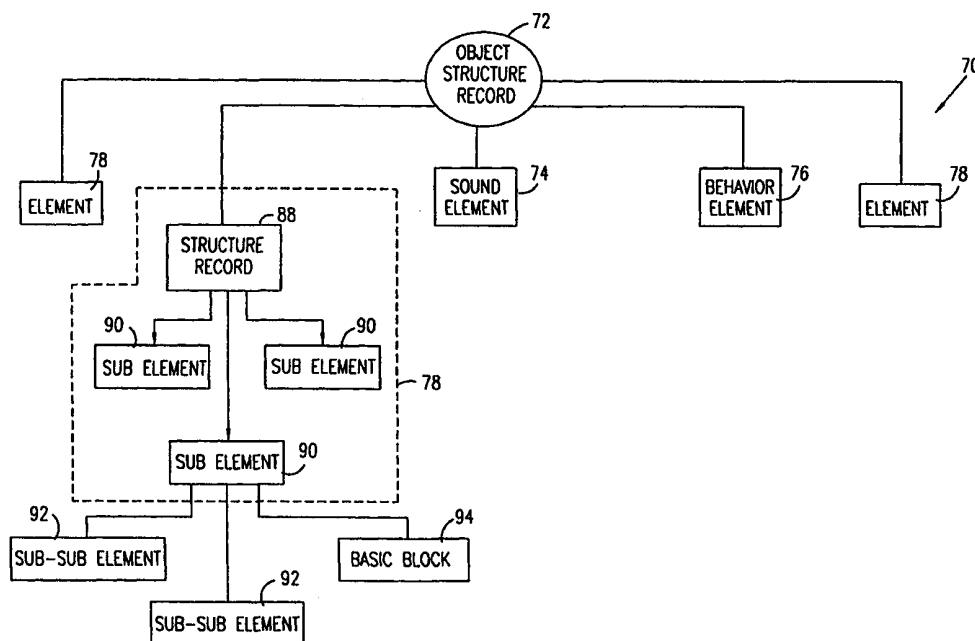




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06T 13/00, 15/70, G06F 15/173, 15/177, 17/30		A1	(11) International Publication Number: WO 00/34921
			(43) International Publication Date: 15 June 2000 (15.06.00)
(21) International Application Number: PCT/IL99/00544		Wormiza Street 10, 62642 Tel Aviv (IL). TAYAR, Gil [IL/IL]; Jabotinski Street 69, 53319 Givatayim (IL).	
(22) International Filing Date: 15 October 1999 (15.10.99)		(74) Agents: COLB, Sanford, T. et al.; Sanford T. Colb & Co., P.O. Box 2273, 76122 Rehovot (IL).	
(30) Priority Data: 09/209,385 10 December 1998 (10.12.98) US		(81) Designated States: AE, AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), DM, EE, EE (Utility model), ES, FI, FI (Utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).	
(71) Applicant (for all designated States except US): ZAPA DIGITAL ARTS LTD. [IL/IL]; Yehuda Halevi Street 14, 65137 Tel Aviv (IL).			
(72) Inventors; and			
(75) Inventors/Applicants (for US only): GEVER, Eyal [IL/IL]; Akiva Arye Street 27, 62154 Tel Aviv (IL). AVIGUR, Racheli [IL/IL]; Golomb Street 26, 44357 Kfar Saba (IL). AVRAHAM, Tomer [IL/IL]; Aminadav Street 16, 67066 Tel Aviv (IL). BERGMAN, Orit [IL/IL]; Komemiyut Street 49, 69011 Tel Aviv (IL). DOTAN, Sharon [IL/IL]; Bar Ilan Street 9, 54019 Givat Shmuel (IL). FRIEDMAN, Guy [IL/IL]; Shtand Street 7, 64374 Tel Aviv (IL). GILL, Doron [IL/IL]; Arlozorov Street 24, 53373 Givatayim (IL). HERMONI, Nir [IL/IL]; Bar Lev Street 8, 60408 Or Yehuda (IL). RESHEF, Eilon [IL/IL]; Melchet Street 31, 63825 Tel Aviv (IL). SEGAL, Eran [IL/IL]; Arba Aratsot Street 39, 62746 Tel Aviv (IL). SOLOMOVICI, Amira [IL/IL];			
		Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.	

(54) Title: FAST TRANSMISSION OF GRAPHIC OBJECTS



(57) Abstract

A method for rendering an animation object (70), including a plurality of sub-objects (74, 76, 78), on a display (30) of a client computer (26) under control of a remote server (24). Program code (72) which describes a structure of the object is transferred from the server to the client. A program description of at least one of the sub-objects of the object is accessed within a local memory (32) associated with the client, and the object is displayed such that the sub-object is coupled to the structure in accordance with the program code.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakistan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

FAST TRANSMISSION OF GRAPHIC OBJECTS

FIELD OF THE INVENTION

The present invention relates generally to computer animation, and specifically to interactive rendering of computer graphic objects from a remote source.

BACKGROUND OF THE INVENTION

In recent years, the use of the World Wide Web (WWW) has been rapidly growing. In use of the WWW, a client computer user connects to a Web site on a server and requests that the server download and display Web pages on the client computer. In response to the request, the server sends to the client computer graphic files, which include media elements (pictures, animations, texts, sound), scripts of instructions to be performed by an Internet browser program running on the computer, and in some cases executable software code, e.g., Java code. In addition, the server usually sends a file in HTML format which describes the layout of the computer graphic files on the screen. Generally, the Internet browser program operates independently of other programs on the client computer and, in particular, limits the access of scripts from the server to files stored in the memory of the client computer, and especially to graphic objects stored in the memory.

In order to attract users, sites must be interesting, and it is typically desired that sites display graphics and animation. However, the time required to download and display Web pages which include animation, including three-dimensional (3D) animation, on the client computer is relatively large, which is inconvenient to users and may discourage them from using such sites. Frequently it is desired to render animated advertisements along with user-chosen data from a Web site. Such advertisements must not require long transmission periods, or the visitor at the site may leave the site and be discouraged from returning.

