

-1-

BIT-ACCURATE FILM GRAIN SIMULATION METHOD BASED ON PRE-COMPUTED TRANSFORMED COEFFICIENTS

CROSS-REFERENCE TO RELATED APPLICATIONS

5

This application claims priority under 15 U.S.C. 119(e) to U.S. Provisional Patent Application Serial No 60/628,837, filed November 17, 2004, the teachings of which are incorporated herein.

10 TECHNICAL FIELD

This invention relates to a technique for simulating film grain in an image.

BACKGROUND OF THE INVENTION

15

Motion picture films comprise silver-halide crystals dispersed in an emulsion, coated in thin layers on a film base. The exposure and development of these crystals form the photographic image consisting of discrete tiny particles of silver. In color negatives, the silver undergoes chemical removal after development and tiny blobs of dye occur on the sites where the silver crystals form. These small specks of dye are commonly called 'grain' in color film. Grain appears randomly distributed on the resulting image because of the random formation of silver crystals on the original emulsion. Within a uniformly exposed area, some crystals develop after exposure while others do not.

Grain varies in sizes and shapes. The faster the film, the larger the clumps of silver formed and blobs of dye generated, and the more they tend to group together in random patterns. The grain pattern is typically known as 'granularity'. The naked eye cannot distinguish individual grains, which vary from 0.0002 mm to about 0.002 mm. Instead, the eye resolves groups of grains, referred to as blobs. A viewer identifies these groups of blobs as film grain. As the image resolution becomes larger, the perception of the film grain becomes higher. Film grain becomes clearly noticeable on cinema and high-definition images, whereas film grain progressively loses importance in SDTV and becomes imperceptible in smaller formats.

-2-

Motion picture film typically contains image-dependent noise resulting either from the physical process of exposure and development of the photographic film or from the subsequent editing of the images. The photographic film possesses a characteristic quasi-random pattern, or texture, resulting from physical granularity of the photographic emulsion.

5 Alternatively, a similar pattern can be simulated over computed-generated images in order to blend them with photographic film. In both cases, this image-dependent noise is referred to as grain. Quite often, moderate grain texture presents a desirable feature in motion pictures. In some instances, the film grain provides visual cues that facilitate the correct perception of two-dimensional pictures. Film grain is often varied within a single film to provide various
10 clues as to time reference, point of view, etc. Many other technical and artistic uses exist for controlling grain texture in the motion picture industry. Therefore, preserving the grainy appearance of images throughout image processing and delivery chain has become a requirement in the motion picture industry.

Several commercially available products have the capability of simulating film grain,
15 often for blending a computer-generated object into a natural scene. Cineon® from Eastman Kodak Co, Rochester New York, one of the first digital film applications to implement grain simulation, produces very realistic results for many grain types. However, the Cineon® application does not yield good performance for many high-speed films because of the noticeable diagonal stripes the application produces for high grain size settings. Further, the
20 Cineon® application fails to simulate grain with adequate fidelity when images are subject to previous processing, for example, such as when the images are copied or digitally processed.

Another commercial product that simulates film grain is *Grain Surgery*™ from Visual Infinity Inc., which is used as a plug-in of Adobe® After Effects®. The *Grain Surgery*™ product appears to generate synthetic grain by filtering a set of random numbers. This
25 approach suffers from disadvantage of a high computational complexity.

None of these past schemes solves the problem of restoring film grain in compressed video. Film grain constitutes a high frequency quasi-random phenomenon that typically cannot undergo compression using conventional spatial and temporal methods that take advantage of redundancies in the video sequences. Attempts to process film-originated
30 images using MPEG-2 or ITU-T Rec. H.264 | ISO/IEC 14496-10 compression techniques usually either result in an unacceptably low degree of compression or complete loss of the grain texture.

Thus, there exists a need for a technique simulating film grain, especially a technique that affords relatively low complexity.

BRIEF SUMMARY OF THE INVENTION

5

In example embodiments, there is provided a method for simulating a film grain pattern. The method begins by obtaining a set of bit-accurate transformed coefficients. The set of bit-accurate transformed coefficients then undergoes filtering. Thereafter, the filtered set of bit-accurate transformed coefficients undergoes a bit-accurate inverse transform to yield
10 a film grain pattern.

In a first aspect of the invention, there is provided a method of creating a pattern of film grain, comprising the steps of: (a) establishing a set of bit-accurate transformed coefficients; (b) frequency filtering set of bit-accurate transformed coefficients; and (c) performing a bit-accurate inverse transformation on the frequency filtered transformed
15 coefficients to yield a film grain pattern.

