

テレビ走査型計算機グラフィック表示装置を 利用した図形編集システム GEST†

平石裕実†† 青木 豊†† 矢島 脩 三††

本論文は、利用者が意図した図形を容易にかつ迅速に作成、編集することのできる図形編集システム GEST (A Graphic Editing System Using a Television-Based Computer Graphics System) について述べている。

GEST は筆者らが開発したテレビ走査型計算機グラフィック表示装置 DIGRAPH を利用するもので、ライトペンによる会話的なカラー図形の編集ができるという従来にない特色を有している。また、ひとまとまりの図形に名前を付けて利用者定義図形として登録すると、利用者定義図形単位の追加や削除が行える。このような図形編集機能や図形登録機能は、従来の図形編集システムにおいて、リスト構造のような複雑なデータ構造を用いて実現されてきた。しかし、本論文では、DIGRAPH の特徴であるライトペンによるカラー図形の同定機能を有効に利用することで、これらの機能が GEST において簡単な構成と処理により実現できることを示した。すなわち、GEST は構成の単純なディスプレイファイルと利用者定義図形のための表だけを用いて、これらの機能を容易に実現している。また、GEST はぬりつぶした図形（三角形、楕円など）も扱えるため、面図形の作成も容易である。

現在 GEST を学会発表に使用するカラースライドの作成や論理回路図の作成などに実際に利用し、その有効性を示している。

1. ま え が き

近年の計算機技術の発達により、CRT (陰極線管) を用いるグラフィック表示装置は計算機援用設計 CAD をはじめとするさまざまな分野において会話的に利用されている。その利用形態は種々異なっているが、利用者が意図した図形を作成、表示するという比較的簡単な利用法でも、学会等の発表用のスライド作成、CAI プログラムの表示画面の構成、デモンストレーション用プログラムの開発等幅広い応用範囲が考えられる。本図形編集システム GEST[†](A Graphic Editing System Using a Television-Based Computer Graphics System) は上記の応用を目的としており、テレビ走査型計算機グラフィック表示装置 DIGRAPH[‡] を利用して開発したものである。

この種の図形編集システムは、ランダム走査型やスタ走査型のグラフィック表示装置を利用してさまざま開発されてきた³⁾⁻⁵⁾。しかし、従来の表示装置はライトペンで指示されたカラー図形の同定が困難であるという欠点を持っている。そのため、CRT の会話型

利用に威力を発揮するライトペンが使えるカラー図形編集システムを開発するのは困難で、実現のためには複雑なソフトウェア処理が必要であった。ところが、GEST で利用する DIGRAPH はカラーテレビモニタを用いているにもかかわらず、ライトペンにより図形を同定することができるという特徴を持つ。そのため、GEST はライトペンを用いた会話的なカラー図形の作成、編集を可能にしており、従来の図形編集システムにない特色を有している。また GEST ではひとまとまりの図形を利用者定義図形として登録することができ、登録後は利用者定義図形単位の追加や削除が可能である。このような図形登録機能や図形編集機能は、従来の図形編集システムにおいてリスト構造のような複雑なデータ構造で実現されてきた。しかし、GEST はこれらの機能を簡単な構成と処理により実現しており、従来のシステムと異なる大きな特色となっている。これは、DIGRAPH の特徴であるライトペンによる図形同定機能を有効に利用することで可能となったもので、GEST では構成の単純なディスプレイファイルと利用者定義図形のための表だけを用いて前述の機能を実現している。なお、GEST が提供する基本図形にはぬりつぶした図形（三角形、楕円など）が含まれており、面図形の作成も容易になっている。

以下、第 2 章では GEST の機能や特徴について述

† GEST: A Graphic Editing System Using Television-Based Computer Graphics System DIGRAPH by HIROMI HIRAISHI, YUTAKA AOKI and SHUZO YAJIMA (Department of Information Science, Faculty of Engineering, Kyoto University).

