



- (51) International Patent Classification:
G06F 9/24 (2006.01) *G06F 9/44* (2006.01)
- (21) International Application Number:
PCT/US2009/050483
- (22) International Filing Date:
14 July 2009 (14.07.2009)
- (25) Filing Language: English
- (26) Publication Language: English
- (71) Applicant (for all designated States except US):
HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 11445 Compaq Center Drive W., Houston, TX 77070 (US).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): **GILLESPIE, Kurt, David** [US/US]; 11445 Compaq Center Drive W., Houston, TX 77070 (US).
- (74) Agents: **WASSON, Robert, D.** et al.; Hewlett-packard Development Company, L.P., Intellectual Property Administration, P.o Box 272400, Mail Stop 35, Fort Collins, CO 80527-2400 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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,

[Continued on next page]

(54) Title: MODIFYING A VIDEO RESOLUTION OF VIDEO DATA IN A PRE-OPERATING SYSTEM ENVIRONMENT

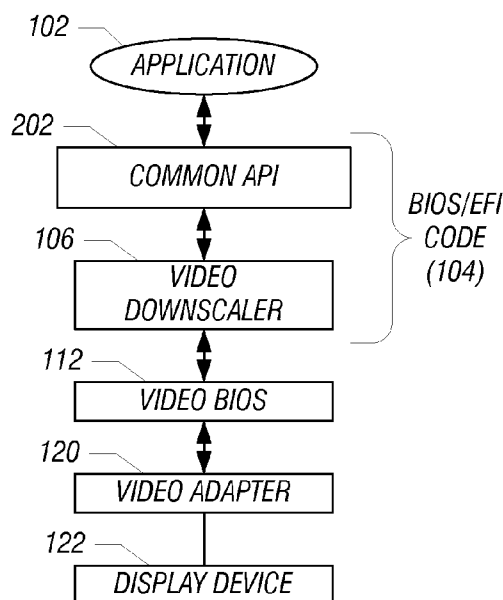


FIG. 2

(57) Abstract: Code (104) is run in a computer (100) to establish a pre-operating system (OS) environment. Video data is generated in the pre-OS environment for display at a display device (122), where the video data has a first video resolution. A video resolution of the video data is modified to provide a second video resolution, where the video data having the second video resolution is output for display at a display device.



ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, **Published:**

MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM,
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
ML, MR, NE, SN, TD, TG).

— *with international search report (Art. 21(3))*

- 1 -

Modifying A Video Resolution Of Video
Data In A Pre-Operating System Environment

Background

[0001] When a computer is initially started, code is executed to perform boot services, which include initializing and testing computer components, and other tasks. At the end of the boot procedure, an operating system (OS) is loaded.

[0002] In the pre-OS environment (prior to loading of the OS), certain applications may run, and such applications may output video data for display at a display device of the computer. For example, some applications may output status information during the boot procedure to indicate whether any faults are detected.

[0003] Some applications that operate in a pre-OS environment are designed to output video data having a particular video resolution. In some cases, there may be incompatibilities between the resolution specified by the application video data and the resolution supported by a display device of a computer.

Brief Description Of The Drawings

[0004] Some embodiments of the invention are described with respect to the following figures:

Fig. 1 is a block diagram of an exemplary computer that incorporates a video downscaler according to an embodiment;

Fig. 2 is a block diagram illustrating various layers including the video downscaler according to an embodiment;

Fig. 3 is a flow diagram of system operation according to an embodiment; and

Fig. 4 is a flow diagram of a process performed by the video downscaler according to an embodiment.

Detailed Description

[0005] Upon startup of a computer, a boot procedure is performed in which various tests, initializations, and other tasks are performed. Traditionally, such a boot procedure is

- 2 -

performed by the basic input/output system (BIOS) code. The BIOS code traditionally identifies, tests, and initializes system components. The BIOS code can also perform other tasks. At the end of the boot procedure, the BIOS code loads an operating system (OS).

