

COLOR TABLE COMPRESSION

Background

[0001] Color management systems deliver a controlled conversion between color representations of various devices, such as image scanner, digital camera, computer monitors, printers, and corresponding media. Device profiles provide color management systems with information to convert color data between color spaces such as between native device color spaces and device-independent color spaces, between device-independent color spaces and native device color spaces, and between source device color spaces and directly to target device color spaces.

Brief Description of the Drawings

[0002] Figure 1 is a block diagram illustrating an example memory device having a compressed color table.

[0003] Figure 2 is a block diagram illustrating an example system implementing the compressed color table on the memory device of Figure 1.

[0004] Figure 3 is a block diagram illustrating an example method of generating the compressed color table of Figure 1.

[0005] Figure 4 is a block diagram illustrating an example method of compressing the color table incorporating features of the example method of Figure 3.

[0006] Figure 5 is a block diagram illustrating an example method having additional features of the example method of Figure 3.

[0007] Figure 6 is a block diagram illustrating an example memory device including a compressed color table of the memory device of Figure 1 operably coupled to an example printing device.

[0008] Figure 7 is a block diagram illustrating an example method of decoding the memory device of Figure 6.

Detailed Description

[0009] In the following detailed description, reference is made to the accompanying drawings, which form a part hereof, and in which are shown by way of illustration as specific examples in which the disclosure may be practiced. It is to be understood that other examples may be utilized and structural or logical changes may be made without departing from the scope of the present disclosure. The following detailed description, therefore, is not to be taken in a limiting sense, and the scope of the present disclosure is defined by the appended claims. It is to be understood that features of the various examples described herein may be combined, in part or whole, with each other, unless specifically noted otherwise.

[0010] A color space is a system having axes and that describes color numerically. Some output devices, such as printing devices, may employ a type of cyan-magenta-yellow-key (black) (CMYK) color space, while some software applications and display devices may employ a type of red-green-blue (RGB) color space. For example, a color represented in the CMYK color space has a cyan value, a magenta value, a yellow value, and a key value that combined numerically represent the color.

[0011] A color profile is a set of data that characterizes a color space. In one example, a color profile can describe the color attributes of a particular device or viewing specifications with a mapping between the device-dependent color space, such as a source or target color space, and a device-independent color space, such as profile connection space (PCS), and vice versa. The mappings may be specified using tables such as look up tables, to which interpolation is applied, or through a series of parameters for transformations. Devices and software programs – including printing devices, monitors, televisions, operating systems, browsers, and other device and software – that capture or display color can include profiles that comprise various combinations of hardware and programming. An ICC profile is an example color profile that is a set of data that characterizes a color space according to standards promulgated by the

International Color Consortium (ICC). Examples of this disclosure using ICC profiles, however, are for illustration only, and the description is applicable to other types of color profiles or color spaces.

[0012] The ICC profile framework has been used as a standard to communicate and interchange between various color spaces. An ICC output profile includes color table pairs, so-called A2B and B2A color look up tables, where A and B denote the device-dependent and the device-independent color spaces, respectively. For different devices, there are different look up table rendering intent pairs. For example, an ICC profile allows for three color table pairs, enumerated from 0 to 2, enabling the user to choose from one of the three possible rendering intents: perceptual, colorimetric, or saturation. ICC profiles are often embedded in color documents as various combinations of hardware and programming to achieve color fidelity between different devices, which increases the total size of these documents. The size of color tables will also increase with finer sampling of the spaces and larger bit depths.

[0013] Color tables that provide transformations between various color spaces are extensively used in color management, common examples being the transformations from device independent color spaces (such as CIELAB, i.e., $L^*a^*b^*$) to device dependent color spaces (such as RGB or CMYK) and vice versa. The mappings may be specified using tables such as one or more single dimensional or multidimensional look-up tables, to which interpolation can be applied, or through a series of parameters for transformations. A color table can include an array or other data structure on a memory device that replaces runtime computations with a simpler array indexing operation as a color look-up table. For the purposes of this disclosure, color tables can also include monochromatic and gray scale color tables.

