

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
2 November 2006 (02.11.2006)

PCT

(10) International Publication Number
WO 2006/115384 A1

(51) International Patent Classification:
G06T 7/40 (2006.01)

(21) International Application Number:
PCT/KR2006/001592

(22) International Filing Date: 27 April 2006 (27.04.2006)

(25) Filing Language: Korean

(26) Publication Language: English

(30) Priority Data:
10-2005-0034823 27 April 2005 (27.04.2005) KR

(71) Applicant (for all designated States except US): **LG ELECTRONICS INC.** [KR/KR]; 20, Yoido-dong, Youngdungpo-ku, Seoul 105-875 (KR).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **HWANG, Jong Hyun** [KR/KR]; 530, 3gu, Yongmi-ri, gwangtan-myeon, paju-shi, Gyeonggi-do 413-855 (KR).

(74) Agent: **HAW, Yong Noke**; 8th Fl. Songchon Bldg., 642-15, Yeoksam-dong, Kangnam-ku, Seoul 135-080 (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, US, UZ, VC, VN, YU, ZA, ZM, ZW.

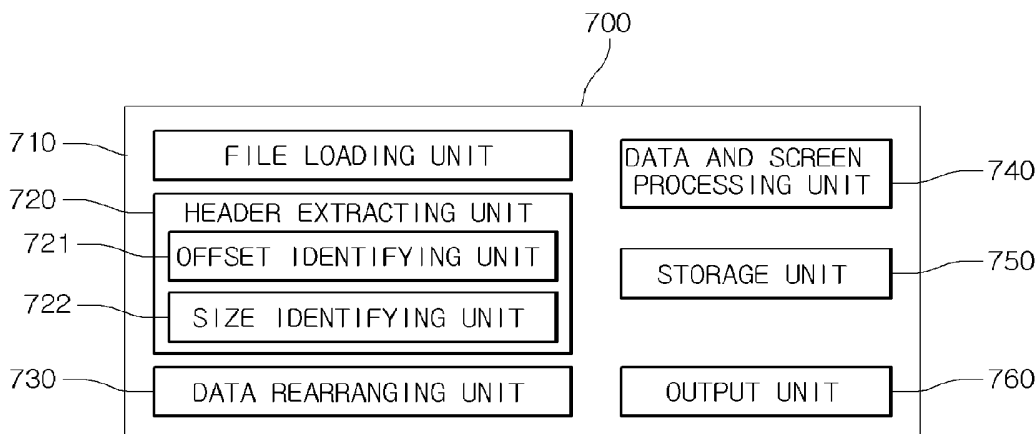
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, 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
- with amended claims

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.

(54) Title: APPARATUS AND METHOD OF CONVERTING DATA



(57) Abstract: A data converting apparatus and a method of converting data are provided. The data converting apparatus includes a header extracting unit extracting a file size information, and an offset information about starting of an actual data in a header of a file, and erasing the header of the file; a data rearranging unit extracting data representing an actual image in the file and rearranging the extracted data using at least one of the file size information and the offset information extracted by the header extracting unit; and a data processing unit converting the rearranged data into a display output format. The method of converting data having a header includes: erasing the header of the data; rearranging the data except for the header; and converting the rearranged data into a corresponding output format.

Description

APPARATUS AND METHOD OF CONVERTING DATA

Technical Field

- [1] The present invention relates to a data conversion, and more particularly, to a data converting apparatus and a method of converting data to output the converted output format.

Background Art

- [2] To display graphic files (GCF, PCX, JPEG, BMP, etc.) having an image form as a kind of the data on an output unit, various codecs that decode corresponding files are required to express the graphic files on an output unit.
- [3] The codecs express and decompress the graphic files in digital. Various kinds of graphic files including a bitmap file, respectively, have codecs for each corresponding compression format. Additionally, the compressed files are displayed on a screen using the corresponding codecs.
- [4] Accordingly, when the files contain same graphic images, each has a respectively different size according to a type of a codec. Even though same graphic images are displayed, each image quality can be different. Moreover, when there is no corresponding codec for a specific graphic file, the image of the graphic file can not be displayed.
- [5] For example, a codec for a compression format of the bitmap file is necessary so that a computer can analyze the bitmap file to display a corresponding image on a screen.
- [6] As described above, the bitmap file stores an image by dividing the image into a plurality of dots, contains various colors, and also is suitable for a colorful and flexible graph file such as a photo file.
- [7] However, the bitmap has a limitation in processing a delicate line or form. When enlarging or reducing an image, a quality of the image becomes coarse or damaged. Additionally, since a bitmap file has information in a pixel unit, the size of the bitmap file is large.
- [8] Accordingly, a codec, which decodes the bitmap file to be appropriate for characteristics of the output unit, is required to display the bitmap file on a screen.
- [9] The codecs express and compress the graphic files in digital. Additionally various kinds of graphic files (JPG, GIF, BMP, etc.) have corresponding codecs, respectively, for each compression format. Thus, the compressed file can be displayed on a screen using a corresponding codec.
- [10] Accordingly, even though files have a same graphic, the size of each file and an

image displayed on the screen are different, respectively, according to a type of a codec. Moreover, in case there is no corresponding codec for a specific graphic file, the image of the graphic file can not be displayed.

[11] In the bitmap, a codec corresponding to a compression format of a bitmap file is required so that a computer can analyze the bitmap file to display an original graphic file on a screen.

[12]

Disclosure of Invention

Technical Problem

[13] Accordingly, the present invention is directed to a data converting apparatus and a method of converting data that substantially obviate one or more problems due to limitations and disadvantages of the related art.

[14] An object of the present invention is to provide a data converting apparatus converting a structure of a graphic file and displaying an image without a corresponding exclusive codec for the graphic file when displaying the graphic file in an image files on a screen, and a method of converting data.

[15] Another object of the present invention is to provide a data converting apparatus converting a graphic file into a necessary format for an output by rearranging an actual data portion except for a header when displaying the graphic file.

[16]

Technical Solution

[17] Additional advantages, objects, and features of the invention will be set forth in part in the description which follows and in part will become apparent to those having ordinary skill in the art upon examination of the following or may be learned from practice of the invention. The objectives and other advantages of the invention may be realized and attained by the structure particularly pointed out in the written description and claims hereof as well as the appended drawings.

[18] To achieve these objects and other advantages and in accordance with the purpose of the invention, as embodied and broadly described herein, there is provided a data converting apparatus including: a header extracting unit extracting a file size information, and an offset information about starting of an actual data in a header of a file, and erasing the header of the file; a data rearranging unit extracting data representing an actual image in the file and rearranging the extracted data using at least one of the file size information and the offset information extracted by the header extracting unit; and a data processing unit converting the rearranged data into a display output format.

[19] In another aspect of the present invention, there is provided a method of converting

data having a header, the method including: erasing the header of the data; rearranging the data except for the header; and converting the rearranged data into a corresponding output format.

- [20] In a further another aspect of the present invention, there is provided a method of converting data having a header, the method including: reading the data except for the header; extracting a predetermined information in units of the read data; and converting the extracted information into a corresponding output format.

Advantageous Effects

- [21] In a data converting apparatus and a method of converting data of a bitmap file, an actual data of a bitmap file that stores an actual image upside down is rearranged to display an image on a screen without a codec. Additionally, data is converted and provided according to a graphic mode that a display supports.

