Data base and Computer Graphics

Makoto IKEDA and Jiro FUJIMOTO

Department of Electrical Engineering

Abstract

This paper discusses introducing the data base's conception to the computer graphics. No data base system in the computer graphics has not been established because of the problem of the data structure. In this paper we assume the system is operated by the minicomputer, and the data is from the geometric figures with a loop structure. The data base system suggested here is composed of the graphics translation program, the data base manipulation program and the basic data element of the geometric figures. The language used here is PASCAL.

1. Introduction

The technical development of the computer system is outstanding and it's applied to every field. However Data will be increasingly regarded as a vital corporate resource which must be organized so as to maximize their value. The vast majority of the information is not yet computerized. However the cost of data-storage hardware is droping more rapidly than other costs in data processing. Under this condition for the system designers efficient and effective information management manners are needed. Then the conception of a data base has been discussed. Recently, many Data base management systems are made. For example CODASYL type, IMS type, TOTAL types are famous put into practical use. However, as regards computer graphics no data base system has not been established because of the data structure problem. This paper discusses the data base system about the computer graphics but the general graphics is not discussed here. Our subject is to focus on the geometric figures especialy with a loop structure as the first step of a data base system for computer graphics. Our data base system suggested here are composed of graphics translation program, data base manipulation program and a data base.

This system after accomplished will be included in operating system in the computer, however as we assume our system is operated by the minicomputer, the system is considered from the effective and efficient usage of mainstorage.
The language used here is Pascal.

2 The outline of a data base system

2–1 The conventional graphics processing algorithm
Fig. 2–1 shows the basic relationships among graphics processing program, graphics processing workareas and input/output. As known well, this can be constructed by the

Received September 28, 1978

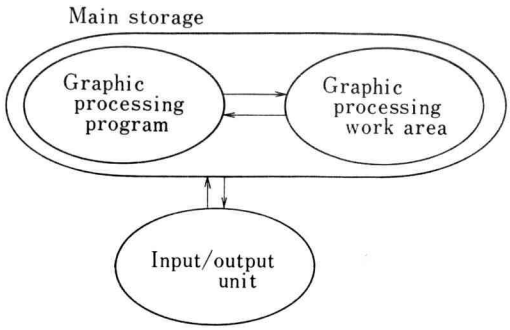

Fig. 2–1 Basic relationships

conventional batch processing.

In user software by this processing the following two demerit lie.

1) To put the graphics processing workarea in main storage means wastefulness of main storage and for us, users, it can not be said to be efficient.

2) In case we need the revival of graphics displayed once in the same processing, we need the program and data again used before under the condition of 1).

These problems concludes we do not use the main storage efficiently whether the sub storage unit can be used or not.

2–2 Graphics Processing algorithm using the disk

Fig. 2–2 shows we can use a main stroage efficiently as a workarea lies in a sub storage not in a main storage. But one important problem still remains. That is the problem about efficiency of using data files. Because if we want to use data file efficiently, we need to consider the data structure and the method to store a data necessary for graphics processing in sub storage. To solve this problem we suggest using the concept of a data base.

2–3 A Data Base system

Researchs to introduce the concept of a data base to computer graphics have begun recently but the definite idea has not been established. The problem to be solved is the basic data structure. Here we try to discuss a data base system for graphics processing of geometric figures especially with a loop structure by mini-computer.

Fig. 2–3 (a), (b) show the relation and algorithm using the concept of a data base.

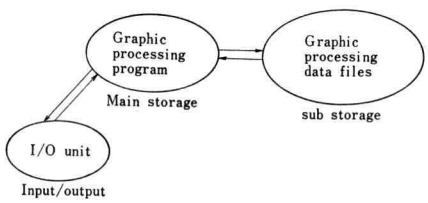

Fig. 2–2 relation with data file

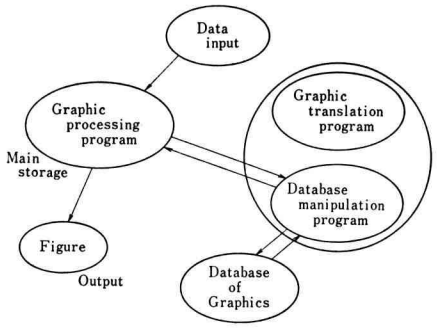

Fig. 2–3 (a) relationships using of the data base

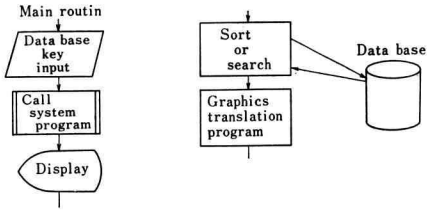

Fig. 2–3 (b) Algorithm

Though it is impossible to simplify the data structure in the algorithm of Fig. 2–2, in the system of Fig. 2–3, firstly we decompose the function of sub storage, and we make a data management independently to sort and search the data to use a data efficiently. Secondly we discuss the data structure for irreducible minimum elements of datum necessary for graphics processing.