‡† 京都大学工学部情報工学科

べ、第3章では DIGRAPH と GEST のシステム構成について概説する。第4章では GEST の編集機能を取り上げ、その実現方法について詳述する。

2. 図形編集システム GEST の機能

図形編集システム GEST は、第1章で述べたように、講演会や学会発表用等のスライド作成、CAI プログラムの表示画面の構成、デモンストレーション用プログラムにおける表示画面の作成等への応用を考え、利用者が意図した図形や絵を容易に作成、編集できるように開発したものである。また、必要と思われる機能を最小限にとどめ、DIGRAPH の特徴を最大限に生かすことにより、できる限り簡単な構成でそれらの機能を実現することをねらいとした。

GEST は以下に述べるような機能・特徴を持っている。

- (i) ライトペンを用いたメニュー方式により会話的にカラー図形の編集が行える。
- (ii) システムが提供する基本図形として、点・線分・文字のほかに、ぬりつぶした図形（三角形、長方形、楕円、円）をサポートする。
- (iii) ひとまとまりの図形を利用者定義図形として登録でき、それらを基本図形と同様に扱える。
- (iv) 基本図形および利用者定義図形の追加や削除といった図形編集機能が簡単なデータ構造で実現されている。

GEST へのコマンドは、メニューの中から適当なものをライトペンで指示することにより与えられる。図1にメニューと使用例を示す。

GEST は表示図形の編集のみを取り扱うので、それ

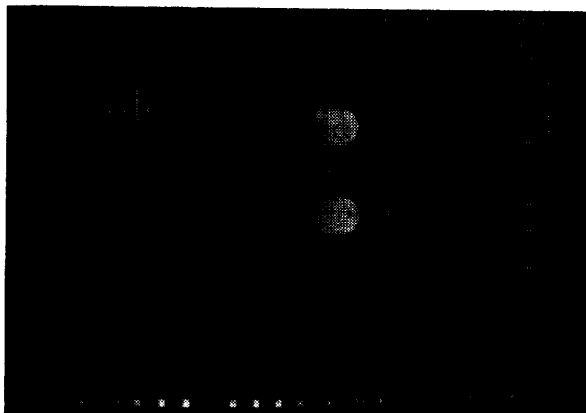


図1 GEST のメニューと使用例

Fig. 1 Menu of GEST and an example of its use.

ほど複雑なデータ構造は必要でない。しかも、DIGRAPH のライトペンによる図形同定機能を用いることにより、図形の同定を非常に簡単に実現することができ。そのため、GEST は従来多く用いられてきたリスト構造などを使わず、DIGRAPH が直接実行できるディスプレイファイルと、利用者定義図形に関する情報を記憶した簡単な表のみを用いて図形の編集を実現することが可能となった。図形編集機能については第4章で詳述する。

3. 図形編集システム GEST のシステム構成

GEST は、小型計算機 PDP 11/40 をホスト計算機とし、装置シミュレータ⁶⁾を併用するテレビ走査型計算機グラフィック表示装置 DIGRAPH を用いる。本装置シミュレータはアセンブリ言語で作成され、DIGRAPH のハードウェアの一部をシミュレートする。そのため、GEST は本装置シミュレータを用いて DIGRAPH を利用し、図形の表示やライトペンによる入力などを行う。

GEST は PDP 11/40 の拡張された BASIC⁷⁾を用いて開発されているが、主記憶容量の制限によりオーバーレイ構造をとっている。オーバーレイの部分をすべて含めると、GEST の大きさは BASIC のステートメント数で約 1,900 である。また、GEST は利用者が作成、編集した図形に対応して、DIGRAPH のためのディスプレイファイルを作成するが、そのためのバッファ領域の大きさは約 1 K 語（1 語は 16 ビット）である。

3.1 テレビ走査型計算機グラフィック表示装置 DIGRAPH

GEST で利用する DIGRAPH は従来のランダム走査型計算機グラフィックス技術とテレビ走査型グラフィックス技術とを結合させて、両者の持つ特徴を取り入れた新たな方式のグラフィック表示装置である。本装置は 20 インチの高解像度カラーテレビモニタを使用し、次のような特徴を持っている。

(i) 表示部にカラーテレビモニタを利用しているにもかかわらず、計算機側からはランダム走査型グラフィック表示装置と同等に扱うことができ、ライトペンによる図形の同定が容易である。

(ii) ライトペンの検出すべき色の指定が可能。GEST はこれらの特徴を生かすように開発されているので、その構成は大変簡単なものになっている。

DIGRAPH のシステム構成図を図2に示す。DIGRAPH は、要求される速度に対応してハードウェア

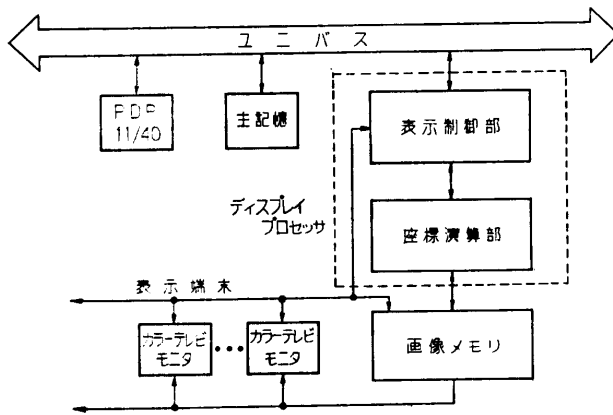


図 2 DIGRAPH のシステム構成図

Fig. 2 System configuration of television-based computer graphics system DIGRAPH.