[22]

Brief Description of the Drawings

- [23] The accompanying drawings, which are included to provide a further understanding of the invention and are incorporated in and constitute a part of this application, illustrate embodiment(s) of the invention and together with the description serve to explain the principle of the invention. In the drawings:
- [24] Fig. 1 is a block diagram of a structure in a bitmap file;
- [25] Fig. 2 is a table of header information in a bitmap file;
- [26] Fig. 3 is a view of an original image file and a stored image file;
- [27] Fig. 4 is a view of header information with a loaded original bitmap file in (a), (b), and (c) according to an embodiment of the present invention;
- [28] Fig. 5 is view of a new data file rearranged in an original address form by extracting a data portion except for a header of a bitmap file in (a) and (b);
- [29] Fig. 6 is view of processes converting a rearranged data into a corresponding output format;
- [30] Fig. 7 is a block diagram of a structure in an apparatus converting a bitmap file;
- [31] Fig. 8 is a flowchart of a method of converting a bitmap file according to an embodiment of the present invention;
- [32] Fig. 9 is a flowchart of a method of rearranging the data of Fig. 8;
- [33] Fig. 10 is a flowchart of a method of converting and storing the data of Fig. 8; and
- [34] Figs. 11 and 12 are firmware codes performing a data conversion.

[35]

Best Mode for Carrying Out the Invention

- [36] Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings. Wherever

possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

[37] A data converting apparatus and a method of converting data without a codec will be described. The conversion of a bitmap (BMP) file (a kind of a graphic file) will be described as an example.

[38] The data includes files with an image, a graphic, and a text.

[39] Fig. 1 is a block diagram of a structure in a bitmap file.

[40] Referring to Fig. 1, a bitmap file mainly includes a file information part 110, a bitmap information part 120, a palette 130, and an image data array unit 140.

[41] The file information 110 stores information about a bitmap file. The information includes a display identifier BM letter representing the bitmap file, an integer of a four byte representing the size of the bitmap file, two fields for a reserved value, and an offset information on how far from an actual image bit.

[42] Additionally, the bitmap information part 120 stores an information on the bitmap file itself. The information includes the size of a structure, a width and height of an image, the numbers of bit planes, the number of bits at each pixel, a compression format, the size of an image, a vertical and horizontal resolution, the number of actually used colors, and an important color index.

[43] Except for a true color, the palette 130 includes each color information according to the number of colors. For example, in a case of four grays, four color information (white 1, black 1, and gray 2) can be, respectively, represented with two bit.

[44] In the related art bitmap file (i.e., bmp file), the image data array part 140 stores pixel values of a row data in an image data without compression.

[45] That is, in a case of four grays, four color information (white 1, black 1, and gray 2) can be, respectively, represented with two bit. In a case of eight grays, eight color information can be, respectively, represented with three bit.

[46] Fig. 2(a) is a table of the file information part 110 in Fig. 1. Fig. 2(b) is a table of the bitmap information part 120 in Fig. 1.

[47] As illustrated in Fig. 2, the file information part 110 (BitMap File Header) includes 14 bytes, and the bitmap information part 120 (BitMap Information Header) includes 40 bytes.

[48] Accordingly, the header information can be erased on the basis of the header information, and then the actual data information can be extracted. For rearrangement, an address position is changed and stored.

[49] Fig. 3(a) is a view of an original image file 330, and Fig. 3(b) is view of an image stored upside down in a hexadecimal and the header information of Fig. 3(a).

[50] As illustrated in Fig. 3, when a red portion on a top of an original image file 330 in Fig. 3(a), it is stored on a bottom of Fig. 3(b).

- [51] Generally, when the bitmap file is stored on memory, it is stored from a left bottom, and also a horizontal line is stored in a multiple of four bytes. For example, when an image expressed with eight bits in each pixel has a horizontal line of thirty pixels, the number of efficiently required bytes is thirty two bytes not thirty bytes.
- [52] As illustrated in Fig. 3(b), a header information 310 of the image is stored on a top of Fig. 3(b).
- [53] The header information 310 includes information 310c on an offset to the actual image bit, information 310a on an image width of 10 bytes, information 310b on an image height of 10 bytes, information 310d on 24 bits (3 bytes) in each pixel, and information 310e on an RGB value of each pixel having an identical value.
- [54] In the bitmap file of a hexadecimal, FF, 80, 00, C0 represent RED, GREEN, BLACK, and GRAY, respectively.
- [55] Accordingly, when only an image information in the stored information with the header information is stored, the header information is separated, and then only the data portion is read and rearranged in an original form.
- [56] The data can include a file with an image, graphic, and text. An image, graphic, or bitmap file is used as data to describe the present invention for convenience, but it is apparent that the present invention can be identically applied to other data.
- [57] Fig. 4 is view of a bitmap file conversion.
- [58] When a bitmap file in RGB (Red, Green, and Blue) can be displayed on a full dot LCD of an output format without a codec, Data of the bitmap file is rearranged and converted into a predetermined gray (e.g., four grays).
- [59] That is, an example image is shown in Fig. 4(a). The image in an original bitmap includes R(410a), G(410b), and B(410c). Additionally, Fig. 4(b) is a view of a stored image. The stored image becomes an opposite position with respect to the actual image 330a like a data arrangement 330b of Fig. 3. However, this arrangement is limited to a bitmap image, and can be different in other image formats. The bitmap image 420 includes R(420a), G(420b), B(420c), and also an address structure includes an image data 430e and the header 430d in the image 430 of Fig. 4(c).
- [60] In Fig. 5(a), the data portion 430e except for the header 430d in the bitmap file of Fig. 4(b) is extracted, and then the new data file is rearranged like a table 550 of Fig. 5(b). Since addresses of the bitmap format are opposite to that of the actual image, it is convenient for encoding and decoding in the data structure. This can be applied to other files (e.g., jpg, png, gif, tif, tiff, etc.).
- [61] Accordingly, the rearranged data file is outputted by converting a pixel information having an address into a format corresponding to a predetermined output form.
- [62] First, general description will be made.
- [63] A required data is processed to display the rearranged data on the Full dot LCD, and

when the LCD supporting four level grays is used, the processes are as follows.

- [64] One byte value is selected from an RGB data of three bytes, which are used to express one pixel. Since each byte represents an RGB value to express a color, the RGB value is identical in a case of black and white. Thus, one byte in three bytes of the RGB data can represent the black and white.
- [65] For example, four expression methods are required to express four grays with one byte, and in this case, one byte (which is 8 bits and can express 256 colors) is unnecessary to express the four grays. Only two bits can represent the four grays. However, when a high resolution (e.g., eight grays or sixteen grays) is displayed, more bits are required. The expression of four grays is described using two bits below.
- [66] (1,1): gray-scale 3 Black
- [67] (1,0): gray-scale 2 Gray 1
- [68] (0,1): gray-scale 1 Gray 1
- [69] (0,0): gray-scale 0 White
- [70] Four expressions, that is, (0,0), (0,1), (1,0), and (1,1) represent White, Gray2, Gray1, and Black, respectively. The Gray-scale 0 to the Gray-scale 3 are predetermined values. One byte extracted from the three bytes of the RGB is hexadecimal values. Since a certain reference is required to convert a hexadecimal value into a binary value, the hexadecimal value is a predetermined value.
- [71] When a gray-scale 0 value is designated as below 0xFF and above 0xBF, one byte is converted into two bits, that is, (0,0) when the one byte includes small numbers such as 0xFE, 0xE0, etc. Consequently, eight bits (one byte) becomes two bits.
- [72] Transmission to an LCD data RAM is performed by one byte, and data of the one byte is eight bits. Thus, the one byte includes four pixels information. As described above, the data is sequentially stored from left to right. When one byte data is transmitted from an arbitrary pixel X point, one byte is formed of data of pixels (i.e., X, X + width, X + 2width, and X + 3width).
- [73] After one byte is selected from the one pixel with three bytes in Fig. 6(a), a hexadecimal value is converted into a binary value (2 bits) to display four grays as illustrated in Fig. 6(b).
- [74] Referring to Fig. 6(c), the converted binary in four selected bytes (address 32, 34, 24, and 27) is combined to represent one byte I.
- [75] Fig. 6(d) is a view of a reconverted image for four grays.
- [76] As illustrated in Fig. 6, the reconverted image 660 includes a converted R 610a, G 610b, and B 610c.
- [77] Additionally, simple structures 430 and 550 in Figs. 4 and 5 are views of the rearranged data. Referring to the address structure 430 of the bitmap file 420 in Figs. 4(b) and 4(c), the size information in the header includes a data size information in

