



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11) **EP 0 670 560 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention  
of the grant of the patent:  
**31.10.2001 Bulletin 2001/44**

(51) Int Cl.7: **G06T 15/40**

(21) Application number: **95102795.2**

(22) Date of filing: **27.02.1995**

(54) **A method for sorting polygon data, a video game machine employing the same and a computer program performing the method**

Verfahren zur Sortierung von Polygondaten, Videospielmaschine zu dessen Anwendung und Rechnerprogramm um dieses Verfahren auszuführen

Procédé de tri de données de polygones, appareil de jeu vidéo utilisant un tel procédé et un programme d'ordinateur pour mettre en oeuvre ce procédé

(84) Designated Contracting States:  
**DE ES FR GB**

(30) Priority: **01.03.1994 JP 5672394**

(43) Date of publication of application:  
**06.09.1995 Bulletin 1995/36**

(73) Proprietor: **SEGA ENTERPRISES, LTD.**  
**Tokyo 144 (JP)**

(72) Inventor: **Shimizu, Yusuke,**  
**c/o Sega Enterprises, Ltd.**  
**Ohta-ku, Tokyo 144 (JP)**

(74) Representative: **Prüfer, Lutz H., Dipl.-Phys. et al**  
**PRÜFER & PARTNER GbR,**  
**Patentanwälte,**  
**Harthäuser Strasse 25d**  
**81545 München (DE)**

(56) References cited:  
**EP-A- 0 529 658**                      **EP-A- 0 531 157**  
**US-A- 4 821 210**

- **ACM TRANSACTIONS ON GRAPHICS, vol. 11,**  
**no. 1, 1 January 1992, pages 1-11, XP000281943**  
**SHARIR M ET AL: "A SIMPLE**  
**OUTPUT-SENSITIVE ALGORITHM FOR HIDDEN**  
**SURFACE REMOVAL"**

**EP 0 670 560 B1**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

**Description**

## BACKGROUND OF THE INVENTION

## Field of the Invention

**[0001]** The present invention relates to a method for sorting data for polygons a game machine that employs the method a computer program and a computer program product relating to the method; and in particular, the present invention relates to a method by which a memory area for storing data for polygons can be reduced and high speed sorting of the data for polygons can be performed, and to a game machine that employs that method.

## Related Arts

**[0002]** In recent video game machines, to provide more realism for users, a figure that is expressed by three-dimensional coordinates is considered as an integrated assembly of a plurality of polygons, and three-dimensional coordinate data for each polygon are transformed into two-dimensional coordinate data for a display device, such as a CRT, that has a two-dimensional display plane. The video game machines use the transformed two-coordinate data for a display on a two-dimensional display plane.

**[0003]** To display polygons in such video game machines, the order in which polygons are to be drawn in a frame buffer memory must be determined based on their relative distances in the direction of the depth of a screen. That is, it is necessary to register data for each polygon in a memory, to sort the data for drawing and read them in the order that corresponds to their relative distances in the direction of the depth of a screen, beginning with data having the greatest distance and to employ the data to draw polygons in the frame buffer memory.

**[0004]** To do this, Quick-Sort, or other sorting sequences are conventionally employed as general sorting methods. By employing such a method, the coordinates of all the polygons are transformed, and the obtained data for polygons are sorted and stored in the order that corresponds to their relative distances in the direction of the depth of a display screen.

**[0005]** With such a general method, after the coordinates of all the polygons are transformed for sorting, it is necessary for the obtained data to be temporarily stored in a memory.

**[0006]** Therefore, a separate, large-capacity memory area is required for the storage of the transformed data. In addition, a procedure is required for the sorting of all the transformed data in descending order from the furthest distance in the direction of depth of a display screen. As the sorting process takes much time, the above described general sorting method is not appropriate for video game machines for which high speed

processing is required.

**[0007]** Further, it is of the same situation as the above regarding sorting data for polygons that the data sorted are read out and drawn in the a frame memory in the order that corresponds to their relative distances in the direction of the depth of a screen, beginning with data having the smallest distance.

**[0008]** In ACM Transactions on Graphics (Vol. 11, No. 1, January 1992, pages 1-11) an output sensitive algorithm for hidden surface removal in computer graphics is disclosed.

**[0009]** According to the method disclosed in the document the view of a scene is created by adding triangles in groups from bottom up in an incremental fashion. The data within each group is presented.

**[0010]** This method, however, does not solve the above-mentioned problems.

## SUMMARY OF THE INVENTION

**[0011]** It is, therefore, an object of the present invention to provide a sorting method for data for polygons that can reduce the size of the memory that is used and that can shorten the time required for sorting, a video game machine that can employ that method, a computer program which when running on a computer performs such a sorting method and a computer program product comprising such a computer program.

**[0012]** It is another object of the present invention to provide both a sorting method for data for polygons by which the order in which polygons are to be drawn in a frame buffer memory can be determined, and a video game machine that can employ that method.

**[0013]** It is an additional object of the present invention to provide both a sorting method for data for polygons by which parallel processing can be performed when a DSP or a plurality of processors is employed, and a video game machine that can use that method.

**[0014]** It is a further object of the present invention to provide both a sorting method for data for polygons, by which not only can drawing data be registered in a video RAM but also a linked list can be formed, using hardware jump commands, for which merely a block table need be prepared in a work RAM, that enables the generation of polygon drawing commands while a sorting process is being performed, and a video game machine that can employ that method.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0015]**

Fig. 1 is a diagram for explaining the principle of the present invention;

Fig. 2 is a block diagram illustrating a machine according to one embodiment for which the present invention is applied;

Fig. 3 is a flowchart showing the processing for a

method according to one embodiment of the present invention;

Fig. 4 is a diagram illustrating an example where polygon data are divided into groups;

Fig. 5 is a diagram showing a distance example for the data for polygons in the direction of a Z1 axis;

Figs. 6A and 6B are explanatory diagrams for data for polygons that correspond to Figs. 4 and 5;

Fig. 7 is a flowchart showing the procedures for registering a linked list;

Figs. 8A and 8B are diagrams for explaining registration state 1 of a linked list;

Figs. 9A and 9B are diagrams for explaining registration state 2 of a linked list;

Figs. 10A and 10B are diagrams for explaining registration state 3 of a linked list;

Figs. 11A and 11B are diagrams for explaining registration state 4 of a linked list;

Fig. 12 is a flowchart showing a linked list connection;

Fig. 13 is a diagram for explaining the sorting results for registered linked list; and

Fig. 14 is an explanatory diagram for the drawing of polygons No. 1 through No. 10.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

**[0016]** Before the preferred embodiments of the present invention are described, the principle of the present invention will now explained, while referring to Fig. 1, in order that the present invention may be correctly understood. The same reference numbers and symbols are used to denote identical or corresponding components in the following drawings.

**[0017]** In Fig. 1, A through E denote a plurality of blocks (BLKs) that are grouped together. The entries No. 1 through No. 10 denote data for those polygons that are to be displayed, with the numerals 1 through 10 corresponding to the order in which data are read from a game program and processed.

**[0018]** Data for polygons that are read from the game program are grouped in accordance with the distance of the polygons between two points: a predetermined coordinate point and a coordinate position for each polygon. The data groups are registered to corresponding blocks A through E. For example, the data for polygons No. 1, No. 3, and No. 10 are registered as belonging to block A.

**[0019]** In the process by which polygons and the data for them, which are assembled into groups, are registered in corresponding blocks, the registration of link connection (I) is performed in the blocks in accordance with the order in which the polygons and the data for them are registered. For example, in block A, the link connection is performed in the order that begins with the data for polygon No. 1 followed by the data for polygon No. 3 and then No. 10.

**[0020]** When the data for all the polygons have been grouped and registered in the corresponding blocks, the registration of the linked connection (II) between the blocks is performed.

