

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
19 October 2006 (19.10.2006)

PCT

(10) International Publication Number
WO 2006/109978 A1

(51) International Patent Classification:
H03M 7/30 (2006.01)

(21) International Application Number:

PCT/KR2006/001323

(22) International Filing Date: 11 April 2006 (11.04.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/669,927	11 April 2005 (11.04.2005)	US
60/676,972	3 May 2005 (03.05.2005)	US
60/700,320	19 July 2005 (19.07.2005)	US

(71) Applicant: SAMSUNG ELECTRONICS CO., LTD.
[KR/KR]; 416, Maetan-dong, Yeongtong-gu, Suwon-si,
Gyeonggi-do 442-742 (KR).

(72) Inventors: AHN, Jeong-Hwan; 103-1007 Dongsuwon
LG Village 1-cha Apt., Mangpo-dong, Yeongtong-gu,
Suwon-si, Gyeonggi-do 443-706 (KR). HAN, Mahn-Jin;
3 Fine Hill, 180-4, Jeil-ri, Yangji-myeon, Yongin-si,
Gyeonggi-do 449-825 (KR). KIM, Do-Kyoon; 3rd
Floor, 131-3, Bundang-dong, Bundang-gu, Seongnam-si,
Gyeonggi-do 463-832 (KR).

(74) Agent: Y.PLEE, MOCK & PARTNERS; The Ko-
ryo Building, 1575-1, Seocho-dong, Seocho-gu, Seoul
137-875 (KR).

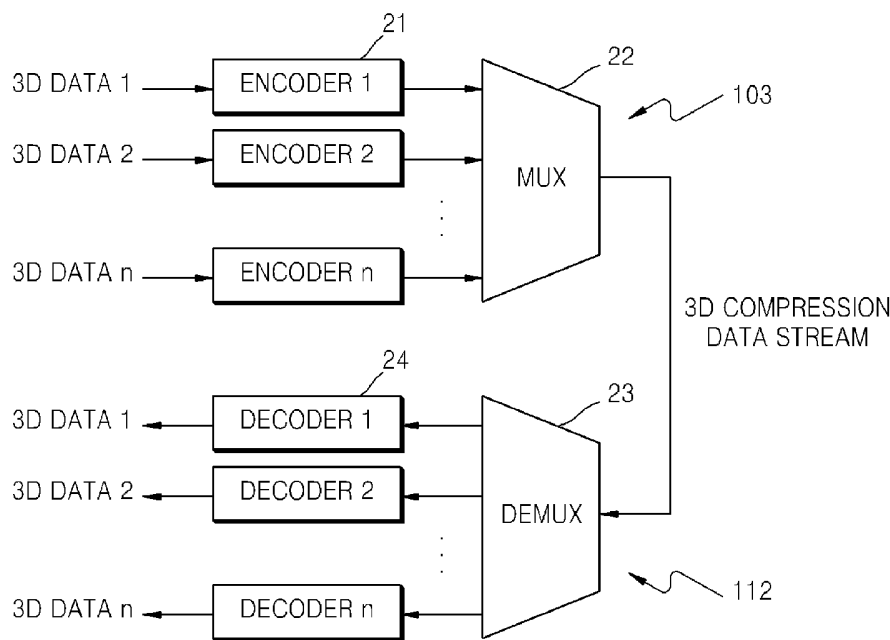
(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN,
CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI,
GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE,
KG, KM, KN, KP, KZ, LC, LK, LR, LS, LT, LU, LV, LY,
MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO,
NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK,
SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC,
VN, YU, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,
ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,
RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA,
GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— with international search report

[Continued on next page]

(54) Title: METHODS AND APPARATUSES FOR GENERATING AND RECOVERING 3D COMPRESSION DATA



(57) Abstract: A method of generating three-dimensional (3D) compression data, and an apparatus to perform the method, the method including generating configuration information of the 3D compression data; generating a plurality of pieces of individual 3D compression data to be included in the 3D compression data; and generating the 3D compression data by multiplexing the pieces of individual 3D compression data, combining the 3D compression data with the configuration information, and outputting the result of the combination.

WO 2006/109978 A1



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Description

METHODS AND APPARATUSES FOR GENERATING AND RECOVERING 3D COMPRESSION DATA

- [1] This application claims the benefit of US Provisional Application No. 60/669,927, filed on April 11, 2005, US Provisional Application No. 60/676,972, filed on May 3, 2005, and US Provisional Application No. 60/700,320, filed on July 19, 2005, in the United States Patent & Trademark Office, the disclosures of which are incorporated herein by reference.

Technical Field

- [2] The present invention relates to a method of generating three-dimensional (3D) graphic compression data, and an apparatus to perform the method, and a method of recovering 3D graphic compression data, and an apparatus to perform the method, and, more particularly, to a method of generating 3D graphic compression data, and an apparatus to perform the method, and a method of recovering 3D graphic compression data, and an apparatus to perform the method, in a mobile environment.

Background Art

- [3] The current state of the evolution of mobile devices allows some services, such as 3D games, 3D avatars, or 3D shopping, which were previously considered to be impossible or impractical in a mobile environment, to be provided. As a result, such services have been recognized as new business models to be used in mobile communications or content providers.
- [4] However, the mobile devices are limited in terms of computational complexity and power consumption, and a bandwidth of a wireless network is also limited. Thus, an efficient method of representing and compressing 3D graphic data is needed to overcome these limitations in a mobile environment.

Disclosure of Invention

Technical Problem

- [5] The present invention provides a method of generating three-dimensional (3D) graphic compression data, and an apparatus to perform the method, and a method of recovering 3D graphic compression data, and an apparatus to perform the method, in order to compactly represent the 3D graphic data and effectively transmit the data via a wireless network in a mobile environment that is restricted in power consumption and computational complexity.

Technical Solution

- [6] Additional aspects and/or advantages of the invention will be set forth in part in the description which follows and, in part, will be apparent from the description, or may be

learned by practice of the invention.