[0006] More recently, an Extensible Firmware Interface (EFI) has been defined to provide code that performs tasks traditionally performed by BIOS code in addition to other tasks. The EFI code provides boot services as well as other services that are not available or standardized with traditional BIOS code. The EFI code also includes a boot manager for loading an operating system. The EFI specification is provided by the Unified EFI (UEFI) Forum.

[0007] The EFI is an interface between the operating system (and/or application software) and platform firmware, where platform firmware refers to programmable content embedded in hardware components, such as microprocessors, application-specific integrated circuits (ASICs), programmable logic devices, peripheral devices, and so forth. The platform firmware can include code to perform management of health of the computer. The platform firmware can also include certain BIOS code, such as BIOS code to perform power management tasks or to provide video capabilities.

[0008] Prior to loading of the operating system, during execution of the BIOS code or EFI code, various applications may run in the pre-OS environment, such as by using EFI-provided APIs (application programming interfaces) and functionality. The pre-OS environment refers to an environment after the BIOS/EFI code has started running but prior to loading of the OS. Examples of applications that can run in the pre-OS environment include a preboot application (such as an application that allows access, in the pre-OS environment, to email or calendar data associated with an application such as Microsoft Outlook), a test application (that performs certain tests in the pre-OS environment), a graphical package for system setup, and other applications. Most of the applications that run in the pre-OS environment produce video data that is to be output on a display device.

[0009] When applications that are able to output video data in a pre-OS environment are developed, developers may assume that the applications are to operate in computers having display devices that support at least a certain video resolution (*e.g.*, 1024x768). However, such an assumption may be incorrect, and the applications may in fact run in computers with

- 3 -

display devices that do not support the particular resolution assumed for the applications. For example, certain computers may employ smaller display devices, such as “netbooks” (which are cheaper computers designed for wireless communication and access to the Internet). For example, some “netbooks” support only a video resolution of 800x600. Other computers may support a video resolution of 1024x720, which is less than 1024x768.

[0010] Although example video resolutions have been given above, it is noted that other implementations can employ other video resolutions.

[0011] Typically, it would be expensive to redesign applications that operate in the pre-OS environment to output video data at a lower video resolution. Redesigning applications can be time consuming, since developers have to change the code of the applications, followed by application testing.

[0012] In accordance with some embodiments, to address the issue of an application outputting video data, in a pre-OS environment, at a video resolution that is greater, in height or width, than the video resolution supported by a display device of a computer, a video resolution modifier module is provided that executes in the pre-OS environment for modifying a resolution of video data from the application to be consistent with the resolution supported by the display device of the computer. One type of the video resolution modifier module is a downscaler, which reduces the video resolution of the video data from the application. In the ensuing discussion, reference is made to use of a downscaler; however, it is noted that in alternative embodiments, the same techniques can be applied to a video resolution modifier module that can increase the video resolution of video data from the application.

[0013] Generally, in accordance with some embodiments, an application that executes in the pre-OS environment outputs video data at a first resolution. This video data at the first resolution is received by the downscaler, which applies video resolution modification to reduce the resolution to a second resolution that is consistent with a resolution supported by the display device of the computer. The downscaler then outputs the modified video data having the second resolution to video firmware, which is the interface to a video adapter of the computer, to cause the modified video data to be output on the display device of the computer.

- 4 -

[0014] Fig. 1 is a block diagram of an exemplary computer 100, which includes software applications 102 and BIOS/EFI code 104. The BIOS/EFI code 104 includes code 108 to perform boot services and a video downscaler 106. Note that the BIOS/EFI code 104 can also include other code that is not depicted in Fig. 1 for performing other tasks.

[0015] The computer 100 also includes an operating system 124 that is loaded by the BIOS/EFI code at the end of a boot procedure.