[0014] Printing devices – including printers, copiers, fax machines, multifunction devices including additional scanning, copying, and finishing functions, all-in-one devices, or other devices such as pad printers to print images on three dimensional objects and three-dimensional printers (additive manufacturing devices) – employ color management systems to deliver a controlled conversion between color representations of various devices, such as image scanner,

digital camera, computer monitors, printers, and software applications. In one example, printing devices often employ color tables including multidimensional color look-up tables to provide transformations between different color spaces such as from input device-independent colors to CMYK ink amounts for printing on media. For devices such as color printers or other printing devices, the color tables are often embedded in memory devices storing the printer firmware or other hardware, where the color tables consume computer memory in storage devices. The trend toward finer sampling of the spaces and larger bit depths also results in an increase in table sizes, further exacerbating concerns related to costs of and available space on the computer memory. Additionally, the concerns of efficient memory use and storage space consumption are applicable for color tables that are embedded in color documents such as ICC source profiles. In applications where embedded profiles are used, the embedded profiles represent an overhead.

[0015] Figure 1 illustrates an example memory device 100 including a compressed difference color table 102 and corrective information 104. A difference color table that is to be compressed includes a plurality of difference nodes in which each difference node represents a value that is a difference of a value of a node an original color table and a value of corresponding node of a reference table. The difference color table is compressed at a selected compression amount, such as a compression ratio. The plurality of difference nodes includes a set of nodes having a color difference outside an error threshold at the selected compression ratio. The corrective information corresponds with the set of nodes of the color table.

[0016] In one example, the memory device may be for a supply component. For example, the memory device may be for a print cartridge for a printing device. In some examples, the memory device may be included on a supply component for a printing device. In some examples, a reference table, for use with reconstruction of a color table corresponding with the difference color table, may be included on a separate memory device located on the printing device. The color table can include a multidimensional color look-up tables to provide transformations between different color spaces such as from input device-

independent colors to CMYK ink amounts. As this color transform may be ink-dependent, such as dependent on the particular formulation of the ink included in a supply component of a print cartridge including ink, the compressed difference table can be stored on a memory device located on the print cartridge for use with the printing device having the reference table.

[0017] As used herein, a printing device supply component may correspond to a component from which consumable printing material may be supplied to a printing device for use thereof. Some examples of a printing device supply component may be referred to as a print cartridge, where a print cartridge may be replaceable and may be a two dimensional or three dimensional print cartridge. Examples of printing device supply components and print cartridges may comprise a print material reservoir to store a reserve of print material for use in printing operations when replaceably coupled to a printing device/system. Examples of print material, as used herein, may include consumable materials such as consumable fluids and/or consumable powders. Examples of print material include ink, toner, gloss, varnish, powders, sealants, colorants, and/or other such materials for printing. For example, a print cartridge may include fluid ink corresponding to at least one color (or two or more colors) in which a printing device may print. In other examples, a print cartridge may include toner corresponding to at least one color (or two or more colors) in which a printing device may print. In some examples, such supply components and print cartridges thereof may be referred to as “replaceable supplies.”

[0018] The example memory device 100 can be implemented to include a combination of one or more volatile or nonvolatile computer storage media. Computer storage media may be implemented as any suitable method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. A propagating signal by itself does not qualify as storage media or a memory device. The memory device can be included as part of a system including a processor and memory for storing a set of computer instruction for controlling the processor to perform a color transformation.

[0019] Figure 2 illustrates an example system 200 employing the memory device 100. System 200 includes a memory device 202, which can correspond with memory device 100, for a supply component 204 that can be used with a printing device 206. An example of a printing device supply component 204 is a consumable, or a replaceable element on the printing device 206, such as an ink cartridge, a fuser, a photoreceptor, a toner cartridge, or other element. The memory device 200 includes corrective data 208 corresponding with nodes of a reference table 210 for the printing device 206. In one example, the reference table 210 is stored on a memory device on the printing device 206 that is a separate hardware component from memory device 202. The reference table 210 can be stored with the firmware of the printing device 206. The corrective data 208 includes the compressed difference color table 102 and corrective information 104 of memory device 100 to transform the nodes of the reference table 210 to a reconstructed color table 212 to provide transformations between different color spaces such as from input device-independent colors to printing-device dependent colors. In one example, the compressed difference color table is reconstructed and combined with the reference table 210 to generate an initial reconstructed color table. The corrective information is then applied to the initial reconstructed color table to generate a reconstructed color table.