3 Data independence and Data base

As in Fig. 3(a), the software executed the input/output operations of the storage devices and little else. The coding written into the application programs took care of the data organization. If a change was made to the data organization or to the storage units, application programmer had to modify his programs accordingly, recompile them, and then test modifications. Fig. 3 (b) shows the software made it possible to change the physical data layout without changing the logical view of the data, provided that there was no change in record contents or in the fundamental structuring of the files. It was desirable to isolate the application program not only from changes in file hardware and the effect of increasing file size but also from additions to the data that are stored, such as new fields and new relationships. Data base software in Fig. 3(c) attempts to accomplish this. When the early data-base systems had been in use for a short time it became apparent that a further level of data independence was needed. The overall logical structure of the data became complex in many cases, and as the data base grew, the overall logical structure inevitably changed. It became important that it should be able to change without forcing a change in the many application programs which used it. In some systems, change in the overall logical structure of the data has become a way of life; it is needed logical and physical data independence.

Fig. 3(d) illustrates the concept of logical and physical independence. The block in the center of the illustration represents the overall logical structure of the data. This view may be entirely different from the physical structure of the data and different from any individual application programmer's view of the data. The data-base software will, in effect, derive the application programmer's data from the overall logical structure and will then map the overall logical structure into the physical representation.

File Access Method (Predominant in the late 1960's)

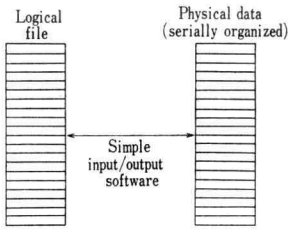

Fig. 3 (a)

- Files organized in serial manner.
- Physical data structure essentially the same as logical file structure.
- Batch-processing with no real-time access.
- Several copies stored of the same file because previous generations of data are kept.
- Software handles only input/output operations.
- Application programmer designs the physical file layouts and embeds them in the application programs.
- If the data structure or storage device are changed, the application program must be rewritten, recompiled, and retested.
- Data is usually designed and optimized for one application.
- Hence the same data is rarely used across applications.
- High level of redundancy between data files.

File Access Method (Predominant in the late 1960's)

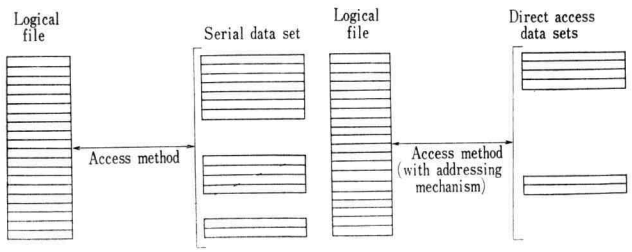

Fig. 3 (b)

- Serial access or random access is possible to records (not fields).
- Processing is batch, on-line, or real-time.
- Logical and physical file organization are distinguished but the relationship between them is fairly simple.
- Storage units can be change without changing the application program.
- Data structures are usually serial, indexed sequential, or simple direct access.
- Multiple-key retrieval is generally not used.
- Data security measures may be used (but are rarely very secure).
- Data still tends to be designed and optimized primarily for one application.
- Much data redundancy still exists.
- If hierarchical files and used, the programmer usually has to construct father-son relationships.
- The software provides" access" methods but not" data managlment".