width 6 and height 6 and information on offset 6. Accordingly, an actual data begins with an address number 6, and is stored upside down like Fig. 4.

[78] Accordingly, after an address number 30 to address number 35 (width six) in a bottom line are stored, an address number 24 to an address 29 (width 6) are stored in a top line to rearrange the actual data using an original image. This process repeats for storing.

[79] Using the processes above, as illustrated in Fig. 5(b), the data is rearranged and stored in an address number structure 550 corresponding to new data file 540.

[80] When a display device can support an RGB color in the address structure 550 of the new data file 540, an image can be displayed on the screen. When an output format of the display device is a full dot LCD, the data is converted into four grays one more, and then the reconverted file 550 can be displayed like Fig. 6(d).

[81] Fig. 7 is a block diagram of a structure in a bitmap file converting apparatus.

[82] Referring to Fig. 7, the bitmap file converting apparatus displays a bitmap file on the full dot LCD. The bitmap file converting apparatus includes a file loading unit 710 loading a bitmap file, a header extracting unit 720 extracting the size of the file and an offset information on beginning of an actual data after extracting only the header of the bitmap file loaded by the file loading unit 710, a data rearranging unit 730 rearranging the actual data of the bitmap file to be in an original image order using the file size and the offset information extracted from the header extracting unit 720, a data and screen processing unit 740 converting the data rearranged by the data rearranging unit 730 into four gray data to be displayed on the full dot LCD, a storage unit 750 storing the data processed by the data and screen processing unit 740, and an output unit 760 outputting the data stored on the storage unit 750 into the screen.

[83] At this point, the header extracting unit 720 includes a size identifying unit identifying the size information of the file and an offset identifying unit 721 identifying information on beginning of an actual data.

[84] Additionally, the header extracting unit 720 extracts and removes the header from the loaded data. The size identifying unit 722 identifies the size information of a corresponding bitmap file in the extracted header. The offset identifying unit 721 identifies the offset information on a distance between the header and the actual data.

[85] The data rearranging unit 730 rearranges the data of the bitmap file using the size of the actual file identified by the size identifying unit 722 and the offset information identified by the offset identifying unit 721.

[86] When the data rearranging unit 730 rearranges the data of the bitmap file, an actual data portion 430e of the bitmap file 430 stored upside down in Fig. 4 is rearranged into an original image 540 of Fig. 5, which is identical to the image 410 of Fig. 4. Detailed description will be made separately.

- [87] When an output unit is a color monitor supporting an RGB, the rearranged data can be instantly displayed. Additionally, when the output unit is the full dot LCD supporting four grays, the data with the RGB color needs to be converted into four grays.
- [88] Accordingly, the data and screen processing unit 740 for an LCD converts the data having the RGB value, which is rearranged by the data rearranging unit 730, into four gray data. The processing method will be described separately like the data rearrangement method.
- [89] The four gray data processed by the data and screen processing unit 740 is stored on the storage unit 750, and the output unit 760 displays the data stored on the storage unit 750 on the screen.
- [90] A method of converting a bitmap file will be described in more detail.
- [91] Fig. 8 is a flowchart of a method of converting a bitmap file according to an embodiment of the present invention. Fig. 9 is a flowchart of a method of rearranging the data of Fig. 8. Fig. 10 is a flowchart of a method of converting and storing the data of Fig. 8.
- [92] Referring to Figs. 7 to 10, the file loading unit 710 loads the bitmap file to display the bitmap file on the full dot LCD in operation S801. Additionally, the header extracting unit 720 extracts a header of the data that is loaded using the offset information.
- [93] In operation S802, the size identifying unit 722 of the header extracting unit 720 identifies the size information on the height and the size of the bitmap file from the extracted header. The offset identifying unit 721 identifies the offset information on beginning of the actual data.
- [94] Referring to figs. 4 and 7, the file loading unit 710 loads a structure 430 of an original bitmap file. The header extracting unit 720 extracts the header. The size identifying unit 722 identifies a width 6 and height 6 information of the actual data size. The offset identifying unit 721 identifies an offset 6 which is a distance between the actual data and the header.
- [95] After identifying the size information and the offset information in the operation S802, the data rearranging unit 730 rearranges the upside down bitmap file, and the rearrangement order is described in more detail with reference to Fig. 9.
- [96] Let's assumed that width is 6, height is 6, and offset is 6.
- [97] Referring to Fig. 9, when the data rearranging unit 730 rearranges the data of the bitmap file, a data starting address is searched first in operation S901. By the assumption above, the data starting address is an initial address $((6-1)*6 = 30)$ by the offset 6.
- [98] After the initial address is found, data (Addr + Width) of the initial address plus the

width is stored in operation S702. That is, data of addresses 30 to 35 is stored.

[99] After storing the data, an initial address is set to store the right above data by reducing a height value 1 in operation S903. Whether the result of the reducing of the height value is zero or not is determined in operations S904. When the result is not zero, the data is stored by repeating the operations from the S901 operation.

[100] More specifically, referring to the bitmap file address structure 430 of Fig. 4(c), after the data of the addresses 30 to 35 in a lowest bottom is stored in the operation S902, the height 6 is reduced to the height 5, which is in right above of the lowest bottom, to store the data of addresses 24 to 29 in the operation S903. Then, whether the result of the reducing of the height value is zero or not is determined in operations S904. An initial address 24 is calculated through the operation S901, and then the data of addresses 24 to 29 is stored repeatedly.

[101] When the height is 0 in the operation S904, it is determined that all data arrangement is finished, and also rearrangement operations terminate.

[102] After data rearrangement in the operation S803 of Fig. 8 is performed by a flowchart of Fig. 9, the data and screen processing unit 740 performs reprocessing operations to display the rearranged data on the full dot LCD with four grays in operation S804.

[103] Fig. 10 is a view of the rearranged data reprocessing operation S804.

[104] Referring to Fig. 10, one pixel includes an RGB value of three bytes in the bitmap file of the RGB color, and one of RGB is read to display on the full dot LCD of four grays in operation S1001.

[105] When an RGB value in one pixel represents a black and white in four grays, the RGB value is same, and thus any one of the RGB value can be selected to identify a color. Additionally, three bytes for one pixel are reduced to one byte.