量との兼ね合いで種々の構成が可能となるように設計されている。現在は画像メモリを中心とする少量のハードウェアと、表示制御部と座標演算部からなるディスプレイプロセッサ部をシミュレートする装置シミュレータにより構成されている。この装置シミュレータは必要とされる速度に応じて段階的にハードウェアに置き換えることが可能である。装置シミュレータの機能としては、表示命令を解釈実行して図形の表示を行うほか、ライトペンによる図形同定機能をもシミュレートする。なお、現在可能な表示の精度と色は

512×512 および 8 色である。

図形同定機能は、ライトペンよりの座標入力装置（ジョイスティックやタブレット等）あるいは利用者プログラムより与えられた座標を持つ図形に関する情報（色、種類、表示命令のディスプレイファイル上の番地など）を求める機能である。したがって、DIGRAPHを利用する場合、図形同定に関する処理は一切不要であり、GESTの構成を単純化することが可能となった。現在DIGRAPHにはライトペンが接続されており、GESTではライトペンを入力装置として多用している。従来のライトペンでは得られる座標値がかなり不正確であるが、DIGRAPHではテレビ画面の1フレーム期間中に得られるライトペンヒットパルス列より正確にライトペンの位置を求める方式⁹⁾を採用しており、GESTにおいて十分利用できる操作性を有している。

また、装置シミュレータはDIGRAPHを利用するアプリケーションプログラムとは独立したプログラムになっている。したがって、DIGRAPHのハードウェア化にともなって装置シミュレータが変更されても、そのアプリケーションプログラムの変更は不要であるという利点を持つ。しかも、DIGRAPHの効率向上によりアプリケーションプログラムの効率も向上する。次節でGESTと本装置シミュレータの関係について述べる。

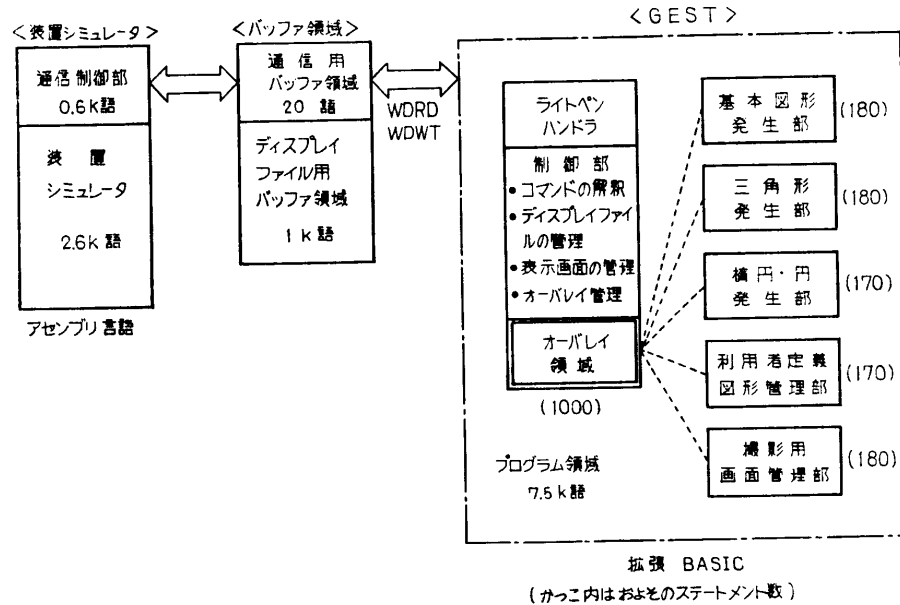


図 3 全体の構成

Fig. 3 Overlay structure of GEST and the relationship between GEST and the device simulator.