Early Data-Base Systems (Predominant in the early 1970's)

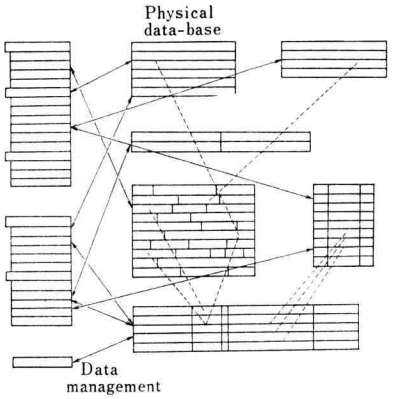

Fig. 3(c)

- Multiple different logical files can be derived from the same physical data.
- The same data can be accessed in different ways by application with different requirements.
- Software provides the means to lessen data redundancy.
- Data elements are shared between diverse application.
- Absence of redundancy facilitates data integrity.
- The physical storage organization is independent of the application programs.
- It may be chaged often in order to improve the data-base performance without application program modification.
- The data is addressable at the field or group level.
- Multiple-key retrieval can be used.
- Complex forms of data organization are used without complicating the application programs.

4 Data base and Computer Graphics

The fundamental idea of data structure by C.W. Bachman is "The real society consists of various kinds of Entity and Relation among these." The resource of data is called Entity and a group of these Entities is called Entity Set. As one example, we consider a regular triangle. This regular triangle itself is Entity and a group of triangles can be Entity Set. As said before, there exists Relation among Entities. In this example since a triangle as entity is included in geometric figure as Entity we obtain the following expression.

$$A \subset U \qquad (4\text{–}1)$$

where u: a universal set of geometric figures

$A = \{a \mid a \in U \text{ and } a \text{ is a triangle}\}$

And this corresponding relation between two entities can be expressed in the following data structure. Entity Set corresponding to that tail of the arrow in the Fig. 4–1 is called Owner and Entity Set to the head of the arrow is called Member.

Today's Requirement in Data-Base Systems

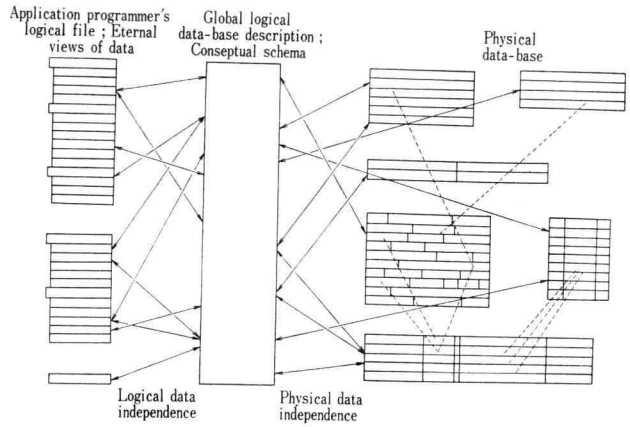

Fig. 3 (d)

- Software provides logical as well as physical data independence, allowing a global logical view of the data to exist independently of certain changes in the application programs' views of data or th physical data layouts.
- The data base may evolve without incurring high maintenance costs.
- Facilities are provided for a Data-Base Administrator to act as controller and custodian of the data, and ensure that its organization is the best for the users as a whole.
- Effective procedures are provided for controlling the privacy, security, and integrity of the data.
- Inverted files are used on some systems to permit rapid data base searching.
- Data bases are designed to provide answers to unanticipated forms of information request.
- Data migration is facilitated.
- The software provides a data description language for the Data-Base Administrator, a command language for the application programmer, and sometimes a data interrogation language for the user.

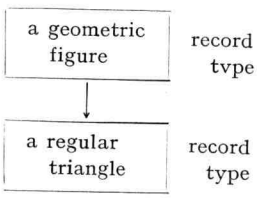

Fig. 4-1 Set tupe

Fig. 4–1 indicates the relationships shown by the arrow between two record typies and is called Set Type. Fig. 4–2 shows a Set type described on data base and the association between the respective record occurence (Entity) is called Set occurence.

In Fig. 4–2, each triangle can be expressed by scaling, parrallel translation and rotation of a regular triangle as a base. We can use the same manner in case of a quadangle. This indicates it is possible to express every geometric figure in the manner to store all equilateral polygones as a base in a file. To execute this, we consider the following DBMS (Data Base Manegement System).

In Fig. 4–3, user program is the area given to users. When a user program demands a data, a data base can be manipulated in the system program and necessary data are

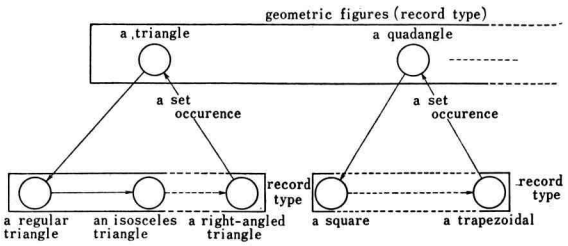

Fig. 4–2 Set Occurence

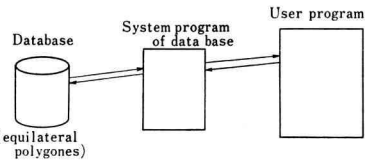

Fig. 4-3 DBMS

transmitted to a user program.