In a second aspect of the invention, there is provided an apparatus comprising one of a processor and dedicated logic circuit for creating a pattern of film grain by (a) establishing a set of bit-accurate transformed coefficients; (b) frequency filtering set of bit-accurate transformed coefficients; and (c) performing a bit-accurate transformation on the frequency
20 filtered transformed coefficients to yield a film grain pattern.

In a third aspect of the invention, there is provided an apparatus for creating a pattern of film grain, comprising: means for establishing a set of bit-accurate transformed coefficients; means for frequency filtering set of bit-accurate transformed coefficients; and means for performing a bit-accurate transformation on the frequency filtered transformed
25 coefficients to yield a film grain pattern.

DETAILED DESCRIPTION OF THE DRAWINGS

FIGURE 1 depicts a block schematic diagram of the combination of a transmitter and
30 receiver in a film grain processing chain useful for practicing the technique of the present principles;

FIGURE 2 depicts, in flow chart form, the steps of a method for creating a bit-accurate film grain pattern using a Gaussian random number generator;

FIGURE 3 depicts, in flow chart form, the steps of a method for creating a bit-accurate film grain pattern using a Gaussian random number look up table;

5 FIGURE 4 depicts, in flow chart form, the steps of a method for creating a bit-accurate film grain pattern creation using a single image of Gaussian noise;

FIGURE 5 depicts, in flow chart form, the steps of a method for creating a bit-accurate film grain pattern creation using a set of pre-computed Discrete Cosine Transformed (DCT) coefficients of a single image of Gaussian noise; and

10 FIGURE 6 depicts, in flow chart form, the steps of a method for creating a bit-accurate film grain pattern using pre-computed DCT coefficients of several images of Gaussian noise.

DETAILED DESCRIPTION

15 To understand the technique of the present principles for creating a bit-accurate film grain pattern, a brief overview of film grain simulation will prove helpful. FIGURE 1 depicts

-4-

a block schematic diagram of a transmitter 10, which receives an input video signal and, in turn, generates a compressed video stream at its output. In addition, the transmitter 10 also generates information indicative of the film grain (if any) present in the sample. In practice, the transmitter 10 could comprises part of a head-end array of a cable television system, or
5 other such system that distributes compressed video to one or more downstream receivers 11, only one of which is shown in FIG. 1. The transmitter 10 could also take the form of encoder that presents media like DVDs. The receiver 11 decodes the coded video stream and simulates film grain in accordance with the film grain information and decoded video, both received from the transmitter 10 or directly from the media itself in the case of a DVD or the
10 like, to yield an output video stream that has simulated film grain. The receiver 11 can take the form of a set-top box or other such mechanism that serves to decode compressed video and simulate film grain in that video.

The overall management of film grain requires the transmitter 10 (i.e., the encoder) provide information with respect to the film grain in the incoming video. In other words, the
15 transmitter 10 "models" the film grain. Further the receiver 11 (i.e., decoder) simulates the film grain according to the film grain information received from the transmitter 10. The transmitter 10 enhances the quality of the compressed video by enabling the receiver 11 to simulate film grain in the video signal when difficulty exists in retaining the film grain during the video coding process.

20 In the illustrated embodiment of FIG. 1, the transmitter 10 includes a video encoder 12 which encodes the video stream using any of the well known video compression techniques such as the ITU-T Rec. H.264 | ISO/IEC 14496-10 video compression standard. Optionally, a film grain remover 14, in the form of a filter or the like depicted in dashed lines in FIG. 1, could exist upstream of the encoder 12 to remove any film grain in the incoming video stream
25 prior to encoding. To the extent that the incoming video contains no film grain, no need would exist for the film grain remover 14.

A film grain modeler 16 accepts the input video stream, as well as the output signal of the film grain remover 14 (when present). Using such input information, the film grain modeler 16 establishes the film grain in the incoming video signal. In its simplest form, the
30 film grain modeler 16 could comprise a look up table containing film grain models for different film stocks. Information in the incoming video signal would specify the particular film stock originally used to record the image prior to conversion into a video signal, thus

-5-

allowing the film grain modeler 16 to select the appropriate film grain model for such film stock. Alternatively, the film grain modeler 16 could comprise a processor or dedicated logic circuit that would execute one or more algorithms to sample the incoming video and determine the film grain pattern that is present.

