US 20040228527A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2004/0228527 A1

(43) Pub. Date: Nov. 18, 2004

Ioucha et al.

(54) METHOD AND APPARATUS FOR BLOCK BASED IMAGE COMPRESSION WITH MULTIPLE NON-UNIFORM BLOCK ENCODINGS

(76) Inventors: **Konstantine Ioucha**, San Jose, CA (US); **Andrew S.C. Pomianowski**, Palo Alto, CA (US); **Raja Koduri**, Santa Clara, CA (US)

Correspondence Address:
COUDERT BROTHERS LLP
333 SOUTH HOPE STREET
23RD FLOOR
LOS ANGELES, CA 90071 (US)

(21) Appl. No.: **10/778,902**

(22) Filed: **Feb. 13, 2004**

Related U.S. Application Data

(60) Provisional application No. 60/447,862, filed on Feb. 13, 2003.

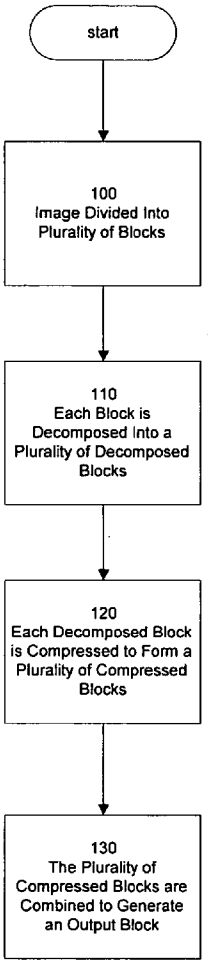
Publication Classification

(51) **Int. Cl.**[7] **G06K 9/00**; G06K 9/36

(52) **U.S. Cl.** **382/166**; 382/232

(57) **ABSTRACT**

Embodiments of the present invention are directed to a method and apparatus for block based image compression with multiple non-uniform block encodings. In one embodiment, an image is divided into blocks of pixels. In one embodiment the blocks are four pixels by four pixels, but other block sizes are used in other embodiments. In one embodiment, a block of pixels in the original image is compressed using two different methods to produce a first and second compressed block. Thus, each block in the original image is represented by two, typically different, compressed blocks. In one embodiment, color associated with a pixel is determined by combining the compressed information about the pixel in the first compressed block with information about the pixel in the second compressed block. In another embodiment, global information about the image is combined with the information in the first and second compressed blocks.



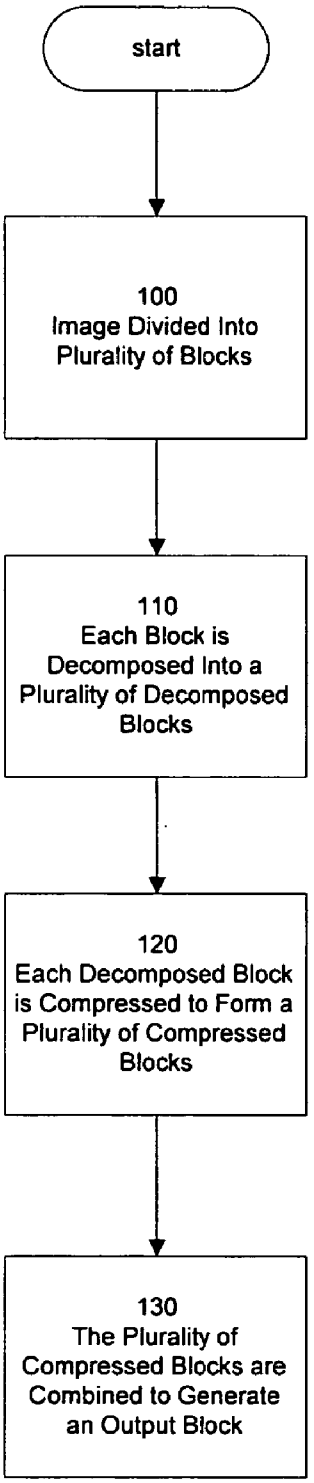

start
100
Image Divided Into Plurality of Blocks
110
Each Block is Decomposed Into a Plurality of Decomposed Blocks
120
Each Decomposed Block is Compressed to Form a Plurality of Compressed Blocks
130
The Plurality of Compressed Blocks are Combined to Generate an Output Block

Figure 1

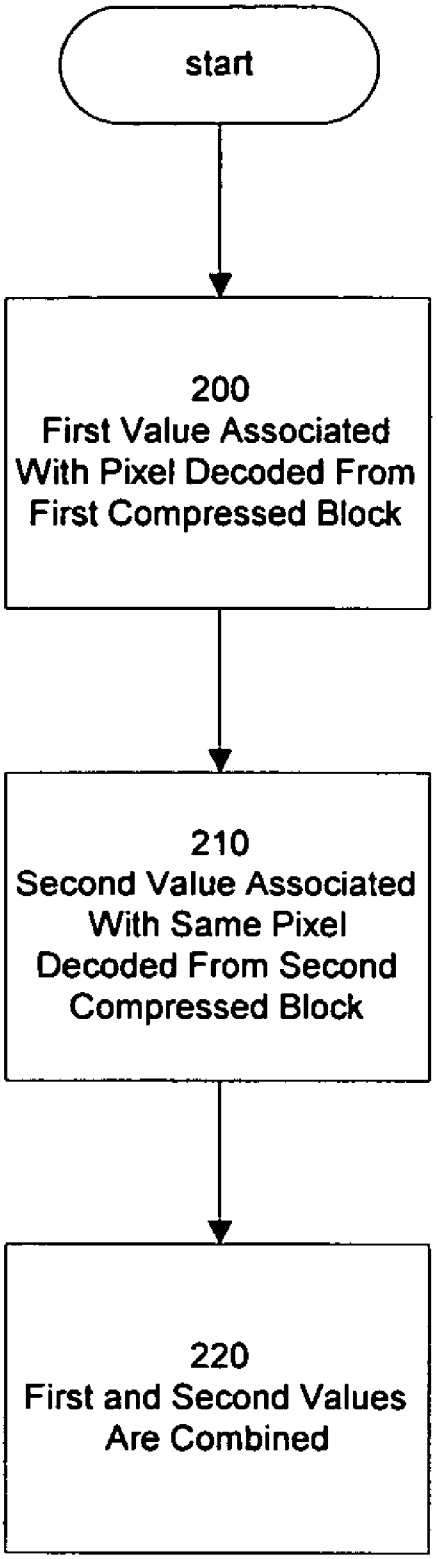

start
200
First Value Associated With Pixel Decoded From First Compressed Block
210
Second Value Associated With Same Pixel Decoded From Second Compressed Block
220
First and Second Values Are Combined

Figure 2

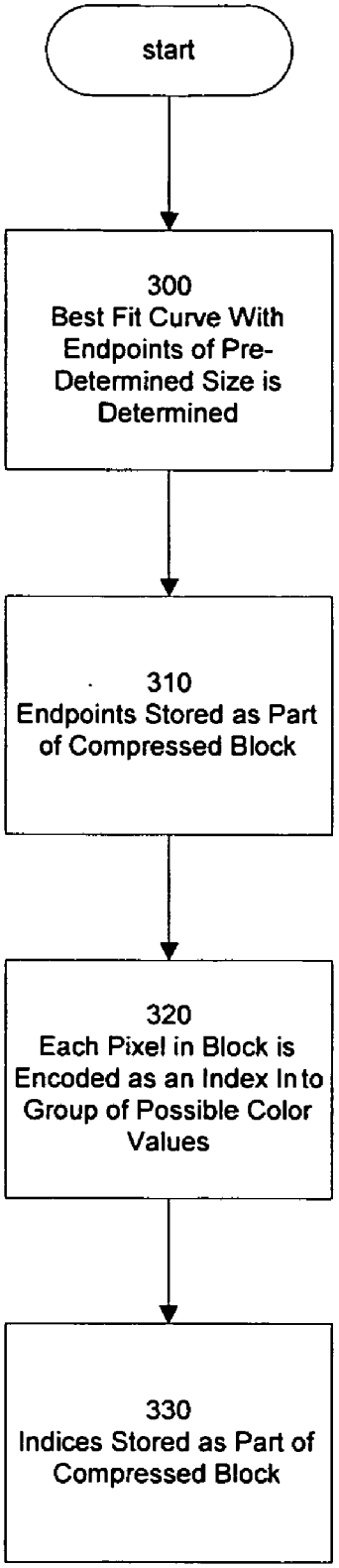

start
300
Best Fit Curve With Endpoints of Pre-Determined Size is Determined
310
Endpoints Stored as Part of Compressed Block
320
Each Pixel in Block is Encoded as an Index Into Group of Possible Color Values
330
Indices Stored as Part of Compressed Block

Figure 3

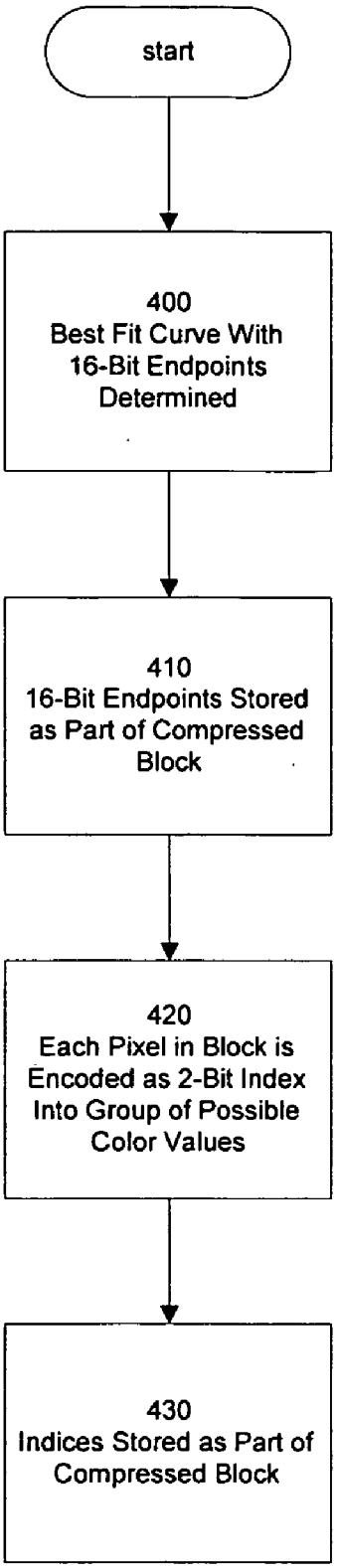

start
400
Best Fit Curve With 16-Bit Endpoints Determined
410
16-Bit Endpoints Stored as Part of Compressed Block
420
Each Pixel in Block is Encoded as 2-Bit Index Into Group of Possible Color Values
430
Indices Stored as Part of Compressed Block

