

METHOD AND SYSTEM FOR IMAGE FORMAT CONVERSION

BACKGROUND OF THE INVENTION

TECHNICAL FIELD

5 The invention relates to image processing. More particularly, the invention relates to a method and system for converting image data, for example, between a bitplane (planar) format and a CMYK (chunky) format.

DESCRIPTION OF THE PRIOR ART

10 Computer images are arranged as a grid of pixels in which each pixel is stored in memory as a numerical representation of the pixel's attributes, such as color and intensity. Several different color models are currently being used to define such colors mathematically.

15 The RGB model, representing the three primary colors red, green, and blue, is an additive model. RGB is used in most computer monitors, scanners, televisions, graphics cards, and lighting effects. The CMYK model is a subtractive system, based on the absorption of ink and paint into paper. Most printers use the CMYK model.

20 Because most computers and scanners use RGB, and most printers use CMYK, a conversion process, also referred to as rendering, must be used to print an image in CMYK that is displayed on a computer screen in RGB. This conversion may be performed in the user's computer, in the scanner, or in the printer. An example of a software application for RGB to CMYK conversion in a computer is Photoshop, by Adobe Systems. After the image is converted, it is then saved as a CMYK version and may be output to the printer as desired. Rendered images may be stored in different
25 formats in a computer's memory. Two such formats are bitplane (planar) and CMYK (chunky) formats.

30 Fig. 1 is an illustration of a portion of an image 10 that is arranged in a chunky data format. In chunky data format, the data for each pixel are stored as "chunks." Each word 12, 14, 16, 18 corresponds to one pixel in the image and includes one byte for each of four color planes. For example, the first word 12 includes a cyan byte C_0 , a magenta byte M_0 , a yellow byte Y_0 , and a black byte K_0 . In the Fig. 1, the bytes are

incremented from left to right in a word, and from top to bottom to provide a group of words.

Chunky format is very efficient when rendering an image because a processor only needs to perform a single word write operation to store the pixel in memory once it has generated the pixel.

Image files generally contain large quantities of data. Overall system performance can be significantly affected when transferring such large image files. It is therefore of critical importance to conserve bandwidth in the image transfer process. Chunky data are readily transmitted quickly and efficiently over a system bus, for example, a PCI bus. Thus, once the image has been rendered in a frame buffer, it can be rapidly transmitted to a printer because the data can be sent as discrete chunks and reassembled at the printer to produce the image. However, chunky data are not processed as efficiently by the printer as are planar data.

A typical color laser printer has one drum upon which an image is formed during an electrostatic dye transfer process. Such printer prints CMYK color planes in four passes and therefore can only accept one color separation at a time. Because each pixel typically includes data for all four color planes when in chunky format, the printer must access the storage memory multiple times to print the image. For example, when a pixel is transmitted to the printer, the printer must discard all data in the word that comprises the pixel except for data relating to the specific color plane that is currently being printed. Thus, up to three-quarters of all data is discarded for each pixel that is printed, and each word must be transferred to the printer four separate times to print a full-color image.

For example, an A4 page having a resolution of 400 dots per inch (dpi) can be stored in approximately 60 megabytes (MB) of memory. However, if the data are stored in chunky format, the printer must access memory four times per pixel, i.e. one for each of the CMYK color planes. The printer first accesses the data to print one color plane, and discards all of the data for each such access that are not concerned with the color plane currently being printed. Thus, 240 MB (60 MB x 4) of memory is being transferred to print the same A4 page when the data are arranged in a chunky format. This larger memory transfer can slow system performance and printing.

Fig. 2 is an illustration of a portion of an image 10 that is arranged in a planar data format. In the planar format, the image is represented by several overlaying color planes 20, 22, 24, 26 of data. Thus, in a CMYK image, each pixel is separated by color. For example, all cyan information $C_0, C_1, C_2 - C_n$ is combined into a cyan pixel

plane. Magenta information $M_0 - M_n$, yellow information $Y_0 - Y_n$, and black information $B_0 - B_n$ is combined into separate pixel planes.

5 In the planar format, pixel data for each color are stored in a separate section of the memory, or in a separate buffer. A four-pass printer separately prints the contents of each buffer to reproduce the color image. The planar format is therefore very efficient when used in printing. In contrast to the previous example concerning chunky format, the four memory buffers are accessed one time each in planar format, without discarding color data. Thus, only 60 MB of memory is required to transfer and print the same A4 image in a planar format.

