: MS-DOS レジデントプログラム入門, 技術評論社 (1989).
- 10) MS-DOS 3.1 プログラマーズリファレンスマニュアル, Microsoft & NEC (1985).
- 11) Bresenham, J. E.: Algorithm for Computer Control of Digital Plotter, *IBM Syst. J.*, Vol. 4, No. 1, pp. 25-30 (1965).
- 12) Sproull, R. F. and Sutherland, I. E.: *A Clipping Divider*, pp. 765-775, FJCC, Thompson Books, Washington, D. C. (1968).

付 録 1

ここでは 4.4 節の RS, GS 問題の回避方法の具体例を述べる。例えば、筆者らが使用している FMR60HX 用の `terminfo` のエントリの一部として次のような記述がある。

```
cub 1=^], cuu 1=^^,
```

`cub 1` はカーソル左移動, `cuu 1` はカーソル上移動であり, `^]` は GS (1 DH), `^^` は RS (1 EH) を示している。このままではホスト側からカーソル移動命令が送出されると、画面制御の前に、本論文の図形インタフェースが図形命令として解釈実行してしまう。そこで、前述のエントリ部分を次のようにエスケープシーケンスで記述することによって全く不都合なく動作している。

```
cub 1=\[D, cuu 1=\[A,
```

付 録 2

[図形表示インタフェースの仕様]

コマンド名	tekplot. exe
動作機種	FMR シリーズ (1120×750 pixels) PC9801 シリーズ (640×400 pixels)
動作 OS	MS-DOS 2.1 以上
起動方法	tekplot [-r][<code>-c</code>][<code>-s</code> <figure>] [<file>]
起動オプション	-r: プログラムの解放 -c: グラフィック画面消去 -s: (x, y) 座標値の右シフトビット数

ファイルを指定するとそのファイルの内容を図形命

令として図形表示する。ファイルを指定しない場合はメモリに常駐し、MS-DOSの文字出力システムコールをフックして図形端末エミュレータとして動作する。

オプション [-r] は、常駐しているインタフェースを開放する。オプション [-c] は画面を消去する。オプション [-s] とそれに続く1桁の数字は座標値の右シフトビット数を指定する。これはパーソナルコンピュータの表示ドット数よりも大きな仮想座標に基づいて描かれた図形の表示時に有効である。4014-1 図形端末の仮想座標系は (4096×4096) の大きさを持つ。もちろん、1ビット右シフトは仮想座標値を1/2にして表示する。

記述言語 C

処理系 Turbo C 2.0

(FMR 用はクロスコンパイルによる)

オブジェクトサイズ

PC9801 シリーズ用: 13,270 Bytes

FMR シリーズ用: 13,384 Bytes

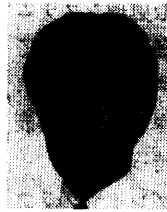
常駐時占有メモリ

PC9801 シリーズ用: 21,744 Bytes

FMR シリーズ用: 21,856 Bytes

(平成3年8月5日受付)

(平成3年11月5日採録)



岡田 稔 (正会員)

昭和59年名古屋工業大学工学部電気卒業。平成元年名古屋大学大学院情報工学専攻博士課程修了。昭和63年・平成元年度日本学術振興会特別研究員。平成2年名古屋大学情報処理教育センター助手。この間、デジタル画像処理、計算幾何学、コンピュータ・グラフィックスとその医用応用に関する研究に従事。電子情報通信学会、日本ME学会各会員。



熊谷 毅 (正会員)

昭和51年名古屋大学工学部電気卒業。昭和53年同大学院修士課程修了。昭和54年宇都宮大学工学部情報助手。昭和63年名古屋大学情報処理教育センター助手。現在に至る。工学博士。計算機アーキテクチャ、プログラミング言語処理系などに興味を持つ。電子情報通信学会、ACM、IEEE Computer Society 各会員。



三輪 和久

昭和59年名古屋大学工学部応用物理卒業。平成元年同大学院情報工学専攻博士課程修了。工学博士。現在、名古屋大学情報処理教育センター助手。認知科学、人工知能、教育工学などの研究に従事。特に、帰納的推論、仮説形成検証過程、プログラミングにおける知識獲得等に興味がある。電子情報通信学会、人工知能学会、日本認知科学会、The Cognitive Science Society 各会員。



櫻井 桂一

昭和48年名古屋大学工学部電子卒業。昭和50年同大学院修士課程修了。名古屋大学プラズマ研究所助手。同大工学部講師を経て、昭和63年同大情報処理教育センター助教授。工学博士。プラズマ物理、核融合の研究に従事。日本物理学会、電気学会、プラズマ・核融合学会各会員。