**[0021]** As is shown in Fig. 1, polygon data No. 8, which has been registered last in block E, is linked to polygon data No. 6, which has been registered first in block D. Polygon data No. 7, which has been registered last in block D, is linked to polygon data No. 2 in block C. Polygon data No. 2 in block C is linked to polygon data No. 4, which has been registered first in block C. Polygon data No. 9, which has been registered last in block B, is linked to polygon data No. 1, which has been registered first in block A.

**[0022]** When all the polygon data have been sorted in the above described manner, the data for the polygons are read out in the order in which they are linked together and the polygons are drawn in a frame buffer memory.

**[0023]** In Fig. 1, the linked connection (II) between the blocks are registered in the order of the distances between two coordinate positions of data for polygons. Namely, the order of drawing data for polygons begins with the group for which the distance between the two positions is the greatest, in the case that the data are read out and drawn in a frame buffer memory in the order of the linked connections.

**[0024]** However, the order of drawing data for polygons in a frame buffer memory according to the present invention is not restricted to the above, and it is also possible to read out and draw the data for polygons in the frame buffer memory in the order beginning with the group for which the distance between the two positions is the smallest.

**[0025]** In this case, it is not necessary to draw again data on the portions in a frame buffer memory, which have been already drawn and therefore, the speed of drawing the data in the frame buffer memory may become fast.

**[0026]** Fig. 2 is a block diagram illustrating a video game machine, according to one embodiment, that employs a method of the present invention according to the above described principle. In Fig. 2, a CPU 1 executes a game program that is stored in a ROM 2.

**[0027]** The CPU 1 performs a transformation process where data for polygons at three-dimensional coordinates (referred to as second three-dimensional coordinates in this application), which are represented, for example, by global coordinates that will be described later, are transformed into other three-dimensional coordinate data (referred to as first three-dimensional coordinates in this application); and executes a sorting process. For an application of the present invention, the coordinate system employed for the first and the second three-dimensional coordinates may be either the polar or the spherical coordinate system.

**[0028]** In the ROM 2 are a control program for executing a game, and data for polygons that are represented by, for example, the second three-dimensional coordi-

nates.

**[0029]** The polygon data include XYZ coordinate data for the vertexes of the polygons on the second three-dimensional coordinates (i.e., for a polygon with four vertexes as an example, XYZ coordinate data that correspond to the individual vertexes of the polygon), color data (only luminance data when it is for monochrome), and a coordinate transformation parameter, which is employed to form a transformation expression for transforming second three-dimensional XYZ data into first three-dimensional X1Y1Z1 data, while a specific point, such as the point of view of a player, is employed as an origin.

**[0030]** In a work RAM 3, data are temporally stored while a game is being executed. Also, a table (hereafter referred to as a "BLK-TBL") is provided for performing the method of the present invention that will be described later.

**[0031]** An input device 4, such as a joy stick, is operated by a game player to input required data when the controlled shifting of a sprite is involved, etc.

**[0032]** In a video RAM (VRAM) 5 are stored polygon data that are obtained by transforming their coordinates into the first three-dimensional coordinates using the method of the present invention. The data for polygons that are stored in the video RAM 5 are X1Y1Z1 coordinate data for the vertexes of a polygon that is to be displayed, and color data (only luminance data are required for a monochrome display). A command table is also provided in the video RAM 5 to perform the method of the present invention, which will be described later.

**[0033]** A video display processor (VDP) 6, which is activated at the time image frames are switched, processes data, for polygons, that are stored in the video RAM 5, and draws figures in a frame buffer memory (FB) 7 in consonance with the pixel data for one screen.

**[0034]** The frame buffer memory 7 has two pages. Pixel data for one screen that are written and stored in one memory page are read out in order and displayed in a display device 9. During this period, pixel data for one page are written and stored in the other memory page.

**[0035]** In other words, the two pages of the frame buffer memory 7 are used for alternately reading and writing data, so that sequential screens can be displayed on a monitor.

**[0036]** Data for polygons for one screen that are stored in the frame buffer memory 7 serve as data for the individual pixels (dots).

**[0037]** A video signal generator 8 reads the pixel (dot) data that are written to and stored in the frame buffer memory 7 and converts these data into video signals. The video signals from the video signal generator 8 are transmitted to the display device 9, such as a CRT.

**[0038]** According to the principle involved in the display of a plurality of polygons, which constitutes a display screen, on a two-dimensional display device 9, such as a CRT, in the frame buffer memory 7 polygons

are sequentially drawn and drawn over beginning with that polygon that is positioned the furthest from the first three-dimensional coordinate origin, for example, from the point of view of a player. This is done because it prevents a nearer polygon from being hidden by a polygon that is more distant in the direction of the depth of a screen.

**[0039]** Fig. 3 is a flowchart showing the processing, for one embodiment according to the method of the present invention, by which is provided such a display. During an initialization process (step S1), performed first is coordinate calculation, by which polygon data of the second three-dimensional coordinates are transformed, with the use of transformation parameter, into the first three-dimensional coordinates that have the point of view of a player as an origin; then a sorting process; and after that the setting of the video RAM 5 to its the initial state.

**[0040]** After the initialization, the second three-dimensional coordinates are transformed into the first three-dimensional coordinates (step S2). In this transformation, which is performed under the control of the CPU 1, polygon data, such as global coordinates, which are the second three-dimensional coordinates that are included in a game program, are transformed, by using a transformation parameter that is also included in the game program, into data for polygons for the first three-dimensional coordinate system that employs the point of view of a player as an origin.

**[0041]** Following in sequence, block identification is performed (step S3). The contents of the block identification (step S3) are as follows. According to the present invention, data for polygons that are obtained by performing the transformation into the first three-dimensional coordinates are grouped according to distance in the direction of, for example, the Z1 axis.

**[0042]** At this time, it is possible to employ a method that adopts the average value of the distances of a plurality of vertexes of individual polygons in the direction of the Z1 axis, or a method that adopts the representative distance value of one of the vertexes, i.e., the shortest distance value for a vertex in the direction of the Z1 axis.

**[0043]** Fig. 4 is a table for explaining a group example where data for polygons are divided into a plurality of blocks (A through E) according to different distances in the direction of the Z1 axis. That is, distance 0 to 1 in the direction of the Z1 axis is group A, distance 1 to 2 is group B, distance 2 to 3 is group C, distance 3 to 4 is group D, and distance 4 to 5 is group E.

**[0044]** Fig. 5 is a table for explaining an example of distance data, where ten polygons (No. 1 through No. 10) are supposed to be displayed and their distances in the direction of the Z1 axis are shown. For example, polygon 1, which has a distance of 0.5 in the direction of the Z1 axis, belongs to group A (see Fig. 4). Similarly, polygon 6, which has a distance of 3.3 in the direction of the Z1 axis, belongs to group D.

**[0045]** As described above, the distances of the polygons in the direction of the Z1 axis are obtained by employing, for all the polygons in common, the average of the distances of a plurality of vertexes of each polygon in the direction of the Z1 axis, or the shortest distance of the vertex in the direction of the Z1 axis.

**[0046]** Figs. 6A and 6B are diagrams for explaining data for polygons relative to Figs. 4 and 5. Fig. 6A is a diagram in which polygons No. 1 through No. 10, which have four vertexes, are arranged in the order of their distances along the Z1 axis. Fig. 6B, a graph in which the vertical line indicates distances in the direction of the Z1 axis and the horizontal line indicates the positions of polygons, shows the relationship between the distances for the polygons in the direction of the Z1 axis, which are explained while referring to Fig. 5, and groups A through E.

**[0047]** The order of the numerals of polygons No. 1 through No. 10 corresponds to the order of the data for polygons that are read by the CPU 1 from a game program that is stored in the ROM 2, and are transformed into the first three-dimensional coordinates by employing the transformation parameter, which is included in the game program.