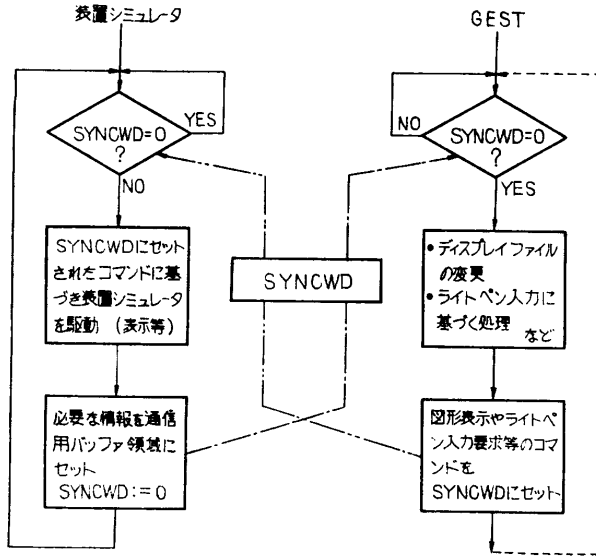


図 4 GEST と装置シミュレータとの通信

Fig. 4 Communication between GEST and the device simulator.

3.2 GEST のシステム構成と装置シミュレータ

GEST のオーバレイ構造および装置シミュレータを含めた全体の構成を図 3 に示す。GEST と通信制御部を付加した装置シミュレータは 2 つの独立したプログラムであり、PDP 11/40 上で並行に動作できるようになっている。GEST は装置シミュレータと通信を行うことにより図形の作成や編集を進めるが、この通信は主記憶の特定領域にあるバッファ領域を介して行う。そのため、GEST は拡張された BASIC に含まれる WDRD と WDWT (主記憶の絶対番地を読み書きする機能) を用いる。

通信用バッファ領域中の SYNCWD は、図 4 に示すように、GEST から装置シミュレータにコマンドを与えるのに用いられる。その主なコマンドは、画面消去、図形表示、ライトペンの入力要求などである。ライトペンの入力要求に対して、装置シミュレータが GEST に与える情報は、

- (i) ライトペンで指示した図形の始点座標、色と種類、その図形に対応する表示命令が格納されている主記憶の番地。
- (ii) ディスプレイサブルーチン時の退避用スタックの内容 (復帰番地)。

などである。これらの情報は図形を編集する際に必要なものであるが、特に (i) の始点座標と (ii) の情報は GEST の編集機能を簡単な構成で実現するために必要となった情報である。そのため、この 2 つの情報は

従来装置シミュレータから得られないものであったが、装置シミュレータを拡張して、GEST がこれらの情報も利用できるようにした。

4. 編集機能の実現とその評価

4.1 ディスプレイファイルの構成と画面管理

図形の編集や利用者定義図形の取り扱いを複雑なデータ構造を用いずにできる限り簡単な構成で実現するため、GEST は作成される図形のディスプレイファイルを中心に図形の編集を進める。基本図形は DIGRAPH の表示命令と 1 対 1 に対応しているが、利用者定義図形は複数の図形から構成され、それらをひとまとまりにして扱わねばならない。したがって、ライトペンで図形が指示された場合、それが基本図形であるか利用者定義図形の一部であるか区別できる必要がある。そこで、ディスプレイファイルを 図 5 のように構成し、利用者定義図形を構成する複数の図形の表示命令で 1 つのディスプレイサブルーチンを形成する。これにより、ライトペンで指示された図形の表示命令がディスプレイサブルーチンの内部にあるかどうかを判定するだけで、その図形が基本図形か利用者定義図形の一部かを簡単に決定できる。4.3 節で述べるように、この判定は装置シミュレータ内の退避用スタックを用いて容易に行える。そのため、3.2 節で述べたように装置シミュレータを拡張して、GEST から退避用スタックにアクセスできるようにした。

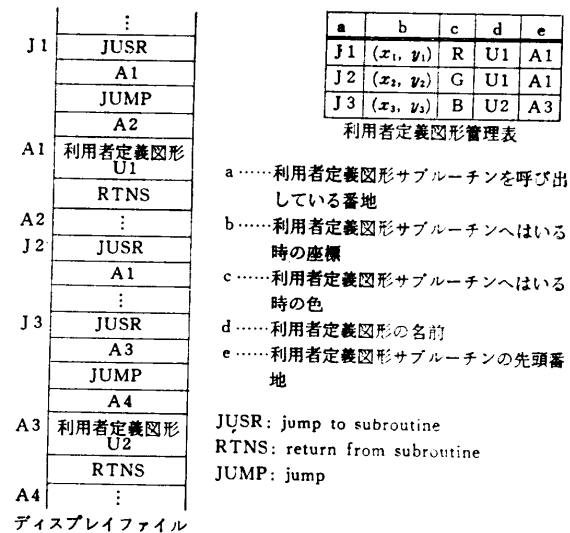


図 5 ディスプレイファイルと利用者定義図形管理表

Fig. 5 Construction of display file and user-defined picture management table.