- [7] According to an aspect of the present invention, there is provided a method of generating three-dimensional (3D) compression data, the method comprising generating configuration information of the 3D compression data; generating a plurality of pieces of individual 3D compression data to be included in the 3D compression data; and generating the 3D compression data by multiplexing the pieces of individual 3D compression data, combining the 3D compression data with the configuration information, and outputting the result of the combination.
- [8] According to another aspect of the present invention, there is provided a method of recovering three-dimensional (3D) compression data, the method including separating the 3D compression data and configuration information of the 3D compression data from a received signal; separating a plurality of pieces of individual 3D compression data and individual configuration information of the pieces of individual 3D compression data from the 3D compression data; and decoding the pieces of individual 3D compression data by using data type information of the 3D compression data, which is included in the individual configuration information.
- [9] According to another aspect of the present invention, there is provided an apparatus to generate three-dimensional (3D) compression data, the apparatus comprising a plurality of encoders which encode received 3D data according to different methods and generate different types of pieces of individual 3D compression data; and a multiplexer which generates configuration information of the 3D compression data and individual configuration information of the pieces of individual 3D compression data, receives the pieces of individual 3D compression data from the encoders and respectively combines the pieces of individual 3D compression data with the corresponding individual configuration information, generates the 3D compression data by multiplexing the combined data, and combines the 3D compression data with the configuration information and outputs the result of the combination.
- [10] According to another aspect of the present invention, there is provided an apparatus to recover three-dimensional (3D) compression data, the apparatus comprising a demultiplexer which separates the 3D compression data and configuration information of the 3D compression data from a received signal, separates a plurality of pieces of individual 3D compression data and individual configuration information of the pieces of individual 3D compression data from the 3D compression data, and analyzes a data type code comprising data type information of the individual 3D compression data, the data type code being included in the individual configuration information; and a plurality of decoders which recover 3D graphic data by decoding the pieces of individual 3D compression data according to the corresponding data type code.
- [11] According to another aspect of the present invention, there is provided a method of

generating three-dimensional (3D) compression data, the method comprising generating configuration information of a 3DObjectSequence, which is the 3D compression data; generating 3DObjects, which are a plurality of pieces of individual 3D compression data that are included in the 3D compression data; and generating the 3D compression data by multiplexing the pieces of individual 3D compression data, combining the 3D compression data with the configuration information, and outputting the result of the combination. The generation of configuration information of the 3D compression data, the generation of the pieces of individual 3D compression data, and the generation of the 3D compression data are realized by:

```
[12]   3DObjectSequence() {
[13]   bit(32) 3dc_object_sequence_start_code ;
[14]   bit(8) p rofile_and_level_indication ;
[15]   do {
[16]   3DObject() ;
[17]   bit(32) *next;
[18]   } while ( next != 3dc_object_sequence_end_code) ;
[19]   bit(32) 3dc_object_sequence_end_code ;
[20]   },
```

[21] wherein 3dc_object_sequence_start_code denotes code representing a start of the 3D compression data, p rofile_and_level_indication denotes profile and level identification information of the 3D compression data, and 3dc_object_sequence_end_code denotes code representing an end of the 3D compression data.

[22] According to another aspect of the present invention, there is provided a method of generating three-dimensional (3D) compression data, the method comprising generating configuration information of the 3D compression data; generating a combination of individual configuration information of an individual 3D compression object included in the 3D compression data with the individual 3D compression object corresponding to the individual configuration information, and repeatedly generating the combination; and multiplexing the combinations to obtain a piece of data, and including the configuration information in the obtained data.

[23] According to another aspect of the present invention, there is provided a method of recovering three-dimensional (3D) compression data, the method comprising separating the 3D compression data and configuration information of the 3D compression data from a received signal; separating combinations of individual configuration information and individual 3D compression objects from the 3D compression data, and separating the individual 3D compression objects and the individual configuration information from the separated combinations; and recovering 3D data by decoding the individual 3D compression objects using data type in-

formation of the individual 3D compression objects, the data type information being included in the individual configuration information .

Advantageous Effects

- [24] According to the present invention, in a mobile environment being limited in power consumption and computational complexity, it is possible to effectively compress 3D graphic data and transmit the compressed 3D graphic data via a wireless network without an MPEG-4 system.

Description of Drawings

- [25] These and/or other aspects and advantages of the invention will become apparent and more readily appreciated from the following description of the embodiments, taken in conjunction with the accompanying drawings of which:

[26] FIG. 1 is a schematic block diagram illustrating a three-dimensional (3D) graphic system according to an embodiment of the present invention;

[27] FIG. 2 is a detailed block diagram illustrating an AFX encoder and an AFX decoder according to an embodiment of the present invention;

[28] FIG. 3 is a diagram illustrating a structure of compression data according to an MPEG-4 profile based on an MPEG-4 system;

[29] FIG. 4 is a diagram illustrating a structure of a 3D compression data stream according to an embodiment of the present invention;

[30] FIG. 5 is a diagram illustrating the structure of a bitstream multiplexed by a multiplexer of FIG. 2;

[31] FIG. 6A is a diagram illustrating an example of a scene; and

[32] FIG. 6B is a graph illustrating the structure of objects included in the scene illustrated in FIG. 6A.

Mode for Invention

- [33] Reference will now be made in detail to the embodiments of the present invention, examples of which are illustrated in the accompanying drawings, wherein like reference numerals refer to the like elements throughout. The embodiments are described below to explain the present invention by referring to the figures.

[34] FIG. 1 is a schematic block diagram illustrating a three-dimensional (3D) graphic system according to an embodiment of the present invention. 3D graphics may include mobile 3D applications, such as, for example, a 3D game, a 3D graphic user interface (GUI), or a 3D avatar. Referring to FIG. 1, the system includes a 3D graphic authoring device 10 and a 3D graphic operation device 11.

[35] The 3D graphic authoring device 10 includes an authoring tool 101, an application implementation unit 102, and an AFX (Animation Framework eXtension) encoder 103.

[36] The authoring tool 101 creates 3D resources. The 3D resources may include, for

example, a 3D mesh, a key-frame sequence, or bone information.

[37] The application implementation unit 102 outputs scene graph information used to display content stored in a data structure in a screen according to a user's input, an interaction with a user, scene management, Open GL, which is a graphic library, and GFX or DirectX, which are application program interfaces.

[38] The AFX encoder 103 compresses some of the 3D resources from the authoring tool 101, or the information from the application implementation unit 102, and transmits the compression result.

[39] The 3D graphic operation device 11 includes an application execution unit 111 and an AFX decoder 112.

[40] The application execution unit 111 receives and executes the information from the application implementation unit 102, or executes the 3D resources recovered by the AFX decoder 112.

[41] FIG. 2 is a detailed block diagram illustrating the AFX encoder 103 and the AFX decoder 103 illustrated in FIG. 1, according to an embodiment of the present invention. The AFX encoder 103 includes a plurality of encoders 21 and a multiplexer (MUX) 22. The AFX decoder 112 includes a demultiplexer (DEMUX) 23 and a plurality of decoders 24.

[42] Each of the encoders 21 compresses received 3D data into a compressed bitstream, and the multiplexer 22 multiplexes the compressed bitstreams to a compressed data stream. Each encoder 21 encodes the received 3D data according to the type of an object, e.g., through, for example, 3D mesh coding (3DMC), interpolator coding (IC), wavelet subdivision surface (WSS), or bone-based animation (BBA).

[43] The demultiplexer 23 demultiplexes the compressed data stream into the compressed bitstream, and each of the decoders 24 recovers the original data from the compressed data.