4–2 Data structure of data base

Data may consist of numerical values, names, alphanumeric characters, codes, or, in general, it may be any "representation in a precise, formalized language of some facts or concepts" (Knuth). According to the definition above, the data base or the logical file of a graphic representation, for example, are such data structures.

Data can be considered as a collection of data items, that is a set; and the relationships between the elements of such a set can consequently be expressed by the mathematical concept of relations between the objects of sets. For example, we define a data structure formally as a pair

$$(S.\ \rho).$$

where $S=\{s1, s2, \ldots, sn\}$ is a set of data objects and $\rho=\{R1, R2, \ldots, Rn\}$ is a set of binary relations.

A necessary data structure as a data base of geometric figures is shown in Fig. 4–4 in a record type.

This can be also expressed in record occurence as in Fig. 4–5

The role of a calc key is to search a necessary data for figures designated by a user program.

The value of the strating point of calc	Number of the side	The angle between sides	the length of side

Fig. 4-4 Data Structure in a record type

0001	3	60°	10 cm
0002	4	90°	10 cm

Fig. 4–5 A record occurence of a regular triangle and a square

5 Realization of a data base system

Systems programing to execute the data base system for geometric figures is composed of graphics translation program and a data base manipulation program.

5–1 A data base manipulation program

To manipulate a data base we need the following algorithm.

1) To sort a data to take out a data stored in a data base in a natural order

2) To search data from a data base indicated by the value of the key when the value of the calcu key is input.

In Fig. 5–1, the program written in Pascal of sorting is shown.
In Fig. 5–2 the program of searching is shown.
In Fig. 5–3 the algorithm to manipulate a data base by using two program of sorting and searching is shown.

```
procedure sort (l, r: integer);
var  i, j: index: x,w: item;
begin i:=l: j:=r;
  x:=m[(lr) div 2];
    repeat
        while m[i]. key<x. key do i:=i+l;
      while x. key<m[j]. key do j:=j−l;
    if i=j then
      begin w:=m[i]; m[i]:=m[j];
          m[j]:=w; i:=i+l; j:=j−1;
      end>
    until i>j
      if l<j then sort (l, j);
      ij i<r then sort (i, r)
end;
```

Fig. 5–1 The program of sorting in Pascal

```
procedure search (k: integer, var s: integer);
var s,i,j: integer;
begin i:=l; j:=n; s:=0:
  while (i=j) and (k< >m[i]. key) do
    begin mid:=(i+j) div 2;
      if k−m [mid]. key=0
        then s:=mid
        else if k−m [mid]. key<0
          then j:=mid
          else i:=mid+1
```

```
      end
  end;
```

Fig. 5–2 The program of searching in Pascal

```
program databasemanipulation (input, output);
const n=1000;
type  index=0..n;
      iutem=record key:integer;
                   numberside: 3..100;
                   angle:integer;
                   lengthside:integer
            end;
var  m:array [1..n) of item;
procedure enter;
  var     i,k,n,a,l: integer;
  begin
    for i:=l to n do
        begin read (k);
          while k<>0 do
            begin read (n, a, l);
              with m[i] do
                begin key:=k;
                      numberside:=n;
                      angle:=a;
                      lengthside:=1
                end
            end
        end
  end;
  procedure sort (l, r: index);
    ⋮
  procedure search (k: integer, var s: integer);
    ⋮
  begin
    enter;
      sort (l, n);
      read (datakey);
      search (datakey, s)
  end.
```

Fig. 5–3 The data base manipulation program

5–2 Graphics translation progarm

To get the directing figure from the data sorted and searched, we suggest the following graphics translation program in Fig. 5–4

```
program graphictranslation (input, output);
var xnew, ynew, xold, yold: real;
    command: char;
procedure move;
var transx, transy: real;
begin
    read (transx, transy);
    xnew: =xold+transx;
    ynew: =yold+transy;
        ⋮
end;
procedure scal
var magnification: real;
begin
  read (magnibication);
    xnew:=xold * magnfiication;
    ynew:=yold * magnification;
        ⋮
end;
procedure rotat;
var rotationangle: real;
begin
  read (rotationangle);
    xnew: =xold * cos (rotationanlge)-yold * sin(rotationangle);
    ynew: =xold*sin (rotationangle)+yold*cos(rotationangle);
        ⋮
end;
begin
  read (command);
    if command=move then
      move
    else
      if command=scal  then
        scal
          else
            rotat
  end.
```