**[0048]** After the grouping process by block identification has been performed (step S3, see Fig. 3), linked list registration is performed (step S4).

**[0049]** Fig. 7 is a flowchart showing the detailed procedures for registering the linked list (step S4). Figs. 8A through 11B are diagrams for explaining the statuses of the registration in the video RAM 5 and the work RAM 3 during the linked list registration (step S4).

**[0050]** In Figs. 8A, 9A, and 10A are shown the contents that are to be registered in a command table that is prepared in the video RAM 5. Addresses in the video RAM 5 (VRAM Adr.), the number of the data (Data No.), for a polygon, that are to be stored at that address, and linking destination addresses are indicated.

**[0051]** In Figs. 8B, 9B, and 10B are shown the contents of the BLK-TBL that is stored in the work RAM 3. Among the data, for polygons, that are classified and registered in blocks A through E, the data that are registered first and the data that are registered last are employed as the start data and as the end data, and their corresponding addresses are stored in the video RAM 5.

**[0052]** Referring back to Fig. 7, the data, for polygons, that are transformed into the first three-dimensional coordinates are first registered in the command table in the video RAM 5 (step S41). It is presumed that the address in the video RAM 5 at which the polygon data are registered is Adr1, and the block of the classified group is blk1.

**[0053]** When the example shown in Figs. 8A and 8B is referred to for the above data, for polygons, the data is for the first polygon, No. 1, and the address Adr1 that is stored in the video RAM 5 is V1. The block blk1 of the classified group belongs to block A because the distance for polygon No. 1 in the direction of the Z1 axis is

0.5.

**[0054]** Then, as the Data No. that corresponds to address V1 in the (VRAM Adr.) column of the command table of the video RAM 5 in Fig. 8A, a "1" is registered. It is therefore indicated that the data for the first polygon, No. 1, has been registered and is stored at address V1 in the video RAM 5.

**[0055]** For registration of linking destinations in the command table, all data for polygons are divided into groups, and until the data are registered, the linking destinations are determined only within the individual groups in the order of registration. Since there is no linking destination in the block A group at the time the data for the first polygon, No. 1, is registered, "none" is then entered in the Linking Destination column.

**[0056]** It should be noted that coordinate data of vertexes for each polygon and color data are included in the data for polygons that are to be registered and stored in the video RAM 5.

**[0057]** Following this, the End column that corresponds to block A for the BLK-TBL in the work RAM 3 is referred to (step S42). As the data for polygon No. 1 is the data that is first registered, the End column that is referred to is accordingly "none."

**[0058]** When the corresponding End column in the BLK-TBL in the work RAM 3 is referred to and "none" is indicated there, Adr1 is registered in the Start column and the End column of the BLK-TBL (step S43). As Adr1 corresponds to V1, as previously described, V1 is entered in the Start and the End column for block A, as is shown in Fig. 8B.

**[0059]** Tables in Figs. 9A and 9B show the status where the data (Data No.: 2) for the second polygon, No. 2, is registered. This registration status is entirely the same as that when the data (Data No.: 1) for the first polygon, No. 1 is registered in the command table of the video RAM 5 and the address is entered in the Start and the End column of the BLK-TBL of the work RAM 3, as explained while referring to Figs. 8A and 8B.

**[0060]** The only difference here is that the data for the second polygon belongs to the group in block C and is stored at address V2 in the video RAM 5.

**[0061]** In Fig. 9A, therefore, "2" is entered as the Data No. that corresponds to address V2 in the (VRAM Adr.) column in the command table, and "none" is entered in its Linking Destination column. Further, in Fig. 9B, V2 is registered in the Start and the End column that correspond to the group in block C in the BLK-TBL of the work RAM 3.

**[0062]** Next, the data for the third polygon, No. 3, which is determined to belong to group A by referring to Figs. 4 and 5, is processed. Following the procedures for the registration of a linked list in Fig. 7, first, "3" is registered as data No. that corresponds to address V3 (which corresponds to Adr1 in this case) in the (VRAM Adr.) column of the command table in the video RAM 5, as is shown in Fig. 10A (step S41). It is therefore indicated that the third polygon, No. 3, has been registered

and is stored at address V3 in the video RAM 5.

**[0063]** Since it is determined that polygon No. 3 belongs to group A by referring to Figs. 4 and 5, the End column for block A in the BLK-TBL in the work RAM 3 is referred to (step S42). The state of the End column for block A that is referred to is as shown in Fig. 9B, i. e., address V1 (which corresponds to Adr2 in this case) has been registered.

**[0064]** After the process at step 42, therefore, the process at step 44 is performed. More specifically, at step S44, address V3, which corresponds to Adr1, is entered in the Linking Destination column for address V1, which corresponds to Adr2, of the command table in the video RAM 5 (Fig. 10A). At the same time, the entry for block A in the End column of the BLK-TBL in the work RAM 3 is updated from address V1 to V3 (Fig. 10B).

**[0065]** The above process is repeated until the data for all the polygons are registered. A check is then performed to determine whether or not all the polygons have been registered (step S5). When all the polygons have not yet been registered, program control returns to step S2 and the above registration process is repeated.

**[0066]** The tables in Figs. 11A and 11B show the registration status of the data (Data No.: 10) for the last, tenth polygon, No. 10. The distance data for polygon No. 10 in the direction of the Z1 axis is 0.3, and the data for polygon No. 10 thus belong to group A (see Fig. 3).

**[0067]** Since there has been no registration of polygon data that belong to block A since the data for the third polygon, No. 3, was registered, address V10 is entered in the Linking Destination column that corresponds to address V3 in the video RAM 5. As the data for the tenth polygon, No. 10, are the last data to be registered, and as there is no linking destination, a linking destination address is not registered in the Linking Destination column that corresponds to address V10 in the video RAM 5 in Fig. 11B.

**[0068]** The entry of the End address column in the video RAM 5, which corresponds to block A in the work RAM 3 in Fig. 11B, is updated to address V10 during the process at step S44, in the same manner as was previously explained for the third polygon, No. 3.

**[0069]** When all the data entries for the polygons have been completed in the above described manner, the connection of the blocks to form a linked list is performed (an address for the data for a polygon that is the first that is registered in a block, the block having a distance in the direction of the Z1 axis that is shorter by one rank than the distance of the immediately preceding block, is entered in the Linking Destination column that corresponds to the address in the video RAM 5, wherein data for polygons are registered at the end of each block) (step S6).

**[0070]** Fig. 12 is the detailed flowchart for explaining the procedures involved in the formation of a linked list (step S6). For these procedures, a software process is executed by the CPU 1 by referring to the contents of

the BLK-TBL in the work RAM 3.

**[0071]** A start address is registered for a block that has the longest distance in the direction of the Z1 axis in the BLK-TBL of the work RAM 3, at the first linking destination in a command table of the video RAM 5, that is, the Linking Destination column corresponding to the address V0 (step S61).

**[0072]** In the BLK-TBL shown in Fig. 11B when the registration of all the polygon data is completed, the block E has the longest distance in the direction of the Z1 axis, and its corresponding start address in the video RAM 5 is V5. Therefore, address V5 is registered in the Linking Destination column for address V0, as is shown in Fig. 13.

**[0073]** Then, the contents of the start column for the block nearer next to the one being processed in BLK-TBL are registered at the linking Destination of data on the command table in the video RAM 5, which is pointed to by the END column of the being processed block (step S62).

**[0074]** More specifically, supposing that block E is being processed, the data address that is entered in the End column for block E in the BLK-TBL, and as a linking destination in the command table in the video RAM 5, is V8 (see Fig. 11B). The block nearer next to block E in the BLK-TBL is D, and its start address is V6.

**[0075]** V6 is therefore registered in the Linking Destination column that corresponds to address V8 in the video RAM 5 (see Fig. 13).