10 However, planar format is not used to advantage in the rendering process, because the processor must perform four write operations for each pixel rendered to store the pixel in memory. This is due to the fact that the processor must access all four memory buffers to obtain the data necessary to complete all four color planes, *e.g.* in a CMYK image. The time required to render an image where the data are arranged in a planar format can be approximately 30 percent greater than the time required to render the image where the data are arranged in a chunky format. Therefore, a planar format is not as efficient for rendering an image as a chunky format.

15 It is known in the prior art to use software to convert between a chunky format and a planar format. However, such software conversion techniques are both memory and time consuming. For example, data stored in planar format at a printer are not transferred as efficiently as data stored in chunky format data.

20 There has therefore been a need for an improved method for converting between chunky and planar formats. It would therefore be an advantage to provide a method and system for converting between a planar format and a chunky data format. It would be yet another advantage if the method and system were able to convert between these formats bidirectionally.

SUMMARY OF THE INVENTION

30 The invention provides a method and system for converting image data between a planar format and a chunky format. In the preferred embodiment of the invention, an image is initially rendered from RGB data to CMYK data and stored in memory in a chunky format. A barrel shifter is used to sort each byte of the image by color plane. The barrel shifter separates the bytes by color plane and outputs the sorted bytes in a planar format. In the preferred embodiment of the invention, the barrel shifter may be constructed as part of an application specific integrated circuit (ASIC) that interfaces a processor to the system memory.

The invention preferably includes a plurality of memory cells that store the color-sorted bytes. The memory cells are partially interconnected, although two addresses of each memory cell are not connected in common with any of the other memory cells. The non-interconnected addresses function as a set of selection lines, whose binary values determine the number of positions that the data stored in the memory are to be shifted.

The invention includes a circuit that obtains an address for the first pixel of the chunky data in an image. The processor sends this first word via a processor interface to the circuit. If no offset is needed, the barrel shifter does not shift the word which represents the first pixel. The pixel is therefore separated into four bytes and each byte is stored by color plane in a corresponding one of the memory cells.

The second pixel of the image has an address with an offset of +1 word (+4 bytes). The barrel shifter shifts each of the bytes within the word by one position, thereafter transferring each of the bytes into respective memory cells. The barrel shifter shifts each of the bytes in the third word by two positions, and each of the bytes in the fourth word by three positions. The four bytes of each word are thereby separated into each of the respective CMYK color planes and stored in corresponding memory cells by color planes as appropriate. The process then repeats for the next pixels in the image.

Thus, the first memory cell contains, *e.g.* a cyan byte from each word, the second memory cell contains a magenta byte from each word, the third memory cell contains a yellow byte from each word, and the fourth memory cell contains a black byte from each word. Thus, the chunky format data have been converted to planar data as stored in the respective memory cells. This planar data may then be output to a printer, *e.g.* for four pass printing. The invention therefore provides a technique that uses to advantage the more efficient chunky format while rendering an image, and the more efficient planar format during printing.

In an alternative embodiment of the invention, the invention is bidirectionally operative. Thus, bytes that have been sorted by color plane and that are therefore arranged in a planar format may be input to a barrel shifter to sort the bytes, and thereby produce image data in a chunky format.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is an illustration of a portion of an image arranged in a chunky data format;

5 Fig. 2 is an illustration of a portion of an image arranged in a planar data format;

Fig. 3 is a schematic block diagram of the system for image format conversion according to the invention;

Fig. 4 is a flow chart illustrating a method for performing image format conversion according to the invention;

10 Fig. 5 is a block schematic diagram illustrating memory cell addressing according to the invention;

Fig. 6 is a diagram that shows color plane byte shifting for four pixels according to the invention;

15 Fig. 7 is a block schematic diagram showing operation of a barrel shifter for a cyan color separation according to the invention;

Fig. 8 is a block schematic diagram showing operation of a barrel shifter for a cyan color separation according to the invention;

Fig. 9 is a block schematic diagram showing operation of a barrel shifter for a magenta color separation according to the invention; and

20 Fig. 10 is a block schematic diagram showing operation of a barrel shifter for a yellow color separation according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

25 The invention provides a method and system for converting image data between a planar format and a chunky format. In the preferred embodiment of the invention, an image is initially rendered from RGB data (*e.g.* a scanned RGB image) to CMYK data and stored in computer memory in a chunky format.

30 Fig. 3 is a block diagram of the system 30 for image format conversion, according to the invention. A processor 32 is connected to a barrel shifter 34 via a processor interface 36. A barrel shifter is a circuit that cyclically shifts input data by a number of positions dictated by the word address, *i.e.* the binary value of a set of selection lines.