[0020] Figure 3 illustrates an example method 300 of generating the corrective data 208 on memory device 202, which can include the compressed difference table 102 and corrective information 104 of memory device 100. The method 300 allows the storing one or more color look-up tables (CLUTs) in environments where a memory device, such as flash memory on printer cartridges, is relatively expensive and limited. The method 300 supports lossy compression for relatively high compression ratios to meet storage space limitations and achieves relatively small color differences.

[0021] In one example, a color table environment such as a printing device may include a plurality of multidimensional color tables that correspond with different media and neutral axes of a color gamut included in a color profile. In general, a profile can include N color tables to be processed, such as $CLUT_1$, $CLUT_2$, . . . , $CLUT_N$, and the input color space includes J_{in} channels. In one example,

multiple color tables representing different rendering intents can be included with one ICC profile. Additionally, the output color space includes J_{out} channels, and in many examples of an ICC profile J_{in} and J_{out} can be 3 or 4 channels. For each output channel, the corresponding lookup table contains $M^{J_{in}}$ nodes. For example, each color table can include M^3 nodes for each of the C, M, Y, and K four colorants corresponding with each ink color used in the printing device. Additionally, each type of media used in the printing device can include a set of color tables.

[0022] In the example method, a reference table $CLUT_{ref}$ is generated at 302. The reference table includes nodes having a preselected, or predetermined values. In one example, the reference table is generated from averaging the N color tables of the profile. The each of the N original color tables is subtracted from the reference table to obtain a corresponding difference color table $difCLUT_i$ at 304. For example, $difCLUT_i = CLUT_i - CLUT_{ref}$, $i = 1, 2, \dots, N$. The N difference color tables are compressed, such as with a lossy compression, at 306. The compressed difference table is reconstructed and applied to the corresponding original difference color table to generate corrective information at 308. The compressed difference table and the corrective information are stored on a memory device at 310, such as on memory device 202 on supply component 204. The reference table can be stored on the printing device 206 where memory space can be more plentiful and contribute less to the overhead than on a supply component 204. In some examples, one or more of the compressed difference table, corrective information, and the reference table can be further compressed with a lossless compression.

[0023] The reference table $CLUT_{ref}$ in one example can be generated at 302 from averaging the N original color tables, or a subset of the original color tables such as related or logically grouped color tables, from the profile. For example,

$$CLUT_{ref} = \frac{\sum_{i=1}^N CLUT_i}{N}$$

[0024] Each of the original color tables includes a plurality of nodes in which each node is at a location and includes a value. In one example, each of the multidimensional original color tables used to create the reference table can include M^3 nodes. The values of each node at node location j of each of the

original color tables are added together and then divided by the number of values to provide a value at the corresponding node in the reference table. Thus, the value at node location j of each of the original color tables are added together, divided by the number of color tables N to provide the value at node location j of the reference color table, in which j is the node location from 1 to M^3 .

[0025] The difference color table generated at 304 can include a plurality of difference nodes in which each node includes a value that represents a difference of a value of a node of the original color table and a value of a node of a reference table. In one example, the original color table and reference table each include M^3 nodes. The value of each node of the original color table is subtracted from the value of the corresponding node in the reference table to provide a value at the corresponding node in the difference table. Thus, the value at node location j of the original color table is subtracted from the value at node location j of the reference table to provide the value at node location j of the difference color table, in which j is the node location from 1 to M^3 .

[0026] In one example of compressing a difference color table at 306, the nodes of the difference color table are transformed and processed to obtain quantized coefficients. The particular transformation and processing can be determined by a selected amount of compression, such as compression ratio, desired and can include a selected step size Δ . The quantized coefficients can be reordered into a one-dimensional bitstream using a multi-dimensional reordering, such as a three-dimensional zigzag ordering. The bitstream of quantized coefficients can be provided as a binary file and further compressed with a lossless compression. The resulting compressed bitstream can be stored as a file on the memory device 100 included in compressed difference color table 102 at 310.

[0027] In the example of compressing the difference table at 306, a coefficient bit assignment table is calculated from the quantized coefficients. The coefficient bit assignment table can be applied to the quantized coefficients to reconstruct the coefficients in decompression. The three-dimensional zigzag ordering can introduce a large amount of redundancy to the coefficient bit

assignment table, and the coefficient bit assignment table can be further compressed with the lossless compression. The compressed coefficient bit assignment table can be included as part of the compressed difference color table 102 at 310 or stored separately with the firmware on printing device 206.