[44] FIG. 3 is a diagram illustrating a structure of compression data according to an MPEG-4 profile based on an MPEG-4 system. Referring to FIG. 3, according to the specification of a system part of the ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) 14496-1, an MPEG-4 system (not shown) inserts an object descriptor 32, which describes the characteristics of a media object (audio or video) and the characteristics of an elementary stream, into an elementary stream 31 comprised of access units (AUs). Configuration information 33 is inserted into the object descriptor 32.

[45] The specification of the system part of the ISO/IEC 14496-1 specifies separate encoding of video data and audio data, and multiplexing and synchronizing of the encoded data in order to store multimedia data in a file or a recording medium or transmit it via a network. It is an MPEG-4 system that defines the data multiplexing.

Video or audio binary data that has yet to be multiplexed by the system is referred to as the elementary stream.

[46] In the MPEG-4 standard, object encoding is used to multiplex audio and video data. In this case, both audio data and video data are processed as a single object, and the system multiplexes and synchronizes the objects.

[47] The MPEG-4 standard defines BIFS (Binary Format for Scenes) as a feature used in a scene description by combining and processing a plurality of objects. For example, moving picture data and audio data of a person or a background are encoded as separate objects. Thus, an application can be made, so that the encoded objects are displayed to overlap each other, or a user can move any object.

[48] However, when an application is downloaded and stored by using most basic tools in a mobile environment, the profile illustrated in FIG. 3 is tightly coupled with an MPEG-4 system having a BIFS structure. That is, the profile is inefficient since the MPEG-4 system must be prepared to decode a bitstream encoded by an MPEG-4 AFX tool, thereby increasing the load on the hardware.

[49] Therefore, in an embodiment of the present invention, a 3D compression profile that does not require an MPEG-4 system is adopted. In 3D compression data according to an embodiment of the present invention, configuration information, i.e., header information of an object, is inserted into a start of a bitstream, and, optionally, repeatedly inserted into the bitstream after the start of the bitstream.

[50] FIG. 4 is a diagram illustrating a structure of a 3D compression data stream according to an embodiment of the present invention. Referring to FIG. 4, an elementary system includes objects, each having a plurality of access units, and configuration information Cfg of each object.

[51] The data structure illustrated in FIG. 4 is obtained by making an object by encoding 3D data by each of the encoders 21 of FIG. 2 according to the type of the object, and inserting configuration information into each of the objects and multiplexing the objects by the multiplexer 22. Here, the types of the objects may include, for example, 3D mesh coding (3DMC), a coordinate interpolator (CI), an orientation interpolator (OC), a wavelet subdivision surface (WSS), a position interpolator (PI), and bone-based animation (BBA), which are suggested in the MPEG-4 standard.

[52] FIG. 5 is a diagram illustrating a data structure of a bitstream multiplexed by the multiplexer 22 of FIG. 2. The illustrated bitstream is a 3DCObjectSequence that includes a 3DCObjectSequence header 51 and a 3DCObject 52. The 3DCObjectSequence header 51, which is generated by the multiplexer 22, includes profile and level identification of the bitstream. The 3DCObjectSequence header 51 may further include user data defined for a specific application by a user, e.g., scene graph information for the bitstream, which will be described in detail with reference to

FIGS. 6A and 6B.

- [53] FIG. 6A is a diagram illustrating an example of a scene. FIG. 6B is a graph illustrating the structure of objects included in the scene illustrated in FIG. 6A. Referring to FIG. 6A, the scene includes objects 1 through 3 defined in world coordinates 61, and the object 3 includes detailed objects 3-1, 3-2, and 3-3. For example, the objects 1 through 3 may be a building, a tree, and a chair, respectively, and the detailed objects 3-1 through 3-3 of the object 3 may respectively be a ball on the chair and the legs of the chair.
- [54] The scene illustrated in FIG. 6A may be represented as graph information as illustrated in FIG. 6B. Referring to FIG. 6B, the world coordinates 61 are linked to objects 1 through 3, and the object 3 is linked to detailed objects 3-1, 3-2, and 3-3.
- [55] The scene graph information illustrated in FIGS. 6A and 6B may be included as user data in the 3DObjectSequence header 51 of FIG. 5. For example, the objects 1 through 3, linked to the world coordinates 61, may be included in the user data as the objects for configuring a scene, e.g., a building, a tree, and a chair, and the detailed objects 3-1 through 3-3, which comprise the chair 3, may also be included in the user data as a ball on the chair and the legs of the chair, respectively.
- [56] The 3D compression object 3DObject 52 includes a plurality of 3DObject headers 521, and a plurality of individual 3DObjects 522. Here, each of the 3DObjects 522 is a 3D compression bitstream, such as 3DMC Simple_3DMC, IC (not shown), BBA Simple_BBA, WSS (not shown), and PI Simple_PI, which is encoded by the encoders 21.
- [57] The 3DObject header 521, which is generated by the multiplexer 22, includes a '3dc_object_verid' field describing the version number of a tool list of a 3D compression object type, and a '3dc_object_type_start_code' field that is used in resynchronization and describes the object type of a 3D compression stream to be transmitted and the type of a decoder matching the 3D compression stream. For example, if the '3dc_object_type_start_code' field is a 'Simple_3DMC' field, the demultiplexer 23 analyzes the '3dc_object_type_start_code' field and outputs the 3D compression stream to a 3DMC decoder, which is one of the decoders 24, and the 3DMC decoder decodes the received 3D compression stream. New object types and a corresponding '3dc_object_type_start_code' field may be defined and extended to define more object types and corresponding decoders.
- [58] The 3DObject header 521 may further include user data that a user defines for a specific application for the 3D compression object. For example, in FIGS. 6A and 6B, a building, a tree, and a chair may be included in a 3DObject header 1, a 3DObject header 2, and a 3DObject header 3, respectively.
- [59] A process of generating a 3D compression object sequence by using the

multiplexer 22 is expressed with the following code:

```
[60]     3DObjectSequence() {
[61]     bit(32) 3dc_object_sequence_start_code;
[62]     bit(8) profile_and_level_indication;
[63]     bit(32) *next;
[64]     while (next== user_start_code){
[65]     user_data();
[66]     }
[67]     do {
[68]     3DObject();
[69]     bit(32) *next;
[70]     } while ( next != 3dc_object_sequence_end_code);
[71]     bit(32) 3dc_object_sequence_end_code;
[72]     }
```

[73] In the above code, 3dc_object_sequence_start_code is a hexadecimal number system of a bit string '000001A0', and starts a 3D compression session. In this case, the Mnemonic is bslbf. The Mnemonic describes the type of data available for a coded bitstream, and bslbf stands for a bit string, left bit first.

[74] p rofile_and_level_indication is an 8-bit integer used to signal the profile and level identification of the bitstream, and the meaning of each bit value is indicated in the following table:

[75]

[Table 1]

Profile/Level	Code
Reserved	00000000
core profile/level 1	00000001
core profile/level 2	00000010
Reserved	00000011
:	:
Reserved	11111111

[76]

[77] In this case, the mnemonic is uimbsf, which stands for unsigned integer, most significant bit first.