Figure 4

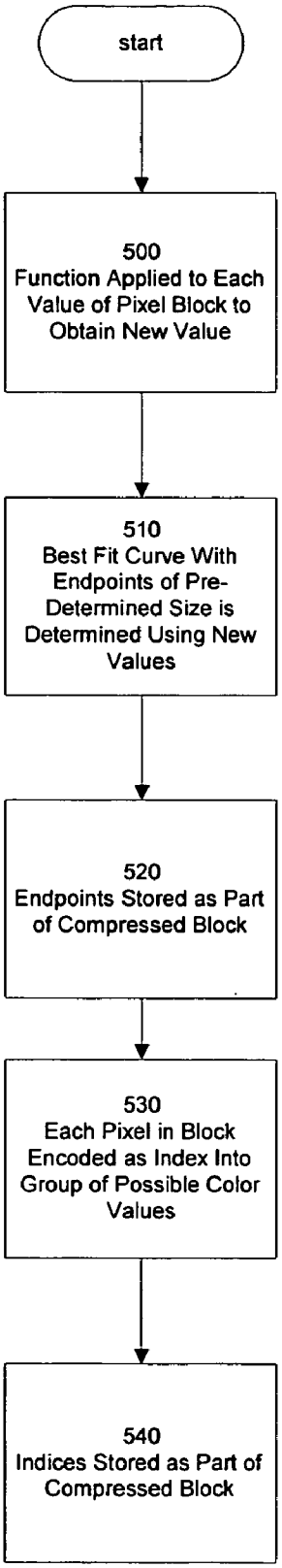

start
500
Function Applied to Each Value of Pixel Block to Obtain New Value
510
Best Fit Curve With Endpoints of Pre-Determined Size is Determined Using New Values
520
Endpoints Stored as Part of Compressed Block
530
Each Pixel in Block Encoded as Index Into Group of Possible Color Values
540
Indices Stored as Part of Compressed Block

Figure 5

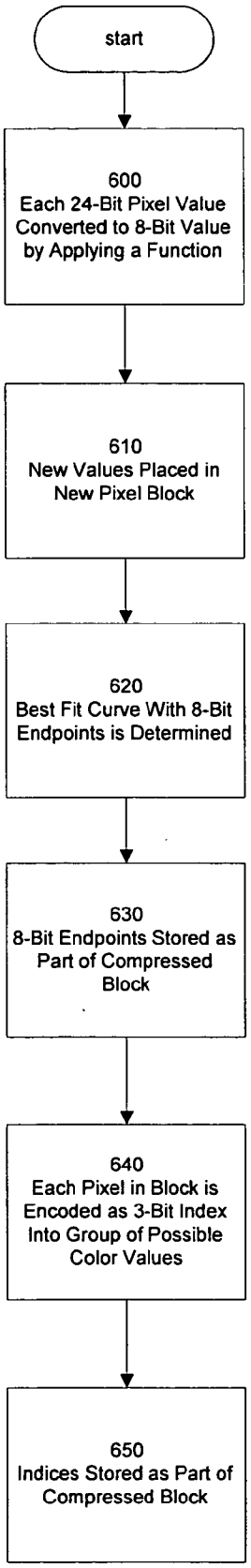

start
600
Each 24-Bit Pixel Value Converted to 8-Bit Value by Applying a Function
610
New Values Placed in New Pixel Block
620
Best Fit Curve With 8-Bit Endpoints is Determined
630
8-Bit Endpoints Stored as Part of Compressed Block
640
Each Pixel in Block is Encoded as 3-Bit Index Into Group of Possible Color Values
650
Indices Stored as Part of Compressed Block

Figure 6

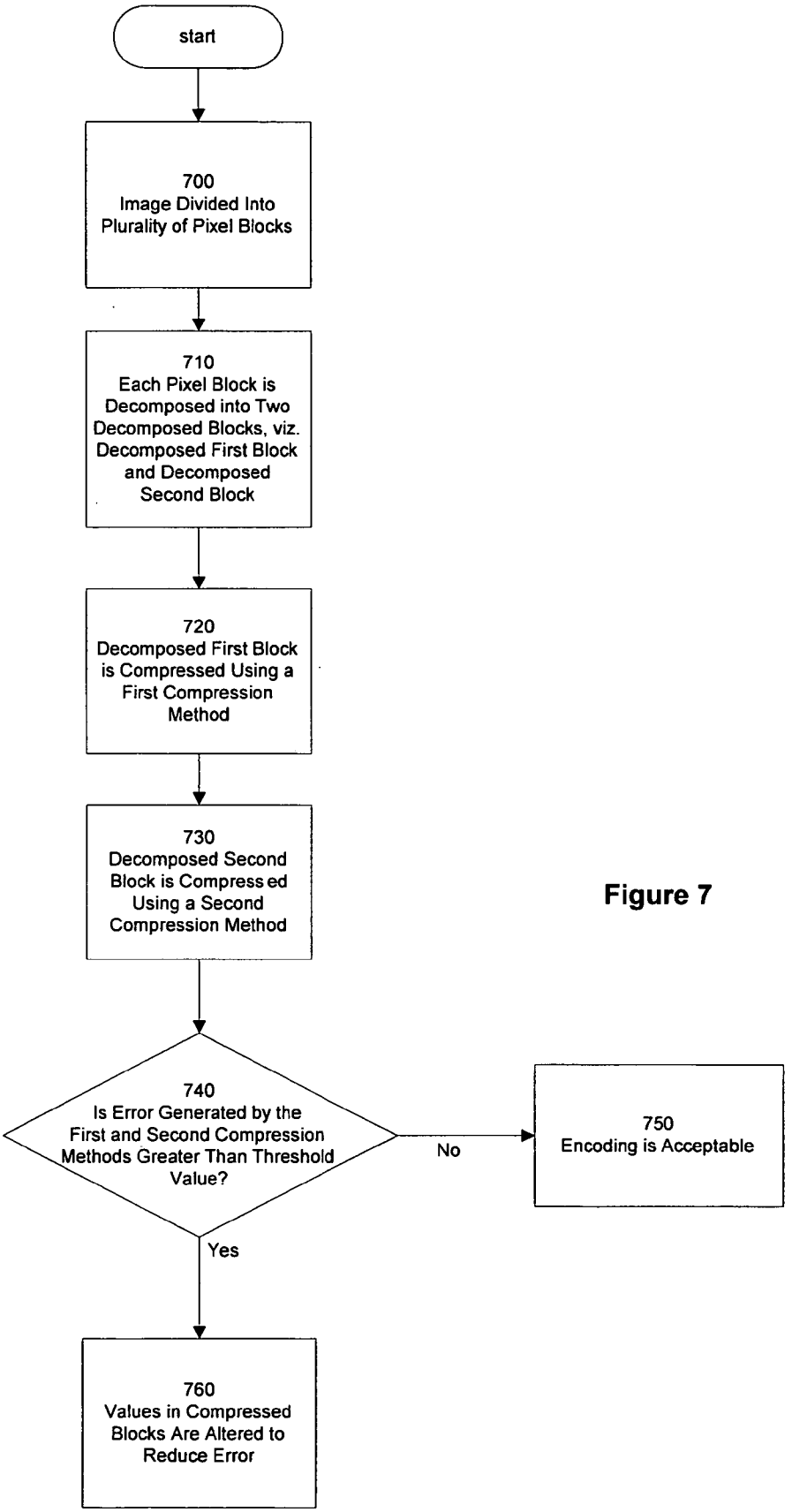

start
700
Image Divided Into Plurality of Pixel Blocks
710
Each Pixel Block is Decomposed into Two Decomposed Blocks, viz. Decomposed First Block and Decomposed Second Block
720
Decomposed First Block is Compressed Using a First Compression Method
730
Decomposed Second Block is Compressed Using a Second Compression Method
740
Is Error Generated by the First and Second Compression Methods Greater Than Threshold Value?
No
750
Encoding is Acceptable
Yes
760
Values in Compressed Blocks Are Altered to Reduce Error