[0028] Many color devices can employ multiple coefficient bit assignment tables. For example, a coefficient bit assignment table can be created for each colorant. For a printing device using C, M, Y, K colorants, four separate coefficient bit assignment tables can be used. Additional parameters may dictate eight coefficient bit assignment tables can be used in a printing device, each coefficient bit assignment table corresponding with a different set of quantized coefficients of a compressed original color table.

[0029] Lossless compression and lossy compression are forms of data compression, which includes encoding information using fewer bits than the original representation. In lossless compression, no digital difference exists between the original data and the reconstructed compressed data. In lossy compression, a portion of the original data is lost upon reconstruction of the compressed data.

[0030] A variety of lossy and lossless compression systems can be employed in method 300. In one example, the lossy compression applied at 306 can be implemented using a discrete cosine transform, or DCT, which expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies, although other systems can be employed. DCT compression can be particularly apt for examples in which color tables may be expressed in multiple dimensions. For example, an ICC profile may include a three-dimensional or a four-dimensional color table, and the lossy compression can be performed using a three-dimensional or four-dimensional DCT process, accordingly. Another lossy compression system could be based on wavelets, such as the SPIHT (Set Partitioning In Hierarchical Trees) and SPECK (Set Partitioned Embedded bloCK). Lossless compression can be implemented using a variety of lossless systems including Lempel-Ziv-Markov chain Algorithm process (or LZMA), GZIP (or GNU-zip) process, or other suitable lossless systems.

[0031] In order to determine the corrective information at 308, the compressed difference color table is reconstructed to generate an initially reconstructed difference color table. In one example, the reconstructed difference table and reference table each include M^3 nodes, and the value of each node of the reconstructed difference table is added to the value at the corresponding node in the reference table to provide a value at the corresponding node in the initially reconstructed table. The values of the nodes of initially reconstructed table are compared to the corresponding nodes in the corresponding original difference color table. Due to the lossy compression at a selected amount of compression, the values of the nodes of the initially reconstructed difference color table may differ from the values of the corresponding node in the original difference color table. The amount of difference between the values of initially reconstructed difference color table and the original difference color table can be referred to as an error, and may change for different amounts of compression.

[0032] An error threshold, such as a selected value, is defined for a difference between values in the nodes of the initially reconstructed difference color table and the values in the corresponding nodes of the original difference color table. For the set of nodes of the initially reconstructed difference color table having values outside the defined error threshold, corrective information is generated.

[0033] For example, corrective information generated at 308 can include a residual value to be added to each node of the initially reconstructed difference table having a value outside the scope of the defined error threshold to bring the value of that node within the error threshold, such as equal to the value of the corresponding node in the original difference color table, or within a more stringent error threshold, as well as the node location j corresponding with the residual value. In one instance, such as for nodes on or proximate the neutral axis, the residual values correct generally all error. In another instance, such as for nodes in the red or dark blue color space, the residual values correct error to be at or about the error threshold.

[0034] The nodes of the difference table, when compressed and reconstructed, can include a set of nodes that include a color difference falling outside of the error selected threshold at a selected amount of compression. Corrective

information for the set of nodes can include a set of residual values to bring the nodes of the reconstructed difference color table equal to or within the error threshold or within a more stringent error threshold. Corrective information can also include the node locations corresponding with the residual values. In some examples, the corrective information is compressed with a lossless compression. Upon combination of the reconstructed difference color table and the reference color table, such as in printing device 206, the corrective information can be applied such as by adding the residual values to the nodes at the respective locations to obtain a reconstructed color table having all nodes within the error threshold.

[0035] In another example, the initially reconstructed difference table is combined with the reference table and subtracted from the original color table to determine which nodes are outside of the selected error threshold.

[0036] The example method 300 can be implemented to include a combination of one or more hardware devices and programs for controlling a system, such as a computing device having a processor and memory, to perform method 300 to compress a color table into a file or a bitstream. For example, method 300 can be implemented as a set of executable instructions stored in a computer memory device for controlling the processor to perform method 300. Other methods of the disclosure can be implemented as a combination of hardware and programming for controlling a system as well. A color table can include an array or other data structure on a memory device that replaces runtime computations with a simpler array indexing operation as a color look up table.