Various compression methods have been suggested to reduce the data volume of a transmitted Web page and thus reduce the amount of time required to display the page on the computer. However, even using such compression methods, available network transmission bandwidth is typically still much lower than required to transmit large high quality animation streams. The users therefore must either settle for reduced quality of the animation stream or limit its size and/or length. In addition, some compression methods require lengthy decompression, which wastes much of the time saved by the compression.

Another method, referred to as caching or data pushing, described, for example, in U.S. Patent 5,727,129, which is incorporated herein by reference, includes "pushing" data in advance while the connection of the client computer to the network is idle. The computer, or the server, chooses data which may be of interest to the user and sends this data to the computer in advance of a specific request from the user, while the connection is idle. However, in many cases, there is no way to predict what the user will want to see. The number of Web pages which may be stored on the computer is limited, and the user may want updated versions of the page. Furthermore, there is not always an idle line available for significant periods of time.

Some Web browsers allow server-originated code to access the hard disk of the computer running the browser. Such access is normally performed using programs known as "plug-ins" or ActiveX controls. In most cases, when such access is attempted, the user of the computer is typically warned of a potential security violation, arising from such access.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide improved methods and apparatus for fast rendering of an animation object, preferably a three-dimensional (3D) object, on a remote computer graphic display.

In preferred embodiments of the present invention, an animation image of an object comprises a set of sub-objects (also referred to herein as elements) which are arranged in a predetermined hierarchy and animated responsive to the hierarchy. A structure record (also referred to as a "skeleton") describes relationships between the sub-objects in the hierarchy. Preferably, the structure record is written in accordance with a common script language, such as JavaScript. A plurality of sub-object records describe the sub-objects. At least some of the sub-object records are stored in a local memory associated with a client computer.

When the animation image is to be rendered on a display associated with the client computer, a server sends the structure record describing the animation image to the client computer, together with required sub-object records, if any, which are not stored in the client computer's local memory. Typically, only the structure record and a minimal number of sub-object records need be sent from the server. Thus, in most cases remote rendering of animation objects requires relatively low bandwidth, since the heavy graphical parts of the objects are stored permanently in the memory associated with the client computer.

In the context of the present patent application, the term "object" is used to refer to a modular, graphic and/or mathematical description of an image (generally a representation of a physical object, which may be naturalistic or fanciful). The skeleton of the object preferably defines the parts of the object and the joints connecting those parts. The hierarchical structure of objects allows a user to exchange sub-objects between objects that share a common hierarchy.

In some preferred embodiments of the present invention, the object represents a human, humanoid or animal character, and the sub-objects describe organs (such as limbs) and sub-organs (such as parts of the limbs) of the character. Other optional sub-objects describe attributes such as sound and rules governing behavior of the object, such as motions of the image.

Preferably, the objects also include sub-objects that define their response to sensitivity conditions that may arise externally to the object itself, such as the presence of other objects in the computer display. Furthermore, objects may include sub-objects that have learning capabilities, as are known in the art, whereby the behavior of the object is modified adaptively in response to inputs from the computer-animated environment. Preferably, some of the sub-objects are themselves formed of other sub-objects. In a preferred embodiment of the present invention, the objects comprise Smart Objects as are described in PCT publication WO97/35280, which is incorporated herein by reference.

Preferably, the client computer runs a scene manager program, which displays the animated images on a screen associated with the client computer. The scene manager is preferably part of a Web browser, which downloads graphic pages from a network. The client computer preferably also runs an element manager program, which keeps track of the locations of the records of sub-objects stored on the client computer. The scene manager interacts with the element manager to properly render the object using the received records and the stored records.

Preferably, the element manager maintains a look-up table which is used to find the location in the memory of sub-objects referenced by the received structure record. Most preferably, the element manager receives the name of a desired sub-object and returns a pointer to the stored location of the object based on the name. If the desired sub-object is not held in the memory of the client computer, the element manager returns a negative answer. Alternatively or additionally, the element manager orders the requested element from the

server and provides the scene manager with a pointer to an input directory in which received sub-objects are stored.

5 In some preferred embodiments of the present invention, the element manager is included within the Web browser. Thus, the user does not need to install any "plug-in" to the browser. Preferably, the browser as supplied to the client computer includes a set of records describing commonly-used sub-objects. Most preferably, the browser and set of records are pre-installed in a computer with dedicated three-dimensional graphics capabilities.

10 In some preferred embodiments of the present invention, when a sub-object not already in the memory is received by the client computer, the sub-object is stored in the memory and its address is added to the look up table.

15 In some preferred embodiments of the present invention, when a server instructs the client computer to render a new object, the server sends the computer a list of the sub-objects to be used in constructing the transmitted object, and the computer responds with a list of the sub-objects which the computer has and/or with a list of the sub-objects which the computer does not have. The server responds by sending the sub-objects missing from the client computer. Preferably, while waiting for the computer's response, the server sends the structure record and/or records describing sub-objects which are least likely to be already stored in the memory of the computer.

20 Preferably, when a computer responds that it does not have a copy of a sub-object which itself is formed of sub-sub-objects, the server sends a second list which lists the sub-sub-objects which form the sub-object and requests that the computer identify those sub-sub-objects which it does not have. This process may continue for the sub-sub-objects until there are no further sub-objects or until the records describing the sub-objects are so small that it is simpler to send the records, rather than to ask if they are needed. Alternatively or additionally, 25 the server only asks the client computer about sub-objects for which there is a high probability that the computer has records describing its sub-sub-objects.

30 Alternatively or additionally, the client computer determines the required sub-objects from the received structure record. Preferably, the JavaScript of the structure record describes the animation conditionally, based upon the presence or lack of sub-objects in the client computer. The JavaScript preferably uses "If-Then" directives to reference the sub-objects, wherein a referenced sub-object is first accessed in the local memory using the element

manager, and if it is not found in the local memory, the sub-object is requested from the server either by the scene manger or by the element manager.