[78] user_start_code is a hexadecimal number system of a bit string '000001A4', and indicates the start of the user data. The user data is continuously generated until another start code is received. In this case, mnemonic is bslbf.

[79] 3dc_object_sequence_end_code is a hexadecimal number system of a bit string '000001A1', and ends the 3D compression session. In this case, the mnemonic is bslbf.

[80] To generate the 3D compression object sequence, when the 3D compression

session starts, user data is generated according to a given profile and level during a period of time in which a next bit is user_start_code, and 3DCObject is generated during a period of time in which a next bit is not 3dc_object_sequence_end_code.

[81] A process of generating the 3D compression object is expressed with the following code:

```
[82]     3DCObject() {
[83]     bit(32) 3dc_object_start_code;
[84]     bit(1) is_3dc_object_identifier;
[85]     bit(3) 3dc_object_verid;
[86]     bit(4) 3dc_object_priority;
[87]     bit(32) *next;
[88]     while ( next== user_start_code){
[89]     user_data();
[90]     }
[91]     bit(32) *next;
[92]     if (next == 'Simple_3DMC') {
[93]     bit(32) 3dc_object_type_start_code;
[94]     3D_Mesh_Object();
[95]     }
[96]     else if (next == 'Simple_WSS') {
[97]     bit(32) 3dc_object_type_start_code;
[98]     Wavelet_Mesh_Object()
[99]     }
[100]    else if (next == 'Simple_CI') {
[101]    bit(32) 3dc_object_type_start_code;
[102]    CompressedCoordinateInterpolator()
[103]    }
[104]    else if (next == 'Simple_OI') {
[105]    bit(32) 3dc_object_type_start_code;
[106]    CompressedOrientationInterpolatort()
[107]    }
[108]    else if (next == 'Simple_PI') {
[109]    bit(32) 3dc_object_type_start_code;
[110]    CompressedPositionInterpolator()
[111]    }
[112]    else if (next == 'Simple_BBA'){
[113]    bit(32) 3dc_object_type_start_code;
[114]    bba_object()
```

[115] }

[116] }

[117] In the above code, 3dc_object_start_code is a hexadecimal number system of a bit string '000001A2', and starts a 3D compression object. In this case, the mnemonic is bslbf.

[118] is_3dc_object_identifier is a 1-bit code, and set to '1' when the version identification and priority of a 3D compression object are defined. In this case, the mnemonic is uimsbf.

[119] 3dc_object_verid is a 4-bit code that represents the version number of the 3D compression object. The meanings of the bit values are shown in the following table. If 3dc_object_verid does not exist, the value of 3dc_object_verid is '0001'. In this case, the mnemonic is uimsbf.

[120]

[Table 2]

3dc_object_verid	Meaning
0000	reserved
0001	object type listed in Table 3
0010-1111	reserved

[121] Table 3 shows a tool list included in each object type. Current object types may be extended until new tools or functionalities are introduced.

[122]

[Table 3]

AFX Tool	3D Compression Object Type						
	Simple 3DMC	Simple CI	Simple PI	Simple OI	Simple WSS	Simple BBA	reserved
3DMC	X						
CI		X					
PI			X				
OI				X			
WSS					X		
BBA						X	
reserved							

[123]

[124] 3dc_object_priority is a 3-bit code representing the priority of the 3D compression object. 3dc_object_priority has a value between 1 and 7. Here, '1' represents the highest priority, and '7' represents the lowest priority. '0' is a reserved value. In this case, the mnemonic is uimsbf.

[125] 3dc_object_type_start_code is a 32-bit bit string. The first 24 bits of 3dc_object_type_start_code are '0000 0000 0000 0000 0000 0001', which is used in

resynchronization. The last 8 bits include values of 'A6' or 'AB', and represent the object types listed in Table 4. In this case, the mnemonic is bslbf.

[126] A related decoder is called and the compressed stream is decoded according to the last 8 bits of 3dc_object_type_start_code. If more object types are further defined in Table 3, they are reflected in Table 4.

[127]

[Table 4]

3dc_object_type_start_code	Code (hexadecimal)
Reserved	A5
Simple 3DMC	A6
Simple CI	A7
Simple PI	A8
Simple OI	A9
Simple WSS	AA
Simple BBA	AB
Reserved	AC through AF

[128]

[129] To generate the 3D object, when a session starts, user data is generated according to the version and priority of the 3D compression object during a period of time in which a next bit is user_start_code. Next, an object is generated according to the type of the corresponding 3D object.

[130] A process of generating the user data is expressed with the following code:

```
[131] user_data() {
[132]   bit(23) *next;
[133]   while(next !=0){
[134]     bit(8) user_data_bits;
[135]     bit(23) *next;
[136]   }
[137] }
```

[138] user_data_bits is an 8-bit integer that is defined for a specific application by a user. If there are a series of user_data_bits, a bit string of 23 or more consecutive '0's is not allowed.

[139] In addition to the above-described embodiments, the method of the present invention can also be implemented by executing computer readable code/instructions in/on a medium, e.g., a computer readable medium. The medium can correspond to any medium/media permitting the storing and/or transmission of the computer readable code. The code/instructions may form a computer program.

[140] The computer readable code/instructions can be recorded/transferred on a medium in a variety of ways, with examples of the medium including magnetic storage media

(e.g., ROM, floppy disks, hard disks, etc.), optical recording media (e.g., CD-ROMs, or DVDs), and storage/transmission media such as carrier waves, as well as through the Internet, for example. The medium may also be a distributed network, so that the computer readable code/instructions is stored/transferred and executed in a distributed fashion. The computer readable code/instructions may be executed by one or more processors.

- [141] As described above, according to the present invention, in a mobile environment being limited in power consumption and computational complexity, it is possible to effectively compress 3D graphic data and transmit the compressed 3D graphic data via a wireless network without an MPEG-4 system.
- [142] Although a few embodiments of the present invention have been shown and described, it would be appreciated by those skilled in the art that changes may be made in these embodiments without departing from the principles and spirit of the invention, the scope of which is defined in the claims and their equivalents.