35 In the preferred embodiment of the invention, the barrel shifter is constructed as part of an application specific integrated circuit (ASIC) 38 that interfaces the processor to the memory. The ASIC may also include memory interface 40 and a PCI bus interface 42. It should be appreciated that, while the preferred embodiment of the

invention is discussed in connection with an ASIC, the actual hardware implementation of the invention is considered to be a matter of choice to one skilled in the art.

The barrel shifter is connected to four memory storage cells 50, 52, 54, 46, via a memory interface 40. In the preferred embodiment of the invention, the memory cells are dynamic random access memory (DRAMs), such as single in-line memory modules (SIMMS) or dual in-line memory modules (DIMMS).

The preferred embodiment is adapted for use with a four-byte barrel shifter having four associated memory storage cells, and therefore four lines joining the elements in the data bus. However, alternative embodiments of the invention are adapted for use with any appropriate number of memory cells and lines. Thus, the following discussion is provided for purposes of example and not as a limitation on the scope of the invention.

Fig. 4 is a flow chart illustrating one cycle of a method for image format conversion according to the invention. The circuit in which the invention is implemented obtains the address for the first word (first pixel) of the chunky data (200). In the example, this word address is offset 0. The processor sends this first word via the processor interface to the circuit in which the invention is implemented (205). The barrel shifter in the circuit does not shift the word (210). The word $C_0M_0Y_0K_0$ is therefore separated into four bytes. Referring to Fig. 3, byte C_0 is transferred to memory cell 50, byte M_0 to memory cell 52, byte Y_0 to memory cell 54, and byte K_0 to memory cell 56.

The circuit then obtains the address for the second word (second pixel) of the chunky data (215). In the example, this word address is offset +1 Word (+4 bytes). The processor sends this second word via the processor interface to the circuit (220). The barrel shifter shifts each of the bytes in the word one position (225). The word $K_1C_1M_1Y_1$ is therefore separated into four bytes, with byte C_1 transferred to memory cell 50, byte M_1 to memory cell 52, byte Y_1 to memory cell 54, and byte K_1 to memory cell 56.

The circuit next obtains the address for the third word (third pixel) of the chunky data (230), offset +2 Words (+8 bytes). The processor sends this third word via the processor interface to the circuit (235). The barrel shifter shifts each byte in the word two positions (240). The word $Y_2K_2C_2M_2$ is separated into four bytes, with byte C_2 transferred to memory cell 50, byte M_2 to memory cell 52, byte Y_2 to memory cell 54, and byte K_2 to memory cell 56.

The circuit then obtains the address for the fourth word (fourth pixel) of the chunky data (245), offset +3 Words (+12 bytes). The processor sends this fourth word

via the processor interface to the circuit (250). The barrel shifter shifts each byte in the word three positions (255). The word $C_3M_3Y_3K_3$ is separated into four bytes, with byte C_3 transferred to memory cell 50, byte M_3 to memory cell 52, byte Y_3 to memory cell 54, and byte K_3 to memory cell 56.

5 The next word address returns to offset 0 (260) and the process is repeated until all of the chunky data have been stored as planar data in the memory cells.

Fig. 5 is a diagram illustrating memory cell addressing according to the invention. In the preferred embodiment of the invention, the lowest two address lines, A_0 and A_1 , of each memory cell 50, 52, 54, 56 are not interconnected with the address lines of any other memory cell. These "non-common" address lines function as a set of selection lines 60, 62, 64, 66, 68, 70, 72, 74 whose binary values determine the number of positions that the data to be stored in the memory cell are shifted. In this way, each byte in each color plane is stored in its proper position in relation to each other byte in the respective color plane. The remaining addresses 76, A_2 through A_{10} in the example, are interconnected. These addresses are used to store the bytes of each word as they are output from the barrel shifter.

For example, the barrel shifter does not shift the input data if the binary values (decimal equivalents are shown in parenthesis) stored in A_0 and A_1 of memory cells 50, 52, 54, and 56 are "00" (zero), "01" (one), "10" (two), and "11" (three), respectively. The barrel shifter rotates the input data once if the stored binary values are "11" (three), "00" (zero), "01" (one), and "10" (two), respectively.

Table 1 is a function table for the four-byte barrel shifter of the preferred embodiment of the invention.

TABLE 1

Memory Cell:	50	52	54	56
No shift	"00" (0)	"01" (1)	"10" (2)	"11" (3)
Rotate once	"11" (3)	"00" (0)	"01" (1)	"10" (2)
Rotate twice	"10" (2)	"11" (3)	"11" (0)	"01" (1)
Rotate three times	"01" (1)	"10" (2)	"11" (3)	"00" (0)

The barrel shifter thereby sorts the input chunky format data by color plane to output planar format data. In the preferred embodiment of the invention, these data are output via the PCI interface 42 to a printer (not shown) via four lines 76, 80, 82, 84 of the PCI data bus.