[0037] Figure 4 illustrates an example method 400 of generating a compressed difference tables, such as compressed difference table 102. Method 400 includes applying a DCT process, such as a multidimensional DCT process, to the difference color tables at 402. Particularly, a J_{in} -dimensional DCT process can be applied to the difference color tables at 402. In general, a profile can include N difference color tables to be processed, such as $difCLUT_1$, $difCLUT_2$, . . . , $difCLUT_N$, and the input color space includes J_{in} channels. Additionally, the output color space includes J_{out} channels, and in many examples of an ICC profile J_{in} and J_{out} can be 3 or 4 channels. For each output channel, the

corresponding lookup table contains $M^{J_{in}}$ nodes. Applying the J_{in} -dimensional DCT transform to the difference color tables $difCLUT_i$, at 402 provides for DCT coefficients, which in the example includes as many coefficients as there are nodes in the difference color table and thus the corresponding original color table.

[0038] The DCT process at 402 yields AC coefficients and DC coefficients that are quantized and processed at 404. Informally, a coefficient that scales the constant basis function is referred to as the DC coefficient, while the other coefficients are referred to as AC coefficients. The AC coefficients are quantized using a fixed step size Δ , and rounded to the nearest integer in an example at 404. Additionally, the DC coefficients are also rounded to the nearest integer at 404, so they are effectively quantized to step size $\Delta = 1$. Quantization yields J_{in} -dimensional quantized coefficients.

[0039] In further processing at 404, the J_{in} -dimensional quantized coefficients are reordered into a one-dimensional data stream of a selected order. The selected order can be based upon a multidimensional zigzag ordering, such as a three-dimensional zigzag ordering, which can be used to reorder the quantized coefficients because the energy after the DCT transform is concentrated in the low frequency domain. In performing a three-dimensional ordering, traversals can be configured such that the planes $i + j + k = c$ are visited in increasing order of c and a two-dimensional zigzagging is performed within each plane. Such traversals of the quantized coefficients from low-to-high frequency can introduce a large amount of redundancy to the coefficient bit assignment table, which can provide efficient packing of the data in compression. The resulting one-dimensional data stream of quantized coefficients can be written to a binary file.

[0040] The one-dimensional data stream of quantized coefficients in the binary file can be compressed with a lossless compression, such as LZMA described above or another lossless compression at 406 to create the compressed quantized coefficients.

[0041] The quantized coefficients are applied to calculate a coefficient bit assignment table at 408, which can be used for decoding the compressed

difference color table. The coefficient bit assignment table stores the information related to how many bits are assigned to each coefficient. For example, $\lceil \log_2 L \rceil$ bits are used to quantize a real number in the range -0.5 to $L - 0.5$ to an integer value, in which $\lceil \log_2(L) \rceil$ represents a ceiling function of $\log_2(L)$ and a ceiling function maps the real number to the smallest subsequent integer. An additional bit is provided to the sign because the coefficient can be a negative number. Each output channel can correspond to a separate coefficient bit assignment table. Accordingly, a profile having J_{out} output channels will include J_{out} coefficient bit assignment tables. The nodes in each coefficient bit assignment table correspond with the nodes of the difference color table.

[0042] An example process can be applied to calculate a coefficient bit assignment table for each of the J_{out} output channels at 408. For a given output channel, the quantized DCT coefficient of the output channel is denoted as Q_{ij} , in which i (from 1 to N) is the color table number and j (from 1 to $M^{J_{in}}$) is the node number. The number of bits B_{ij} needed for Q_{ij} is $B_{ij} = 0$ if Q_{ij} is 0 and $B_{ij} = \lceil \log_2 |Q_{ij}| \rceil + 1$ if Q_{ij} is not 0.

[0043] In one example, a fixed number of bits a can be assigned to every node of the coefficient bit assignment table and used to determine the size of each coefficient bit assignment table. The value of the coefficient bit assignment table at node location j , or L_j , can be determined from the largest number of bits B_{ij} needed for each i (from 1 to N) difference color table. The fixed number of bits a can be determined from the largest number of $\lceil \log_2 (L_j) \rceil$ as determined for each j (from 1 to $M^{J_{in}}$). In the example, the total size of one coefficient bit assignment table for an output channel is thus $aM^{J_{in}}$ bits. The above process can be repeated to determine the size for each output channel, and the total size is the sum of the sizes for the J_{out} coefficient bit assignment tables.