Claims

- [1] A method of generating three-dimensional (3D) compression data, the method comprising:
generating configuration information of the 3D compression data;
generating a plurality of pieces of individual 3D compression data to be included in the 3D compression data; and
generating the 3D compression data by multiplexing the pieces of individual 3D compression data, combining the 3D compression data with the configuration information, and outputting the result of the combination.
- [2] The method of claim 1, wherein the configuration information comprises profile and level identification information of the 3D compression data.
- [3] The method of claim 2, wherein the configuration information further comprises user data defining applications of the 3D compression data.
- [4] The method of claim 3, wherein the user data comprises scene graph information representing objects included in a scene by using a graph in response to the 3D compression data being graphic data.
- [5] The method of claim 1, wherein the generation of the pieces of individual 3D compression data comprises:
generating individual configuration information of the pieces of individual 3D compression data;
generating the pieces of individual 3D compression data according to a data type; and
combining the individual configuration information with the pieces of individual 3D compression data.
- [6] The method of claim 5, wherein the individual configuration information comprises:
a version number of a tool list of the data type; and
data type code containing information regarding the data type.
- [7] The method of claim 6, wherein the individual configuration information further comprises user data defining applications of the individual 3D compression data.
- [8] The method of claim 5, wherein the pieces of individual 3D compression data are generated by encoding 3D data according to an encoding method corresponding to code representing the data type.
- [9] The method of claim 8, wherein the data type code further comprises a plurality of bits used in resynchronization.
- [10] A method of recovering three-dimensional (3D) compression data, the method comprising:

separating the 3D compression data and configuration information of the 3D compression data from a received signal;
separating a plurality of pieces of individual 3D compression data and individual configuration information of the pieces of individual 3D compression data from the 3D compression data; and
decoding the pieces of individual 3D compression data using data type information of the 3D compression data, which is included in the individual configuration information.

- [11] The method of claim 10, wherein the individual configuration information comprises profile and level identification information of the 3D compression data.
- [12] The method of claim 11, wherein the configuration information further comprises user data defining applications of the 3D compression data.
- [13] The method of claim 10, wherein the individual configuration information comprises:
a version number of a tool list of a data type of the of the pieces of individual 3D compression data; and
a data type code containing information regarding the data type.
- [14] The method of claim 13, wherein the individual configuration information further comprises user data defining applications of the pieces of individual 3D compression data.
- [15] The method of claim 13, wherein the data type code further comprises a plurality of bits used in resynchronization.
- [16] An apparatus to generate three-dimensional (3D) compression data, the apparatus comprising:
a plurality of encoders which encode received 3D data according to different methods, and generate different types of pieces of individual 3D compression data; and
a multiplexer which generates configuration information of the 3D compression data and individual configuration information for the pieces of individual 3D compression data, receives the pieces of individual 3D compression data from the encoders and respectively combines the pieces of individual 3D compression data with the corresponding individual configuration information, generates the 3D compression data by multiplexing the combined data, and combines the 3D compression data with the configuration information and outputs the result of the combination.
- [17] The apparatus of claim 16, wherein the multiplexer generates the configuration information so that the configuration information comprises profile and level

- identification information of the 3D compression data.
- [18] The apparatus of claim 17, wherein the multiplexer generates the configuration information so that the configuration information further comprises user data, which defines applications of the 3D compression data.
- [19] The apparatus of claim 16, wherein the multiplexer generates the individual configuration information so that the individual configuration information comprises a version number of a tool list of a data type of the pieces of individual 3D compression data and a data type code containing information of the data type of the pieces of individual 3D compression data.
- [20] The apparatus of claim 19, wherein the multiplexer generates the individual configuration information so that the individual configuration information further comprises user data defining applications of the pieces of individual 3D compression data.
- [21] The apparatus of claim 16, wherein the encoder generates the individual 3D compression data by encoding the received 3D data according to an encoding method matching the data type code of the pieces of individual 3D compression data.
- [22] The apparatus of claim 16, wherein the multiplexer generates the data type code of the individual 3D compression data so that the data type code further comprises a plurality of bits used in resynchronization.
- [23] The apparatus of claim 16, wherein the multiplexer generates the configuration information so that the configuration information further comprises user data defining applications of the 3D compression data.
- [24] An apparatus to recover three-dimensional (3D) compression data, the apparatus comprising:
a demultiplexer which separates the 3D compression data and configuration information of the 3D compression data from a received signal, separates a plurality of pieces of individual 3D compression data and individual configuration information of the pieces of individual 3D compression data from the 3D compression data, and analyzes a data type code comprising data type information of the individual 3D compression data, the data type code being included in the individual configuration information; and
a plurality of decoders which recover 3D graphic data by decoding the pieces of individual 3D compression data according to the corresponding data type code.
- [25] At least one computer readable medium storing instructions that control at least one processor to perform a method of generating three-dimensional (3D) compression data, the method comprising:
generating configuration information of the 3D compression data;

generating a plurality of pieces of individual 3D compression data to be included in the 3D compression data; and

generating the 3D compression data by multiplexing the pieces of individual 3D compression data, combining the 3D compression data with the configuration information, and outputting the result of the combination.

[26] At least one computer readable medium storing instructions that control at least one processor to perform a method of recovering three-dimensional (3D) compression data, the method comprising:

separating the 3D compression data and configuration information of the 3D compression data from a received signal;

separating a plurality of pieces of individual 3D compression data and individual configuration information of the pieces of individual 3D compression data from the 3D compression data; and

decoding the pieces of individual 3D compression data using data type information of the 3D compression data, which is included in the individual configuration information.

[27] A method of generating three-dimensional (3D) compression data, the method comprising:

generating configuration information of a 3DCObjectSequence, which is the 3D compression data;

generating 3DCObjects, which are a plurality of pieces of individual 3D compression data that are included in the 3D compression data; and

generating the 3D compression data by multiplexing the pieces of individual 3D compression data, combining the 3D compression data with the configuration information, and outputting the result of the combination,

wherein the generation of configuration information of the 3D compression data, the generation of the pieces of individual 3D compression data, and the generation of the 3D compression data are realized by:

```
3DCObjectSequence() {
  bit(32) 3dc_object_sequence_start_code;
  bit(8) profile_and_level_indication;
  do {
    3DCObject();
    bit(32) *next;
  } while ( next != 3dc_object_sequence_end_code);
  bit(32) 3dc_object_sequence_end_code;
}
```

and wherein 3dc_object_sequence_start_code denotes code representing a start

of the 3D compression data, `profile_and_level_indication` denotes profile and level identification information of the 3D compression data, and `3dc_object_sequence_end_code` denotes code representing an end of the 3D compression data.

- [28] The method of 27, wherein the configuration information further comprises user data defining applications of the 3D compression data, wherein the user data is realized by:

```
user_data() {
  bit(23) *next;
  while(next !=0){
    bit(8) user_data_bits;
    bit(23) *next;
  }
}
```

wherein `user_data_bits` is an 8-bit integer that is defined for a specific application by a user.