[106] One byte in the operation S1001 needs to be transmitted to RAM storing the LCD data, and then is displayed on the full dot LCD. For this, the one byte value is converted into two bits in operation S1002.

[107] The reason of the converting the one byte into the two bits is to express four gray colors of a gray-scale 3 (black), a gray-scale 2 (brighter than the gray-scale 3), a gray-scale 1 (brighter than the gray-scale 2), and a gray-scale 0 (white) in the full dot LCD.

[108] That is, the four gray colors can be expressed as follows.

[109] Gray-scale 3 = (1,1)

[110] Gray-scale 2 = (1,0)

[111] Gray-scale 1 = (0,1)

[112] Gray-scale 0 = (0,0)

[113] Accordingly, the one byte in the operation S1001 is converted into the two bits to express four grays.

- [114] The data is transmitted to the RAM to be displayed on the full dot LCD in one byte unit.
- [115] At this point, since one byte includes eight bits, four pixels (each pixel with two bits) can be stored, and then transmitted. The data and screen processing unit 740 converts the continuous four pixels into one byte data in operation S1003, and then stores the one byte in the storage unit 750 in operation S1004.
- [116] In the operation S1001 to the operation 1004, the bitmap file in the RGB colors is converted into the data in four grays, and the converted data is stored in the storage unit 750. Then, the output unit 760 transmits the data into the RAM of the full dot LCD by one byte unit, and thus, the data is displayed on the display in operation S805.
- [117] Figs. 11 and 12 are firmware codes performing a data conversion.
- [118] As illustrated in Figs. 11 and 12, the firmware codes include loading a bitmap information having an original image and header information, rearranging the loaded information except for the header information, and converting the rearranged information into an output format.
- [119] As described above, the present invention relates an apparatus and a method of converting a bitmap file by extracting an actual data in the bitmap file and then rearranging the data. Thus, the bitmap file can be displayed on the screen without a codec. In the case of a four-gray LCD, the bitmap file with the RGB color is converted into four gray data and then the data is stored, which can support the four grays. Moreover, the size of the data is reduced and the data can be displayed on the LCD without a codec.
- [120] It is apparent that the various embodiments are possible when selectively including components within the scope of the present invention.
- [121] For example, in a case of four grays, the gray information can be stored in one byte and not three bytes at each pixel. Additionally, the one byte with the gray information can be converted into two bits. Thus, one twelfth of memory capacity can be reduced.
- [122] On the other hand, in a case of eight grays, the gray information can be stored in one byte. Additionally, the one byte with gray information can be converted into three bits (two of three bits in one byte). Accordingly, one sixth of the memory capacity can be reduced.
- [123] It will be apparent to those skilled in the art that various modifications and variations can be made in the present invention. Thus, it is intended that the present invention covers the modifications and variations of this invention provided they come within the scope of the appended claims and their equivalents.

Industrial Applicability

- [124] The present invention is applicable to fields for converting data and outputting the

converted data.

Claims

- [1] A method of converting data having a header, the method comprising:
erasing the header of the data;
rearranging the data except for the header; and
converting the rearranged data into a corresponding output format.
- [2] The method according to claim 1, wherein the data except for the header is rearranged by a predetermined method using a data starting information in the header.
- [3] The method according to claim 2, wherein the data except for the header is rearranged upside down and left to right using the data starting information in the header.
- [4] A method of converting data having a header, the method comprising:
reading the data except for the header;
extracting a predetermined information in units of the read data; and
converting the extracted information into a corresponding output format.
- [5] The method according to claim 4, wherein the predetermined information unit of a stored data is three bytes per one pixel.
- [6] The method according to claim 5, wherein each one byte is extracted from each pixel information in the three bytes, and then read.
- [7] The method according to claim 4, wherein each gray information in the extracted one byte is read according to a gray information to be outputted, and represented using a predetermined byte unit.
- [8] The method according to claim 7, wherein a supportable gray bit information is represented with more than one byte in the gray information of each pixel converted into more than one byte.
- [9] The method according to claim 8, wherein a respectively different gray information is extracted from each common gray information in each byte, and stored in more than one byte.
- [10] The method according to claim 8, wherein each of the four grays is represented in four of two bits, i.e., one byte, and the four grays are represented in one byte in a case of the four grays.
- [11] A data converting apparatus comprising:
a header extracting unit extracting a file size information, and an offset information about starting of an actual data in a header of a file, and erasing the header of the file;
a data rearranging unit extracting data representing an actual image in the file and rearranging the extracted data using at least one of the file size information

- and the offset information extracted by the header extracting unit; and
a data processing unit converting the rearranged data into a display output format.
- [12] The data converting apparatus according to claim 11, further comprising:
a data and screen processing unit reconvertng the rearranged data into a color mode that a display device supports; and
a storage unit storing the reconverted data of the data and screen processing unit.
- [13] A method of converting data, comprising:
extracting a file size information and an offset information about starting of an actual data in a header of a file;
erasing the header using the extracted offset information; and
rearranging the actual data of the file in a predetermined order using the extracted file size information, and storing the rearranged data.
- [14] The method according to claim 13, further comprising: reconvertng and outputting the rearranged data to be appropriate for a color information that a display device supports.
- [15] The method according to claim 13, wherein the files size information represents an actual height and width information of the data in a bitmap file during the extracting of the information.
- [16] The method according to claim 13, wherein the rearranging of the actual data comprises:
calculating a data starting address using the height and width information of the extracted file;
loading and storing sequentially the data stored from the calculated starting address to an address of the width information; and
loading and storing sequentially the data by reducing the height sequentially and using the reduced height information.
- [17] The method according to claim 16, wherein the loading of the data calculates a starting address using the reduced height information and the width information until the reduced height becomes zero, and loads sequentially the data, stored from the calculated starting address over an address of the width information, from left to right, and then stores the loaded data.
- [18] The method according to claim 13, wherein the file is a graphic file including bmp, jpg, png, gif, tif, tiff.

AMENDED CLAIMS
received by the International Bureau on 09 October 2006 (09.10.06)

【CLAIMS】**【Claim 1】**

A method of converting data having a header, the method comprising:

receiving the data having the header; rearranging the data except for the header; and

converting the rearranged data into a desired output format.

【Claim 2】

The method according to claim 1, wherein rearranging the data includes rearranging the data except for the header based on data starting information provided in the header.

【Claim 3】

The method according to claim 2, wherein rearranging the data further includes rearranging the data except for the header in an upside down and left to right manner based on the data starting information in the header.

【Claim 4】

A method of converting data having a header, the method comprising:

reading the data;

extracting predetermined information from the read data; and

converting the extracted information into a specific output format.

【Claim 5】

The method according to claim 4, wherein the predetermined information is stored in units of three bytes per one pixel.

【Claim 6】

The method according to claim 5, wherein extracting the predetermined information includes extracting one byte from the three bytes for each pixel.

【Claim 7】

The method according to claim 4, wherein gray information in the extracted one byte is read according to gray information to be outputted and is represented using a predetermined byte unit.

【Claim 8】

The method according to claim 7, wherein gray bit information is represented by more than one byte in the gray information for each pixel converted into more than one byte.

【Claim 9】

The method according to claim 7, wherein different gray information is extracted from common gray information in each byte, and is stored in more than one byte.

【Claim 10】

The method according to claim 7, wherein each of four grays is represented using two bits, and the four grays are represented as one byte.