[0044] The J_{out} coefficient bit assignment tables are compressed such as with a lossless compression at 406 to create the compressed coefficient bit assignment table. The total size of the coefficient bit assignment table can be significantly reduced via the lossless compression.

[0045] The selected step size Δ in 404 affects the compression, and a larger step size Δ achieves a larger amount of compression. The selected step size Δ in 404, however, also affects the amount of error between the values of the node in the difference color table and the values of the nodes of the reconstructed compressed difference color table, and a larger step size Δ creates a larger amount of error. For some reconstructed compressed values, the amount of error may be acceptable, such as generally imperceptible, for an application. In such cases, the color difference or amount of error is within a selected error threshold. In other cases, such as for colors around the neutral axis, an amount of error may be too perceptible for the application. In such cases, the amount of error is outside the scope of the selected error threshold. In one example, an amount of error outside the selected error threshold is an amount of error greater than the selected error threshold.

[0046] In order to achieve a high enough step size to provide benefits of lossy compression, one or more nodes may include an amount of error outside the scope of the selected error threshold. Nodes including an amount of error outside the scope of the selected error threshold are determined at 410. For such nodes, corrective information to modify the nodes is generated at 412.

[0047] Figure 5 illustrates a method 500 that can be implemented in determining the nodes of the original color table to be modified with corrective information. In one example, the method 500 is performed using a selected step size and a selected error threshold. The original difference color table is compressed using a lossy compression technique such as DCT in 402, 404, 408 of method 400 at 502. The compressed difference color table is reconstructed via decoding to invert the compression and yield a reconstructed color table at 504. The nodes of the reconstructed difference color table are compared to the original difference color table to determine an amount of error at 506. The amount of error for a node, in one example, is the difference between the value associated with the node in the original difference color table and the value associated with the node of the reconstructed difference color table. In one example of 506, each node of the reconstructed difference color table is compared to its

corresponding node in the original difference color table to determine an amount of error for that node.

[0048] In one example, the processes of 402, 404, and 408 can be reapplied to the original color table at a modified step size to include more nodes within the selected error threshold. For instance, if the amount of nodes of the reconstructed difference color table at having an amount of error outside of the selected threshold is too large for a given application, the processes 402, 404, 408 can be repeated on the original difference color table using a smaller step size Δ .

[0049] The node or nodes that include an amount of error outside the scope of the selected error threshold are identified at 508 and corrective information is included for the identified nodes at 508. In one example of 412, the corrective information for the nodes of the original difference color table identified at 508 include residual values that can be added to (or subtracted from) the reconstructed nodes to bring the reconstructed node to within the selected error threshold for the node. The corrective information includes the residual values, which can then be losslessly compressed. In some examples, the residual values applied to the reconstructed nodes can bring the final values of the reconstructed nodes to within a second, more stringent error threshold than the error threshold associated with the first set of nodes. In another example, the residual values applied to the reconstructed nodes can bring the final value of the reconstructed nodes to be generally the same as the original nodes. The application of method 500 provides the option to store nodes with significant visual importance that may include perceptible error after lossy compression, such as nodes around the neutral axis, without loss. This improves the performance of the decoded compressed color table.

[0050] The compression ratio of the data on the memory device 100 (as compared to the original difference color table) and selected step size Δ are related when nodes exceeding a selected error threshold of color difference are stored without lossy compression at 410. In particular, as the selected step size Δ increases, the compression ratio first increases, reaches a peak value of compression ratio, and then decreases. Not being bound to a particular theory,

as the selected step size Δ increases, the compression becomes more aggressive and the amount of error gets larger. As the number of nodes outside the scope of the selected error threshold increases, more nodes are stored without lossy compression or at a lower compression ratio reducing the efficiency of the color table compression. The peak value of the compression ratio for a selected error threshold provides an optimal compression ratio for a selected error threshold of color difference. Additionally, the peak value of compression ratio increases as the error threshold increases.