[0016] Platform firmware 110 is also provided between the BIOS/EFI code 104 and hardware 114 of the computer 100. The platform firmware 110 can include various code, including video BIOS code 112 (or other type of video firmware) that supports the video capabilities of the computer 100 in the pre-OS environment. Note that once the OS 124 is loaded, a video driver of the OS 124 supports the video capabilities of the computer 100, rather than the video BIOS code 112.

[0017] The hardware 114 of the computer 100 includes a processor 116, storage media 118 (*e.g.*, one or more disk-based storage devices and/or one or more integrated circuit or semiconductor memory devices), and a video adapter 120. The video adapter 120 is connected to a display device 122. The video adapter 120 works in conjunction with the video BIOS code 112 (or with a video driver of the OS 124 when the OS is loaded) to enable provision of video data to the display device 122.

[0018] As further shown in Fig. 2, various components that are employed in the pre-OS environment are shown. The application 102 outputs video data to a common application program interface (API) 202, which can be part of the BIOS/EFI code 104. The common API 202 provides API routines that can be called by applications for drawing video data on a display device. Provision of the common API 202 allows for applications designed for execution in the pre-OS environment to work with various different types of display devices, such as a display device on a notebook computer, an externally connected display device (such as a display device connected through a Universal Serial Bus or USB port), or a display device connected over a network.

[0019] Application video data received by the common API 202 is forwarded to the video downscaler 106, which determines whether or not modification of the video resolution

- 5 -

of the application video data is to be performed. If video resolution modification does not have to be performed, then the video downscaler 106 just forwards the application video data to the video BIOS code 112 without modification. However, if video resolution modification has to be performed, then the video downscaler 106 applies video resolution modification to the application video data, and the modified application video data is then provided by the downscaler 106 to the video BIOS code 112.

[0020] The video BIOS code 112 in turn provides application video data received from the video downscaler 106 to the video adapter 120, which in turn outputs the video data to the display device 122 for display.

[0021] Fig. 3 is a flow diagram of system operation according to an embodiment. Upon system startup, BIOS/EFI code is executed (at 302), such as by the processor 116 of the computer 100 (Fig. 1). Execution of the BIOS/EFI code establishes a pre-OS environment. In the pre-OS environment, an application (102) is run (at 304), where the application 102 outputs video data.

[0022] If video resolution modification is to be performed, the video downscaler 106 modifies (at 306) a resolution of the video data from the application to be consistent with a resolution supported by the display device 122. The modified video data is then sent (at 308) to the video BIOS code 112 to cause the video data to be displayed.

[0023] Fig. 4 is a flow diagram of tasks performed by the video downscaler 106 according to an embodiment. The video downscaler 106, when invoked (such as by the common API 202), queries (at 402) the display device for the supported resolution. Note that querying the display device involves querying the video BIOS code 112, which contains information regarding the video resolution that is supported by the video BIOS code 112. For example, the video BIOS code 112 may support an 800x600 resolution, or alternatively, the BIOS code 112 may support a 1024x720 resolution. Either of these video resolutions may be incompatible with the resolution of the application video data, which may be at 1024x768.

[0024] The video downscaler 106 compares (at 404) the supported resolution to the resolution of the application video data. The video downscaler 106 then determines (at 406) whether video resolution modification is to be performed. If not, then the application video

- 6 -

data can be directly output to the video BIOS code 112 without modification. However, if video resolution modification is to be performed, the video downscaler 106 modifies (at 408) the resolution of the application video data. The modified application video data is then output to the video BIOS code 112.

[0025] In some implementations, such as in embodiments where the supported resolution of the display device is 800x600, the modification by the downscaler 106 may be to transform each block of 4x4 pixels of the video data into a block of 3x3 pixels, where the values assigned to the pixels in the 3x3 block are based on averaging values of the pixels in the 4x4 block. Note that the original video data (which may have a resolution of 1024x768 pixels) can be divided into blocks, such as the 4x4 blocks noted above. After transforming the 4x4 blocks to 3x3 blocks, the resolution of the video data is decreased by about 25% along each of horizontal and vertical directions of the video data. On the other hand, if the supported resolution of the display device is 1024x720, then the video resolution modification may be performed only in the vertical direction, in order to obtain a final resolution less than or equal to the desired resolution in both the vertical and horizontal directions.