【Claim 11】

A data converting apparatus comprising:

a header extracting unit to extract, from a header of a file, a file size information and an offset information regarding a starting location of an actual data;

a data rearranging unit to extract data representing an actual image from the file and to rearrange the extracted data using at least one of the file size information and the offset information extracted by the header extracting unit; and

a data processing unit to convert the rearranged data into a display output format.

【Claim 12】

The data converting apparatus according to claim 11, further comprising:

a data and screen processing unit to reconvert the rearranged data into a color mode that is supported by a display device; and

a storage unit to store the reconverted data of the data and screen processing unit.

【Claim 13】

A method of converting data, comprising:

extracting file size information and offset information about a location of actual data based on a header of a file;

rearranging the actual data of the file using the extracted file size information; and

storing the rearranged data.

【Claim 14】

The method according to claim 13, further comprising:

reconverting and outputting the rearranged data to be appropriate for a color format of a display device.

【Claim 15】

The method according to claim 13, wherein the file size information represents height and width information of the data in a bitmap file.

【Claim 16】

The method according to claim 13, wherein the rearranging of the actual data comprises:

calculating a data starting address using height and width information of the file;

storing the data based on the calculated data starting address, the height information and the width information provided in the header.

【Claim 17】

The method according to claim 16, wherein the storing of the data is based on the calculated starting address, the reduced height information and the width information until the reduced height becomes zero.

【Claim 18】

The method according to claim 13, wherein the file comprises a graphic file including at least one of bmp, jpg, png, gif, tif, or tiff.

【Claim 19】

The method according to claim 13, further comprising erasing the header based on the extracted offset information.

【Claim 20】

The method according to claim 16, wherein storing the data includes sequentially reducing a value of the height information.

【Claim 21】

The method according to claim 1, further comprising displaying an image on a screen in the desired output format based on the converted rearranged data.

【Claim 22】

The method according to claim 1, further comprising erasing the header based on offset information provided in the header, the erasing being performed after receiving the data having the header.

【Claim 23】

The method according to claim 1, further comprising storing the converted rearranged data.

【Claim 24】

The method according to claim 1, wherein rearranging the data is

performed based on height and width information provided within the header.

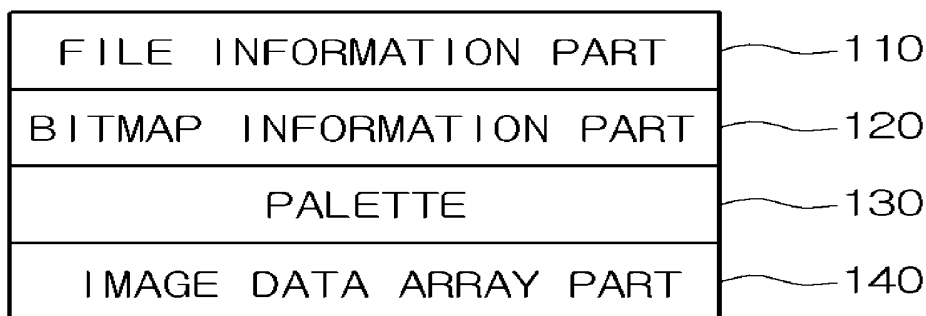
【Claim 25】

The method according to claim 1, wherein rearranging the data further includes calculating a data starting address based on the height and width information.

【Claim 26】

The method according to claim 11, wherein the header extracting unit erases the header of the file.

[Fig. 1]



[Fig. 2]

BitMap File Header : (Size(Byte))

(a)

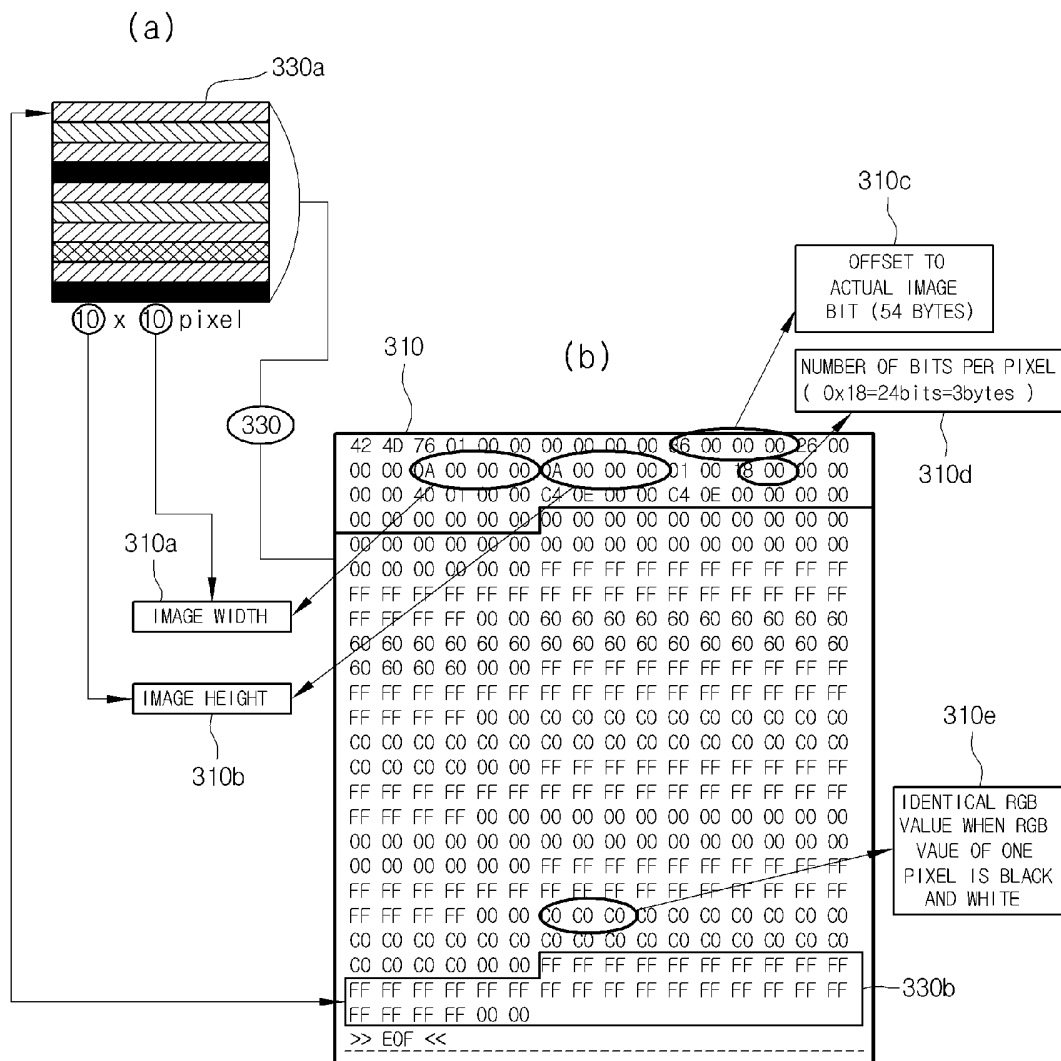
start	size	name	stdvalue	purpose
1	2	bfType	19778	must always be set to 'BM' to declare that this is a .bmp-file.
3	4	bfSize	??	specifies the size of the file in bytes.
7	2	bfReserved1	0	must always be set to zero.
9	2	bfReserved2	0	must always be set to zero.
11	4	bfOffBits	1078	specifies the offset from the beginning of the file to the bitmap data

Bitmap Infor Header : (Size(Byte))