Figure 7

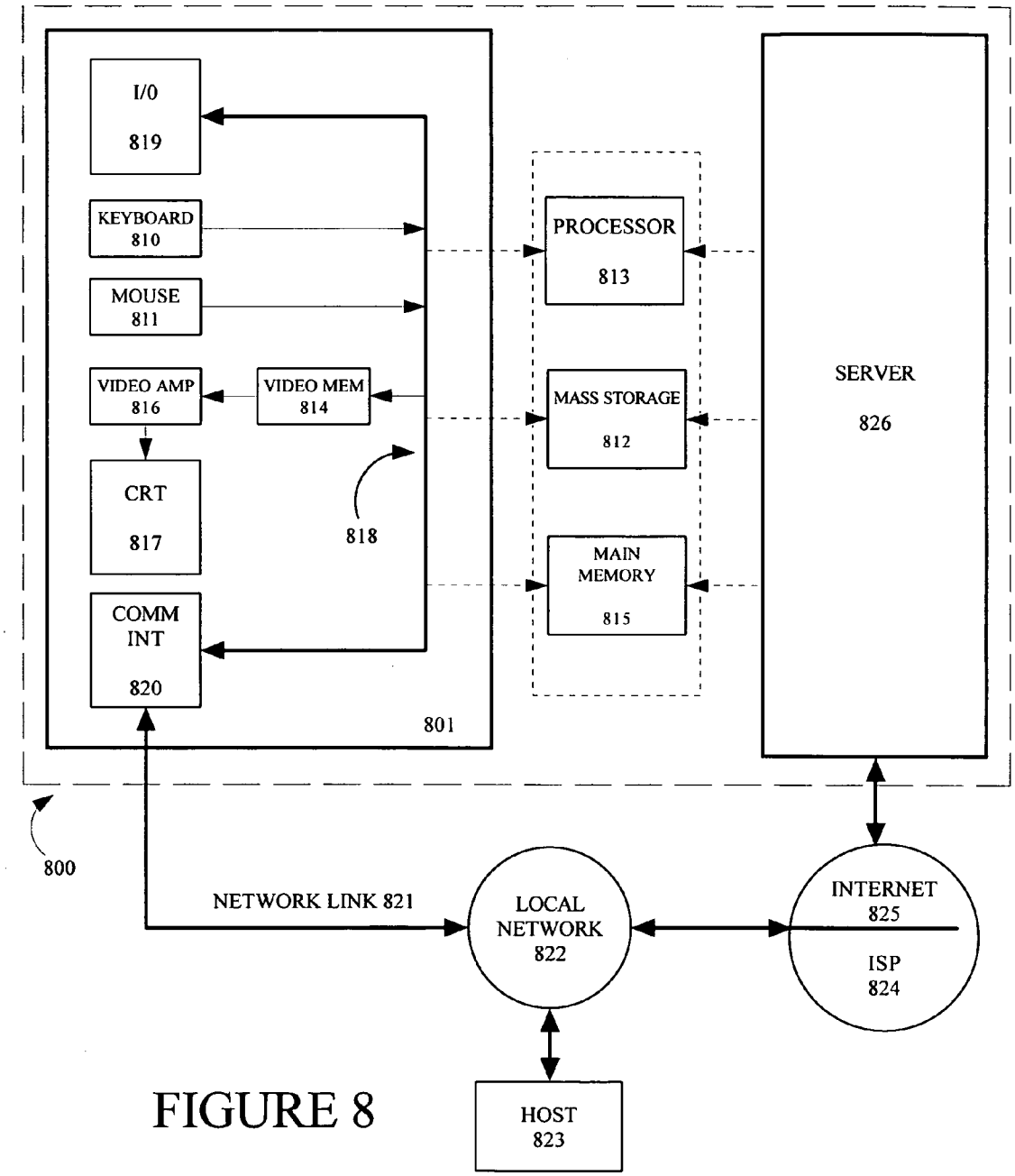

I/0
819
KEYBOARD
810
MOUSE
811
VIDEO AMP
816
VIDEO MEM
814
CRT
817
818
COMM
INT
820
801
PROCESSOR
813
MASS STORAGE
812
MAIN
MEMORY
815
SERVER
826
800
NETWORK LINK 821
LOCAL
NETWORK
822
INTERNET
825
ISP
824
HOST
823

FIGURE 8

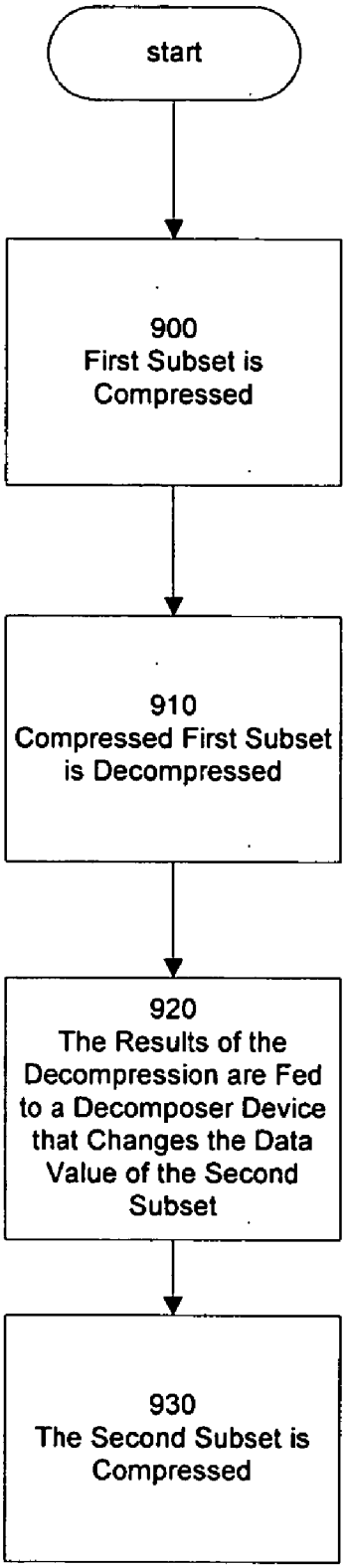

start
900
First Subset is Compressed
910
Compressed First Subset is Decompressed
920
The Results of the Decompression are Fed to a Decomposer Device that Changes the Data Value of the Second Subset
930
The Second Subset is Compressed

Figure 9

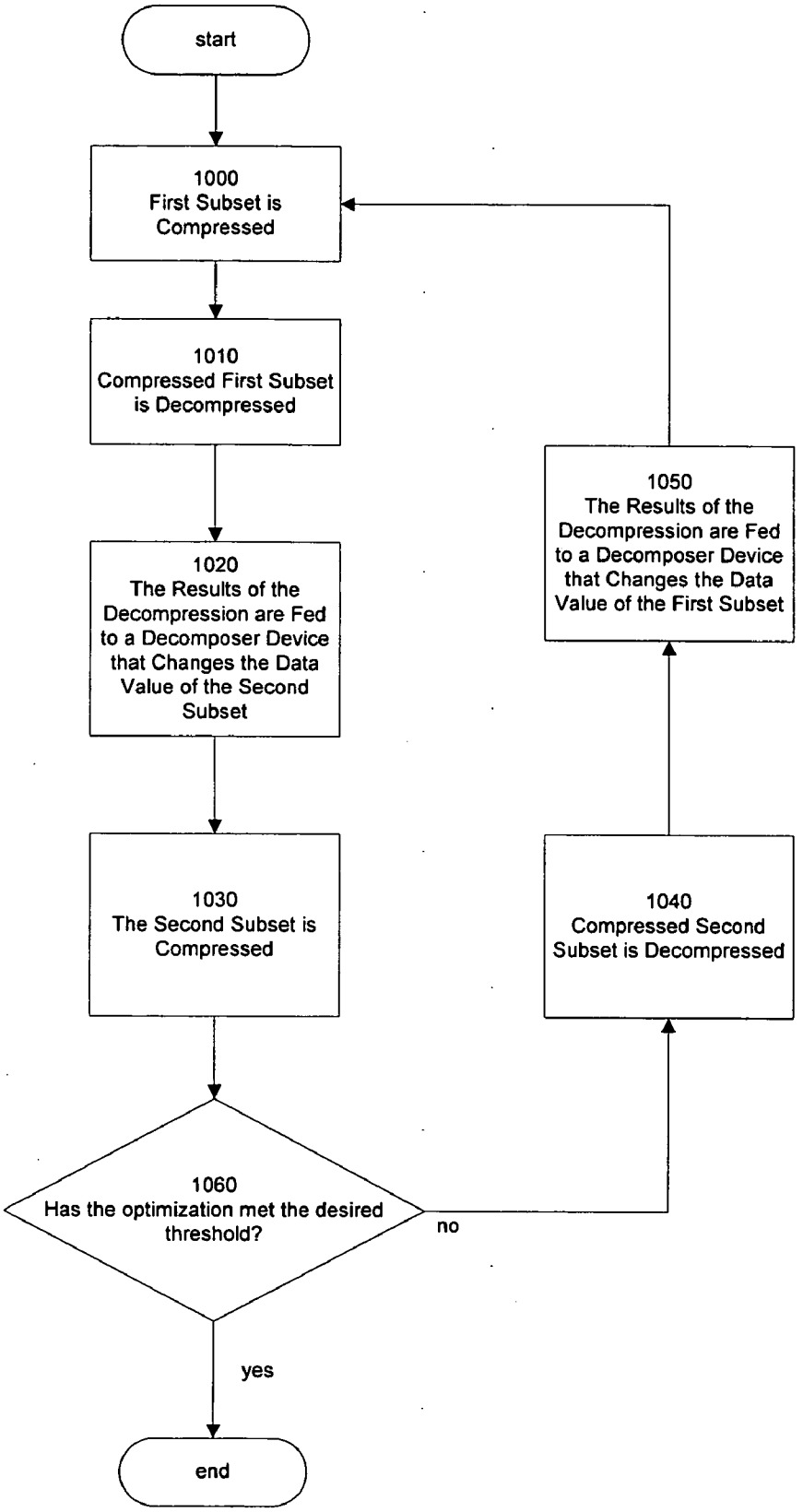

start
1000
First Subset is Compressed
1010
Compressed First Subset is Decompressed
1020
The Results of the Decompression are Fed to a Decomposer Device that Changes the Data Value of the Second Subset
1030
The Second Subset is Compressed
1060
Has the optimization met the desired threshold?
no
yes
end
1040
Compressed Second Subset is Decompressed
1050
The Results of the Decompression are Fed to a Decomposer Device that Changes the Data Value of the First Subset