**[0076]** After the process has been completed up to the final block, END (termination of drawing) is registered in the Linking Destination column of the command table in the video RAM 5 for the address that is indicated in the End column of the nearest block. More specifically, as is shown in the contents of the BLK-TBL in Fig. 11B, the nearest block is block A and in the End column for block A in BLK-TBL is entered V10 (see Fig. 11B).

**[0077]** Thus, END (termination of the drawing) is registered for address V10 in the Linking Destination column of the command table in the video RAM 5.

**[0078]** A linked list, which is sequentially arranged in the descending order of the distances involved, is constructed as described above, and the sorting process is completed.

**[0079]** This process is also performed by the CPU 1. The connections for a linked list can be facilitated by using a jump command that the video display processor (VDP) 6 understands.

**[0080]** As is apparent from Fig. 13, when the sorting process is completed, address V5, at which is stored data No. 5 is registered in the Linking Destination column that corresponds to the initial address V0 in the video RAM 5.

**[0081]** Following the linked list of the ordered addresses in the video RAM 5, the video display processor (VDP) 6 reads the data for polygons for one screen that are stored in the video RAM 5, and transmits pixel data, into which the data have been converted, to the frame

buffer memory 7 for the drawing.

**[0082]** The pixel data for one screen that are drawn in the frame buffer memory 7 are repetitively read out, and converted into video signals by the video signal generator 8. The video signals are thereafter displayed on the display device 9 (step S7).

**[0083]** Figs. 14A and 14B are diagrams for explaining the results when polygons are drawn in the frame buffer memory 7 in the above described manner. The diagram in Fig. 14A shows the results obtained when the polygons are drawn. The polygons are drawn in the frame buffer memory 7 beginning with the one that has the largest distance in the direction of the Z1 axis, as is shown in Fig. 14B.

**[0084]** In other words, polygons are drawn and partially over-written, with the exception of No. 10, in order from No. 5 to No. 8, to No. 6, to No. 7, to No. 2, to No. 4, to No. 9, to No. 1, to No. 3, and to No. 10. As is shown in Fig. 14A, part of each of the polygons that are not over-written are displayed on the display device 9.

**[0085]** Furthermore, the order of drawing polygons in a frame buffer memory may be reversed according to the present invention. As explained above, as is apparent from Fig. 13, when the sorting process is completed, address V5, at which is stored data No. 5 that is at the largest distance in the direction of the Z1 axis, is registered in the Linking Destination column that corresponds to the initial address V0 in the video RAM 5. Accordingly, the video display processor (VDP) 6 reads the data for polygons for one screen, following the linked list of the reversely ordered addresses in the video RAM 5, and transmits the pixel data to the frame buffer memory 7 for drawing.

**[0086]** In this case, the polygons are drawn in the frame buffer memory 7 beginning with the one that has the smallest distance in the direction of the Z1 axis, as is shown in Fig. 14B. Namely, polygons are drawn in the order from No. 10 to No. 3, to No. 1, to No. 9, to No. 4, to No. 2, to No. 7, to No. 6, to No. 8, to No. 5, and it is necessary to indicate portions which have been drawn by indication flags so that the portions are not over-written with polygons that are drawn later. Accordingly, the total speed of drawing polygons in the frame buffer memory 7 becomes fast, as it will be not necessary to redraw the portions that have been already drawn.

**[0087]** When the present invention is applied for a video game machine, the drawing order for polygons in the frame buffer memory 7 should be discussed during the design phase for the polygon data. Drawing in the frame buffer memory 7 is performed beginning with a polygon in a group (block) that is the farthest in the direction of the Z1 axis.

**[0088]** The sorting of polygons based on their distances are not performed in the group. The speed of the sorting process can therefore be increased. The order in which the polygons in the group are drawn in the frame buffer memory 7 is optional. In the example shown in Figs. 14A and 14B, the polygons are first divided into

groups and then drawn in the frame buffer memory 7 by following the order in which the polygons are registered. However, the polygons in the group may be drawn in the opposite order to that in which the grouped polygons are registered.

**[0089]** As described above, according to the embodiment, while the conventional sorting method cannot determine the order for the polygons that are arranged in the direction of the Z1 axis unless all the data are prepared, the present invention merely determines which group each polygon belongs to and then registers it. It is therefore possible to sequentially determine the order in the direction of the Z1 axis for data that is to be processed.

**[0090]** Therefore, time required for sorting can be shortened and also the capacity of RAM is enough to be only for one polygon, that is used to store polygon data after the coordinates are transformed.

**[0091]** Especially when a DSP or a plurality of processors is employed, parallel processing can be performed.

**[0092]** In addition, according to the present invention, data to be drawn are registered in a video RAM, a block table is prepared in a work RAM, and then a linked list is formed by using hardware jump commands so that a sorting process can be performed, while polygon drawing commands are generated.

**[0093]** The above described embodiment has been provided for the explanation of the present invention; the present invention, however, is not limited to this embodiment. Variations based on ideas that are common to the present invention also fall within the scope of the present invention as claimed.

## Claims

1. A sorting method for data of polygons, that determines the order in which polygons (1-10) are to be drawn in a frame buffer memory (7), comprising the steps of:

grouping three-dimensional coordinates of the polygons (S3) in accordance with distances between a viewing origin and a representative distance value of the individual polygons and listing the groups (A - E) of polygons based on a distance from the viewing origin to the representative distance values,

registering and linking the polygons within each group (S4) beginning with a first polygon that is first listed in a corresponding group and ending with the polygon that is last listed in the corresponding group,

linking all the groups (S6), beginning with a group including polygons that have the greatest or smallest distance between the viewing origin and the representative distance value by linking

polygons, one of which is first registered into one group and the other of which is last registered into the other group, so that the groups are ordered; and  
writing into the frame buffer memory (7) the data for the video images in the order in which the polygons are registered and linked.

2. The method according to claim 1 wherein the grouping of the three-dimensional coordinates of the polygons is based on the average value of the distances between the viewing origin and the plurality of vertexes of individual polygons on a Z-axis of XYZ-coordinate system.
3. The method according to claim 1, wherein the grouping of the three-dimensional coordinates of the polygon is based on the shortest distance value for a vertex on a Z-axis of a XYZ-coordinate system.
4. The method according to one of claims 1 to 3, **characterized in that** the step of linking the groups begins with a group including polygons that have the greatest distance between the viewing origin and the representative distance value in order of decreasing distance from the viewing origin.
5. A method according to one of claims 1 to 3, **characterized in that** the step of linking the groups begins with the group including polygons that have the shortest distance between the viewing origin and the representative distance value in order of increasing distance from the viewing origin.
6. The method according to claim 4, further comprising the step of drawing (S7) the plurality of polygons in the frame buffer memory (7) according to the order of linking the groups and the order of linking the polygons in each of the groups.
7. The method according to claim 4, further comprising the step of drawing the plurality of polygons in the frame buffer memory (7) according to the order of linking the groups and the opposite order of linking the polygons in each of the groups.
8. The method according to one of claim 1 to 7, wherein the viewing origin is an imaginary point of view of a game player that is represented by the first three-dimensional coordinates.
9. The method according to one of claims 1 to 8, comprising the step of transforming polygon data that are included in a game program into said three-dimensional coordinates (S2).

10. A video game machine, which includes means for sorting data for polygons (1-10), that determines the order in which said polygons are to be drawn in a frame buffer memory (7), comprising

a CPU (1) executing a game program and which, during the execution of said game program is carrying out the method of claim 1;  
a frame buffer memory (7) storing drawing data that is to be displayed; and  
a video display processor (6) for the drawing, in said frame buffer memory (7), of polygons that are formed into groups (A-E) that are arranged in descending or ascending order beginning with said group for which said distance between the two points is the greatest or the smallest.