- [29] The method of 27, wherein the generating of the 3DCObject comprises: generating individual configuration information of the 3DCObject; and generating the 3DCObjects according to `3dc_object_type_start_code`, which is data type code containing data type information of the 3DCObjects; wherein the generating of the individual configuration information for the 3DCObject and the generating of the 3DCObject are realized by:

```
3DCObject() {
  bit(32) 3dc_object_start_code;
  bit(1) is_3dc_object_identifier;
  bit(3) 3dc_object_verid;
  bit(4) 3dc_object_priority;
  bit(32) *next;
  if (next == 'Simple_3DMC') {
    bit(32) 3dc_object_type_start_code;
    3D_Mesh_Object();
  }
  else if (next == 'Simple_WSS') {
    bit(32) 3dc_object_type_start_code;
    Wavelet_Mesh_Object()
  }
  else if (next == 'Simple_CI') {
    bit(32) 3dc_object_type_start_code;
```

```

CompressedCoordinateInterpolator()
}
else if (next == 'Simple_OI') {
bit(32) 3dc_object_type_start_code;
CompressedOrientationInterpolatort()
}
else if (next == 'Simple_PI') {
bit(32) 3dc_object_type_start_code;
CompressedPositionInterpolator()
}
else if (next == 'Simple_BBA'){
bit(32) 3dc_object_type_start_code;
bba_object()
}
}

```

, and wherein 3dc_object_start_code denotes code representing a start of 3DCObject, is_3dc_object_identifier denotes an identifier of 3DCObject, 3dc_object_verid denotes code representing identification of a version of 3DCObject, and 3dc_object_priority denotes code representing a priority of 3DCObject.

[30] The method of claim 29, wherein the individual configuration information further comprises user data defining applications of the 3DCObject, wherein the user data is realized by:

```

user_data() {
bit(23) *next;
while(next !=0){
bit(8) user_data_bits;
bit(23) *next;
}
}

```

wherein user_data_bits is an 8-bit integer that is defined for a specific application by a user.

[31] The method of claim 29, wherein the 3dc_object_type_start_code comprises: bits '0000 0000 0000 0000 0000 0001' used in resynchronization; and one of values 'A6' through 'AB', which are 8 bits, as the data type information.

[32] A method of generating three-dimensional (3D) compression data, the method comprising:
generating configuration information of the 3D compression data;

generating a combination of individual configuration information of an individual 3D compression object included in the 3D compression data with the individual 3D compression object corresponding to the individual configuration information, and repeatedly generating the combination; and multiplexing the combinations to obtain a piece of data, and including the configuration information in the obtained data.

- [33] A method of recovering three-dimensional (3D) compression data, the method comprising:
separating the 3D compression data and configuration information of the 3D compression data from a received signal;
separating combinations of individual configuration information and individual 3D compression objects from the 3D compression data, and separating the individual 3D compression objects and the individual configuration information from the separated combinations; and
recovering 3D data by decoding the individual 3D compression objects using data type information of the individual 3D compression objects, the data type information being included in the individual configuration information.
- [34] A method of generating three-dimensional (3D) compression data which includes individual 3D compression objects, the method comprising:
generating a plurality of the individual 3D compression objects;
generating individual configuration information of the individual 3D compression objects; and
multiplexing the individual 3D compression objects and corresponding individual configuration information.
- [35] A method of recovering three-dimensional (3D) compression data, the method comprising:
demultiplexing individual 3D compression objects and corresponding individual configuration information from a received signal; and
decoding the individual 3D compression objects according to data type information included in the individual configuration information.