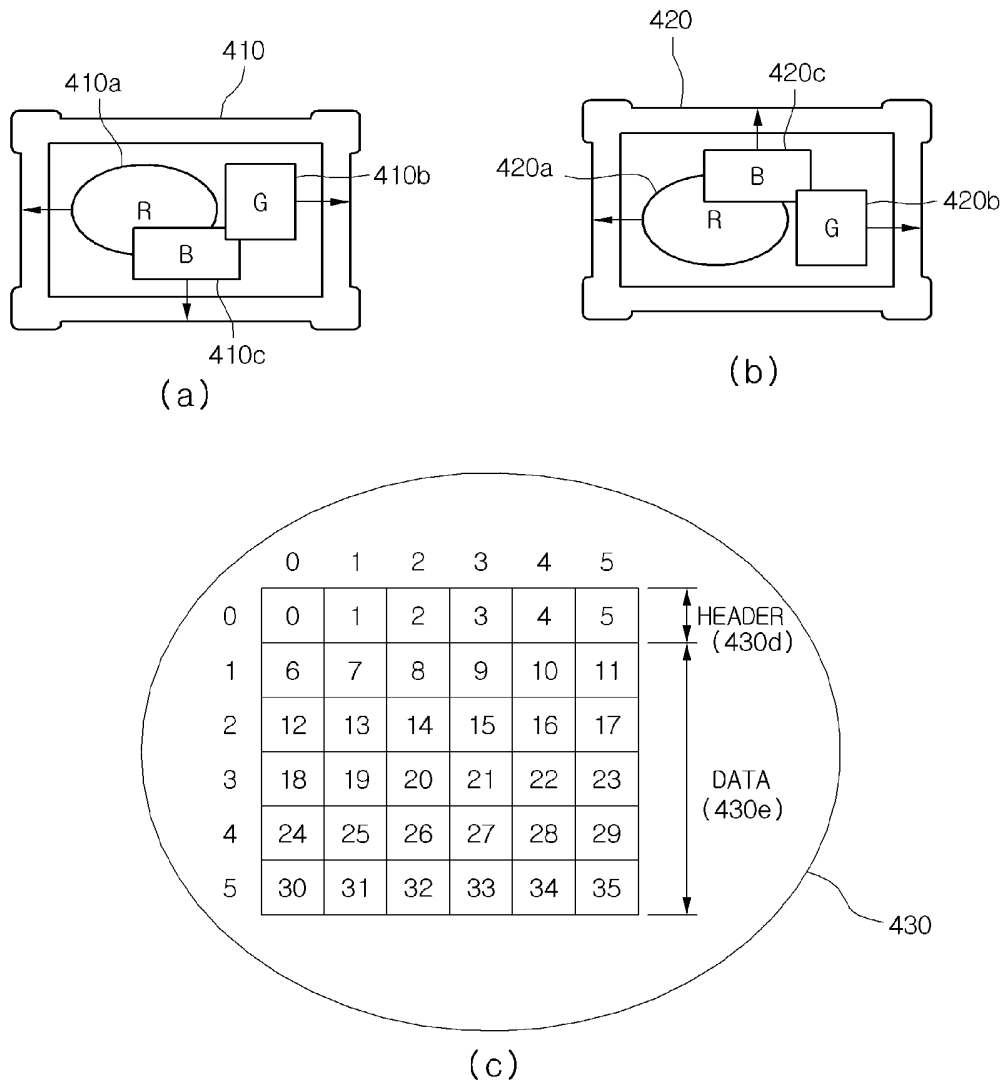
(b)

start	size	name	stdvalue	purpose
15	4	biSize	40	specifies the size of the BITMAPINFOHEADER structure, in bytes.
19	4	biWidth	100	specifies the width of the image, in pixels.
23	4	biHeight	100	specifies the height of the image, in pixels.
⋮				
43	4	biYPelsPerMeter	0	specifies the the vertical pixels per meter on the designated targer device, usually set to zero.
47	4	biClrUsed	0	specifies the number of colors used in the bitmap, if set to zero the number of colors is calculated using the biBitCount member.
51	4	biClrImportant	0	specifies the number of color that are 'important' for the bitmap, if set to zero, all colors are important.

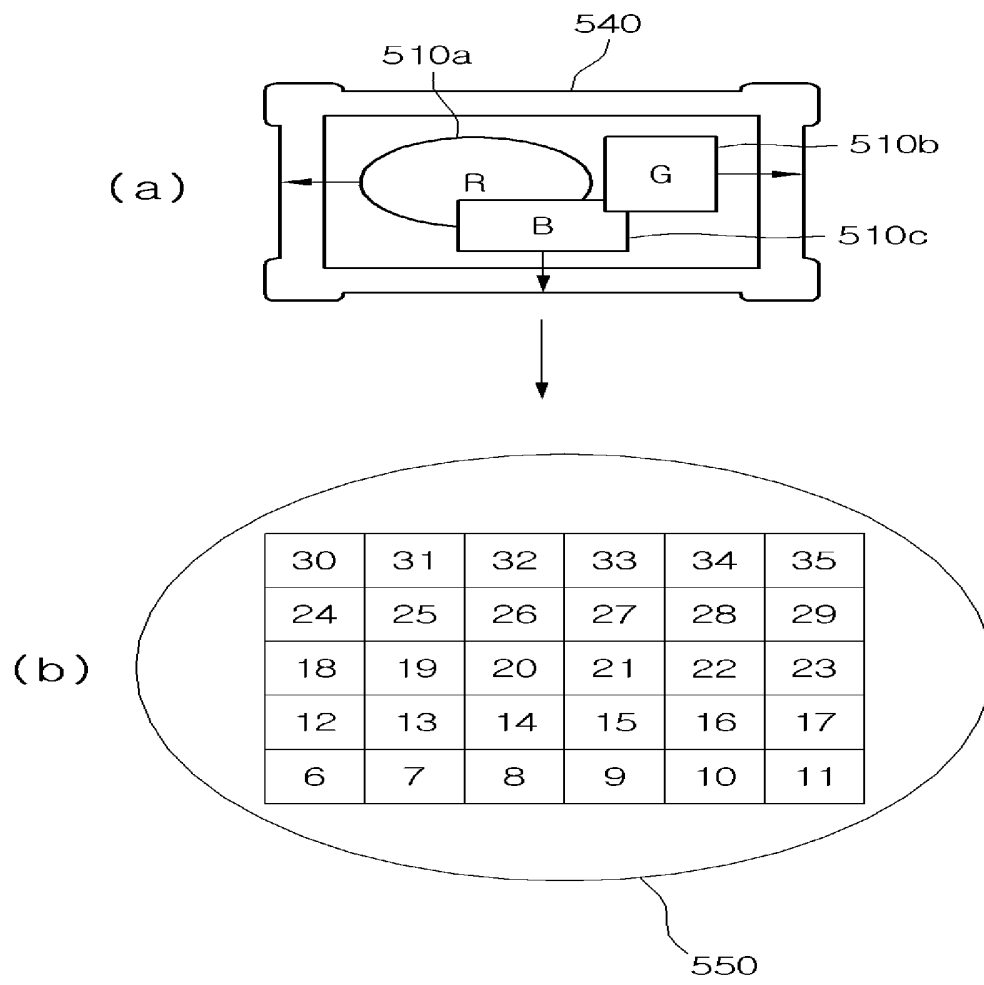
[Fig. 3]