GEST は表示図形に関する情報のみを扱えばよいので、基本図形に関する情報は DIGRAPH の図形同定機能を適当なディスプレイファイルに対して行わせることにより得られる情報で十分である。しかし、同一の利用者定義図形に対するディスプレイサブルーチンを共有し、追加や削除を行うためにはそれだけでは不十分である。そこで、利用者定義図形に関する情報(位置、色、名前、番地)を図5に示す簡単な表(利用者定義図形管理表)に記憶し、編集の際に利用する。なお、利用者定義図形を用いて別の利用者定義図形を構成することも可能で、これはディスプレイサブルーチンのネストにより実現される。

GEST は図形の編集結果を画面に常に表示している。したがって、図形の編集に応じて画面の変更が必要である。図形の編集によりディスプレイファイルは更新されるので、一度画面全体を消去した後、更新したディスプレイファイルを最初から実行させて表示する方法が考えられる。しかし、複雑な図形になるほど表示時間がかかり、しかもメニューまで消去されるのでその再表示が必要となる。そこで、GEST は効率向上のために、変更部分のみの表示を行う。図形の追加は、その図形を画面にそのまま表示すればよい。図形の削除にともなってその図形を消去する場合は、EX-OR モードを用いる。これは、画像メモリ内の色データとディスプレイプロセッサで発生された色データとの間で排他的論理和演算が行われるモードである。消去したい図形と全く同じ図形(基本図形、利用者定義図形のどちらでもよい)をこのモードで重ねて表示すると、色データがすべて0(黒)となり消去が行える。図形を完全に重ねるためには、図形の始点座標がわかればよい。そのため、3.2節で述べたように装置シミュレータを拡張し、ライトペンで指示された図形の始点座標がわかるようにした。これが不明のシステムでは始点座標を知るために、指示された図形の表示命令を表示停止命令に変更し、ディスプレイファイルを最初から実行して停止した点の座標を調べるといった複雑な処理を行わねばならない。なお、利用者定義図形の場合は利用者定義図形管理表からその情報がわかる。

次に、以上述べた消去の方法を比較し評価を行う。簡単のため、 N 個(奇数)の基本図形で構成された図形があり、ディスプレイファイルの中央の表示命令に対応する図形を消去する場合を考える。1つの基本図形の表示のための平均時間を T_d 、画面消去とメニュー

表示に必要な時間をそれぞれ T_e 、 T_m とすると、以下の結果を得る。

(a) 画面消去後、ディスプレイファイルを実行
 $T_e + T_m + (N-1)T_d$

(b) 変更部分のみの表示(EX-OR モードを用いる方法)

(i) 始点座標が不明の時

$$\frac{N-1}{2}T_d + T_d = \frac{N+1}{2}T_d$$

(ii) 始点座標がわかる時(GEST)

$$T_d$$

装置シミュレータを用いている現在、一点表示するのに約 $200 \mu\text{sec}$. かかるので、基本図形が平均 50 点で構成されるとすると、 $T_d \approx 10 \text{ msec}$. となる。 $T_e + T_m \approx 80 T_d$ 、 $N \approx 100$ とすると、(a)、(b)の(i)および(b)の(ii)の概略値はそれぞれ、1.8 sec., 0.5 sec., 10 msec. となる。これより(b)の(ii)がかなり効率的であることがわかる。

4.2 ライトペンによる重なり図形の同定

重なっている図形をライトペンで指示すると、どの図形が指示されたかあいまいであるという問題が起こる。たとえば図6の文字“G”を指示しても、ライトペンの口径には長方形を構成する点も同時にとらえられるため、利用者の意図する図形を決定できない。

特に図形の削除の際に誤った削除が行われる可能性があるため、GEST では削除の対象を会話的に決定する。すなわち、削除すべき図形の候補を画面から1つずつ順に消去し、利用者の応答により実際に削除する図形を定める。図形の消去は4.1節で述べた EX-OR モードを用いて行う。また、正しくない図形を消去した時は画面を復元した後(消去されていた図形が再び画面に表示される)、次の候補の図形を消去する。画面の復元は消去した図形のみをもう一度 EX-OR モードで表示することにより完全に行えるので、図形の消去や画面の復元は短時間でできる。しかし、消去する図形がほかの図形と重なっている場合は、重なり部分がかき消され(復元は完全にできる)。その