11. The video game machine according to claim 10, further comprising  
a work RAM (3) that includes a table in which, for each of said groups (A-E) of said polygons and for the data  
for said polygons, are registered a start address for a first polygon data, which is stored in a video RAM (5), and an end address for a last polygon data, which is stored in said video RAM (5), wherein said video RAM (5) includes a command table in which are registered said addresses, as linking destinations for data for said polygons, by referring to said table in said work RAM (3).
12. The video game machine according to claim 11, wherein said end address and said start address are so registered in said command table in said video RAM (5) that said end address for data for a polygon in one group, which is registered in said table in said work RAM, is linked with said start address for data for a polygon in the nearest group in direction of the viewing origin.
13. A computer program which when running on a computer performs a sorting method for data for polygons having the steps of one of claims 1 to 9.
14. A computer program product comprising program code means stored on a computer readable medium for performing the method of anyone of the claims 1 to 9 when said program product is run on a computer.

#### Patentansprüche

1. Sortierverfahren für Polygondaten, welches die Reihenfolge bestimmt, in der Polygone (1-10) in einem Bildpufferspeicher (7) gezeichnet werden sollen, mit den Schritten:



- Gruppieren dreidimensionaler Koordinaten der Polygone (S3) in Übereinstimmung mit Abständen zwischen einem Betrachtungsursprung und einem repräsentativen Abstandswert der individuellen Polygone und Listen der Gruppen (A-E) der Polygone basierend auf einem Abstand von dem Betrachtungsursprung zu den repräsentativen Abstandswerten, Eintragen und Verknüpfen der Polygone innerhalb jeder Gruppe (S4) beginnend mit einem ersten Polygon, welches zuerst in einer entsprechenden Gruppe aufgelistet ist und endend mit dem Polygon, welches als Letztes in der entsprechenden Gruppe aufgelistet ist, Verknüpfen aller Gruppen (S6), beginnend mit einer Gruppe, die Polygone enthält, die den größten oder kleinsten Abstand zwischen dem Betrachtungsursprung und dem repräsentativen Abstandswert haben, durch Verknüpfen von Polygonen, von denen eines zuerst in eine Gruppe eingetragen ist und von denen das andere zuletzt in die andere Gruppe eingetragen ist, so dass die Gruppen geordnet sind; und Schreiben in den Bildpufferspeicher (7) die Daten der Videobilder in der Reihenfolge, in der die Polygone eingetragen und verknüpft sind.
2. Verfahren nach Anspruch 1, wobei das Gruppieren der dreidimensionalen Koordinaten der Polygone auf dem Mittelwert der Abstände zwischen dem Betrachtungsursprung und der Mehrzahl von Ecken der individuellen Polygone auf einer Z-Achse eines XYZ-Koordinatensystems basiert.
3. Verfahren nach Anspruch 1, wobei das Gruppieren der dreidimensionalen Koordinaten des Polygons auf dem kürzesten Abstandswert für einen Eckpunkt auf einer Z-Achse eines XYZ-Koordinatensystems basiert.
4. Verfahren nach einem der Ansprüche 1 bis 3, **dadurch gekennzeichnet, dass** der Schritt des Verknüpfens der Gruppen beginnt mit einer Gruppe, die Polygone beinhaltet, die den größten Abstand zwischen dem Betrachtungsursprung und dem repräsentativen Abstandswert in der Reihenfolge eines abnehmenden Abstandes von dem Betrachtungsursprung haben.
5. Verfahren nach einem der Ansprüche 1 bis 3, **dadurch gekennzeichnet, dass** der Schritt des Verknüpfens der Gruppen beginnt mit der Gruppe, die Polygone beinhaltet, die den kürzesten Abstand zwischen dem Betrachtungsursprung und dem repräsentativen Abstandswert in der Reihenfolge eines zunehmenden Abstandes von dem Betrachtungsursprung haben.
6. Verfahren nach Anspruch 4, das weiter den Schritt des Zeichnens (S7) der Mehrzahl von Polygone in dem Bildpufferspeicher (7) entsprechend der Reihenfolge der Verknüpfung der Gruppen und der Reihenfolge der Verknüpfung der Polygone in jeder der Gruppen umfasst.
7. Verfahren nach Anspruch 4, das weiter den Schritt des Zeichnens der Mehrzahl von Polygone in dem Bildpufferspeicher (7) entsprechend der Reihenfolge der Verknüpfung der Gruppen und der entgegengesetzten Reihenfolge der Verknüpfung der Polygone in jeder der Gruppen umfasst.
8. Verfahren nach einem der Ansprüche 1 bis 7, wobei der Betrachtungsursprung ein imaginärer Beobachtungspunkt eines Spielers ist, der durch die ersten dreidimensionalen Koordinaten dargestellt ist.
9. Verfahren nach einem der Ansprüche 1 bis 8 mit dem Schritt des Transformierens von Polygondaten, die in einem Spielprogramm enthalten sind, in die dreidimensionalen Koordinaten (S2).
10. Videospielemaschine, die ein Mittel zum Sortieren von Polygondaten (1-10) beinhaltet, das die Reihenfolge bestimmt, in der die Polygone in einem Bildpufferspeicher (7) gezeichnet werden sollen mit einer CPU (1), die ein Spielprogramm ausführt und die während der Ausführung des Spielprogramms das Verfahren nach Anspruch 1 ausführt; einem Bildpufferspeicher (7), der den Zeichenwert, der angezeigt werden soll, speichert; und einem Videoanzeigeprozessor (6) zum Zeichnen von Polygonen in dem Bildpufferspeicher (7), die in Gruppen (A-E) geformt sind, die in absteigender oder aufsteigender Reihenfolge angeordnet sind, beginnend mit der Gruppe, für die der Abstand zwischen den zwei Punkten der Größte oder der Kleinste ist.
11. Videospielemaschine nach Anspruch 10, weiter mit einem Arbeits-RAM (3), welches eine Tabelle beinhaltet, in der, für jede der Gruppen (A-E) der Polygone und für die Polygondaten, eine Startadresse für einen ersten Polygondatenwert, der in einem Video RAM (5) gespeichert ist, und eine Endadresse für einen letzten Polygondatenwert, der in dem Video RAM (5) gespeichert ist, eingetragen sind, wobei das Video RAM (5) eine Befehlstabelle aufweist, in der die Adressen eingetragen sind als Verknüpfungsbestimmungen für Polygondaten, durch Bezug nehmen auf die Tabelle in dem Arbeits-RAM (3).
12. Videospielemaschine nach Anspruch 11, wobei die

Endadresse und die Startadresse derart in die Befehlstabelle in dem Video-RAM (5) eingetragen sind, dass die Endadresse für Daten für ein Polygon in einer Gruppe, die in der Tabelle in dem Arbeits-RAM eingetragen ist, mit der Startadresse für Daten für ein Polygon in der nächstliegenden Gruppe in Richtung auf den Betrachtungsursprung verknüpft ist.

13. Computerprogramm, das, wenn es auf einem Computer läuft, ein Sortierverfahren für Polygondaten mit den Schritten eines der Ansprüche 1 bis 9 ausführt.
14. Computerprogrammprodukt mit einem Programmcodemittel, das auf einem computerlesbaren Medium gespeichert ist, zum Ausführen des Verfahrens nach einem der Ansprüche 1 bis 9, wenn das Programmprodukt auf einem Computer läuft.