Fig. 6 is a diagram that shows color plane byte shifting for four pixels according to the invention. In Fig. 6, a portion of an image 90 is stored in chunky format. Each word 92, 94, 96, 98 contains bytes for each of the four color planes. In the example shown, the bytes for each word are identified by a subscript. Thus, the magenta byte of the first word 92 is denoted M_0 and the magenta byte for the third word 16 is M_2 .

In the preferred embodiment of the invention, the cyan bytes C_0, C_1, C_2, C_3 are separated when the input data are not shifted. The magenta bytes M_0, M_1, M_2, M_3 are separated next, followed by the yellow and black bytes. The separated bytes are then output from the barrel shifter. The barrel shifter thus receives chunky data as an input, and outputs the data in planar format.

Fig. 7 is a block schematic diagram showing operation of a barrel shifter for a cyan color separation according to the invention. The following discussion is best understood if Fig. 6 is also consulted. In this example, the barrel shifter 34 does not shift the input data in accordance with the binary values on the memory selection lines. These values are shown on Fig. 7 as "00" at memory cell 50, "10" at memory cell 52, "01" at memory cell 54, and "11" at memory cell 56. Thus, byte C_0 from the first word, is transmitted on line 118 to the barrel shifter. The byte is output from the barrel shifter on line 100 to memory cell 50. Byte C_1 from the second word, is transmitted on line 120 to the barrel shifter. The byte is output from the barrel shifter on line 102 to memory cell 52. Similarly, byte C_2 , from the third word, is transmitted on line 122 to the barrel shifter. The byte is output from the barrel shifter on line 104 to memory cell 54. Byte C_3 , from the fourth word, is transmitted on line 124 to the barrel shifter. The byte is output from the barrel shifter on line 104 to memory cell 56. In this way, the

cyan color plane is assembled from the chunky image data for output to the printer (e.g. via the PCI interface) in a planar format.

Fig. 8 is a block schematic diagram showing operation of the barrel shifter for a magenta color separation according to the invention. In Fig. 8, input data are shifted one position by the barrel shifter, as shown by the selection line values at memory cells 50 "11", 52 ("00", 54 ("10"), and 56 ("01"). Thus, byte M_0 from the first word is transmitted on line 118 to the barrel shifter, and output from the barrel shifter on line 102 to memory cell 52. Byte M_1 from the second word is transmitted on line 120, and output from the barrel shifter on line 104 for storage in memory cell 54. Byte M_2 from the third word is transmitted on line 122, and output from the barrel shifter on line 106 for storage in memory cell 56. Byte M_3 from the fourth word is transmitted on line 124 to the barrel shifter, and output from on line 100 to memory cell 50.

Fig. 9 is a block schematic diagram showing operation of the barrel shifter for a yellow color separation according to the invention. In Fig. 9, input data are shifted two positions by the barrel shifter, as shown by the selection line values at memory cells 50 ("01"), 52 ("11"), 54 ("00"), and 50 ("10"). Thus, byte Y_0 from the first word is transmitted on line 118 to the barrel shifter, and output from the barrel shifter on line 104 to memory cell 54. Byte Y_1 from the second word is transmitted on line 120, and output from the barrel shifter on line 106 for storage in memory cell 56. Byte Y_2 from the third word is transmitted on line 122, and output from the barrel shifter on line 100 for storage in memory cell 50. Byte Y_3 from the fourth word is transmitted on line 124 to the barrel shifter, and output from on line 102 to memory cell 52.

Fig. 10 is a block schematic diagram showing operation of the barrel shifter for a black color separation according to the invention. In Fig. 10, input data are shifted three positions by the barrel shifter, as shown by the selection line values at memory cells 50 ("10"), 52 ("01"), 54 ("11"), and 56 ("00"). Thus, byte K_0 from the first word is transmitted on line 118 to the barrel shifter, and output from the barrel shifter on line 106 to memory cell 56. Byte K_1 from the second word is transmitted on line 120, and output from the barrel shifter on line 100 for storage in memory cell 50. Byte K_2 from the third word is transmitted on line 122, and output from the barrel shifter on line 102 for storage in memory cell 52. Byte K_3 from the fourth word is transmitted on line 124 to the barrel shifter, and output from on line 104 to memory cell 54.