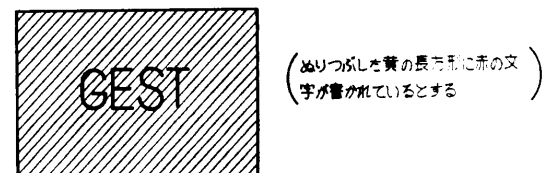


図6 図形の重なり例

Fig. 6 An example of overlapping pictures.

ため GEST には、画面全体を一度消去した後、更新されたディスプレイファイルを最初から実行して正しい表示にするコマンドが設けてある。

4.3 基本図形の追加と削除

GEST は基本図形として、点・線分・面長方形・文字列のほかにもりつぶした三角形・楕円・円などを提供する。これらのうち、前者は DIGRAPH の表示命令に対応するが、後者は対応しない。そのため、GEST は DDA によるアルゴリズム⁸⁾等を用いて、複数の点や線分で後者の図形を発生する。したがって、これらの図形はひとまとまりに扱う必要があるので、GEST 内部ではこれらの図形を利用者定義図形と同様に扱う。そこで、本節では前者の基本図形の追加と削除について述べる。

〈追加〉 必要な処理は以下の2つである。

- (i) ディスプレイファイルの更新。
- (ii) 表示画面の変更 (追加した図形の表示)。

(i) は利用者の指示 (図形の位置、色、種類) をもとに、必要な表示命令を作成してディスプレイファイルの最後に付け加えるだけでよい。ただし、利用者定義図形実現のため、図形の表示位置は相対的となるようにする。(ii) は 4.1 節で述べたように、追加した図形のみを表示する。

〈削除〉 削除すべき図形は 4.2 節で述べたように会話的に決定されるが、それと同時にその図形が基本図形か利用者定義図形の一部かを以下のようにして判定する。

- (i) DIGRAPH に図形同定機能を行わせる。
- (ii) ディスプレイサブルーチンのための退避用スタックにアクセスする。

(iii) スタックが空であれば、指示された図形は基本図形であり、そうでなければ利用者定義図形を構成する図形の1つである。

図形同定機能は、ライトペンで指示された点を構成点として持つ図形の表示命令を、ディスプレイファイルの先頭から順に探索するものである。したがって、ディスプレイサブルーチン内部の表示命令において探索が成功すると、退避用スタックは復帰番地が格納された状態になっている。4.1 節で述べたように、ディスプレイファイル内では利用者定義図形のみがディスプレイサブルーチンとして構成されるので、このスタックをもとに (iii) のように判定が行える。

削除する図形が基本図形である場合、ディスプレイファイルの更新は容易である。図形同定機能により

削除する図形の表示命令のある番地がわかるので、その表示命令を無表示の命令 (表示をせずにビームの移動のみが起こる) に変更すればよい。また、図形の削除にともなってその図形を画面から消去せねばならない。しかし、これは 4.2 節で述べたように、削除する図形を決定する際にすでに EX-OR モードを用いて効率的に行われている。

4.4 利用者定義図形の登録、追加と削除

ひとまとまりの図形に名前を付けて登録したものは利用者定義図形として使用することができ、利用者定義図形単位の追加や削除が可能である。本節では、このような利用者定義図形の登録、追加、削除について述べる。

〈登録〉 利用者定義図形の登録は、GEST が作成したディスプレイファイルと利用者定義図形管理表をディスク上のファイル (ファイル名は利用者定義図形の名前と同一) とすることにより行う。登録は GEST を利用して図形の編集を行っている任意の時点で可能であり、登録を指定した時に画面に表示されている図形がその対象となる。したがって、利用者定義図形を用いて構成した図形を別の利用者定義図形として登録できる。