FIG. 1

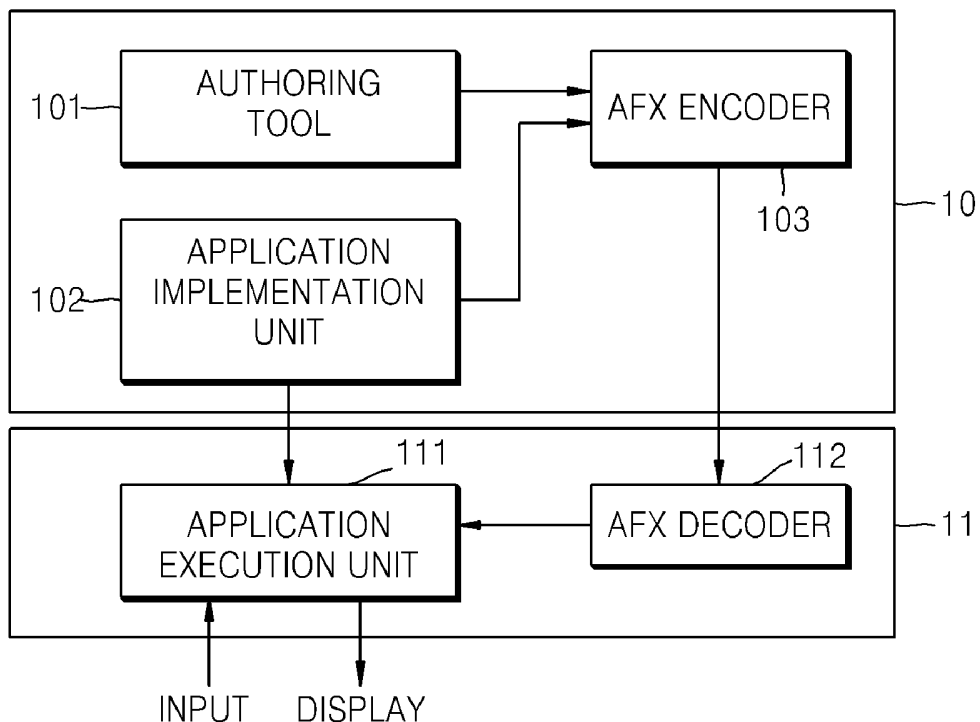


FIG. 2

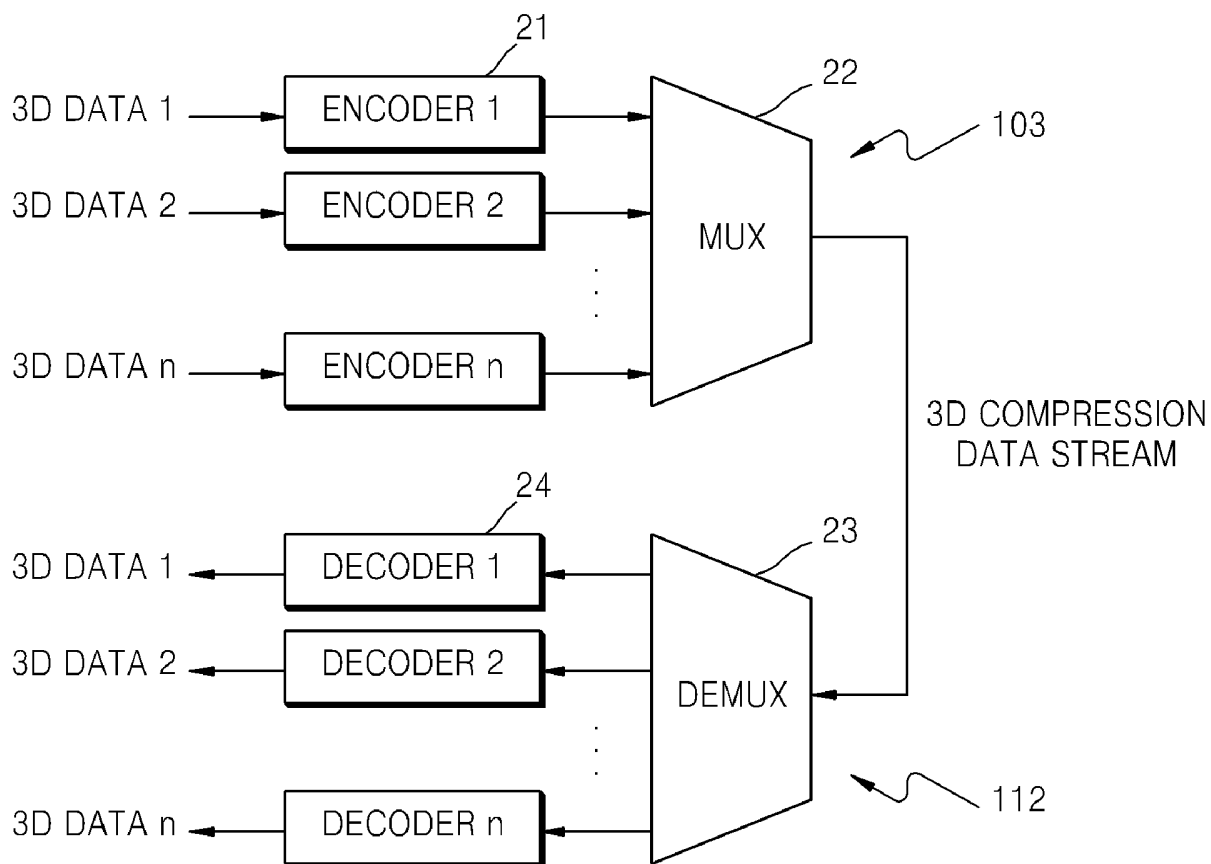


FIG. 3

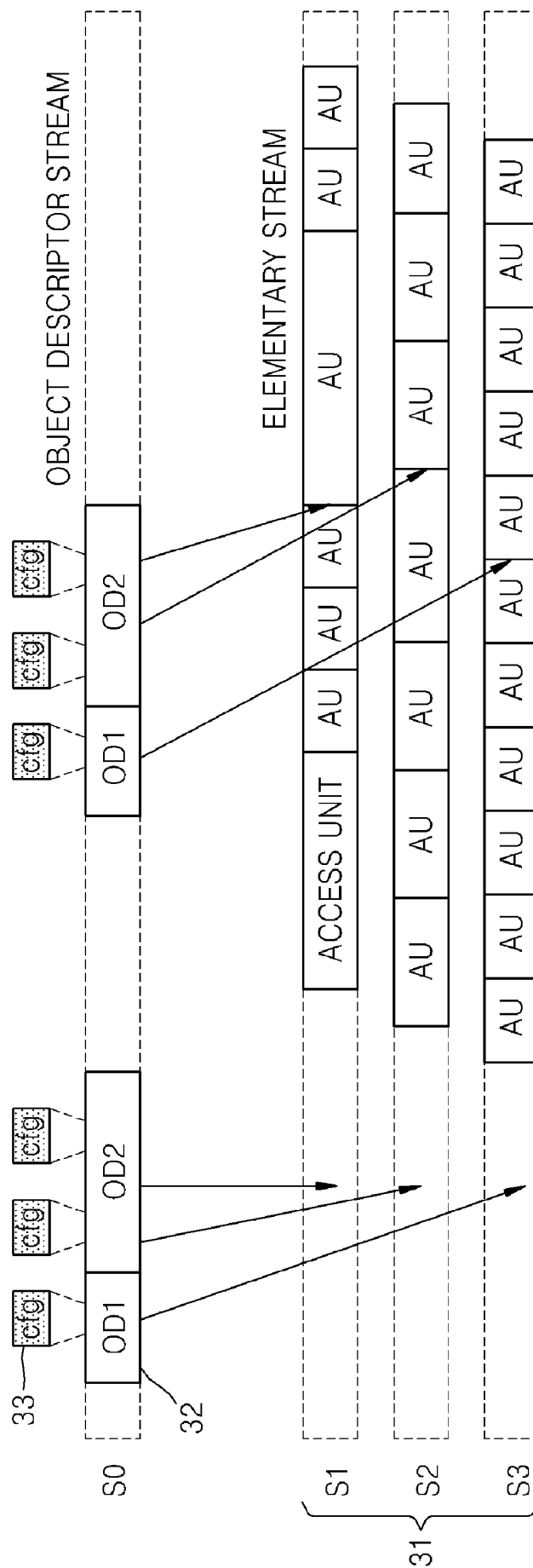


FIG. 4

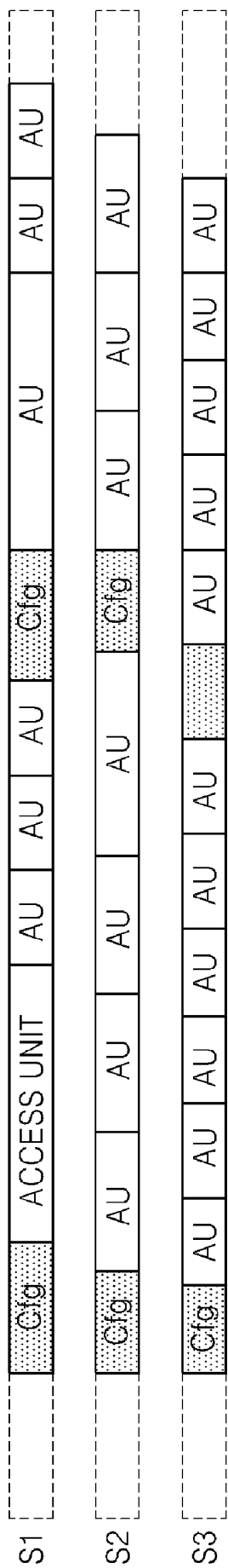


FIG. 5

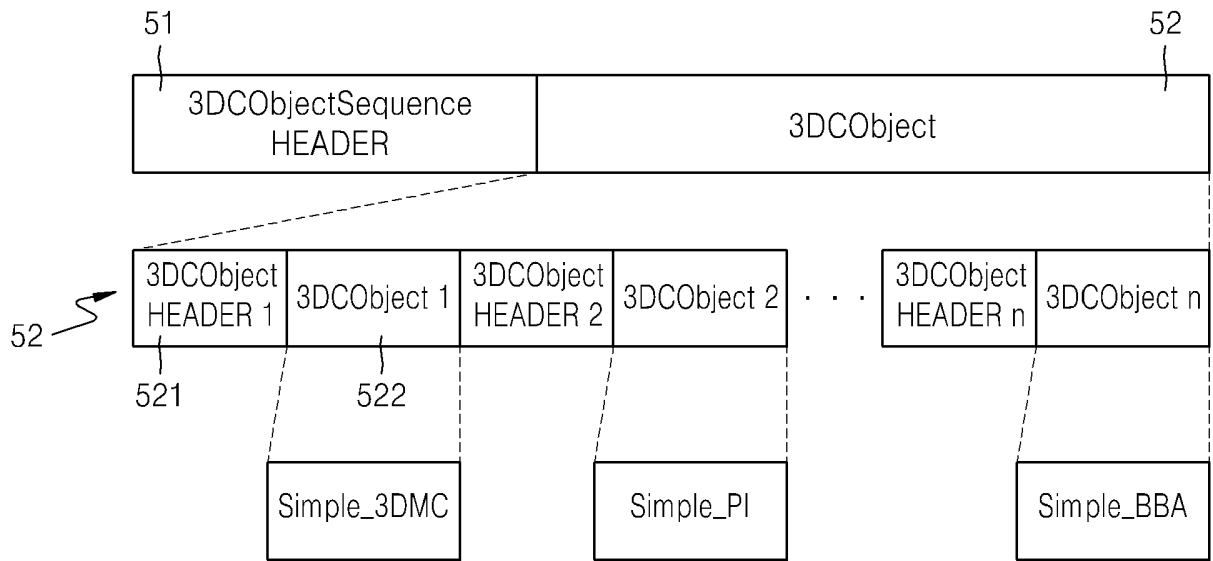


FIG. 6A

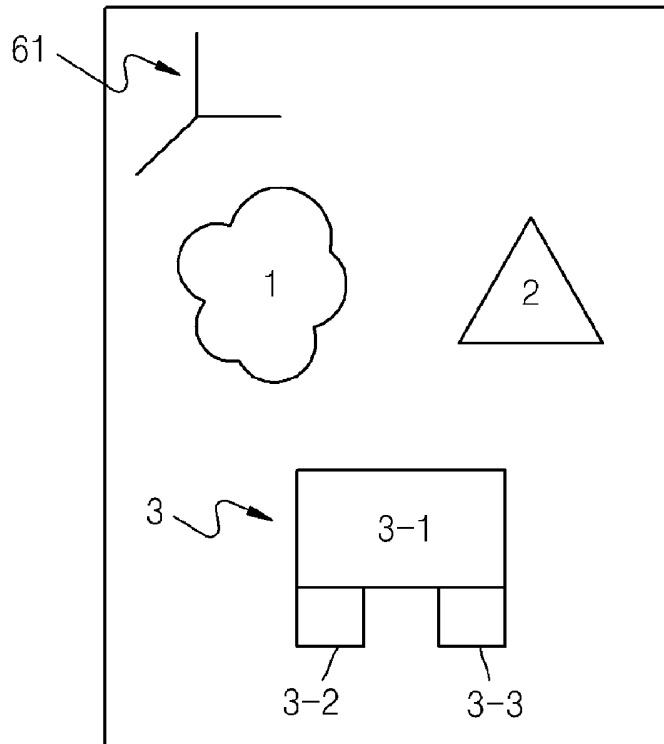
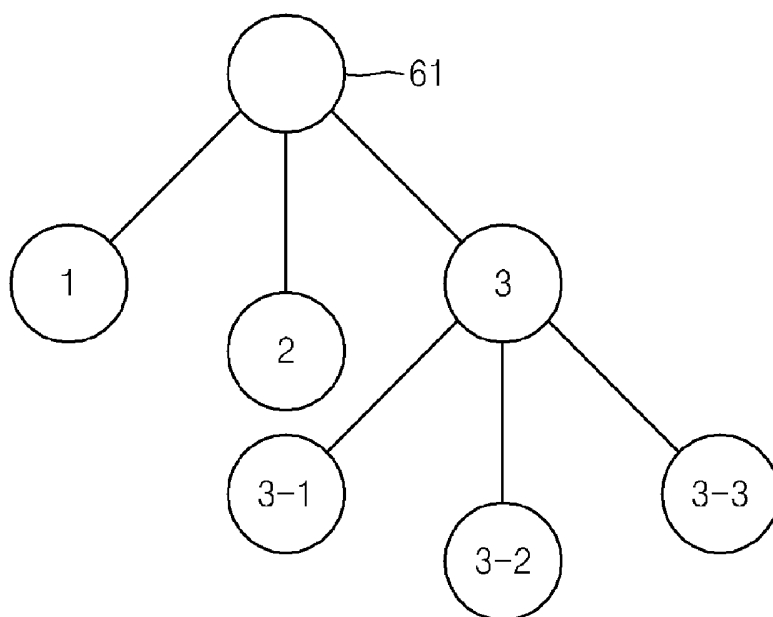


FIG. 6B



INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2006/001323

A. CLASSIFICATION OF SUBJECT MATTER

H03M 7/30(2006.01)i

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)
IPC 8: G06T 15/10, H04N 7/24

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean patents and applications for inventions since 1975
Korean utility models and applications for utility models since 1975

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKIPASS(Searching System of KIPO)



C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2003/0218606 A1 (ZHIRKOV, A.O. et al.) 27 November 2003 See figures 35, 36; paragraphs 517-528; claims 1, 5, 13;	1-35
A	KR 10-2003-05555 A (SAMSUNG ELECTRONICS CO., LTD.) 23 January 2003 See figures 4-6; page 3; claims 1-11.	1-35
A	KR 10-1999-26803 A (SAMSUNG ELECTRONICS CO., LTD.) 15 April 1999 See figure 3; pages 3-4; claims 2-3.	1-35