[0026] In other implementations, other resolutions of the application video data may be employed. Also, different display devices support different resolutions.

[0027] After downscaling, there may be portions on the sides or the top and bottom margins of the video data that are empty. The video downscaler 106 can provide a border around the video data to fill in the empty portions prior to display.

[0028] Note that the modification of the resolution of the application video data is performed transparently to the application. The application may be provided with feedback (such as from the common API 202 in Fig. 2) indicating that the original resolution (*e.g.*, 1024x768) is supported by the EFI environment, but only because of use of the video downscaler. By employing the video downscaler 106 (or other video resolution modifier module) according to some embodiments, the application that outputs video data in a pre-OS environment does not have to be changed, which helps to reduce development costs and avoids delays in product release.

- 7 -

[0029] Instructions of software described above (including the BIOS/EFI code 104, boot services code 108, video downscaler 106, video BIOS code 112, applications 102, and OS 124 of Fig. 1) are loaded for execution on a processor (such as processor 116 in Fig. 1). The processor includes microprocessors, microcontrollers, processor modules or subsystems (including one or more microprocessors or microcontrollers), or other control or computing devices. As used here, a “processor” refers to a single component or to plural components (*e.g.*, one or plural CPUs).

[0030] Data and instructions (of the software) are stored in respective storage devices, which are implemented as one or more computer-readable or computer-usable storage media. The storage media include different forms of memory including semiconductor memory devices such as dynamic or static random access memories (DRAMs or SRAMs), erasable and programmable read-only memories (EPROMs), electrically erasable and programmable read-only memories (EEPROMs) and flash memories; magnetic disks such as fixed, floppy and removable disks; other magnetic media including tape; and optical media such as compact disks (CDs) or digital video disks (DVDs). Note that the instructions of the software discussed above can be provided on one computer-readable or computer-usable storage medium, or alternatively, can be provided on multiple computer-readable or computer-usable storage media distributed in a large system having possibly plural nodes. Such computer-readable or computer-usable storage medium or media is (are) considered to be part of an article (or article of manufacture). An article or article of manufacture can refer to any manufactured single component or multiple components.

[0031] In the foregoing description, numerous details are set forth to provide an understanding of the present invention. However, it will be understood by those skilled in the art that the present invention may be practiced without these details. While the invention has been disclosed with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover such modifications and variations as fall within the true spirit and scope of the invention.

- 8 -

What is claimed is:

- 1 1. A method comprising:
2 executing code (104) providing boot services to establish a pre-operating system (OS)
3 environment in a computer (100);
4 running an application (102) in the pre-OS environment, wherein the application
5 outputs video data for display at a display device (122) of the computer; and
6 modifying, by a video resolution modifier module (106) executing in the computer in
7 the pre-OS environment, a video resolution of the video data from the application to be
8 consistent with a video resolution supported by the display device.
- 1 2. The method of claim 1, wherein modifying the video resolution by the video
2 resolution modifier module comprises modifying the video resolution by the video resolution
3 modifier module that is part of the code.
- 1 3. The method of claim 1, wherein executing the code comprises executing BIOS (Basic
2 Input Output System) code.
- 1 4. The method of claim 1, wherein executing the code comprises executing EFI
2 (Extensible Firmware Interface) code.
- 1 5. The method of claim 1, further comprising sending the video data having the modified
2 resolution to video firmware to allow for display of the video data at the display device.
- 1 6. The method of claim 5, further comprising determining the video resolution supported
2 b the display device by determining the video resolution supported by the video firmware.
- 1 7. The method of claim 6, wherein the video resolution supported by the video firmware
2 is lower than the video resolution of the video data from the application.
- 1 8. The method of claim 1, further comprising the code loading the operating system after
2 the running and modifying tasks.