利用者定義図形はディスプレイサブルーチンとして実現されるが、その時の番地は主記憶上の絶対番地で

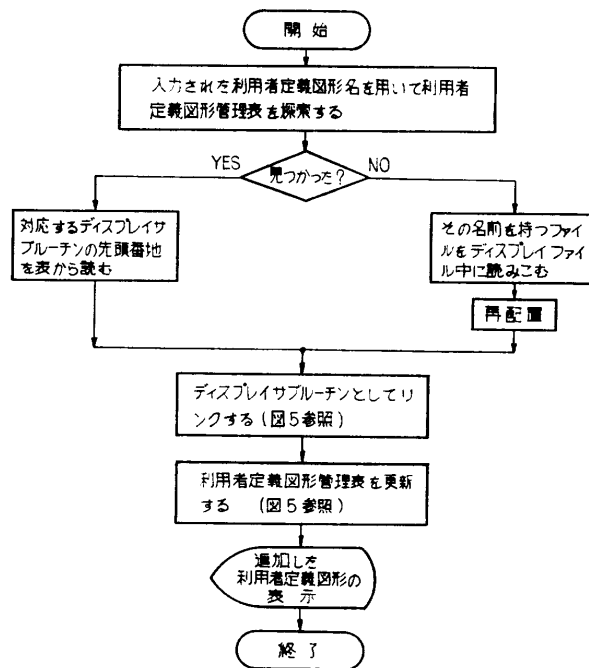


図 7 利用者定義図形の追加

Fig. 7 Addition of a user-defined picture.

ある。そこで、登録の際にはそれらを相対番地に変換し、追加を行う時に再配置する。

なお、完成した図形を登録することにより図形の保存が可能で、ほかの応用プログラムは登録された図形のディスプレイファイルをそのまま利用できる。

〈追加〉 利用者定義図形の追加はその名前を指定して行われる。その処理の概略を図7に示す。GESTは入力された名前をもとに利用者定義図形管理表を探索し、その利用者定義図形に対応するディスプレイサブルーチンが存在するかどうかを判定する。存在する場合は、そのサブルーチンを呼び出す命令をディスプレイファイルに追加するだけでよい。しかし、そうでない時は利用者定義図形のファイルを、ディスプレイファイル中に読みこまねばならない。しかも、そのファイル中に含まれる番地は相対番地であり、再配置の処理が必要である。

利用者定義図形管理表は追加した利用者定義図形に関する情報により更新し、その図形がさらに利用者定義図形を含んでいる時は、その情報も表に記憶する。

〈削除〉 削除の際にライトペンで利用者定義図形の一部が指示された時は、その利用者定義図形全体を削除する。ただし、削除の単位は追加の時と同一とし、一番浅いレベルのサブルーチンで実現された利用者定義図形を削除するものとする。たとえば、図8のように構成された利用者定義図形Aを追加した場合、削除の際に利用者定義図形Dを指示しても、削除はA全体で行う。さもないと、Aが画面上のほかの場所で利用されている時（ディスプレイサブルーチンの共有による）、その部分のDも削除されてしまうからである。これは、Dの削除により共有されたサブルーチンの内容が変更されてしまうことになる。なお、利用者定義図形自身の編集を可能にするため、GESTはその図形の登録を行った時の状態に復旧することができる。

4.3節で述べたように、削除すべき図形が利用者定義図形であるのは、退避用スタックが空でない場合で

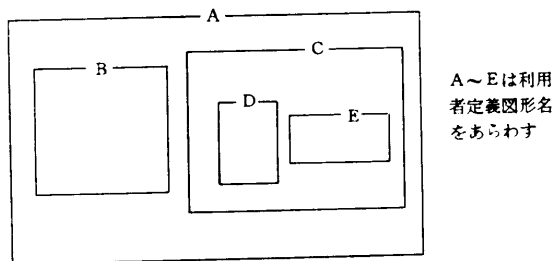


図8 利用者定義図形を用いて構成した利用者定義図形
Fig. 8 A user-defined picture composed of other ones.

ある。この時の処理は、

(i) 退避用スタックの一番底の復帰番地を読み出す。
(ii) その番地を用いて削除すべき利用者定義図形に対応するディスプレイサブルーチンの呼び出し命令を見つけ、それを無効にする (NO—OP 命令に変換する)。

(iii) その番地を用いて利用者定義図形管理表を探索し、見つかった欄を削除する。

退避用スタックにはいくつかの復帰番地が格納されている場合があるが（ディスプレイサブルーチンのネストによる）、一番浅いレベルのディスプレイサブルーチンのみを考えればよいので、退避用スタックの一番底の番地だけが必要である。また、画面からの消去は基本図形の場合と同様で、EX—OR モードによる効率的な消去がすでに行われている。