[0051] In one example, a process of first determining an acceptable error threshold in which nodes exceeding the threshold are stored without lossy compression, and then determining the selected step size Δ that corresponds to the peak compression ratio of the relation can be used to select the optimal compression ratio of the lossy compression.

[0052] Figure 6 illustrates an example memory device 600 including a compressed difference color table 602 and corrective information 604, which can correspond with the compressed difference color table 102 and corrective information 104 of memory device 100. In one example, the compressed difference color table 602 and corrective information 604 can be generated with method 300 and stored on the memory device 600 at 310. In another example, the compressed difference color table 602 and corrective information 604 are reproduced from another memory device that may have information stored at 310.

[0053] The compressed difference color table 602 stored on the memory device 600 includes bits representative of quantized coefficients 606 from a lossy compression of difference color table. The quantized coefficients 606 of compressed difference color table 602 can be produced at 306. In one example, the quantized coefficients 606 are stored as a binary file. In another example, the quantized coefficients 606 have been further compressed with a lossless compression. Memory device 600 can include additional information related to decoding the quantized coefficients, such as step size Δ or other information.

[0054] The corrective information 604 includes residual values 608 be added to a set of nodes of the initially reconstructed difference table to bring the value of

that node within the error threshold. The residual values 608 can be produced at 308. The corrective information 604 can also include node location information 610 for the locations of the set of nodes to be modified with the residual values, such as node location information generated at 308, as to the location of the nodes corresponding with the residual values 608. In one example, the residual values 608 and node location information 610 are stored as a binary file. In another example, the residual values 608 and node location information 610 have been further compressed with a lossless compression.

[0055] In one example, the memory device 600 is included on a supply component 612, such as an ink cartridge, a toner cartridge, etc. The supply component can be operably coupled to a printing device 614. The printing device can include a separate memory device 616 that is operably coupled to the memory device 600. In one example, the separate memory device 616 includes a coefficient bit assignment table (CBAT) 618, such a coefficient bit assignment table generated at 306, which can be applied to decode the quantized coefficients 606. The separate memory device 616 can also include a reference table 620 such as the reference table generated at 302. In other examples, one or more of the CBAT 618 and reference table 620 can be instead included on memory device 600.

[0056] The compressed difference table 602 and corrective information 604 can correspond with one difference color table 622a of a plurality of difference color tables 622a, 622b, ..., 622n. In one example, the color table information related to the ink color of the cartridge is included as a difference color table 622a. Other color difference color tables 622b-622n can include a color table for each media type/color separation method (such as K-Only and CMYK Black color separation methods). The plurality of difference color tables 622a-622n can relate to The N difference color tables, and each of the N difference color tables can include a unique set of quantized coefficients, residual values, and node location information. Further, the separate memory device 616 can include a plurality of CBATs that correspond with the quantized coefficients in each of the plurality of difference color tables 622a-622b for each of the colorants.

[0057] Figure 7 illustrates a method 700 of decoding the compressed color difference table 622a of memory device 600. A standard lossless decompression technique, such as inverse LZMA or inverse GZIP (i.e., the inverse of the lossless compression applied at 406), is applied to losslessly compressed difference color table 602 and corrective information 604 at 702 to provide a binary stream including the quantized DCT coefficients 606, and corrective information 604 such as residual values 608 and nodes locations 610. The coefficient bit assignment table (CBAT) 618 can be used to determine how many bits of the binary stream belong to each node location. The coefficient bit assignment tables 618 are applied to the quantized DCT coefficients 606 to reconstruct the DCT coefficients at 704. An inverse DCT transform is applied to the DCT coefficients at 706. The coefficients are multiplied by the quantizer step size Δ and rounded to the nearest integer to obtain the initially reconstructed difference color tables at 708.

[0058] The corrective information 604 is applied to the initially reconstructed difference color tables at 710 to obtain the decompressed difference color table. In one example the corrective information 604 includes residual values 608 that can be applied to a set of nodes of the initially reconstructed color table to obtain the decompressed difference color table. For example, the residual value for an identified node location is added to the value of the corresponding node location in the initially reconstructed difference table. This process is repeated for each residue value in the corrective information 604.

