# US-5021974-A Method for updating a display bitmap with a character string or the like

(1) TECHNICAL FIELD  
(2) This invention relates generally to a computer system for displaying information on a computer output device (e.g., display or printer) and more specifically a method and apparatus for updating a display bitmap.  
(3) BACKGROUND ART  
(4) The display of information on a computer output device typically occurs in either text or graphics modes. In text mode, information is displayed in a highly structured manner. The output device is logically divided into character cells. The output device displays one character in each character cell. The character cells are organized into a matrix of rows and columns. The output devices typically have a standard character set, which defines the text mode characters. The standard character set contains alphabetic characters, numeric characters, and special characters (e.g., $). Because of this structured approach, information can be displayed quickly. However, typically only the standard character set can be displayed.  
(5) In graphics mode, the output device is treated as a display bitmap (that is, a matrix of rows and columns of pixels). The output device turns on or off each pixel based on the information in the display bitmap. A computer program displays information by updating the display bitmap of the output device. The display bitmap is a storage array of 1's and 0's corresponding to the pixels to be turned on or off. To effect the display of information, the output device's processor reads the display bitmap and turns the corresponding pixels on or off on its display.  
(6) A common usage of computers, especially personal computers, has been word processing. Originally, the word processing programs displayed information in text mode using the standard character set. Competitive pressures have lead the developers of word processing software to provide increasingly sophisticated capabilities. Specifically, the word processing programs allow the documents to be printed in a wide variety of character fonts. Also, desktop publishing type features have been incorporated, such as varying character size, emboldening, and italics. The first word processing software with a multiple font capability would display the document on the screen in single font format. Only when the document was printed would the varied fonts be seen. Thus the user was unable to see what the document looked like until actually printed.  
(7) The developers then provided the capabilities to display the text on the screen in the same format as it would be printed. This feature has been referred to as What-You-See-Is-What-You-Get (WYSIWYG), that is, what the user sees on the screen is what gets printed. In order to implement WYSIWYG, the graphics mode of the output device is used.  
(8) Another common use of computers is for spreadsheet applications. A spreadsheet contains a grid of rectangular cells. Each cell can contain a character string. The spreadsheet program writes a character string to a cell by first blanking (opaquing) the cell and then writing the text to the blanked (opaqued) cell. This blanking and writing often results in a flashing effect. The user of the spreadsheet can perceive that a cell is first blanked then written to. The text is also clipped to the size of the cell. This clipping ensures that the text will not be written outside the cell.  
(9) Another common use of computers is to provide windowing environment for application programs. This windowing environment is common especially with personal computers. In multitasking, several computer programs can be executing concurrently. The programs may be sharing the same output device. The multitasking programs share the output device by writing to a window (a rectangular region) on the device. A multitasking executive ensures that an application does not write outside its assigned window. The executive enforces this policy by defining a clipping rectangle for each application program.  
(10) The conventional methods for displaying a character string within a rectangular region accomplishes the opaquing, clipping, and displaying functions by multiple updates to the display bitmap. The methods include the specifying of a clip region, then the opaquing of a rectangle by updating each byte of the rectangle within the display bitmap, and then displaying the character string by updating a portion of the rectangle within the display bitmap. The prior art methods are inefficient in their use of the computer resources, which leads to slow updating of display bitmaps and generally poor system performance.  
(11) DISCLOSURE OF THE INVENTION  
(12) It is an object of the present invention to provide a method and system for updating a display bitmap with a character string.  
(13) It is another object of the present invention to provide such a method and system that will minimize the number of writes to the display bitmap to effect the update.  
(14) It is another object of the present invention to provide such a system and method that can efficiently update a display bitmap when a clipping rectangle is specified.  
(15) It is another object of the present invention to provide such a system and method that can efficiently update a display bitmap when an opaquing rectangle is specified.  
(16) These and other objects, which will become apparent as the invention is more fully described below, are obtained by an improved method and system for updating a display bitmap. In a preferred embodiment, a character bitmap is comprised of a contiguous set of virtual characters and the width of each character. A visible stack is created for each character string to be displayed, with offset, width and phase for each virtual character in the string. The data in the visible stack is then processed to efficiently update the display bitmap.