Figure 10

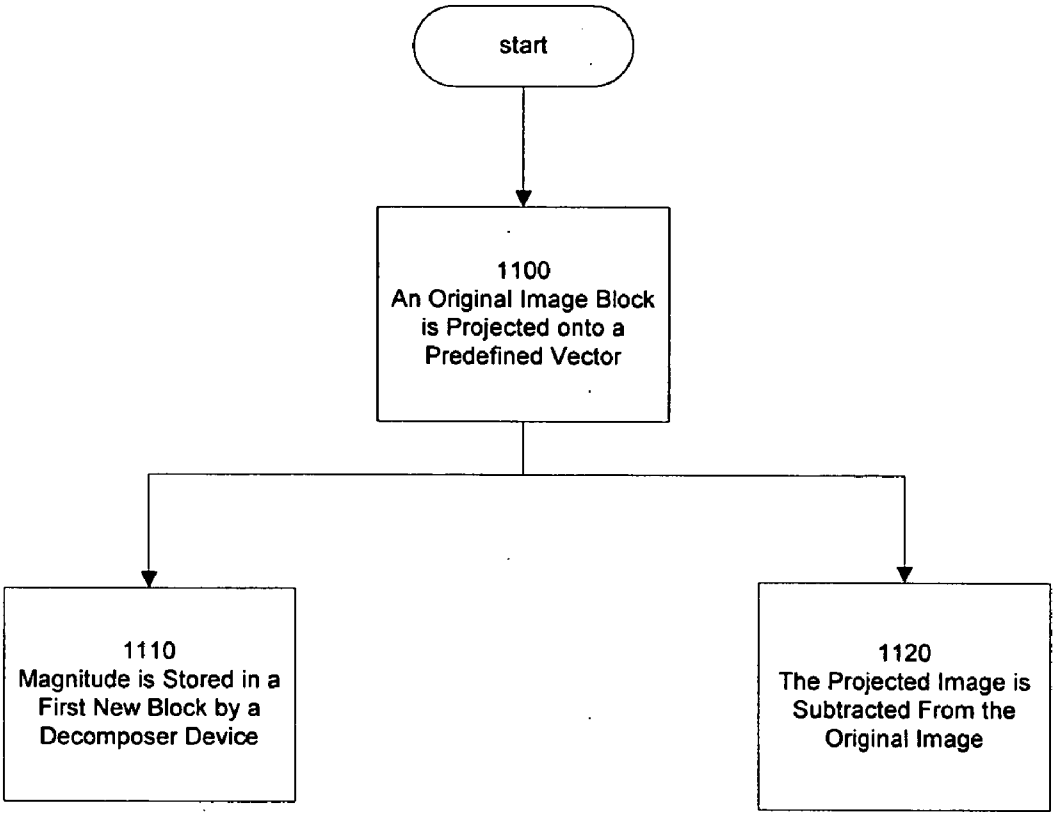

start
1100
An Original Image Block is Projected onto a Predefined Vector
1110
Magnitude is Stored in a First New Block by a Decomposer Device
1120
The Projected Image is Subtracted From the Original Image

Figure 11

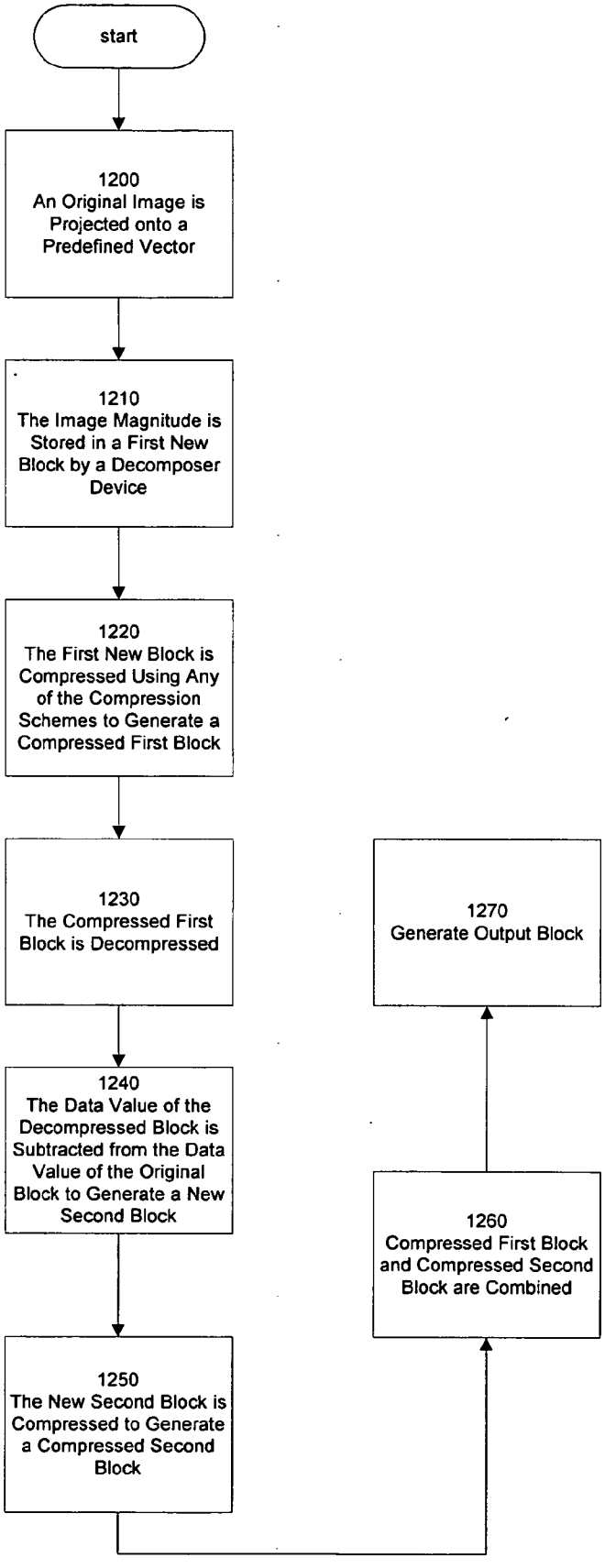

start
1200
An Original Image is Projected onto a Predefined Vector
1210
The Image Magnitude is Stored in a First New Block by a Decomposer Device
1220
The First New Block is Compressed Using Any of the Compression Schemes to Generate a Compressed First Block
1230
The Compressed First Block is Decompressed
1240
The Data Value of the Decompressed Block is Subtracted from the Data Value of the Original Block to Generate a New Second Block
1250
The New Second Block is Compressed to Generate a Compressed Second Block
1260
Compressed First Block and Compressed Second Block are Combined
1270
Generate Output Block

Figure 12

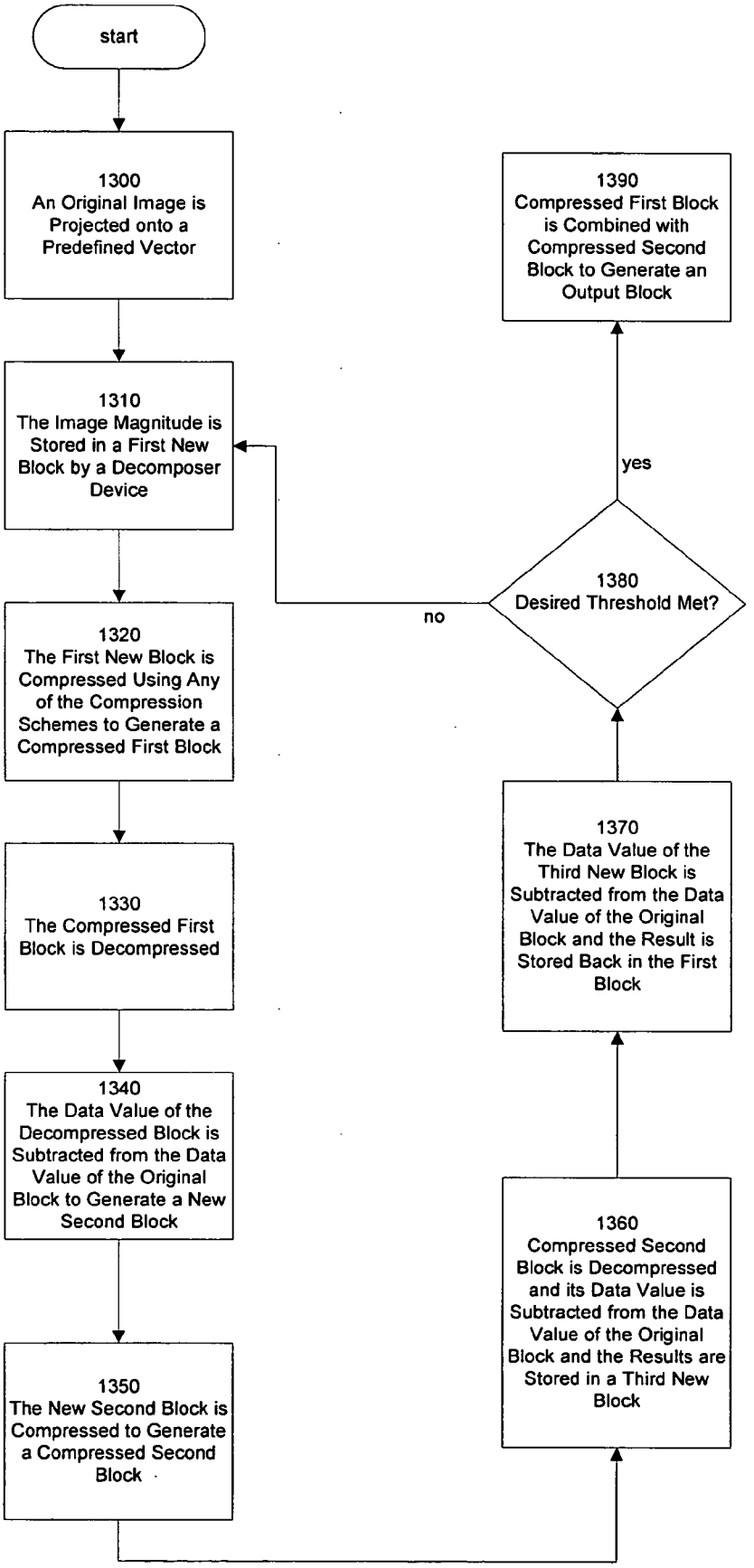

start
1300
An Original Image is Projected onto a Predefined Vector
1310
The Image Magnitude is Stored in a First New Block by a Decomposer Device
1320
The First New Block is Compressed Using Any of the Compression Schemes to Generate a Compressed First Block
1330
The Compressed First Block is Decompressed
1340
The Data Value of the Decompressed Block is Subtracted from the Data Value of the Original Block to Generate a New Second Block
1350
The New Second Block is Compressed to Generate a Compressed Second Block
1360
Compressed Second Block is Decompressed and its Data Value is Subtracted from the Data Value of the Original Block and the Results are Stored in a Third New Block
1370
The Data Value of the Third New Block is Subtracted from the Data Value of the Original Block and the Result is Stored Back in the First Block
1380
Desired Threshold Met?
no
yes
1390
Compressed First Block is Combined with Compressed Second Block to Generate an Output Block

Figure 13

METHOD AND APPARATUS FOR BLOCK BASED IMAGE COMPRESSION WITH MULTIPLE NON-UNIFORM BLOCK ENCODINGS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims the benefit of priority from pending U.S. Provisional Patent Application No. 60/447,862, entitled Method and Apparatus For Block Based Image Compression With Multiple Non-Uniform Block Encodings, filed on Feb. 13, 2003, which is herein incorporated by reference in its entirety.