4.5 考 察

GEST を実際に使用した経験から、以下に述べることが明らかになった。

(1) 位置決め of 正確さ

カーソルの移動速度を選択できるので、位置決めが正確に行える。また、ライトペンで指示された図形の始点や終点にカーソルを移す機能も位置決め to 有効である。

(2) 指示の容易さ

図形の指示はライトペンで会話的に行われるので、大まかな位置を指すだけで正しい図形を選び出すことができる。

(3) EX—OR モードによる消去の高速化

必ずしも高速でない装置シミュレータを用いているGESTにおいて、EX—OR モードを利用することにより図形消去の高速化がはかれた。また、誤って図形を削除した場合にも高速に復元可能となった。

(4) 不足の機能

図形の拡大・縮小や回転の機能、表示図形の一部をまとめて移動したり、利用者定義図形として登録できる機能などがあれば、一層使いやすくなると思われる。

5. む す び

本論文では、DIGRAPH の特徴を生かすことにより、ライトペンを用いた会話的なカラー図形編集システムがGESTのような簡単な構成で実現できることを示した。現在 DIGRAPH のハードウェアの一部は

装置シミュレータによりシミュレートされているので、GEST の図形表示速度や図形の削除に対する速度は必ずしも速くない。しかし、ライトペンにより指示されたそのほかの基本的なコマンドに対しては、約 0.5 秒程度以下の応答速度が得られているので十分実用に耐えるものであり、装置シミュレータを用いた DIGRAPH でもこの種の会話型利用に対して十分実用になることが明らかになった。さらに GEST の利用により、従来表示命令用アセンブラを使用して 4～5 時間要した図形作成が、20 分～1 時間程度で可能となり GEST の実用上の利用価値は大きいものと考えられる。

現在の問題点としては、削除を重ねるにつれて無表示の命令や NO-OP 命令が増え、ディスプレイファイル内の無駄な部分が大きくなるということがあげられる。しかし、余分の表示命令は簡単な処理で取り除けるので、GEST にその機能を組み込むことは容易である。

また、GEST で作成したカラー図形は、現在写真撮影によりハードコピーとしており、講演会や学会発表用のスライド作成等に役立っている。この場合、テレビ画面上に作成されたカラー図形の色相等がスライド撮影した場合に若干変化したり、テレビ画面が湾曲しているために近距離撮影を行うと周辺部が若干丸く歪む等の問題点がある。

今後 GEST には座標変換の機能等も追加し、より充実した図形編集システムへと発展させる予定である。

謝辞 本システムの開発にあたり貴重なご意見を賜った本学上林弥彦助教授、ならびに熱心にご討論頂いた矢島研究室各位に深謝します。なお、本研究は一部文部省科学研究費による。

参 考 文 献

- 1) 平石, 青木, 矢島: テレビ走査型計算機グラフィックスを利用したスライド作成用図形編集システム GEST, 信学技報, EC 78-64 (1979).
- 2) Hiraishi, H., Kawakubo, K. and Yajima, S.: The Design of a Microprogrammed Graphic Display Processor with Pipeline Organization, Proc. 3rd EUROMICRO Symp., pp. 66-73 (1977).
- 3) Sutherland, I. E.: SKETCHPAD A Man-Machine Graphical Communication System, Proc. SJCC, AFIPS, Vol. 23, pp. 329-346 (1963).
- 4) Denes, P. B. and Gershkoff, I. K.: AN INTERACTIVE SYSTEM FOR PAGE LAYOUT DESIGN, Proc. ACM, Annual Conference, pp. 212-221 (1974).
- 5) Parke, F. I. and Koperda, F. R.: INTERACTIVE DESIGN OF VISUAL AIDS, Proc. ACM, Annual Conference, pp. 434-439 (1977).
- 6) 平石, 矢島: ランダム走査機能を有するラスタ走査型計算機グラフィック表示装置の装置シミュレータによる実現, 情報処理学会論文誌, Vol. 22. No. 1, pp. 36-43 (1981).
- 7) 中川, 矢島: Implementation of LABOLINK Communication Control Program by an Extension of Multi-User BASIC, 信学技報, EC 76-84 (1977).
- 8) 穂坂: コンピュータ・グラフィックス, 産業図書, pp. 83-86 (1974).
- 9) 平石, 桑原, 矢島: テレビ走査型計算機グラフィックス表示装置における色指定可能なライトペン方式, 信学技報, IE 79-73 (1979).
(昭和 55 年 11 月 10 日受付)
(昭和 56 年 6 月 16 日採録)