It is often desirable to re-generate the chunky data structure from the color-sorted planar data. For example, the processor may need to override certain data. An alternative embodiment of the invention is provided that is bidirectionally operative and

transparent to the processor. Thus, color-sorted bytes in planar format may be input from the print buffer to the barrel shifter. The barrel shifter is then used to sort the planar bytes to output the image data in a chunky format. This is readily accomplished, for example, by reconfiguring the memory cell select lines accordingly.

- 5 Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention. Thus, the invention may be used with any color conversion scheme that generates a CMYK image. For example, the invention may be used when the image
- 10 is rendered from a HSV (hue, saturation, value) color cone image or an RGB image to a CMYK image. Accordingly, the invention should only be limited by the Claims included below.

CLAIMS

1. A method for converting image data between a planar format and a chunky format, comprising the steps of:

5 sorting chunky format image data by color plane with a barrel shifter; and
 separately outputting the sorted bytes of each color plane from said barrel shifter to provide planar format image data.

2. The method of Claim 1, further comprising the steps of:

10 rendering said image in chunky data format in a processor; and
 transmitting said chunky data over said processor's bus to said barrel shifter.

3. The method of Claim 1, further comprising the step of providing a plurality of memory cells having at least one interconnected address for receiving said sorted bytes.

15 4. The method of Claim 3, wherein each of said memory cells includes at least two addresses that are not interconnected to the addresses of other of said memory cells.

20 5. The method of Claim 4, wherein said barrel shifter sorts said chunky format image data in accordance with binary values provided at said at least two non-interconnected addresses of said memory cells.

25 6. The method of Claim 5, wherein said barrel shifter outputs cyan bytes when said barrel shifter does not shift byte position in response to said binary values.

7. The method of Claim 5, wherein said barrel shifter outputs magenta bytes when said barrel shifter shifts bytes one position in response to said binary values.

30 8. The method of Claim 5, wherein said barrel shifter outputs yellow bytes when said barrel shifter shifts bytes two positions in response to said binary values.

9. The method of Claim 5, wherein said barrel shifter outputs black bytes when said barrel shifter shifts bytes three positions in response to said binary values.



10. The method of Claim 1, further comprising the steps of:
inputting in planar format to said barrel shifter the sorted bytes of each color; and
using the barrel shifter to sort said bytes to output said image in chunky format.

5 11. A method for converting image data between a planar format and a chunky
format, comprising the steps of:
transmitting said chunky data to a barrel shifter;
using said barrel shifter to sort the bytes of said chunky format data by color
plane;

10 separately outputting the sorted bytes of each color plane from said barrel shifter;
providing a plurality of memory cells each having at least first and second
addresses that are not interconnected to implement byte shifting in accordance with said
barrel shifter, and at least a third common address to receive said output bytes in planar
format; and

15 separately outputting the sorted bytes of each color plane from said barrel shifter
to store said image in planar format.

20 12. The method of Claim 11, wherein the number of positions shifted by said barrel
shifter is determined by binary values stored in said memory addresses of said memory
cells that are not interconnected.

13. The method of Claim 12, wherein said barrel shifter outputs cyan bytes when
said barrel shifter does not shift byte position in response to said binary values.

25 14. The method of Claim 12, wherein said barrel shifter outputs magenta bytes
when said barrel shifter shifts one byte position in response to said binary values.

15. The method of Claim 12, wherein said barrel shifter outputs yellow bytes when
said barrel shifter shifts two byte positions in response to said binary values.

30 16. The method of Claim 12, wherein said barrel shifter outputs black bytes when
said barrel shifter shifts three byte positions in response to said binary values.



04/4

17. The method of Claim 11, further comprising the steps of:
inputting in planar format to said barrel shifter the sorted bytes of each color plane; and
using the barrel shifter to sort the bytes to output said image in chunky format.

5

18. A system for converting image data between a planar format and a chunky format, comprising:

a barrel shifter for sorting by color planar the bytes of said image in chunky format and for outputting the sorted bytes in planar format; and

10

a plurality of memory cells each having at least first and second addresses that are not interconnected to implement byte shifting in accordance with said barrel shifter and at least a third common address to receive said output bytes in planar format.

19. The system of Claim 18, wherein said barrel shifter outputs cyan bytes when said barrel shifter does not shift byte position in response to said binary values;

15

wherein said barrel shifter outputs magenta bytes when said barrel shifter shifts one byte position in response to said binary values;

wherein said barrel shifter outputs yellow bytes when said barrel shifter shifts two byte positions in response to said binary values; and

20

wherein said barrel shifter outputs black bytes when said barrel shifter shifts three byte positions in response to said binary values.

20. The system of Claim 18, wherein said barrel shifter is bidirectionally operable to receive said sorted bytes in planar format and to sort said bytes to output said image in chunky format.

25