### Revendications

1. Procédé de tri des données de polygones à déterminer l'ordre par lequel des polygones (1 à 10) sont à dessiner dans une mémoire tampon d'images (7), comprenant les opérations suivantes:

groupement de coordonnées tridimensionnelles desdits polygones (S3) selon les distances entre une origine d'observation et une valeur de stance représentative des polygones individuels, et listage des groupes (A à E) de polygones à la base d'une distance de ladite origine d'observation vers lesdites valeurs de distance représentatives,

enregistrement et établissement des liens parmi lesdits polygones au-dedans de chaque groupe (S4), commençant à partir d'un premier polygone listé à la première position dans un groupe correspondant et se terminant au polygone listé à la dernière position dans le groupe correspondant,

établissement des liens pour tous les groupes (S6), commençant à un groupe renfermant des polygones dont la distance entre l'origine d'observation et la valeur de distance représentative est la plus grande ou la plus petite, moyennant des polygones de lien dont un est enregistré comme premier dans un groupe et dont l'autre est enregistré comme dernier dans l'autre groupe, de façon que tous les groupes soient ordonnés; et

écriture des données pour les images vidéo dans ladite mémoire tampon d'images (7) par l'ordre de l'enregistrement des polygones et d'établissement des liens parmi les polygones.

2. Procédé selon la revendication, dans lequel le groupement desdites coordonnées tridimensionnelles desdits polygones se base sur la valeur moyenne des distances entre l'origine d'observation et la pluralité de pointes des polygones individuels sur un axe des z d'un système de coordonnées XYZ.

3. Procédé selon la revendication 1, dans lequel le groupement desdites coordonnées tridimensionnelles du polygone se base sur la plus petite valeur de distance pour une pointe sur un axe des z d'un système de coordonnées XYZ.

4. Procédé selon une quelconque des revendications 1 à 3, **caractérisé en ce que** l'opération d'établissement des liens des groupes commence à partir d'un groupe renfermant des polygones à la plus grande distance entre l'origine d'observation et la valeur de distance représentative par l'ordre de réduction des distances à partir d'une origine d'observation.

5. Procédé selon une quelconque des revendications 1 à 3, **caractérisé en ce que** l'opération d'établissement de liens des groupes commence à partir du groupe renfermant des polygones à la plus petite distance entre l'origine d'observation et la valeur de distance représentative par l'ordre d'accroissement de la distance à partir d'une origine d'observation.

6. Procédé selon la revendication 4, comprenant en outre l'opération de dessiner (S7) la pluralité de polygones dans ladite mémoire tampon d'images (7) par l'ordre d'établissement de liens des groupes et par l'ordre d'établissement de liens des polygones au-dedans de chacun des groupes.

7. Procédé selon la revendication 4, comprenant en outre l'opération de dessiner la pluralité de polygones dans ladite mémoire tampon d'images (7) par l'ordre d'établissement des liens des groupes et par l'ordre inversé d'établissement des liens des polygones au-dedans de chacun des groupes.

8. Procédé selon une quelconque des revendications 1 à 7, dans lequel l'origine d'observation est un point d'observation imaginaire d'un joueur d'un jeu, qui est représenté par les premières coordonnées tridimensionnelles.

9. Procédé selon une quelconque des revendications 1 à 8, comprenant l'opération de transformation des données de polygones, qui sont comprises dans un programme de jeu, dans lesdites coordonnées tridimensionnelles (S2).

10. Appareil de jeu vidéo, comprenant des moyens de

tri de données de polygones (1 à 10), qui déterminent l'ordre par lequel lesdits polygones sont à dessiner dans une mémoire tampon d'images (7), comprenant

5

un ordinateur (1) exécutant un programme de jeu, qui, au cours de l'exécution dudit programme de jeu, effectue le procédé selon la revendication 1;

une mémoire tampon d'images (7) qui mémorise des données de dessin à afficher; et

10

un processeur d'affichage sur écran (6) pour le dessin, dans ladite mémoire tampon d'images (7), de polygones groupés en groupes (A à E) qui sont arrangés par ordre descendant ou ascendant, commençant à partir du groupe pour lequel la distance entre les deux points à la plus grande ou la plus petite valeur.

15

11. Appareil de jeu vidéo selon la revendication 10, comprenant en outre

20

une mémoire vive de travail (3) qui comprend une table à enregistrer, pour chacun desdites groupes (A à E) desdits polygones et pour les données desdits polygones, une adresse d'origine pour une première donnée de polygone, qui est mémorisée dans une mémoire vive vidéo (5), et une adresse finale pour une dernière donnée de polygone, qui est mémorisée dans ladite mémoire vive vidéo (5), dans lequel ladite mémoire vive vidéo (5) comprend une table de commande à enregistrer lesdites adresses comme destinations de lien pour des données desdits polygones, par référence à ladite table dans ladite mémoire vive de travail (3).

25

30

35

12. Appareil de jeu vidéo selon la revendication 11, dans lequel ladite adresse finale et ladite adresse d'origine sont enregistrées dans ladite table de commande dans ladite mémoire vive vidéo (5) de façon que ladite adresse finale pour des données d'un polygone d'un groupe, qui est enregistré dans ladite table dans ladite mémoire vive de travail, soit liée à ladite adresse d'origine pour des données d'un polygone dans le groupe le plus proche en direction de ladite origine d'observation.

40

45

13. Programme d'ordinateur qui, en exécution sur un ordinateur, effectue un procédé de tri de données de polygones aux opérations selon une quelconque des revendications 1 à 9.

50

14. Produit de programme d'ordinateur, comprenant des moyens de codage de programme, qui est mise en mémoire sur un support lisible par ordinateur, à effectuer le procédé selon une quelconque des revendications 1 à 9, quand ledit produit de programme est exécuté sur un ordinateur.