There is therefore provided, in accordance with a preferred embodiment of the present invention, a method for rendering an animation object, including a plurality of sub-objects, on a display of a client computer under control of a remote server, including:

transferring from the server to the client, program code which describes a structure of the object;

accessing within a local memory associated with the client a program description of at least one of the sub-objects of the object; and

displaying the object such that the sub-object is coupled to the structure in accordance with the program code.

Preferably, the method includes reporting to the server the identity of at least some of the sub-objects stored in the memory. Most preferably, reporting the identity of the sub-objects includes receiving a list of sub-objects from the server and reporting on a correspondence between the list and the sub-objects that are stored in the memory.

Further preferably, the method includes transferring from the server to the client descriptions of sub-objects not already stored in the memory and storing the transferred descriptions in the memory. In a preferred embodiment, at least some of the sub-objects include sub-sub-objects, and transferring descriptions of the sub-objects includes transferring only those sub-sub-objects not stored in the memory.

In another preferred embodiment, the method includes storing in the memory a plurality of descriptions of sub-objects when installing a browser on the computer.

Preferably, accessing the at least one description includes looking up the location of the description in the memory in a look-up table. Alternatively or additionally, accessing the at least one description includes receiving an exact location of the description from an element manager and accessing the exact location.

In a preferred embodiment, the object includes a three dimensional graphical object.

Preferably, at least some of the sub-objects include graphical descriptions of parts of the object. Further preferably, at least some of the sub-objects include sound modules.

Preferably, accessing the program description of at least one of the sub-objects includes calling a routine which returns the address of the object in the local memory. Most preferably, transferring the program code includes transferring a script which references the sub-objects in

the local memory and if the object is not found in the local memory, references the sub-object in an input location.

There is also provided, in accordance with a preferred embodiment of the present invention, apparatus for rendering an image of an animation object under control of a remote
5 server, including:

a memory, which stores descriptions of sub-objects which are used to form the objects;
an animation generator, linked locally to the memory, which receives a skeleton which describes a structure of the object and retrieves the descriptions of at least some of those sub-objects from the memory; and

10 a display on which the image is displayed by rendering the sub-objects responsive to the descriptions thereof and responsive to the skeleton.

In a preferred embodiment, the object includes a three dimensional object.

Preferably, the memory includes a look up table which lists the sub-objects of which there are descriptions in the memory. Further preferably, the animation generator receives
15 from the server a list of the sub-objects in the object and responds by indicating to the server those sub-objects of which descriptions are not stored in the memory.

The present invention will be more fully understood from the following detailed description of the preferred embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic illustration of a system for rendering an animation sequence on a client computer under control of a server, in accordance with a preferred embodiment of the present invention;

5 Fig. 2 is a block diagram illustrating the program structure of an animation object, in accordance with a preferred embodiment of the present invention;

Fig. 3 is a flow chart illustrating a method of rendering an animation object, in accordance with a preferred embodiment of the present invention;

10 Fig. 4 is a schematic block diagram illustrating a data structure in a memory associated with the client computer, in accordance with a preferred embodiment of the present invention; and

Fig. 5 is a flow chart illustrating the actions of a computer in rendering an object under control of a remote server, in accordance with another preferred embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

15 Fig. 1 is a schematic illustration of a system 18 for rendering of an animation sequence, in accordance with a preferred embodiment of the present invention. The sequence is rendered on an animation generator, typically a client computer 26, under control of a server 24 connected to computer 26 by a network 28, preferably the Internet. Server 24 is preferably a Web server, as is known in the art.

20 The animation sequence is preferably described by a script generated in accordance with a general programming language, such as JavaScript, using computer programming methods known in the art. The animation sequence is transmitted from server 24 over a network 28 to a scene manager program, preferably associated with Internet browser software running on computer 26, as described further hereinbelow. The scene manager displays the animation on a display 30 associated with the computer. Preferably, computer 26 further comprises a memory 32 which is used to store building blocks of animation objects, referred to herein as elements, which are accessed in accordance with the script, as described hereinbelow.

30 Fig. 2 is a block diagram illustrating the program structure of a 3D animation object 70, in accordance with a preferred embodiment of the present invention. Object 70 comprises a plurality of elements 78, and a structure record 72 which describes relationships between the

plurality of elements 78. Elements 78 may comprise "organs" which are, in general, coherent groups of 3D shapes that form a part of the physical representation of object 70. Other elements 78 include "sound elements" 74, "behavior elements" 76, "surface detail elements," and other types of groups of data associated with object 70. Elements 78 preferably are made up of a plurality of sub-elements 90 and a sub-structure 88 which describes the relationships between sub-elements 90. Preferably, at least some of sub-elements 90 are made up, in a manner similar to elements 78, of a structure element and sub-sub-elements 92. The sub-sub-elements 92 comprise further sub-elements or basic building blocks 94, such as simple geometrical shapes, single sound notes, basic behavior features, etc. Preferably, object 70 comprises a "Smart Object", as described in PCT publication WO97/35280.

Fig. 3 is a flow chart illustrating a method of rendering animation object 70, in accordance with a preferred embodiment of the present invention. Server 24 transmits to client computer 26 a list of the elements of object 70. In response, computer 26 sends back a list of those elements which it does not have. Alternatively or additionally, computer 26 sends a list of the elements it does have. Meanwhile, server 24 sends structure record 72 to computer 26, since the structure record must be sent in any event. Preferably, if the returned list has not been received yet, server 24 continues sending elements which are most likely to be required, or elements which are chosen randomly. Once the returned list from computer 26 is received, server 24 continues sending only those elements which the computer does not have. Thus, it is possible to take advantage of substantially all the elements which are stored in computer 26, without server 24 knowing beforehand the contents of memory 32 of the computer.

In some preferred embodiments of the present invention, when an element 78 which must be sent to computer 26 includes large sub-elements 90, server 24 sends a list of the sub-elements which it intends to send, and computer 26 responds with a list of those sub-elements it does not have. This process may be continued recursively as illustrated by the pseudo code in Table 1, which is a procedure performed by server 24 when sending the elements of an object 70. The procedure is preferably invoked by the pseudo command: Handle_organ (highest organ in hierarchy).