- 9 -

1 9. The method of claim 1, further comprising:
2 determining whether the video resolution of the video data from the application is
3 supported by the display device,
4 wherein performing the modifying is in response to determining that the video
5 resolution of the video data from the application is not supported by the display device.

1 10. The method of claim 9, further comprising:
2 sending the video data from the application to video firmware without changing the
3 video resolution of the video data in response to determining that the video resolution of the
4 video data is supported by the display device.

1 11. A computer (100) comprising:
2 a processor (116);
3 code (104) executable on the processor to establish a pre-operating system
4 environment in the computer; and
5 a video resolution modifier module (106) executable on the processor, in the pre-OS
6 environment, to:
7 determine if a video resolution of application video data is supported by a
8 display device (122);
9 in response to determining that the resolution of the application video data is
10 supported by the display device, cause display of the application video data at the display
11 device without modifying the video resolution of the application video data; and
12 in response to determining that the video resolution of the application video
13 data is not supported by the display device, modifying the video resolution of the application
14 video data and causing the modified application video data to be displayed at the display
15 device.

1 12. The computer of claim 11, further comprising providing, from an application
2 programming interface, information regarding a resolution that is supported due to
3 performance of the video resolution modifying.

- 10 -

1 13. The computer of claim 11, wherein determining if the video resolution of the
2 application video data is supported by the display device is based on accessing video
3 firmware that enables display of video data at the display device.

1 14. The computer of claim 11, wherein the video resolution of the application video data
2 is modified by downscaling the video resolution of the application video data.

1 15. An article comprising at least one computer-readable storage medium containing
2 instructions that upon execution by a computer (100) cause the computer to:
3 run code (104) to establish a pre-operating system (OS) environment;
4 generate, in the pre-OS environment, video data for display at a display device (122),
5 wherein the video data has a first video resolution; and
6 modify a video resolution of the video data to provide a second video resolution,
7 wherein the video data having the second video resolution is output for display at a display
8 device.

1 16. The article of claim 15, wherein the instructions upon execution cause the computer to
2 further:
3 determine whether the first video resolution is supported by the display device,
4 wherein the modifying is performed in response to determining that the first video resolution
5 is not supported by the display device; and
6 output the video data with the first video resolution for display at the display device in
7 response to determining that the first video resolution is supported by the display device.