5 The receiver 11 typically includes a video decoder 18 that serves to decode the compressed video stream received from the transmitter 10. The structure of the decoder 18 will depend on the type of compression performed by the encoder 12 within the transmitter 10. Thus, for example, the use within the transmitter 10 of an encoder 12 that employs the ITU-T Rec. H.264 | ISO/IEC 14496-10 video compression standard to compress outgoing
10 video will dictate the need for an H.264-compliant decoder 18. Within the receiver 11, a film grain simulator 20 receives the film grain information from the film grain model 16. The film grain simulator 20 can take the form of a programmed processor, or dedicated logic circuit having the capability of simulating film grain for combination via a combiner 22 with the decoded video stream.

15 Film grain simulation aims to synthesize film grain samples that simulate the look of the original film content. As described, film grain modeling occurs at the transmitter 10 of FIG. 1, whereas film grain simulation occurs at the receiver 11. In particular, film grain simulation occurs in the receiver 11 along with the decoding of the incoming video stream from the transmitter 10 upstream of the output of the decoded video stream. Note that the
20 decoding process that occurs in the receiver 11 makes no use of images with added film grain. Rather, film grain simulation constitutes a post-processing method for synthesizing simulated film grain in the decoded images for display. For that reason, the ITU-T Rec. H.264 | ISO/IEC 14496-10 video compression standard contains no specifications regarding the film grain simulation process. However, film grain simulation requires information concerning the
25 grain pattern in the incoming video signal, which information typically undergoes transmission in a Supplemental Enhancement Information (SEI) message when using the ITU-T Rec. H.264 | ISO/IEC 14496-10 video compression standard as specified by the Amendment 1 (Fidelity Range Extensions) of that compression standard.

30 In accordance with the present principles, disclosed herein are several implementations of a method for creating a bit-accurate film grain pattern, such as for storage in a database for film grain simulation in HD DVD systems. Such a film grain pattern database enables film grain simulation in accordance with the parameters in a film grain Supplemental Enhancement

-6-

Information (SEI) message according to the H.264 | MPEG-4 AVC standard as discussed above. More specifically, devices and systems that implement the existing or future HD DVD specifications can employ such a film grain pattern database to store samples of different film grain types, each characterized by different frequency model parameters.

5 As discussed in detail below, the method of the present principles for creating a bit-accurate pattern of film grain makes use of a basic strategy that enables implementation in several different ways within the receiver 11. The basic strategy for creating a bit-accurate film grain pattern begins by establishing a set of integer transformed coefficients, typically, although not necessarily, Discrete Cosine Transformed (DCT) coefficients, typically $N \times N$ in
10 size where N is an integer. The term "image" will some times refer to such a set of integer transformed coefficients. The step of establishing a set or image of integer transformed coefficients can occur in several different ways. For example, the establishing step could entail the processor or discrete logic circuitry in the receiver 11 of FIG. 1 accessing a database of integer-transformed coefficients as described hereinafter. Alternatively, the processor or
15 discrete logic circuitry in the receiver 11 could establish a set or image of Gaussian random numbers, by either reading a database of such numbers or generating them directly. Thereafter, an integer DCT would be performed on the Gaussian random number image to yield a set of integer DCT coefficients.

The next step in the basic process of creating a bit-accurate film grain pattern entails
20 frequency filtering the integer-transformed coefficients according to a desired film grain size and shape. For example, the frequency filtering could employ a predefined set of cut frequencies f_{HL} , f_{VL} , f_{HH} , and f_{VH} that represent cut-off frequencies (in two dimensions) of a filter that characterizes the desired film grain pattern. Following the frequency filtering, the integer transformed coefficients undergo an inverse transform to create the bit-accurate film
25 grain pattern for subsequent storage in a database. In some instances, scaling could occur following the inverse transform of the coefficients.

In practice, the film grain pattern creation method of the present principles makes use of integer DCT and inverse DCT operations, thus assuring bit accuracy which proves useful for verification and testing purposes. However, different implementations of the basic
30 strategy as described below do not necessarily afford bit accuracy between each other. Choosing a particular implementation will depend on the desired tradeoff between memory requirements and computational cost.