Table 1

Procedure Handle_organ (*organ_to_be_handled*){

```

    If organ_to_be_handled exists on client
        Exit
    Else If organ_to_be_handled has sub-organs then {
        For each sub-organ of organ_to_be_handled {
5           Handle_organ (the sub-organ)
        }
    Else
        Send organ to client
    }

```

10

Alternatively or additionally, the original list of elements 78 of object 70 also lists all or some of the sub-elements 90 in the elements, and computer 26 indicates for those elements that it does not have in memory 32, which sub-elements it does have. It is noted that a similar procedure may be used for any depth of hierarchy of object 70.

15 Further alternatively or additionally, at the beginning of each session of rendering objects on computer 26 under control of server 24, the computer sends the server a list of all the elements and sub-elements it has already stored in memory 32. Thereafter, server 24 updates the list each time another object is rendered on computer 26, so that the list transmitted by the server to the computer is kept as short as possible and in some cases may
20 not be necessary.

Fig. 4 is a schematic illustration of memory 32, in accordance with a preferred embodiment of the present invention. Memory 32 includes an area 34 in which descriptions of objects 70, elements 78 and sub-elements 90 are stored. Preferably, whenever any object 70, element 78, or sub-element 90 is downloaded for rendering on the screen of computer 26, it is
25 in addition stored in memory 32 for further use. Alternatively or additionally, a group of commonly-used objects and/or elements are provided along with the browser. Further alternatively or additionally, a group of commonly used objects and/or elements are stored in memory 32 upon manufacture, and are provided to every purchaser of computer 26. Further alternatively or additionally, a user may download from an Internet site groups of objects
30 and/or elements.

Preferably, memory 32 comprises a look-up table 36 which points to the location in area 34 of each object, element and sub-element stored in memory 32. When computer 26

receives a list of objects from server 24, it compares the list to look-up table 36 and responds accordingly. Thereafter, when the browser displays object 70, it accesses look-up table 36 to determine the location of elements it needs for rendering.

Preferably, elements 78 are cataloged in look up table 36 both as elements and
5 according to their sub-elements, so that sub-elements included in more than one element need be sent for rendering only once.

Fig. 5 is a flow chart illustrating the actions of computer 26 during rendering of object 70, in accordance with another preferred embodiment of the present invention. Object 70 is sent in the form of a script to a Web browser on computer 26. The script includes structure
10 record 72 of object 70, in which the elements of the object are referenced along with the connections between them. The browser renders the elements on display 30, as is known in the art. For each element to be rendered, the browser calls an element manager which provides the address of the description of the element.

Preferably, the element manager is stored in memory 32 together with the descriptions
15 of the objects. Preferably, the element manager is an ActiveX element and is provided with the Web browser, with an operating system or with computer 26. Alternatively or additionally, the element manager may be downloaded from a Web site or is provided along with an animation stream.

The element manager preferably consults look-up table 36 to find the element. If the
20 element is in memory 32, the element manager returns a pointer to the exact area 34 where the description of the element resides. Providing the exact address of the element allows the script from server 24 to access the element directly without searching through memory 32, a search which generally causes the Web browser to display an alert message. However, if the element is not in memory 32 the element manager returns a pointer to an input directory in which
25 elements from server 24 are temporarily stored. In addition, the element manager preferably orders the element from server 24, by sending an appropriate message. Preferably, if the element is formed of relatively large sub-elements, and some of the sub-elements are stored in memory 32 (as listed in LUT 36), the element manager orders only those elements which are not in the memory.

30 After the element manager has ordered one of the elements from server 24, computer 26 waits for the server to return the ordered element. Preferably, while the computer is waiting, the element manager proceeds to process the rest of structure record 72.

Alternatively, the element manager waits for the ordered element to be returned before proceeding.

Table 2 is a subroutine which finds the objects referenced by the structure record, in accordance with a preferred embodiment of the present invention.

5

Table 2

```

//*****
// da - DirectAnimation object for creating parts
10 // number_of_parts - the number of parts in the character
// alternate path - the path where all parts will reside if
// not found in local users hard disk
// characterURLs - array with entry for each part.
//   If entry is "", no part is needed.
15 //   If entry is "?", part can be found only in
//       alternatePath, and not on user's local hard disk.
//   otherwise, points to relative path of part in Zapa's
//   folder on user's local hard disk.
//*****
20 function ZapaCreateCharacter (da, number_of_parts, alternatePath, characterURLs)
{
    // create ActiveX object that searches for local content
    // Zapa.LocalContent is the element manager var zapaContents = new
    ActiveXObject("Zapa.LocalContent");
25 // array that will receive DA parts
    var allParts = new Array(skeleton.length);

    var i;

30 // in Zapa's characters, base part is always a null part
    allParts[0] = da.EmptyGeometry;

```

```

for (i = 1; i <= number_of_parts; ++i)
{
    var localFileName = characterURLs[i-1];
    // alternate filename is as below.
5    var alternateFileName = alternatePath + i + ".x";

    if (localFileName == "")
        // if part i is an empty part
        allParts[i] = da.EmptyGeometry;
10    else if (zapaContents != null && localFileName != "?")
    {
        // if part may be in local content and activeX object
        // was created correctly, ask it for local path
        var path = zapaContents.PathOfFile(localFileName,alternateFileName);
15
        // ActiveX returns either local file name or alternate file name
        if (path != "")
            allParts[i] = da.ImportGeometryAsync(path, da.EmptyGeometry).Geometry;
        else
20        allParts[i] = da.EmptyGeometry;
    }
    else
        // if no local content, then just use alternateFileName
        allParts[i] =
25        da.ImportGeometryAsync(alternateFileName,da.EmptyGeometry).Geometry;

```

Thereafter, the elements received from server 24 are preferably passed to long-term storage in memory 32 in appropriate directories. In addition, the elements are preferably cataloged in list 36. The long term storage is preferably performed at a later time when computer 26 is idle. Alternatively or additionally, elements which appear more than once in an animation stream or are likely to appear in animation streams to be accessed in the future are

placed in the long-term storage while the rest of the elements received are left in the temporary directory and are eventually erased. Preferably, each element sent from server 24 is tagged with a "store" / "don't store" label, which instructs computer 26 whether to pass the element to long term storage.