BACKGROUND OF THE INVENTION

[0002] 1. Field of the Invention

[0003] The present invention relates to the field of image compression, and in particular to a method and apparatus for block based image compression with multiple non-uniform block encodings.

[0004] 2. Background Art

[0005] Typically, there is a desire to compress images used for texturing operations in 3D graphics accelerator hardware to improve the performance of rendering and reduce the memory required to store the information. However, prior art image compression methods frequently produce undesired visual artifacts when rendering images, making their application appropriate for a limited number of images. This problem can be better understood with a review of image compression.

[0006] Image Compression

[0007] Prior art compression techniques can be categorized in at least two ways. First, a technique can be either loss-less or lossy. Second, a technique can be either fixed or variable rate. A loss-less compression technique is one in which no information about the image is lost due to the compression. Thus, an image could be compressed and decompressed and the decompressed image would be identical to the original image. A lossy compression technique is one in which some information about the image is lost due to the compression. Thus, compression of an image followed by decompression could result in a decompressed image that is not identical to the original image.

[0008] A fixed rate compression technique reduces the storage requirement of an image by a fixed percentage. Since most image formats do not contain unnecessary data, almost all fixed rate compression techniques are lossy in general (however, they may not be lossy for a particular image). A variable rate compression technique reduces the storage requirement of an image by an amount that is not known at the time when the image is compressed. In fact, the storage requirement may not be reduced at all.

[0009] Because loss-less, variable rate compression techniques are unable to guarantee any rate of compression, graphics systems typically use lossy, fixed rate compression techniques. However, the lossy nature sometimes results in unacceptable image quality (i.e., visual artifacts). Sometimes, it is so difficult to determine when compression will result in unacceptable loss of image quality and when it will not that users and/or systems elect to forgo compression altogether. As systems are developed that utilize larger amounts of information about the system, the desire for compression increases, but the problem of unacceptable image quality loss also increases in prior art compression systems.

SUMMARY OF THE INVENTION

[0010] Embodiments of the present invention are directed to a method and apparatus for block based image compression with multiple non-uniform block encodings. In one embodiment of the present invention, an image is divided into blocks of pixels. In one embodiment the blocks are four pixels by four pixels, but other block sizes are used in other embodiments.

[0011] In one embodiment, a block of pixels in the original image is decompressed into one or more decompressed blocks, each of which represent a partial data value of the original block. Each of these blocks are compressed using the same or different methods for each block to produce one or more compressed blocks, which are combined to produce an output block. Thus, each block in the original image is represented by a plurality, typically different, compressed blocks each representing all the data values of the block. In one embodiment, color (or a similar value) associated with a pixel is determined by combining the compressed information about the pixel in one compressed block with compressed information about the same pixel in another compressed block. In another embodiment, global information about the image is also combined with the information in one or more compressed blocks to produce a desired value for a pixel.

[0012] According to one embodiment, decompression of the original image is optimized to reduce the loss. According to one embodiment, the optimization is performed by conducting a series of functions on a first subset comprising of one or more decompressed blocks and applying the results to a second subset comprising of one or more decompressed blocks different from the ones in the first subset. According to one embodiment, these functions are iterated through the subsets until a predetermined value of the image is obtained, or some other factor is satisfied. According to another embodiment, the optimization is performed by projecting an original block to a predefined vector and storing the original block's magnitude in a first block, and using a device that projects and subtracts the projection block from of the original block and stores the value in a second block. According to another embodiment, the optimization is performed by conducting a series of functions on the first and second blocks obtained in the previous embodiment. According to another embodiment, the optimization of the previous embodiment is iterated in a loop until a desired output image is accomplished or some other factor is met.

[0013] As way of example, two decompressed blocks of an original box of pixels in the image are compressed using two different methods to produce a first and a second compressed block by encoding each pixel value as an index into a color map before combining the two compressed blocks to generate an output block. In one embodiment, the color map maps pixels to a best fit curve with two end points in a color space. In one embodiment the curve is a straight line. In one embodiment, the end points of the best fit curve are values to which the mapping can map. In another embodiment, the mapping can map to an extrapolated value

between two endpoints along the best fit curve. In still another embodiment, the mapping can map to a plurality of different extrapolated values between two endpoints along the best fit curve. In one embodiment, the compressed block is comprised of the two endpoints of the best fit curve and index values for each of the pixels.

[0014] In an example embodiment, the original block is a four pixel by four pixel block wherein each pixel value is represented by 24 bits (e.g., a 24-bit RGB color value). Two, 16-bit endpoints for a best fit curve are stored. Additionally, sixteen 2-bit indices (one for each pixel in the block) are stored. The indices encode which of the two endpoint colors or two interpolated values along the curve are the compressed color for each pixel. The endpoints and the indices comprise a first compressed block.

[0015] In another example embodiment, a compressed block is created by applying a function to each of the values of the original block to obtain new block values. In one embodiment, the function combines components of each value to reduce the amount of space needed to encode the data. In an example embodiment, 24-bit RGB values are combined to create an 8-bit grey scale value (e.g., (R+G+B)/3). In one embodiment, the new block values are compressed by encoding the block as an index for mapping onto a best fit curve. In one embodiment, two endpoints are stored. In another embodiment, the indices map to either the two end points or any of one or more interpolated values along the best fit curve. The two stored endpoints and the block of indices comprise a second compressed block.

[0016] In an example embodiment, the original block is a four pixel by four pixel block wherein each pixel value is represented by 24 bits (e.g., a 24-bit RGB color value). The values in the block are converted to 8-bit grey scale values. Two, 8-bit endpoints for a best fit curve are stored. Additionally, sixteen 3-bit indices (one for each pixel in the block) are stored. The indices encode which of the two endpoint colors or six interpolated values along the curve are the compressed color for each pixel. The endpoints and the indices comprise a third compressed block. Further compressed blocks can be similarly obtained.

[0017] In one embodiment, the error generated by compression is computed and if the error is greater than a threshold value, one or more values in the plurality of compressed blocks are altered to reduce the computed error. In one embodiment, the color values are encoded as YCrCb. In one embodiment, the Y component is encoded in one compressed block and CrCb endpoints are used to encode another compressed block. In another embodiment, the color values are encoded as RGB values. One of the RGB components are encoded in one compressed block and endpoints in the other two components' color spaces are used to encode another compressed block.

[0018] In one embodiment, the function associated with the creation of one or more compressed blocks is not a grey scale function. Instead, a vector that represents the axis of the function is stored in the compressed blocks. In one embodiment, the vector is of unit length. In another embodiment, the vector is of non-unit length. In one embodiment, the vector is a 16-bit axis vector stored in addition to the endpoints and indices as part of the one or more compressed blocks. In another embodiment, two 8-bit Theta-Phi axis components are stored in addition to the endpoints and indices as part of the one or more compressed blocks. The Theta-Phi axis components represent angle and elevation of the axis vector on a hemisphere. In one embodiment, the number of bits per index is reduced to allow storage of the vector value without increasing the size of a compressed block beyond a desired maximum size. In various other embodiments, more than one compressed blocks are encoded to represent each original block. Specific bit values have been for the purpose of example. One of ordinary skill in the art will recognize that embodiments of the present invention may be practiced with original pixel values of any size, using any color encoding scheme, and generating one or more compressed blocks.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] These and other features, aspects and advantages of the present invention will become better understood with regard to the following description, appended claims and accompanying drawings where:

[0020] **FIG. 1** is a flow diagram of the process of compressing an image in accordance with one embodiment of the present invention.

[0021] **FIG. 2** is a flow diagram of the process of determining a color for a pixel in accordance with one embodiment of the present invention.

[0022] **FIG. 3** is a flow diagram of the process of compressing a pixel block in accordance with one embodiment of the present invention.

[0023] **FIG. 4** is a flow diagram of the process of compressing a 24-bit per pixel, four pixel by four pixel block in accordance with one embodiment of the present invention.

[0024] **FIG. 5** is a flow diagram of the process of compressing a pixel block in accordance with one embodiment of the present invention.

[0025] **FIG. 6** is a flow diagram of the process of compressing a 24-bit per pixel, four pixel by four pixel block in accordance with one embodiment of the present invention.

[0026] **FIG. 7** is a flow diagram of the process of reducing the occurrence of visual artifacts in accordance with one embodiment of the present invention.

[0027] **FIG. 8** is a block diagram of a general purpose computer.

[0028] **FIG. 9** is an optimization method according to one embodiment of the present invention.

[0029] **FIG. 10** is another optimization method according to one embodiment of the present invention.

[0030] **FIG. 11** is another optimization method according to one embodiment of the present invention.

[0031] **FIG. 12** is another optimization method according to one embodiment of the present invention.