Fig. 5–4 Graphic translation program

The basic function for Graphics translation program is defined as follow. Now we consider the two dimensional figure displayed on CRT. The x-distance and the y-distance are normalized in the region of $0 \leq x, y \leq 81.9$. When the co-ordinates (x, y) moves to (x', y') by T (the normlazied value), the co-ordinates $(x', y'($ is determined by the following expression.

$$[x,\ y,\ 1]\begin{bmatrix} 1, & 0, & 0 \\ 0, & 1, & 0 \\ Tx, & Ty, & 1 \end{bmatrix} = [x',\ y',\ 1]$$

In case of scaling, the following expression is given.

$$[x,\ y,\ 1]\begin{bmatrix} Sx, & 0, & 0 \\ 0, & Sy, & 0 \\ 0, & 0, & 1 \end{bmatrix} = [x',\ y',\ 1]$$

where S_1 and S_y indicates the facter of scaling

In case of rotating, the following expression is given

$$[x,\ y,\ 1]\begin{bmatrix} \cos\theta, & \sin\theta, & 1 \\ -\sin\theta, & \cos\theta, & 0 \\ 0, & 0, & 1 \end{bmatrix} = [x',\ y',\ 1]$$

where θ is the angle of rotating

By using these three expressions, the geometric figures are determined from the fundamental figure. The example is shown in Fig. 5–5.

6 Graphic and data-base system hardware

System construct in Fig. 6–1. Our graphic and data base system is comprising data base manipulation program, graphic translation program and data base in substorage. CRT and ALU & CU are connected with s.s. by direct memory access. From user program comprising in main memory, users can manipulate efficiently and effectively converse as one handling.

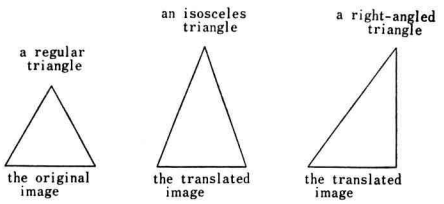

Fig. 5–5 Translation

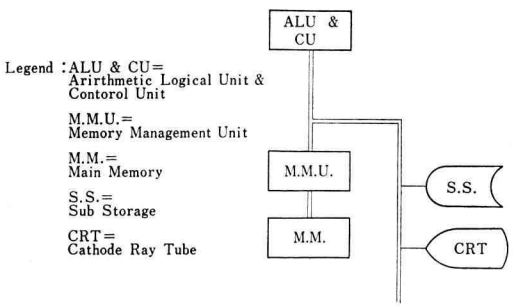

Fig. 6–1 GRAPHIC and DATA-BASE System hardware

7 Conclusion

We discussed the introduction of the data base system to the computer graphics by mini-computer. Difficulties that we found ourselves confronted with is how to idea the data elements of the general figures. In this paper, we imposed restrictions on the geometric figures, however as a next step we must consider the data base and the data structure of the general figures. As our first purpose we think we could succeed in obtaining and introducing the conception of the data base system to the ordinary programing in the mini-computer to use the main storage efficiently and effectively. Finally we want to say we sometimes quoted COMPUTER DATA-BASE ORGANIZATION in the summary explanation of the conception of data base.

References

1) M.D. Prince "Interractive graphics for computer-aided design" Addison-Wesley Publishing Company Inc 1971
2) R. M. Baecker "Picture-driven animation" S J C C 1969
3) W. Mayeda "Graph theory" John Wiley & Sons Inc 1972
4) J. L. Pfalty "Computer data structures" MaCRAW-Hill Inc 1977
5) J. J. Donovan "Systems programming" McGRAW-Hill Inc 1972
6) C. M. Theiss "Computer graphics displays of simulated automobile dynamics" F J C C 1972
7) J. Martin "Computer data-base organization" Prentice-Hall Inc 1975
8) W. K. Giloi "Interactive computer graphics" Prentice-Hall Inc 1978