5 Alternatively or additionally, the elements are tagged with a priority value which states the expected frequency of use of the element. Computer 26 determines, based on the priority value and the available storage room in memory 32, whether or not to pass the element to the long-term storage. Preferably, other variables may also be used to determine whether to pass the element to long-term storage, and a user of computer 26 may change the settings of the
10 variables.

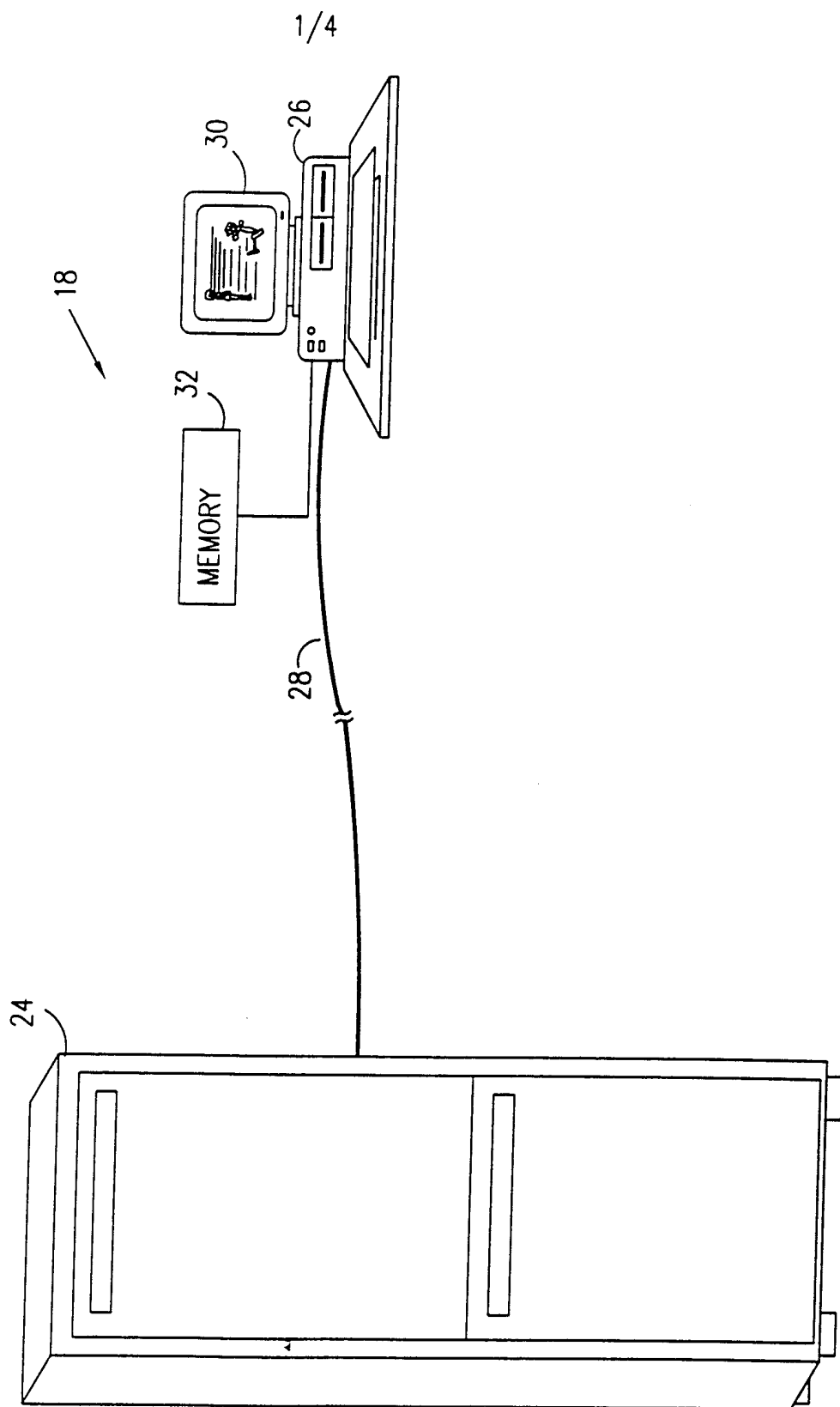
It will be appreciated that the preferred embodiments described above are cited by way of example, and the full scope of the invention is limited only by the claims.

CLAIMS

1. A method for rendering an animation object, including a plurality of sub-objects, on a display of a client computer under control of a remote server, comprising:
 - transferring from the server to the client, program code which describes a structure of
 - 5 the object;
 - accessing within a local memory associated with the client a program description of at least one of the sub-objects of the object; and
 - displaying the object such that the sub-object is coupled to the structure in accordance with the program code.
- 10 2. A method according to claim 1, and comprising reporting to the server the identity of at least some of the sub-objects stored in the memory.
3. A method according to claim 2, wherein reporting the identity of the sub-objects comprises receiving a list of sub-objects from the server and reporting on a correspondence between the list and the sub-objects that are stored in the memory.
- 15 4. A method according to claim 1, and comprising transferring from the server to the client descriptions of sub-objects not already stored in the memory and storing the transferred descriptions in the memory.
5. A method according to claim 4, wherein at least some of the sub-objects comprise sub-sub-objects, and transferring descriptions of the sub-objects comprises transferring only those
- 20 sub-sub-objects not stored in the memory.
6. A method according to claim 1, and comprising storing in the memory a plurality of descriptions of sub-objects when installing a browser on the computer.
7. A method according to claim 1, wherein accessing the at least one description comprises looking up the location of the description in the memory in a look-up table.
- 25 8. A method according to claim 1, wherein accessing the at least one description comprises receiving an exact location of the description from an element manager and accessing the exact location.
9. A method according to claim 1, wherein the object comprises a three dimensional graphical object.