55

FIG. 1

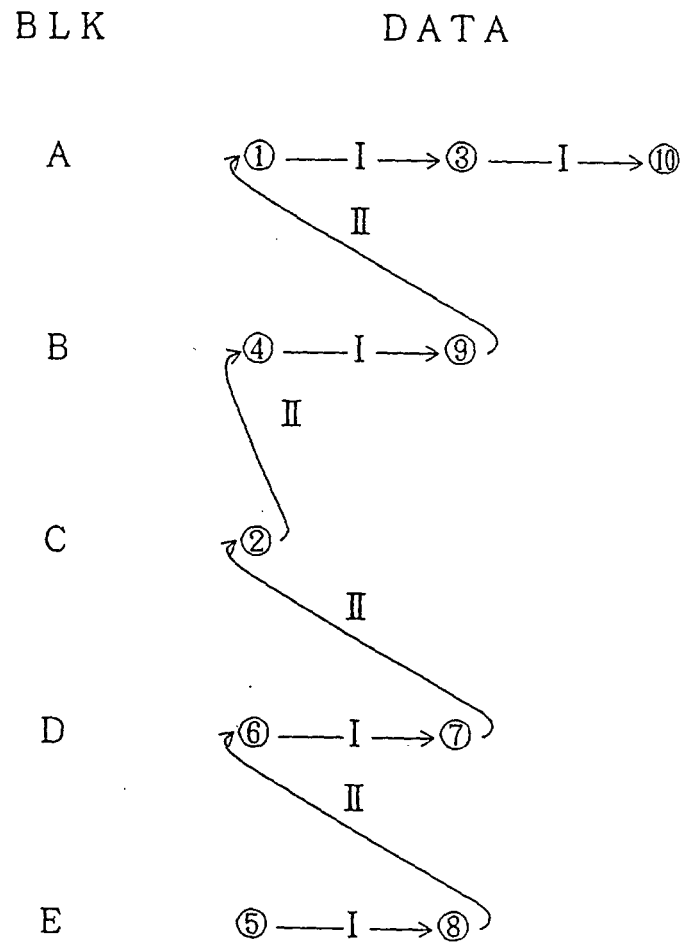


FIG. 2

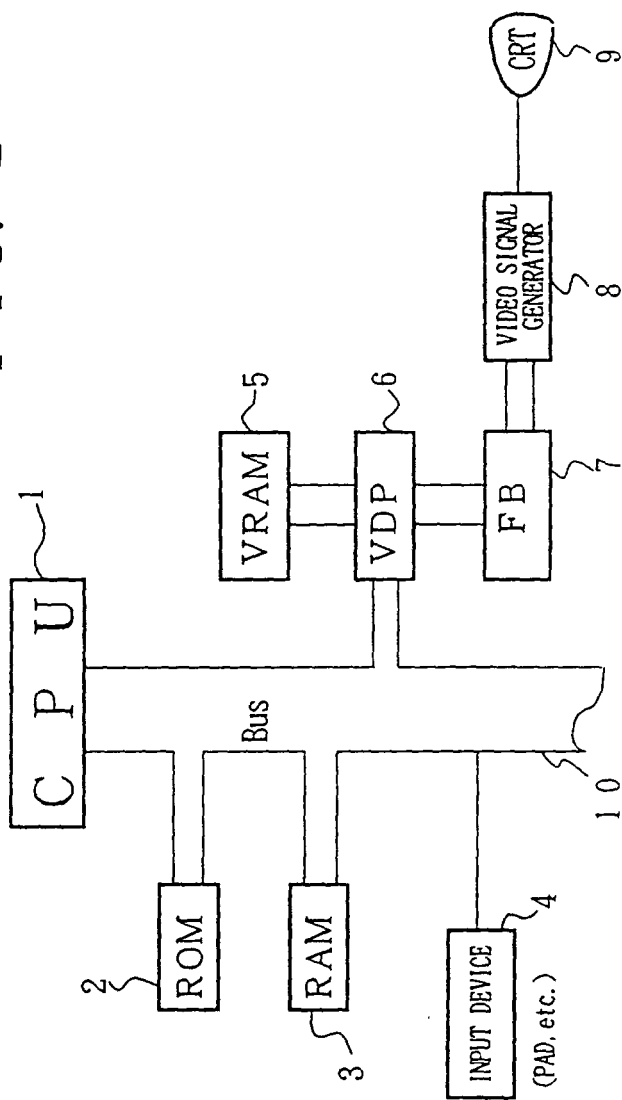


FIG. 3

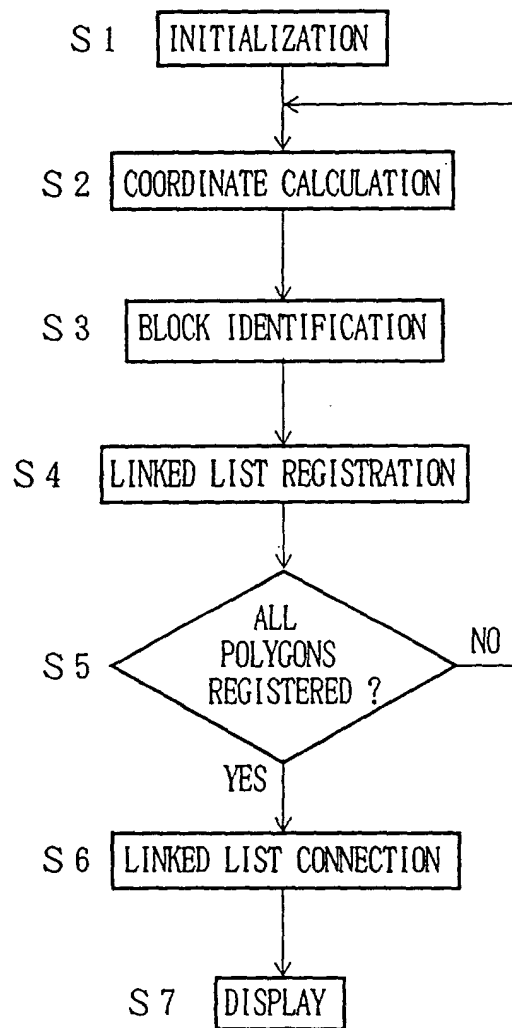


FIG. 4

GROUP	DISTANCE
A	0~1
B	1~2
C	2~3
D	3~4
E	4~5

FIG. 5

No.	DISTANCE
1	0.5
2	2.3
3	0.8
4	1.1
5	4.2
6	3.3
7	3.5
8	4.9
9	1.4
10	0.3

FIG. 6A

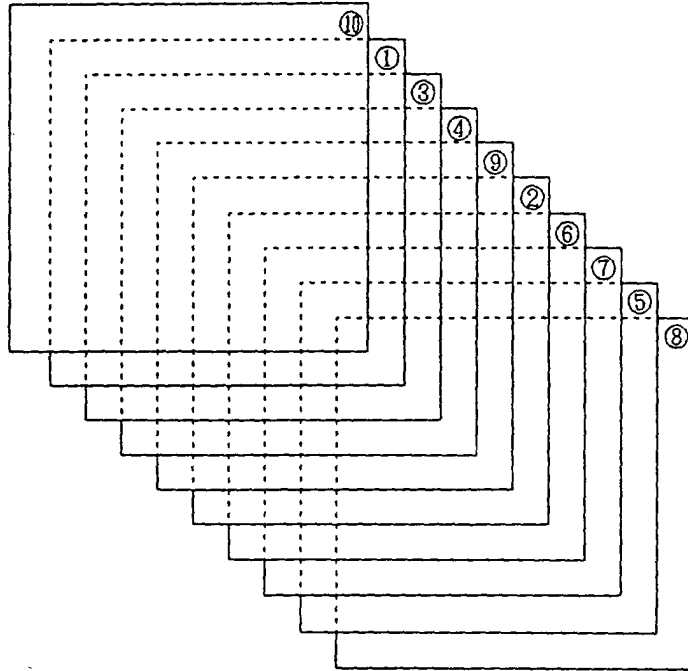


FIG. 6B

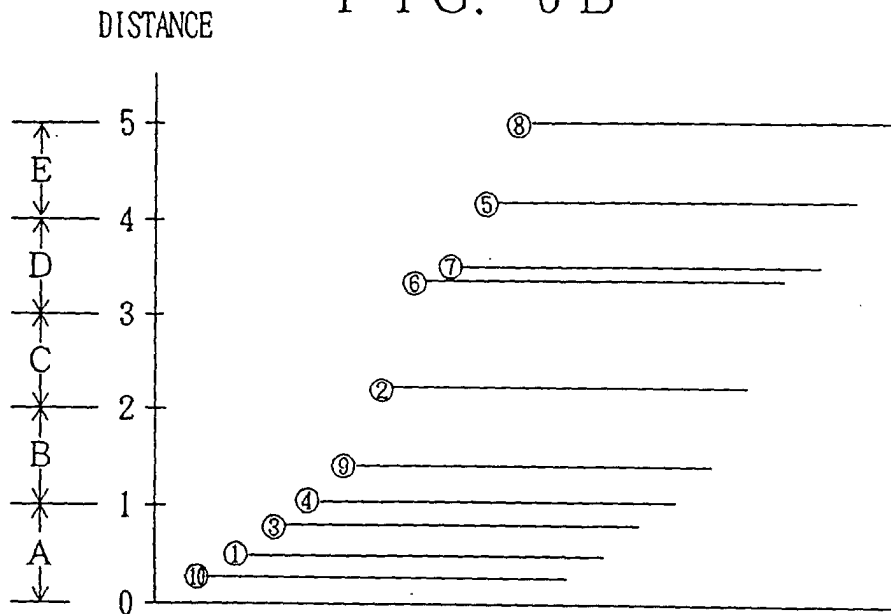




FIG. 7

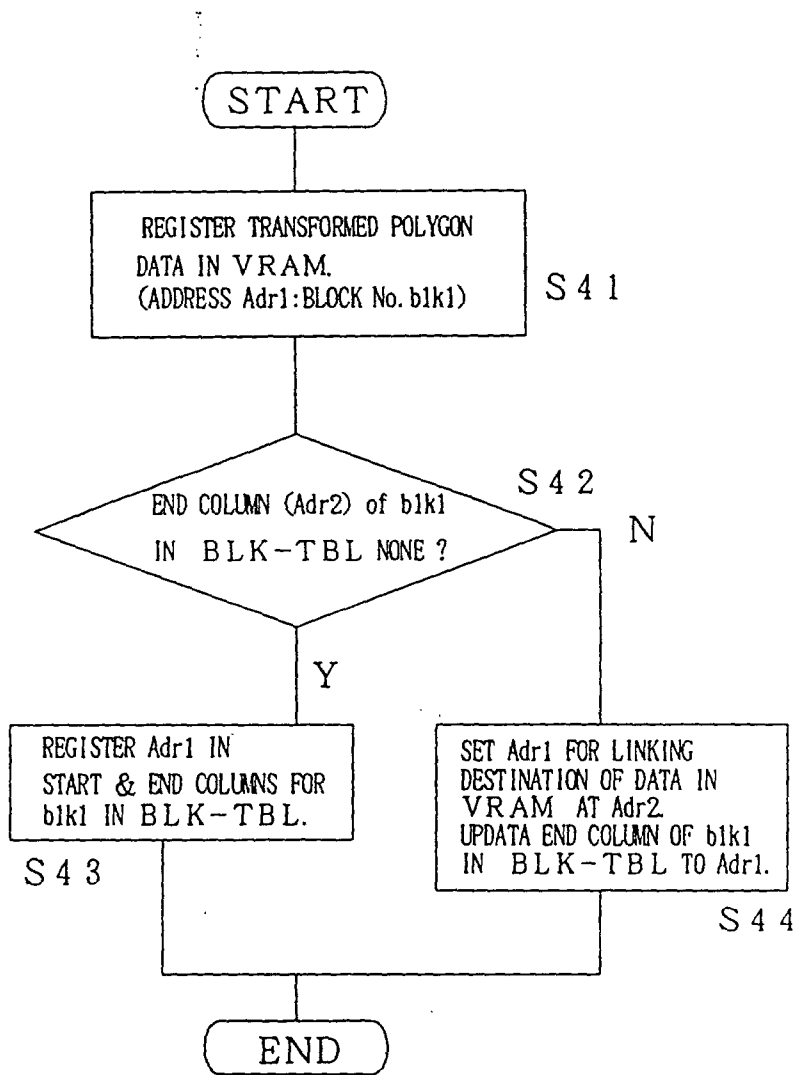


FIG. 8A

VRAM Adr.	DATA No.	LINKING DESTINATION
V0	none	none
V1	1	none

FIG. 8B

BLOCK	START (VRAM Adr. )	END (VRAM Adr. )
A	V1	V1
B	none	none
C	none	none
D	none	none
E	none	none

FIG. 9A

VRAM Adr.	DATA No.	LINKING DESTINATION
V0	none	none
V1	1	none
V2	2	none

FIG. 9B

BLOCK	START (VRAM Adr. )	END (VRAM Adr. )