[0032] **FIG. 13** is another optimization method according to one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0033] The invention is a method and apparatus for block based image compression with multiple non-uniform block

encodings. In the following description, numerous specific details are set forth to provide a more thorough description of embodiments of the invention. It is apparent, however, to one skilled in the art, that the invention may be practiced without these specific details. In other instances, well known features have not been described in detail so as not to obscure the invention.

[0034] Multiple Blocks

[0035] Embodiments of the present invention are directed to a method and apparatus for block based image compression with multiple non-uniform block encodings. In one embodiment of the present invention, an image is divided into blocks of pixels. In one embodiment the blocks are four pixels by four pixels, but other block sizes are used in other embodiments.

[0036] In one embodiment, a block of pixels in the original image is decompressed into one or more decompressed blocks, each of which represent a partial data value of the original block. According to one embodiment, the decompression is based on applying a predefined transform function to each block. According to another embodiment, the decompression is optimized to reduce the loss of image quality. Each of these decompressed blocks are then compressed using the same or different methods for each block to produce one or more compressed blocks, which are combined to produce an output block. Thus, each block in the original image is represented by a plurality, typically different, compressed blocks each representing all the data values of the block. In one embodiment, color (or a similar value) associated with a pixel is determined by combining the compressed information about the pixel in one compressed block with compressed information about the same pixel in another compressed block. In another embodiment, global information about the image is also combined with the information in one or more compressed blocks to produce a desired value for a pixel.

[0037] **FIG. 1** illustrates the process of compressing an image in accordance with one embodiment of the present invention. At block **100**, an image is divided into a plurality of blocks. At block **110**, each block is decompressed into a plurality of decompressed blocks. At step **120**, each of these decompressed blocks is compressed to form a plurality of compressed blocks. At step **130**, the plurality of compressed blocks are combined to generate an output block.

[0038] Optimization of Decompressed Image Block

[0039] According to one embodiment, the decompression of each original block of the image is optimized so that the loss in image quality is reduced to a desired amount, or meets some threshold. According to another embodiment of the present invention, there are several levels of optimization available depending on the threshold or other factors such as time constraints, quality of output image, etc. It should be noted here, that the optimization methods mentioned below take into account only a limited amount of blocks (2 in these cases), but an unlimited amount of blocks can be used without departing from the scope of the present invention.

[0040] A first optimization method uses a first subset comprising of one or more decompressed blocks and a second subset comprising of one or more decompressed blocks different from the ones in the first subset. This optimization method is illustrated in **FIG. 9**. At block **900**, the first subset is compressed. Compression schemes are explained in more detail below. Next, at block **910**, the compressed first subset is decompressed. Next, at block **920**, the results of the decompression are fed to a decompress device that changes the data value of the second subset. Next, at bock **930**, the second subset is compressed. Blocks **900** through **930** are repeated for every subset of an image until the entire image is compressed.

[0041] A second optimization method uses the first optimization method in addition to a final iteration step. This optimization method is illustrated in **FIG. 10**. Blocks **900** through **930** remain the same and are numbered **1000** through **1030**. Next, at block **1060**, if the optimization has met a certain threshold or some other factor, then we stop, else at block **1040** the compressed second subset is decompressed. Next, at block **1050**, the results of the decompression are fed to a decompress device that changes the data value of the first subset, and blocks **1000**, **1010**, **1020**, **1030**, and **1060** are repeated again.

[0042] A third optimization method uses an original image block in conjunction with a series of functions to generate two new blocks. This optimization method is illustrated in **FIG. 11**. At block **1100**, an original image block is projected onto a predefined vector, and at block **1110** its magnitude is stored in a first new block by a decompress device. Concurrently, at block **1120**, the projected image is subtracted from the original image and the result is stored in a second new block.

[0043] A fourth optimization method uses an original image block in conjunction with a series of functions to generate two new blocks. This optimization method is illustrated in **FIG. 12**. At block **1200**, an original image block is projected onto a predefined vector, and at **1210** its magnitude is stored in a first new block by a decompress device. Next, at block **1220**, the first new block is compressed using any of the compression schemes described below to generate a compressed first block. Next, at block **1230**, the compressed first block is decompressed. Next, at block **1240**, the data value of the decompressed block is subtracted from the data value of the original block to generate a new second block. Next, at block **1250**, new second block is compressed to generate a compressed second block. Next, at block **1260**, compressed first block and compressed second block are combined to generate an output block at block **1270**.

[0044] A fifth optimization method uses an original image block in conjunction with a series of functions to generate two new blocks. This optimization method is illustrated in **FIG. 13**. At block **1300**, an original image block is projected onto a predefined vector, and at **1310** its magnitude is stored in a first new block by a decompress device. Next, at block **1320**, the first new block is compressed using any of the compression schemes described below to generate a compressed first block. Next, at block **1330**, the compressed first block is decompressed. Next, at block **1340**, the data value of the decompressed block is subtracted from the data value of the original block to generate a new second block. Next, at block **1350**, new second block is compressed to generate a compressed second block. Next, at block **1360**, compressed second block is decompressed and its data value subtracted from the data value of the original block and the results are stored in a third new block. Next, at block **1370**,

the data value of the third new block is subtracted from the data value of the original block and the result is stored back in the first new block. At block **1380**, if the desired threshold is met, then the compressed first block is combined with the compressed second block to generate an output block at block **1390**, else blocks **1310-1370** are repeated again.

[**0045**] One Compression Method

[**0046**] **FIG. 2** illustrates the process of determining a color for a pixel in accordance with one embodiment of the present invention. At block **200**, a first value associated with a pixel is decoded from a first compressed block. At block **210**, a second value associated with the same pixel is decoded from a second compressed block. At block **220**, the first and second values are combined to generate an output pixel. In one embodiment, the values are combined by averaging. In another embodiment, the values are combined by appending one to the end of another. In other embodiments, other combining functions are used. It should be noted here that there can be more than two compressed blocks each containing a different value associated with the same pixel, but the general idea of combining the values of each compressed block to generate an output pixel remains the same as explained above.

[**0047**] Another Compression Method

[**0048**] In one embodiment, a first compressed block is created by encoding each pixel value as an index into a color map. In one embodiment, the color map maps pixels to a best fit curve with two end points in a color space. In one embodiment the curve is a straight line. In one embodiment, the end points of the best fit curve are values to which the mapping can map. In another embodiment, the mapping can map to an extrapolated value between two endpoints along the best fit curve. In still another embodiment, the mapping can map to a plurality of different extrapolated values between two endpoints along the best fit curve. In one embodiment, the compressed block is comprised of two points of the best fit curve and index values for each of the pixels. In another embodiment, the two points are the two endpoints.

[**0049**] **FIG. 3** illustrates the process of compressing a decompressed block in accordance with one embodiment of the present invention. At block **300**, a best fit curve with endpoints of a pre-determined size is determined. At block **310**, the endpoints are stored as part of a compressed block. At block **320**, each pixel in the decompressed block is encoded as an index into a group of possible color values. In one embodiment, the possible color values are comprised of the stored endpoints. In another embodiment, the possible color values are comprised of the stored endpoints and one or more interpolated values on the best fit curve. At block **330**, the indices are stored as part of the compressed block.

[**0050**] In an example embodiment, the original block is a four pixel by four pixel block wherein each pixel value is represented by 24 bits (e.g., a 24-bit RGB color value). Two, 16-bit endpoints for a best fit curve are stored. Additionally, sixteen 2-bit indices (one for each pixel in the block) are stored. The indices encode which of the two endpoint colors or two interpolated values along the curve are the compressed color for each pixel. The endpoints and the indices comprise a first compressed block.

[**0051**] **FIG. 4** illustrates the process of compressing a 24-bit per pixel, four pixel by four pixel block in accordance with one embodiment of the present invention. At block **400**, a best fit curve with 16-bit endpoints is determined. At block **410**, the 16-bit endpoints are stored as part of a compressed block. At block **420**, each pixel in the block is encoded as a 2-bit index into a group of possible color values. The possible color values are the two 16-bit endpoints and two interpolated values on the curve. At block **430**, the indices are stored as part of the compressed block. The original block required 384 bits of storage space. The compressed block requires only 64 bits of storage space.

[**0052**] Yet Another Compression Method

[**0053**] In another embodiment, a compressed block is created by applying a function to each of the values of the block to obtain new block values. In one embodiment, the function combines components of each value to reduce the amount of space needed to encode the data. In an example embodiment, 24-bit RGB values are combined to create an 8-bit grey scale value (e.g., (R+G+B)/3). In one embodiment, the new block values are compressed by encoding the block as an index for mapping onto a best fit curve. In one embodiment, two endpoints are stored. In another embodiment, the indices map to either the two points or any of one or more interpolated values along the best fit curve. In one embodiment, the two points are end points. The two stored endpoints and the block of indices comprise a second compressed block.

[**0054**] **FIG. 5** illustrates the process of compressing a decompressed block in accordance with one embodiment of the present invention. At block **500**, a function is applied to each value of the pixel block to obtain a new value. At block **510**, a best fit curve with endpoints of a pre-determined size is determined using the new values. At block **520**, the endpoints are stored as part of a compressed block. At block **530**, each pixel in the block is encoded as an index into a group of possible color values. In one embodiment, the possible color values are comprised of the stored endpoints. In another embodiment, the possible color values are comprised of the stored endpoints and one or more interpolated values on the best fit curve. At block **540**, the indices are stored as part of the compressed block.

[**0055**] In an example embodiment, the original block is a four pixel by four pixel block wherein each pixel value is represented by 24 bits (e.g., a 24-bit RGB color value). The values in the block are converted to 8-bit grey scale values. Two, 8-bit endpoints for a best fit curve are stored. Additionally, sixteen 3-bit indices (one for each pixel in the block) are stored. The indices encode which of the two endpoint colors or six interpolated values along the curve are the compressed color for each pixel. The endpoints and the indices comprise a third compressed block. Further compressed blocks can be similarly obtained.