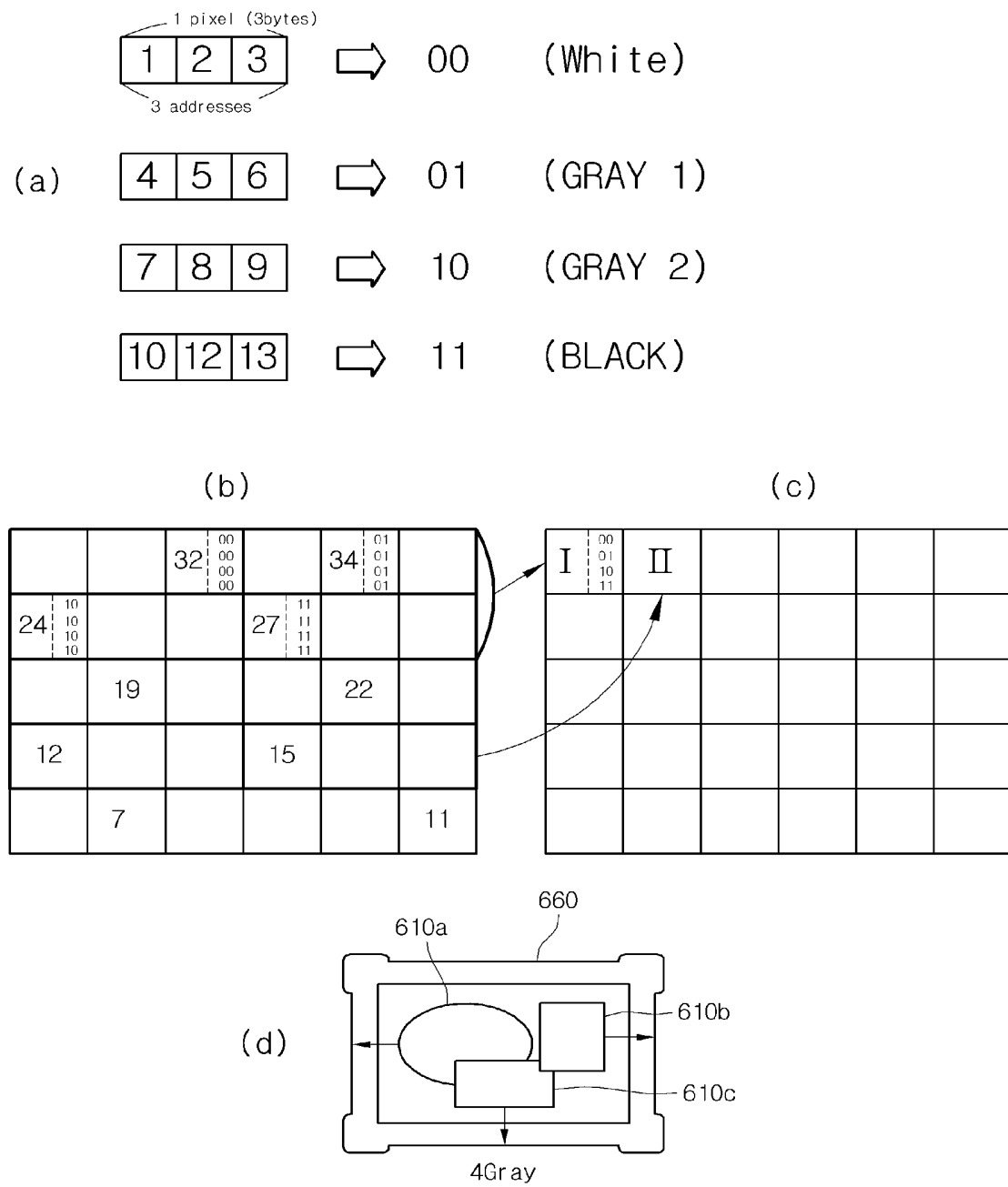
[Fig. 4]



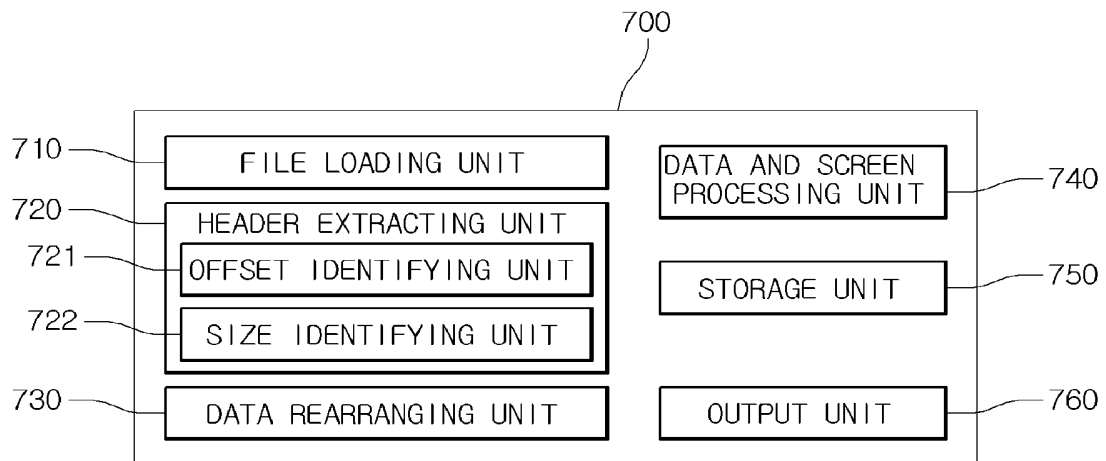
[Fig. 5]



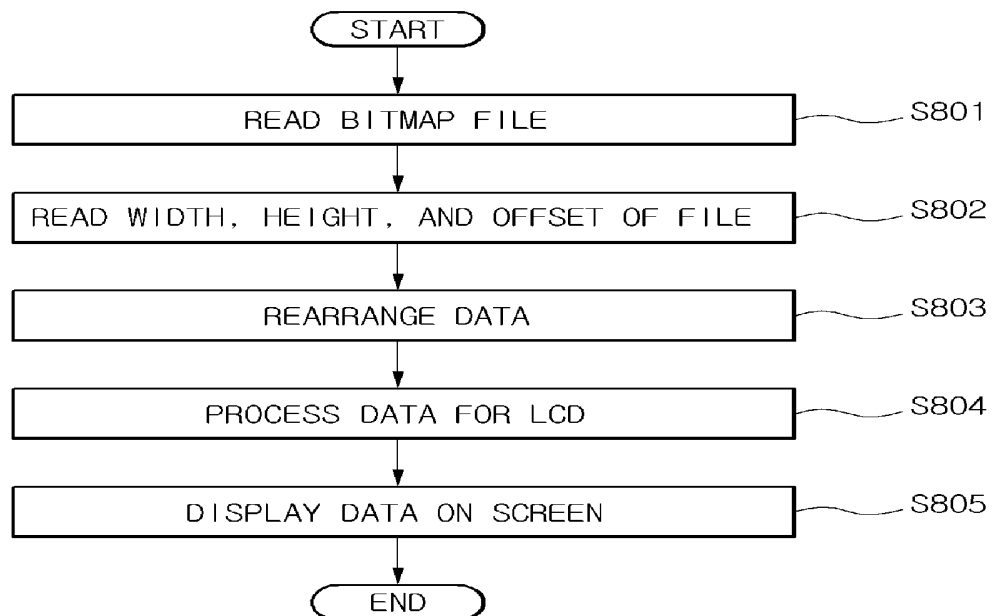
[Fig. 6]