10. A method according to claim 1, wherein at least some of the sub-objects comprise graphical descriptions of parts of the object.
11. A method according to claim 1, wherein at least some of the sub-objects comprise sound modules.
- 5 12. A method according to claim 1, wherein accessing the program description of at least one of the sub-objects comprises calling a routine which returns the address of the object in the local memory.
13. A method according to claim 1, wherein transferring the program code comprises transferring a script which references the sub-objects in the local memory and if the object is
10 not found in the local memory, references the sub-object in an input location.
14. Apparatus for rendering an image of an animation object under control of a remote server, comprising:
a memory, which stores descriptions of sub-objects which are used to form the objects;
an animation generator, linked locally to the memory, which receives a skeleton which
15 describes a structure of the object and retrieves the descriptions of at least some of those sub-objects from the memory; and
a display on which the image is displayed by rendering the sub-objects responsive to the descriptions thereof and responsive to the skeleton.
15. Apparatus according to claim 14, wherein the object comprises a three dimensional
20 object.
16. Apparatus according to claim 14, wherein the memory comprises a look up table which lists the sub-objects of which there are descriptions in the memory.
17. Apparatus according to claim 14, wherein the animation generator receives from the server a list of the sub-objects in the object and responds by indicating to the server those sub-
25 objects of which descriptions are not stored in the memory.