[**0056**] **FIG. 6** illustrates the process of compressing a 24-bit per pixel, four pixel by four pixel block in accordance with one embodiment of the present invention. At block **600**, each 24-bit pixel value is converted to an 8-bit value by applying a function. In one embodiment, the 24-bit pixel value is an RGB value and the function applied is the average of the RGB values. At block **610**, the new values are placed in a new pixel block. At block **620**, a best fit curve with 8-bit endpoints is determined. At block **630**, the 8-bit endpoints are stored as part of a compressed block. At block **640**, each pixel in the block is encoded as a 3-bit index into

a group of possible color values. The possible color values are the two 8-bit endpoints and six interpolated values on the curve. At block **650**, the indices are stored as part of the compressed block. The original block required 384 bits of storage space. The compressed block requires only 64 bits of storage space.

[**0057**] Error Reduction

[**0058**] In one embodiment, the error generated by compression is computed. If the error is greater than a threshold value, one or more values of one or more compressed blocks are altered to reduce the computed error. Thus, the problem of visual artifacts is reduced. **FIG. 7** illustrates the process of reducing the occurrence of visual artifacts in accordance with one embodiment of the present invention. At block **700**, an image is divided into a plurality of pixel blocks. At block **710**, each pixel block is decompressed into two decompressed blocks, viz. decompressed first block and decompressed second block. At step **720**, decompressed first block is compressed using a first compression method. At block **730**, decompressed second block is compressed using a second compression method. At block **740**, it is determined whether the error generated by the first and second compression methods is greater than a threshold value. If the error generated is not greater than the threshold value, at block **750**, the encoding is acceptable. If the error generated is greater than the threshold value, at block **760**, the values in the compressed blocks are altered to reduce the error. It should be noted here that **FIG. 7** is by way of an example that uses only two decompressions of an original pixel block, but any number of decompressions are possible depending on the quality of compression desired, memory constraints of the computer, or other factors without departing from the scope of the present invention.

[**0059**] Alternative Color Encodings

[**0060**] In one embodiment, the color values are encoded as YCrCb. In one embodiment, the Y component is encoded in one compressed block and CrCb endpoints are used to encode another compressed block. In another embodiment, the color values are encoded as RGB values. One of the RGB components are encoded in one compressed block and endpoints in the other two components' color spaces are used to encode another compressed block.

[**0061**] In one embodiment, the function associated with creation of one or more compressed block is not a grey scale function. Instead, a vector that represents the axis of the function is stored in the compressed block. In one embodiment, the vector is of unit length. In another embodiment, the vector is of non-unit length. In one embodiment, the vector is a 16-bit axis vector stored in addition to the endpoints and indices as part of the compressed block. In another embodiment, two 8-bit Theta-Phi axis components are stored in addition to the endpoints and indices as part of the one or more compressed blocks. The Theta-Phi axis components represent angle and elevation of the axis vector on a hemisphere. In one embodiment, the number of bits per index is reduced to allow storage of the vector value without increasing the size of a compressed block beyond a desired maximum size. In various other embodiments, more than one compressed blocks are encoded to represent each original block. Specific bit values have been for the purpose of example. One of ordinary skill in the art will recognize that embodiments of the present invention may be practiced with original pixel values of any size, using any color encoding scheme, and generating one or more compressed blocks.

[**0062**] Embodiment of Computer Execution Environment (Hardware)

[**0063**] An embodiment of the invention can be implemented as computer software in the form of computer readable program code executed in a general purpose computing environment such as environment **800** illustrated in **FIG. 8**. A keyboard **810** and mouse **811** are coupled to a system bus **818**. The keyboard and mouse are for introducing user input to the computer system and communicating that user input to central processing unit (CPU) **813**. Other suitable input devices may be used in addition to, or in place of, the mouse **811** and keyboard **810**. I/O (input/output) unit **819** coupled to bi-directional system bus **818** represents such I/O elements as a printer, ANV (audio/video) I/O, etc.

[**0064**] Computer **801** may include a communication interface **820** coupled to bus **818**. Communication interface **820** provides a two-way data communication coupling via a network link **821** to a local network **822**. For example, if communication interface **820** is an integrated services digital network (ISDN) card or a modem, communication interface **820** provides a data communication connection to the corresponding type of telephone line, which comprises part of network link **821**. If communication interface **820** is a local area network (LAN) card, communication interface **820** provides a data communication connection via network link **821** to a compatible LAN. Wireless links are also possible. In any such implementation, communication interface **820** sends and receives electrical, electromagnetic or optical signals which carry digital data streams representing various types of information.

[**0065**] Network link **821** typically provides data communication through one or more networks to other data devices. For example, network link **821** may provide a connection through local network **822** to local server computer **823** or to data equipment operated by ISP **824**. ISP **824** in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet"**825**. Local network **822** and Internet **825** both use electrical, electromagnetic or optical signals which carry digital data streams. The signals through the various networks and the signals on network link **821** and through communication interface **820**, which carry the digital data to and from computer **800**, are exemplary forms of carrier waves transporting the information.

[**0066**] Processor **813** may reside wholly on client computer **801** or wholly on server **826** or processor **813** may have its computational power distributed between computer **801** and server **826**. Server **826** symbolically is represented in **FIG. 8** as one unit, but server **826** can also be distributed between multiple "tiers". In one embodiment, server **826** comprises a middle and back tier where application logic executes in the middle tier and persistent data is obtained in the back tier. In the case where processor **813** resides wholly on server **826**, the results of the computations performed by processor **813** are transmitted to computer **801** via Internet **825**, Internet Service Provider (ISP) **824**, local network **822** and communication interface **820**. In this way, computer **801** is able to display the results of the computation to a user in the form of output.

[**0067**] Computer **801** includes a video memory **814**, main memory **815** and mass storage **812**, all coupled to bi-

Realtek Ex. 1016
Case No. IPR2023-00688
Page 19 of 23

directional system bus **818** along with keyboard **810**, mouse **811** and processor **813**. As with processor **813**, in various computing environments, main memory **815** and mass storage **812**, can reside wholly on server **826** or computer **801**, or they may be distributed between the two.

[**0068**] The mass storage **812** may include both fixed and removable media, such as magnetic, optical or magnetic optical storage systems or any other available mass storage technology. Bus **818** may contain, for example, thirty-two address lines for addressing video memory **814** or main memory **815**. The system bus **818** also includes, for example, a 32-bit data bus for transferring data between and among the components, such as processor **813**, main memory **815**, video memory **814** and mass storage **812**. Alternatively, multiplex data/address lines may be used instead of separate data and address lines.

[**0069**] In one embodiment of the invention, the microprocessor is manufactured by Intel, such as the 80×86 or Pentium-typed processor. However, any other suitable microprocessor or microcomputer may be utilized. Main memory **815** is comprised of dynamic random access memory (DRAM). Video memory **814** is a dual-ported video random access memory. One port of the video memory **814** is coupled to video amplifier **816**. The video amplifier **816** is used to drive the cathode ray tube (CRT) raster monitor **817**. Video amplifier **816** is well known in the art and may be implemented by any suitable apparatus. This circuitry converts pixel data stored in video memory **814** to a raster signal suitable for use by monitor **817**. Monitor **817** is a type of monitor suitable for displaying graphic images.

[**0070**] Computer **801** can send messages and receive data, including program code, through the network(s), network link **821**, and communication interface **820**. In the Internet example, remote server computer **826** might transmit a requested code for an application program through Internet **825**, ISP **824**, local network **822** and communication interface **820**. The received code may be executed by processor **813** as it is received, and/or stored in mass storage **812**, or other non-volatile storage for later execution. In this manner, computer **800** may obtain application code in the form of a carrier wave. Alternatively, remote server computer **826** may execute applications using processor **813**, and utilize mass storage **812**, and/or video memory **815**. The results of the execution at server **826** are then transmitted through Internet **825**, ISP **824**, local network **822** and communication interface **820**. In this example, computer **801** performs only input and output functions.

[**0071**] Application code may be embodied in any form of computer program product. A computer program product comprises a medium configured to store or transport computer readable code, or in which computer readable code may be embedded. Some examples of computer program products are CD-ROM disks, ROM cards, floppy disks, magnetic tapes, computer hard drives, servers on a network, and carrier waves.

[**0072**] The computer systems described above are for purposes of example only. An embodiment of the invention may be implemented in any type of computer system or programming or processing environment.

[**0073**] Thus, a method and apparatus for block based image compression with multiple non-uniform block encodings is described in conjunction with one or more specific embodiments. The invention is defined by the following claims and their full scope and equivalents.

We claim:

1. A method for compressing an image comprising:

partitioning said image into a plurality of pixel blocks;

decompressing each of said plurality of pixel blocks to generate a plurality of decompress blocks;

compressing each of said plurality of decompress blocks to generate a plurality of compressed blocks; and

combining each of said plurality of compressed blocks to generate an output block.

2. The method of claim 1 wherein said image is a texture.

3. The method of claim 1 wherein said first pixel block is four pixels by four pixels.

4. The method of claim 1 wherein said first pixel block has 24 bits for each pixel.

5. The method of claim 1 wherein said compressing uses one or more compression techniques.

6. The method of claim 1 wherein said step of compressing each of said plurality of decompress blocks comprises:

determining a best fit curve in a color space;

storing two points of said best fit curve; and

storing an index value for a first pixel value in said first pixel block wherein said index maps to a collection of possible colors.

7. The method of claim 6 wherein said collection of possible colors comprises:

colors of said two points.

8. The method of claim 6 wherein said collection of possible colors comprises:

an interpolated value wherein said interpolated value is on said best fit curve.

9. The method of claim 6 wherein said two points are 16-bit values.

10. The method of claim 6 wherein said index value is a 2-bit value.

11. The method of claim 6 wherein said two points are endpoints of said best fit curve.

12. The method of claim 6 wherein said first pixel value is an RGB value.

13. The method of claim 6 wherein said first pixel value is a YCrCb value.

14. The method of claim 1 wherein said step of compressing each of said plurality of decompress blocks comprises:

applying a function to a first pixel value to produce an altered first pixel value;

determining a best fit curve in a color space;

storing two points of said best fit curve; and

storing an index value for said altered first pixel value in said first pixel block wherein said index maps to a collection of possible colors.