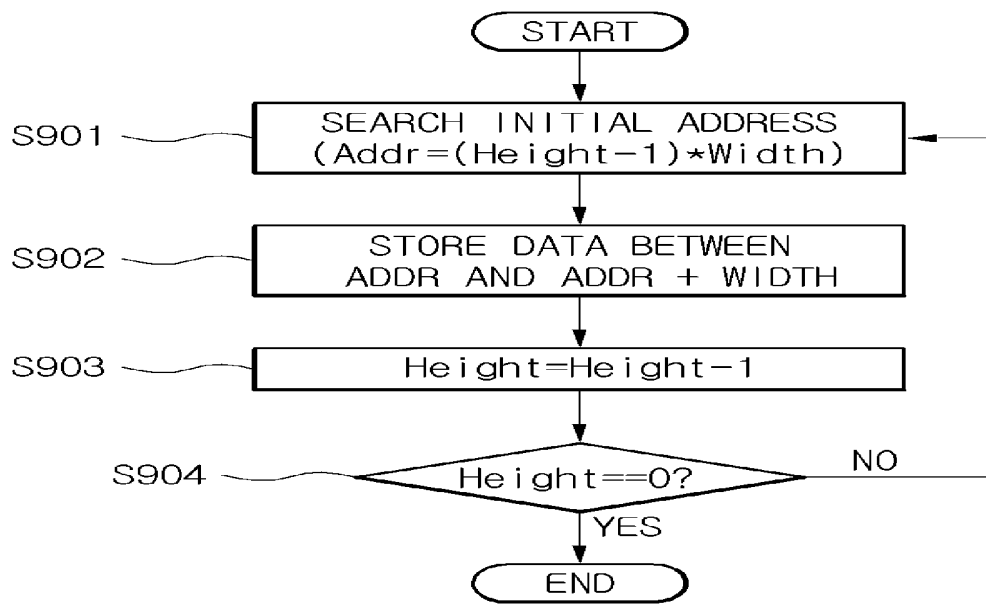
[Fig. 7]



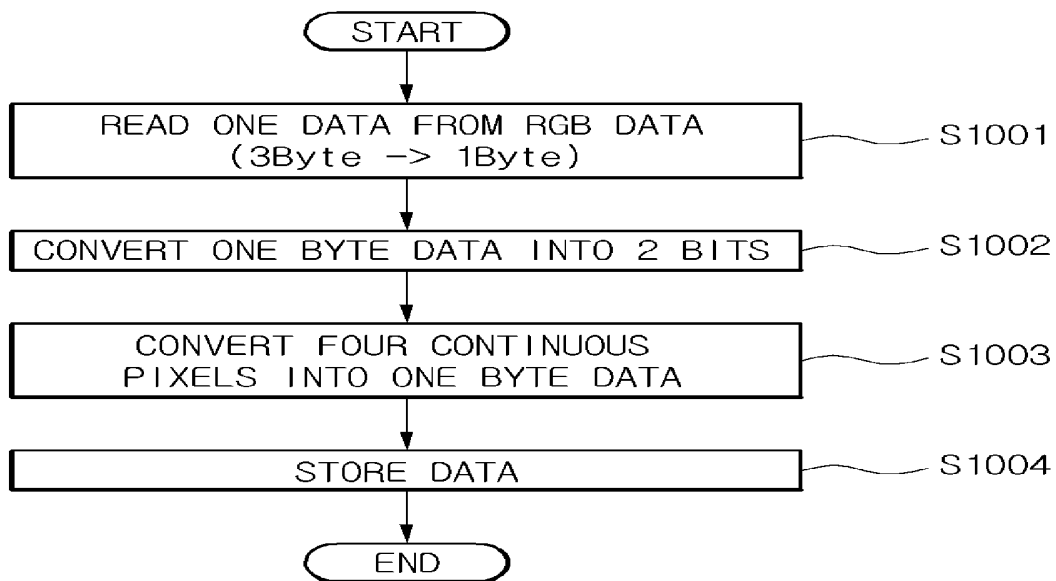
[Fig. 8]



[Fig. 9]



[Fig. 10]



[Fig. 11]

(a)

```

* Reading bitmap information(Offset,Width,Height etc.)
//////////get pixel width of file//////////
    fseek(out1,19,SEEK_SET);
    width1=fgetc(out1);
    if(width1==1) width=256;
    else if(width1==0)
{
    fseek(out1,18,SEEK_SET);
    width0=fgetc(out1);
    width=width0;
}
    else
{
    width=0;
    fprintf(out2,"%s ERROR!!Wn",buf);
}
//////////get pixel height of file//////////
    fseek(out1,22,SEEK_SET);
    height=fgetc(out1);

```

(b)

```

* Rearranging data
//get offset value which indicates the byte distance to
data from first byte of file//
    fseek(out1,10,SEEK_SET);
    offset=fgetc(out1);
    fseek(out1,offset,SEEK_SET);
    for(int a=0; a<height; a++)
{
    for(int b=0; b<width; b++)
    {
        source[a][b] = fgetc(out1);
    }
}

```

[Fig. 12]

(a)

```

* Converting data value for LCD
for(int c=0; c<height; c++)
{
    for(int b=0; b<width; b++)
    {
        ch = source[c][b];
        ch = ch & 0x0f;
        if(ch==0x01) ch = 0;
        else if(ch==0x02) ch = 1;
        else if(ch==0x03) ch = 2;
        else if(ch==0x00) ch = 3;
        fprintf(out2,"%c",ch);
    }
}

```

(b)

```

* Storing 4 pixel information into 1 byte
for(c=height-1; c>=0; c-=4)
{
    for(int b=0; b<width; b++)
    {
        ch1 = source[c][b] & 0x03;
        ch2 = (source[c-1][b]<<2) & 0x0c;
        ch3 = (source[c-2][b]<<4) & 0x30;
        ch4 = (source[c-3][b]<<6) & 0xc0;
        ch = ch1 | ch2 | ch3 | ch4;
        fprintf(out4,"%c",ch);
        Maddr++;
    }
}

```

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2006/001592**A. CLASSIFICATION OF SUBJECT MATTER****G06T 7/40(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)

IPC8 G06T

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

Japanese 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)

WPI, PAJ "data", "image", "convert", "codec", "header", "video", "decode", "offset"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 10-2002-0032020 A (SAMSUNG ELECTRONICS CO., LTD.) 03.05.2002 * See abstract, figures.	1-18
A	KR 10-2004-0039806 A (LG ELECTRONICS INC.) 12.05.2004 * See abstract, figures.	1-18
A	KR 10-2002-0038006 A (ACEINTECH.COM) 23.05.2002 * See abstract, figures.	1-18

☐ 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

"E" earlier application or patent but published on or after the international filing date

"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)

"O" 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

"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

"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

"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

"&" document member of the same patent family

Date of the actual completion of the international search

08 AUGUST 2006 (08.08.2006)

Date of mailing of the international search report

08 AUGUST 2006 (08.08.2006)

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

LEE, Seung Joo

Telephone No. 82-42-481-8186



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2006/001592

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
KR1020020032020	03.05.2002	JP14152725 US20020048323A1	24.05.2002 25.04.2002
KR1020040039806	12.05.2004	None	
KR1020020038006	23.05.2002	None	