[0059] The reference table 620 is added to the decompressed difference table from 710 to obtain final reconstructed color table at 712. In one example, the decompressed difference table and reference table 620 each include M^3 nodes, and the value of each node of the decompressed difference table is added to the value of the corresponding node in the reference table 620 to provide a value at the corresponding node in the final reconstructed color table. Thus, the value at node location j of the decompressed difference table is added to the value at node location j of the reference table 620 to provide the value at node location j of the final reconstructed color table, in which j is the node location

from 1 to M^3 . The final reconstructed J_{in} -dimensional to J_{out} -dimensional color tables can be applied to a color management system.

[0060] The example, method 700 can be implemented to include a combination of one or more hardware devices and programs for controlling a system, such as printing device having a processor and memory, to perform method 700 to decode the compressed color difference table of memory device 600. For example, method 700 can be implemented as a set of executable instructions stored in a computer memory for controlling the processor to receive the compressed color difference table and reference table and perform method 700.

[0061] Although specific examples have been illustrated and described herein, a variety of alternate and/or equivalent implementations may be substituted for the specific examples shown and described without departing from the scope of the present disclosure. This application is intended to cover any adaptations or variations of the specific examples discussed herein. Therefore, it is intended that this disclosure be limited only by the claims and the equivalents thereof.

CLAIMS

1. A print cartridge comprising:
 - a memory device (100; 400; 604) comprising a compressed data structure (102; 402; 602) to construct a one-dimensional color transformation table for a printer, the data structure comprising:
 - quantized coefficients derived from a compression of a difference table (104; 404; 616) including a plurality of difference nodes in which each difference node represents a value that is to be combined with a corresponding node of a reference table, the quantized coefficients useable to produce a reconstructed difference table;
 - a residue table (106; 408; 618) including a plurality of residue nodes in which each residue node is to be combined with a corresponding node of the combination of the reconstructed difference table and the reference table; and
 - a residue bit assignment table (410) useable to decode the residue table (408).
2. The print cartridge of claim 1, wherein the residue bit assignment table (410) is to indicate how many bits are assigned to each residue node of the residue table (408).
3. The print cartridge of claim 1 or 2, wherein the quantized coefficients were generated via a discrete cosine transform, and are to be decompressed using an inverse discrete cosine transform process to thereby obtain the decompressed difference table.
4. The print cartridge of one of claims 1 - 3, wherein each difference node that represents a value that is to be combined with a corresponding node of a reference table comprises a difference of value of a node of an original color table and a value of a corresponding node of a reference table.
5. The print cartridge of one of claims 1 - 4, wherein each residue node that is to be combined with a corresponding node of the combination of the reconstructed difference table and the reference table comprises a value that is a difference of a value of a node of the original color table and a value of a corresponding node of the reconstructed difference table.
6. The print cartridge of one of claims 1 - 5, wherein the memory device (100; 400; 604) comprises a losslessly compressed data structure (102; 402; 602).

7. The print cartridge of claim 6, wherein the lossless compression is a Lempel-Ziv-Markov chain Algorithm compression.
8. The print cartridge of one of any one claims 1 - 7, wherein the quantized coefficients are derived from a lossy compression of the difference table.
9. The print cartridge of claim 6, wherein the quantized coefficients are quantized with a fixed step size using the lossy compression.
10. The print cartridge of one of claims 1 - 9, wherein the quantized coefficients are to be reconstructed using a cosine bit assignment table stored on the printer or cartridge, and multiplied by a fixed step size to obtain the decompressed difference table.
11. The print cartridge of one of claims 1 - 10, wherein the compressed data structure further comprises a coefficient bit assignment table (406) derived from the quantized coefficients, the coefficient bit assignment table (406) useable to decode the quantized coefficients, the coefficient bit assignment table (406) including information related to a number of bits assigned to each of the quantized coefficients.
12. The print cartridge of one of claims 1 to 11, wherein the values of the residue bit assignment table (410) are derived from the residue table (408).
13. The print cartridge of one of claims 1 - 12, comprising black printer material.
14. The print cartridge of one of claims 1 - 13, wherein the memory device (100, 400, 600) is to be operably coupled to another computing device having another memory (706), the other memory (706) storing the reference table.
15. System (600) comprising the print cartridge of one of claims 1 - 14, wherein the memory device (604) is operably coupled to another computing device (702) that includes an inherent print capability, the computing device (702) having a processor (704) and memory (706) to read and apply the compressed color table (602), the memory (706) storing the reference table.