15. The method of claim 14 wherein said collection of possible colors comprises:

colors of said two points.

16. The method of claim 14 wherein said collection of possible colors comprises:

an interpolated value wherein said interpolated value is on said best fit curve.

17. The method of claim 14 wherein said two points are 8-bit values.

18. The method of claim 14 wherein said index value is a 3-bit value.

19. The method of claim 14 wherein said two points are endpoints of said best fit curve.

20. The method of claim 14 wherein said first pixel value has more bits than said altered pixel value.

21. The method of claim 14 wherein said function is an averaging function.

22. The method of claim 21 wherein said function averages a red component, a green component and a blue component.

23. The method of claim 14 wherein said first pixel value is an RGB value.

24. The method of claim 14 wherein said first pixel value is a YCrCb value.

25. The method of claim 1 further comprising:

storing a set of global data for use in decompressing said image.

26. The method of claim 1 wherein said decompressing is optimized for each of said plurality of pixel blocks to reduce loss of image quality.

27. The method of claim 26 wherein optimization of each of said plurality of pixel blocks further comprises:

compressing a first subset comprising of one or more decompress blocks to generate a first compressed subset;

decompressing said first compressed subset;

feeding results of decompressing of first compressed subset to a decompress device that changes the data value of a second subset comprising of one or more decompressed blocks different from first subset; and

compressing second subset.

28. The method of claim 26 wherein optimization of each of said plurality of pixel blocks further comprises:

compressing a first subset comprising of one or more decompressed blocks to generate a first compressed subset;

decompressing said first compressed subset;

feeding results of decompressing of first compressed subset to a decompress device that changes the data value of a second subset comprising of one or more decompressed blocks different from first subset to generate a new second subset;

compressing said new second subset;

decompressing said new second subset;

feeding results of decompressing of new second subset to said decompress device that changes the data value of said first subset; and

repeating steps of compressing, decompressing, and feeding of said first subset and repeating steps of compressing, decompressing, and feeding of said new second subset until a threshold is achieved.

29. The method of claim 26 wherein optimization of each of said plurality of pixel blocks further comprises:

projecting an original block onto a predefined vector to generate a projected;

concurrently storing a magnitude of said original block in a first new block and subtracting said projected block from said original block.

30. The method of claim 26 wherein optimization of each of said plurality of pixel blocks further comprises:

projecting an original block onto a predefined vector to generate a projected;

storing a magnitude of said original block in a first new block;

compressing said first new block to generate a compressed first block;

decompressing said compressed first block to generate a decompressed block;

subtracting data value of said decompressed block from data value of said original block to generate a new second block;

compressing said new second block to generate a compressed second block; and

combining said compressed first block and said compressed second block.

31. The method of claim 26 wherein optimization of each of said plurality of pixel blocks further comprises:

projecting an original block onto a predefined vector to generate a projected;

storing a magnitude of said original block in a first new block;

compressing said first new block to generate a compressed first block;

decompressing said compressed first block to generate a first decompressed block;

subtracting data value of said first decompressed block from data value of said original block to generate a new second block;

compressing said new second block to generate a compressed second block;

decompressing said compressed second block to generate a second decompressed block;

subtracting data value of said second decompressed block from data value of said original block and storing results in a third block;

subtracting data value of said third block from data value of said original block and storing the results in said first new block;

repeating steps of storing, compressing, decompressing, and subtracting of said first new block, repeating steps of compressing, decompressing, and subtracting of said new second block, and repeating steps of subtracting of said third block until a threshold is achieved; and

combining said compressed first block with said compressed second block.

32. An image compression system comprising:

a partitioner configured to partition an image into a plurality of pixel blocks;

a decompression unit configured to decompress each of said plurality of pixel blocks to generate a plurality of decompress blocks;

a compression unit configured to compress each of said plurality of decompress blocks to generate a plurality of compressed blocks; and

a combining unit configured to combine each of said plurality of compressed blocks to generate an output block.

33. The image compression system of claim 32 wherein said image is a texture.

34. The image compression system of claim 32 wherein said first pixel block is four pixels by four pixels.

35. The image compression system of claim 32 wherein said first pixel block has 24 bits for each pixel.

36. The image compression system of claim 32 wherein said compression unit uses one or more compression techniques.

37. The image compression system of claim 32 wherein said first compression unit comprises:

a determiner configured to determine a best fit curve in a color space; and

a storage device configured to store two points of said best fit curve, wherein said storage device is further configured to store an index value for a first pixel value in said first pixel block wherein said index maps to a collection of possible colors.

38. The image compression system of claim 37 wherein said collection of possible colors comprises:

colors of said two points.

39. The image compression system of claim 29 wherein said collection of possible colors comprises:

an interpolated value wherein said interpolated value is on said best fit curve.

40. The image compression system of claim 37 wherein said two points are 16-bit values.

41. The image compression system of claim 37 wherein said index value is a 2-bit value.

42. The image compression system of claim 37 wherein said two points are endpoints of said best fit curve.

43. The image compression system of claim 37 wherein said first pixel value is an RGB value.

44. The image compression system of claim 37 wherein said first pixel value is a YCrCb value.

45. The image compression system of claim 32 wherein said compression unit comprises:

a calculator configured to apply a function to a first pixel value to produce an altered first pixel value;

a determiner configured to determine a best fit curve in a color space; and

a storage unit configured to store two points of said best fit curve, wherein said storage unit is further configured to store an index value for said altered first pixel value in said first pixel block wherein said index maps to a collection of possible colors.

46. The image compression system of claim 45 wherein said collection of possible colors comprises:

colors of said two points.

47. The image compression system of claim 45 wherein said collection of possible colors comprises:

an interpolated value wherein said interpolated value is on said best fit curve.

48. The image compression system of claim 45 wherein said two points are 8-bit values.

49. The image compression system of claim 45 wherein said index value is a 3-bit value.

50. The image compression system of claim 45 wherein said two points are endpoints of said best fit curve.

51. The image compression system of claim 45 wherein said first pixel value has more bits than said altered pixel value.

52. The image compression system of claim 45 wherein said function is an averaging function.

53. The image compression system of claim 52 wherein said function averages a red component, a green component and a blue component.

54. The image compression system of claim 45 wherein said first pixel value is an RGB value.

55. The image compression system of claim 45 wherein said first pixel value is a YCrCb value.

56. The image compression system of claim 32 further comprising:

a second storage device configured to store a set of global data for use in decompressing said image.

57. The image compression system of claim 32 wherein said decompressing unit is optimized for each of said plurality of pixel blocks to reduce loss of image quality.

58. The image compression system of claim 57 wherein optimization of each of said plurality of pixel blocks further comprises:

a first compressing unit configured to compress a first subset comprising of one or more decompress blocks to generate a first compressed subset;

a decompressing unit configured to decompress said first compressed subset;

a feeding unit configured to feed results of decompression of first compressed subset to a decompress device that changes the data value of a second subset comprising of one or more decompressed blocks different from first subset; and

a second compressing unit configured to compress second subset.

59. The image compression system of claim 57 wherein optimization of each of said plurality of pixel blocks further comprises:

a first compression unit configured to compress a first subset comprising of one or more decompressed blocks to generate a first compressed subset;

a first decompressing unit configured to decompress said first compressed subset;

a first feeding unit configured to feed results of decompressing of first compressed subset to a decompress device that changes the data value of a second subset

comprising of one or more decompressed blocks different from first subset to generate a new second subset;

a second compression unit configured to compress said new second subset;

a second decompressing unit configured to decompress said new second subset;

a second feeding unit configured to feed results of decompressing of new second subset to said decompress device that changes the data value of said first subset; and

a repeating unit configured to repeat steps of compress, decompress, and feed said first subset and repeating steps of compress, decompress, and feed said new second subset until a threshold is achieved.

60. The image compression system of claim 57 wherein optimization of each of said plurality of pixel blocks further comprises:

a projection unit configured to project an original block onto a predefined vector to generate a projected;

a device configured to concurrently store a magnitude of said original block in a first new block and subtract said projected block from said original block.

61. The image compression system of claim 57 wherein optimization of each of said plurality of pixel blocks further comprises:

a projection unit configure to project an original block onto a predefined vector to generate a projected;

a storing unit configured to store a magnitude of said original block in a first new block;

a first compression unit configured to compress said first new block to generate a compressed first block;

a decompression unit configured to decompress said compressed first block to generate a decompressed block;

a subtracting unit configured to subtract data value of said decompressed block from data value of said original block to generate a new second block;

a second compression unit configured to compress said new second block to generate a compressed second block; and

a combining unit configured to combine said compressed first block and said compressed second block.

62. The image compression unit of claim 57 wherein optimization of each of said plurality of pixel blocks further comprises:

a projection unit configured to project an original block onto a predefined vector to generate a projected;

a storing unit configured to store a magnitude of said original block in a first new block;

a first compressing unit configured to compress said first new block to generate a compressed first block;

a first decompressing unit configured to decompress said compressed first block to generate a first decompressed block;

a first subtracting unit configured to subtract data value of said first decompressed block from data value of said original block to generate a new second block;

a second compressing unit configured to compress said new second block to generate a compressed second block;

a second decompressing unit configured to decompress said compressed second block to generate a second decompressed block;

a second subtracting unit configured to subtract data value of said second decompressed block from data value of said original block and storing results in a third block;

a third subtracting unit configured to subtract data value of said third block from data value of said original block and storing the results in said first new block;

a repeating unit configured to repeat steps to store, compress, decompress, and subtract of said first new block, repeat steps to compress, decompress, and subtract of said new second block, and repeat steps to subtract said third block until a threshold is achieved; and

a combining unit configured to combine said compressed first block with said compressed second block.

* * * * *