FIG. 1



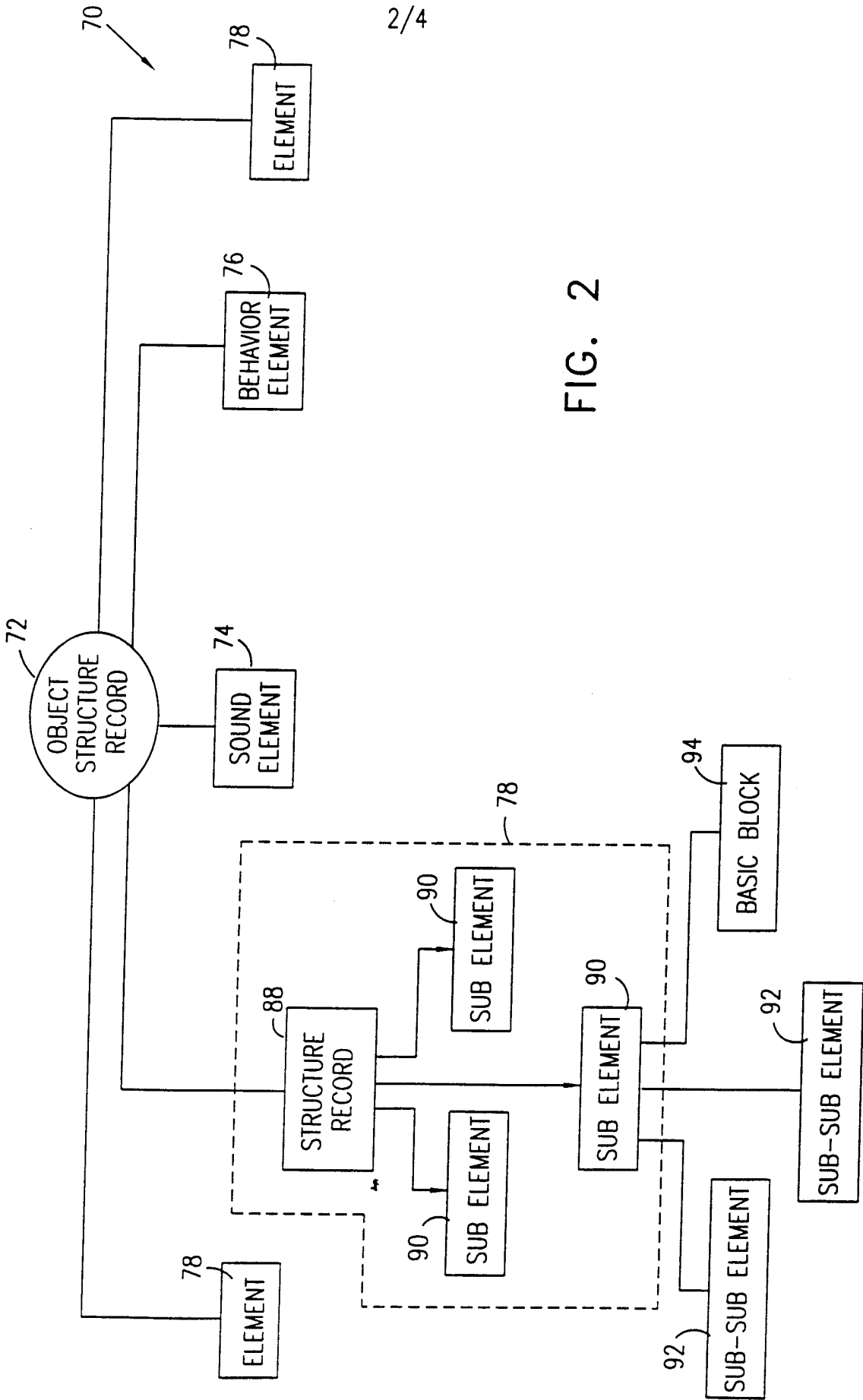


FIG. 2

3/4

FIG. 3

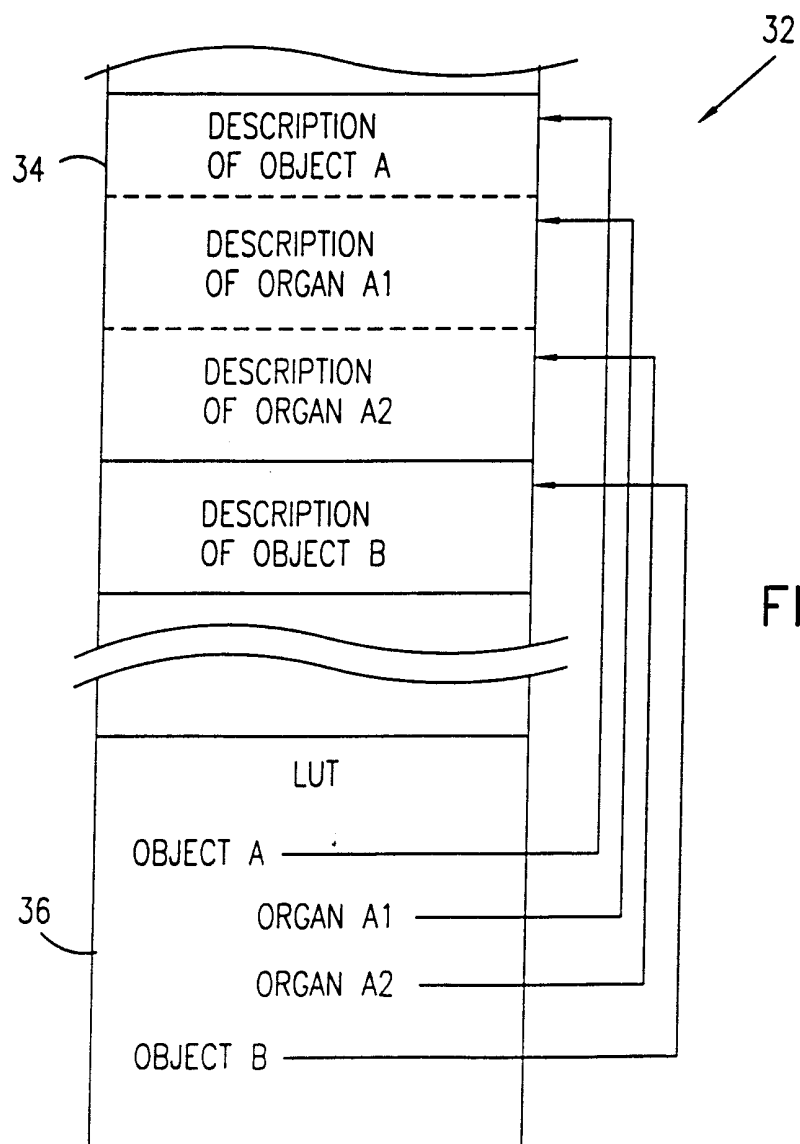
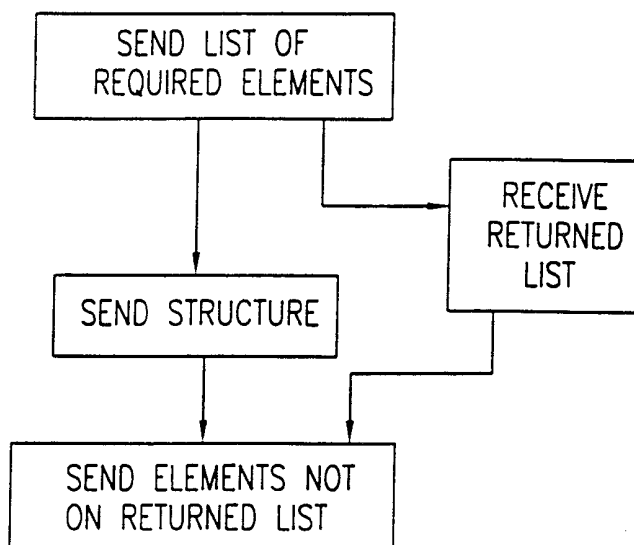
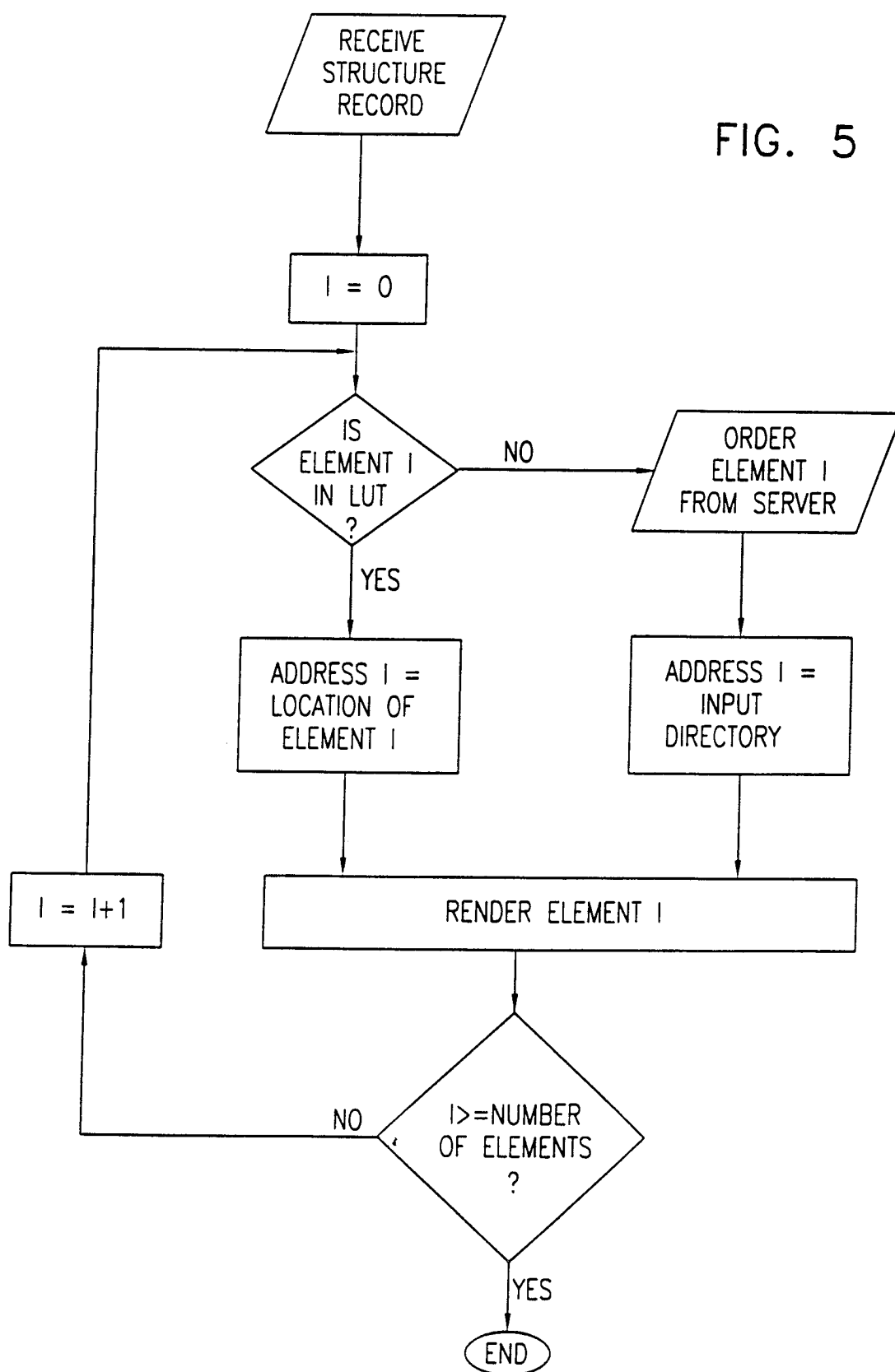