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

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)	"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
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"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 18 JULY 2006 (18.07.2006)	Date of mailing of the international search report 18 JULY 2006 (18.07.2006)
Name and mailing address of the ISA/KR  Korean Intellectual Property Office 920 Dunsan-dong, Seo-gu, Daejeon 302-701, Republic of Korea Facsimile No. 82-42-472-7140	Authorized officer SONG, Byoung Jun  Telephone No. 82-42-481-5677

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2006/001323

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US2003218606AA	27.11.2003	CA2413056AA	27.05.2003
		CA2413058A1	27.05.2003
		CA2514655A1	27.05.2003
		CA2517842A1	27.05.2003
		CN1218282C	07.09.2005
		CN1430183	16.07.2003
		CN1432969A	30.07.2003
		EP01321893A2	25.06.2003
		EP01321894A2	25.06.2003
		EP1321893A2	25.06.2003
		EP1321894A2	25.06.2003
		JP15296755	17.10.2003
		JP16005373	08.01.2004
		JP2003296755A2	17.10.2003
		JP2004005373A2	08.01.2004
		KR1020030043637	02.06.2003
		KR1020030043638	02.06.2003
US2003214502AA	20.11.2003		
KR1020030005555	23.01.2003	None	
KR1019990026803	15.04.1999	None	