1/2

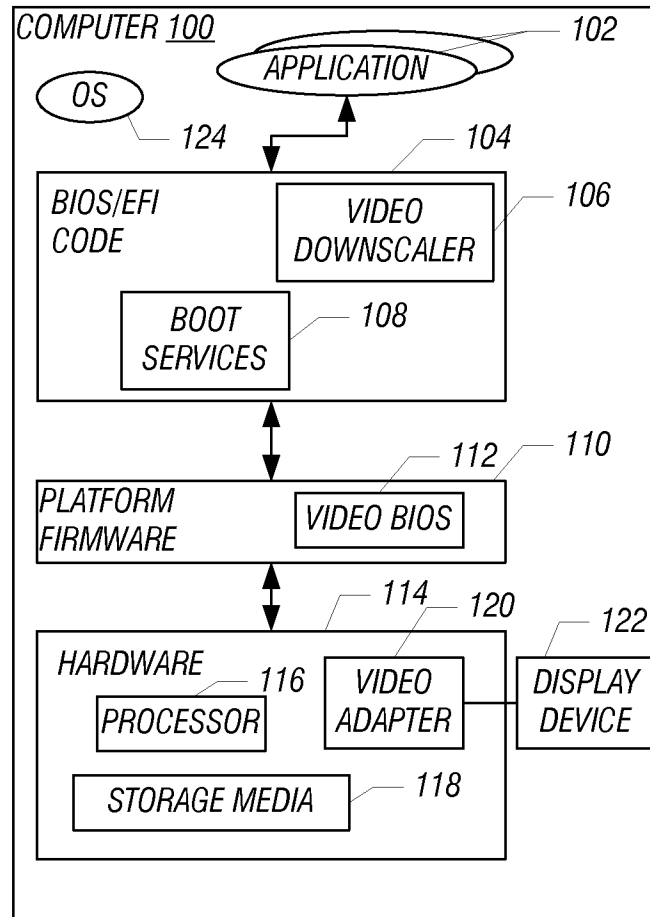


FIG. 1

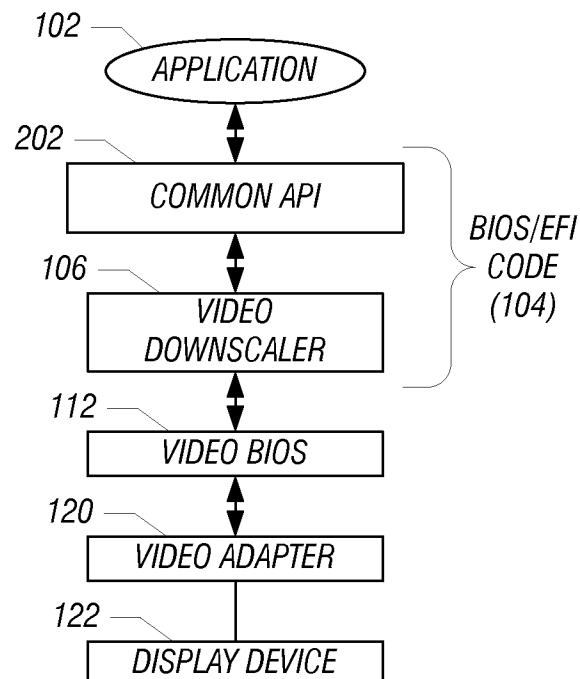


FIG. 2

2/2

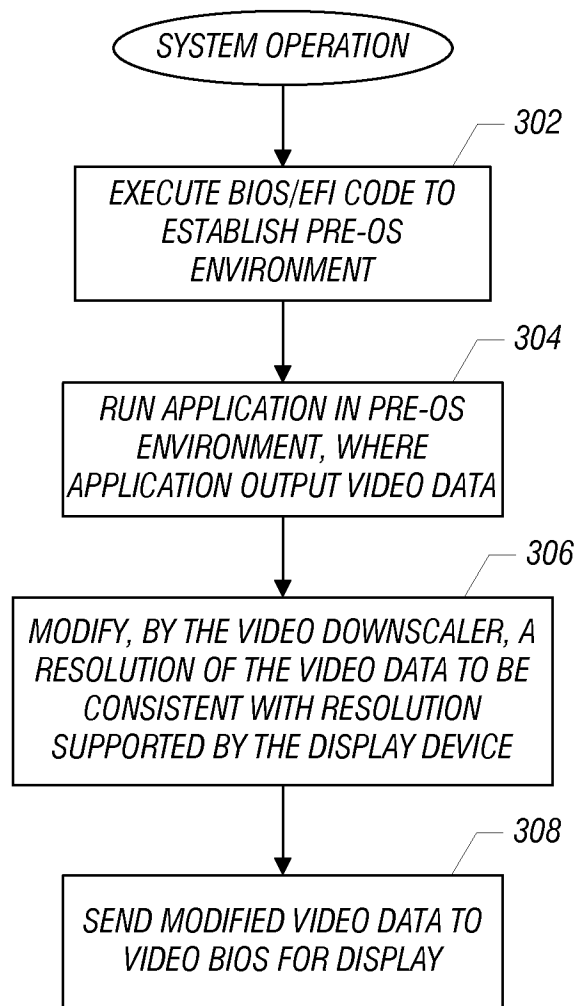


FIG. 3

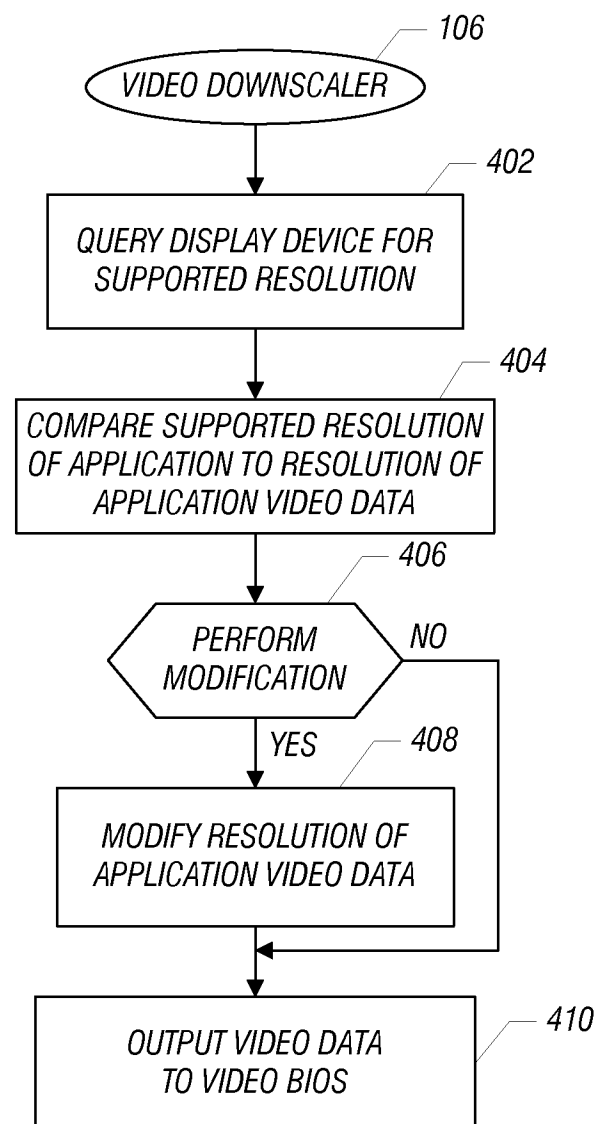


FIG. 4

A. CLASSIFICATION OF SUBJECT MATTER**G06F 9/24(2006.01)i, G06F 9/44(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)

G06F 9/24; G06F 11/00; G06F 3/00; G06T 1/00; G09G 5/00; G09G 5/02

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

(Chinese Patents and application for patent)

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

eKOMPASS(KIPO internal) & Keywords: video data, resolution, pre-boot

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	KR 10-2000-0026284 A (AHN, H. C. et al.) 15 May 2000 See abstract, figs.3-4, p.3	1-3,5-16
A	KR 10-1999-0070717 A (AHN, H. G. et al.) 15 September 1999 See abstract, pp.3-4, fig.4	1-16
A	US 2006-0092187 A1 (WANG, J. F.) 04 May 2006 See abstract, figs.1-4	1-16
A	US 5926166 A1 (KHEDERZADEH, M. et al.) 20 July 1999 See abstract, fig.2, col.2 line 62 - col.3 line 54	1-16
A	US 2008-0010516 A1 (LU, Y. C. et al.) 10 January 2008 See abstract, figs.1-3, paras.[0019]-[0021]	1-16



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

28 JANUARY 2010 (28.01.2010)

Date of mailing of the international search report

29 JANUARY 2010 (29.01.2010)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seo-
gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

Park Ji Eun

Telephone No. 82-42-481-5696



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2009/050483

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
KR 10-2000-0026284 A	15.05.2000	None	
KR 10-1999-0070717 A	15.09.1999	None	
US 2006-0092187 A1	04.05.2006	US 7372457 B2	13.05.2008
US 5926166 A1	20.07.1999	None	
US 2008-0010516 A1	10.01.2008	None	