FIG. 4

4/4

FIG. 5



INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL99/00544

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06T 13/00, 15/70; G06F 15/173, 15/177, 17/30

US CL : 345/419, 473, 474; 209/201, 203, 217, 218, 220, 225, 231; 713/1

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 345/419, 473, 474; 209/201, 203, 217, 218, 220, 225, 231; 713/1

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
DR DOBB'S ELECTRONIC BOOKS ON GRAPHICS

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	EP 804032 A2 (NAKA et al) 29 October 1996, col. 2, lines 35-40; col. 3, lines 43-46; col. 4, lines 30-40.	1-4, 7-16 ----- 5-6, 17
X -- Y	BRP Publications. Fixing Interactive Entertainment Multimedia Daily. January 1997. Vol. 4. No. 1. 763 words, especially lines 50-56.	1-4, 7-16 ----- 5-6, 17
X - Y	US 5,818,463 A (TAO et al), 06 October 1998, col. 1, lines 12, 20-25, 60-65; col. 2, lines 55-60, 62-67; and col. 3, lines 1-14.	1-4, 7-16 ----- 5-6, 17
X - Y	US 5,649,086 (BELFER et al) 15 July 1997, col. 11, lines 4-11.	1-4, 7-16 ----- 5-6, 17

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents	*T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance, the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y"	document of particular relevance, the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"I" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"C" document referring to an oral disclosure, use, exhibition or other means		
"P" document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

10 APRIL 2000

Date of mailing of the international search report

26 APR 2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

MARK R. POWELL

James R. Matthews

Telephone No. (703) 308-0000

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL99/00544

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X - Y	US 5,754,190 (DUTTON et al) 19 May 1998, abstract; Figure 4, col. 1, lines 52-53, 57-60 ; col. 2, 11-45.	1-5, 7-17 ----- 6
X - Y	WO 98/46005 (ELEFThERIADIS et al) 15 October 1998, page 4, lines 15-23.	1-4, 7-16 ----- 5-6, 17
X	STEMPSON, G. Database Management: New Generation Systems Speed Graphics' Role. Computer Graphics World. April 1986.	1-5, 7-13
Y	EP 0841615 A2 (ROWLEY, David J.) 13 May 1998, abstract col. 1, lines 25-55.	5, 17
Y	EP 0753834 A2 (HONDA, Yasuaki) 15 January 1997, page 24, lines 5-11 and 15-49.	6
X,P	US 5,931,906 A (FIDELIBUS, JR et al) 03 August 1999, abstract, col. 3, lines 10-67.	1-17
X,P	US 5,999,944 A (LIPKIN, DANIEL) 07 December 1999, abstract, col. 4, line 35 to col. 6, line 9.	1-17
X - Y	US 5,347,306 A (NITTA, TOHEI) 13 September 1994, abstract, col. 3, lines 10-17; col. 3, lines 42-51; col. 5, lines 4-5 and 13-21.	1-4, 6-16 ----- 5, 17
X,T	US 5,991,798 (OZAKI et al) 23 November 1999, abstract, col. 5, line 5 to col. 7, line 36.	1-13
A	US 4,600,919 A (STERN) 15 July 1986.	1-17
A	US 5,583,994 (RANGAN) 10 December 1996.	1-17
A	US 5,592,626 A (PAPADIMITRIOU et al) 07 January 1997	1-17
A	US 5,659,692 A (POGGIO et al) 19 August 1997 abstract	1-17
A	US 6,038,258 A (CHUN et al) 14 March 2000.	1-17
A	US 5,655,067 A (TAKAHASHI et al) 05 August 1997.	1-17
A	US 5,848,291 A (MILNE et al) 08 December 1998.	1-17
A,P	US 5,877,778 A (DOW et al) 2 March 1999.	1-17

INTERNATIONAL SEARCH REPORTInternational application No.
PCT/IL99/00544**C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0774722 A2 (MICROSOFT) 21 May 1997.	1-17
A	WO 97/35280 A3 (GEVER et al) 25 September 1997.	1-17
A	WO 98/46024 A1 (ELEFThERIADIS et al) 07 April 1997.	1-17
A	WO 98/53406 (GRIFFITHS et al) 26 November 1998.	1-17
A	SINGH, K. et al. Human Figure Synthesis and Animation for Virtual Space Teleconferencing. Proc. Virtual Reality Annual International Symposium 1995, IEEE, pages 118-126.	1-17

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL99/00544

B. FIELDS SEARCHED

Electronic data bases consulted (Name of data base and where practicable terms used):

DIALOG EECOMP COMPSCI, INSPEC, COMPENDEX, USPATFUL, EUROPEAN PATENTS FULL, WORLD
PATENT INDEX, IMPADOC, JAPIO

search terms: animation, bandwidth, skeleton, object, browser, multipath movie, install, transmission, wireframe