A	V1	V1
B	none	none
C	V2	V2
D	none	none
E	none	none

FIG. 10A

VRAM Adr.	DATA No.	LINKING DESTINATION
V0	none	none
V1	1	V3
V2	2	none
V3	3	none

FIG. 10B

BLOCK	START (VRAM Adr.)	END (VRAM Adr.)
A	V1	V3
B	none	none
C	V2	V2
D	none	none
E	none	none

FIG. 11A

VRAM Adr.	DATA No.	LINKING DESTINATION
V0	none	none
V1	1	V3
V2	2	none
V3	3	V10
V4	4	V9
V5	5	V8
V6	6	V7
V7	7	none
V8	8	none
V9	9	none
V10	10	none

FIG. 11B

BLOCK	START (VRAM Adr.)	END (VRAM Adr.)
A	V1	V10
B	V4	V9
C	V2	V2
D	V6	V7
E	V5	V8

FIG. 12

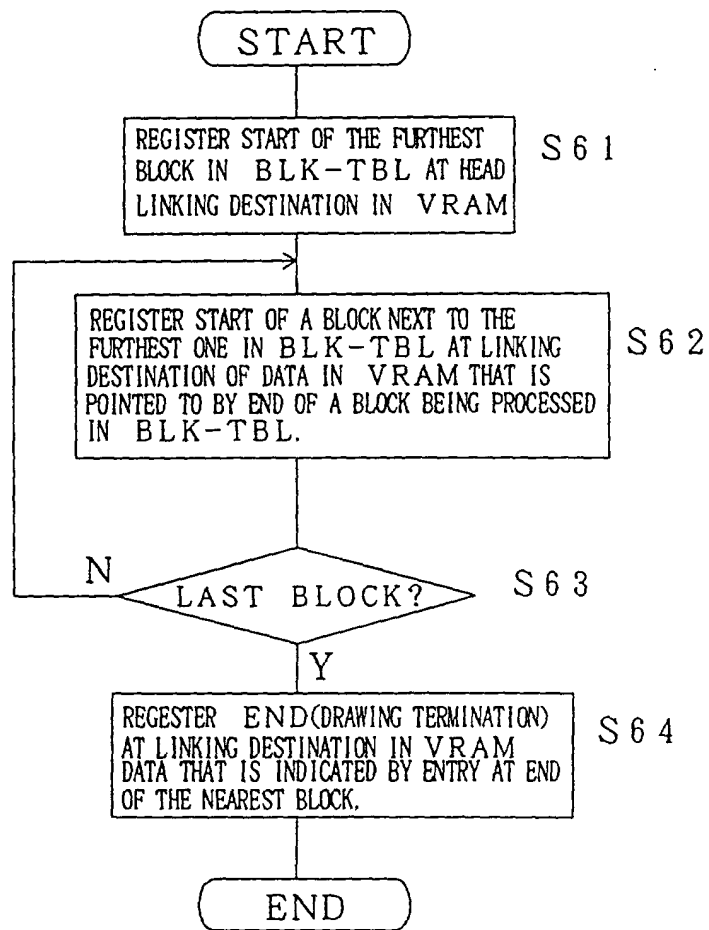


FIG. 13

VRAM Adr.	DATA No.	LINKING DESTINATION
V0	none	V5
V1	1	V3
V2	2	V4
V3	3	V10
V4	4	V9
V5	5	V8
V6	6	V7
V7	7	V2
V8	8	V6
V9	9	V1
V10	10	END

FIG. 14A

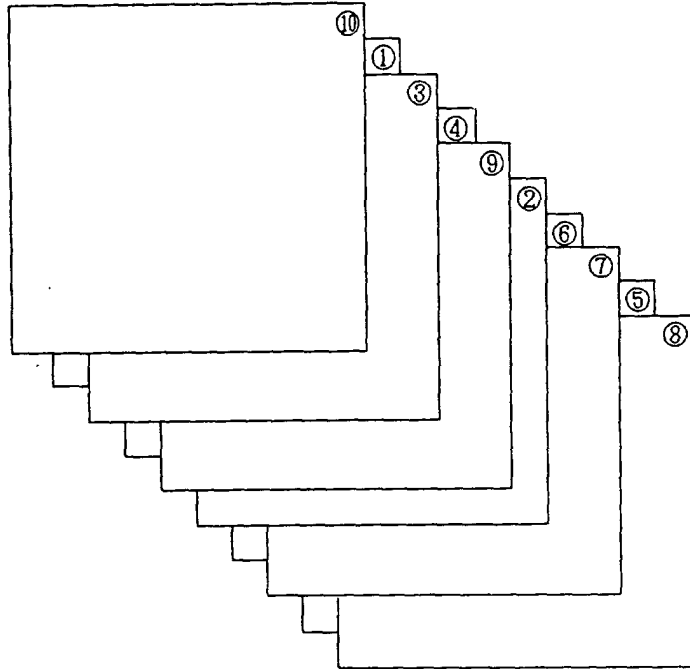


FIG. 14B

DRAWING ORDER