-7-

FIGURE 2 illustrates a first implementation of a method in accordance with the present principles for creating a bit-accurate pattern of film grain. The method of FIG. 2 commences upon execution of the start step 100 during which initialization occurs, although such initialization need not necessarily happen. Next step 101 occurs, initiating entry into a loop that repeats for all possible film grain sizes and shapes. Upon entry into the loop, step 102 occurs to achieve creation of an image of random Gaussian values. In practice, the Gaussian random image has a size of $N \times N$ values. In the method of FIG. 2, creation of the image of Gaussian random values occurs by generating such values via a Gaussian random number generator 103.

The image of Gaussian random values created during step 102 then undergoes a bit-accurate transform, typically by way of an integer Discrete Cosine Transform (DCT) operation during step 104, although other bit-accurate transform techniques exist. The integer DCT coefficients established during step 102 then undergo frequency filtering during step 106. Typically, the frequency filter occurs using a predefined set of cut frequencies f_{HL} , f_{VL} , f_{HH} , and f_{VH} that represent cut-off frequencies (in two dimensions) of a filter that characterizes the desired film grain pattern. Next step 108 occurs during which the frequency filtered, transformed coefficients undergo an inverse bit-accurate transformation, typically, although not necessarily, an integer Inverse Discrete Cosine Transformation (IDCT), to yield a bit-accurate film grain pattern. Under some circumstances, scaling of the inversely transformed block of coefficient following step 108 will prove useful to vary the intensity of the bit-accurate film grain pattern.

Following step 108, step 110 occurs and the bit-accurate film grain that results from the integer IDCT transformation (and scaling if performed) undergoes storage in a database 111. Steps 102-110 undergo repeating during execution of the loop initiated during step 101. The loop continues for all possible film grain sizes and shapes, whereupon loop execution ends during step 112, after which the method ends during step 114.

To achieve bit accuracy, the method of FIG. 2 requires a bit-accurate Gaussian random number generator and bit-accurate DCT and Inverse DCT transform operations. Achieving bit accuracy among different implementations of the present method requires the use of a bit-accurate Gaussian random number generator. Which particular Gaussian random number generator is used remains unimportant as long as all implementations use the same generator. As example, one could use the integer approximation of a bit-accurate Gaussian

-8-

generator described in the publication "Numerical Recipes in C: The Art of Scientific Computing" (ISBN 0-521-41108-5), which is based on the Box-Muller transformation.

In addition to a bit-accurate Gaussian random number generator, a bit-accurate implementation of a Discrete Cosine Transform remains necessary to provide bit accuracy for the whole system. The illustrated embodiments described herein make use of an integer approximation of the DCT. The use of any integer approximation remains possible if the result obtained by the integer transformation lies reasonably close to the result obtained by the floating point DCT. This requirement ensures that the filtering performed in the transformed domain constitutes frequency filtering.

In the illustrated embodiment, an integer version of the DCT occurs by scaling the floating point version of the DCT. The $N \times N$ DCT matrix can be computed as:

```

for ( i = 0; i < N; j++)
    for ( j = 0; j < N; j++)
15 {
    if (i == 0) k = sqrt(1/N);
    else      k = sqrt(2/N);

    C(i,j) = k * cos( ((2*j+1)*i*PI) / (2*N));
20 }

```

where $C(i,j)$ represents the coefficient of the transformation matrix at row j and column i . Then, the integer matrix is computed as:

```

25 for ( i = 0; i < N; j++)
    for ( j = 0; j < N; j++)
    {
        Cint(i,j) = round( C(i,j) * scaling_factor)
    }

```

where $\text{round}(x)$ returns the nearest integer approximation of x , and the `scaling_factor` is a positive integer value. The value of the scaling factor determines the quality of the integer approximation of the DCT (the larger the scaling factor, the better the approximation) as well as the number of bits required to compute the transform (the smaller the scaling factor, the lesser bits are required).

Once the integer approximation of the DCT has been defined, the DCT transform is computed as follows:

-9-

$$B = ((\text{Cint}^T * b * \text{Cint}) + 2^{\text{scaling_factor}}) \gg 2 * \text{scaling_factor}$$

where Cint^T denotes the transposed version of the transformation matrix. Analogously, the
 5. inverse transform is computed as follows:

$$b = ((\text{Cint} * B * \text{Cint}^T) + 2^{\text{scaling_factor}}) \gg 2 * \text{scaling_factor}$$

In a particular embodiment where an 8x8 DCT is used, the integer approximation would be:

10

$$\text{Cint}_8^T = \begin{pmatrix} 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 8 & 7 & 4 & 2 & -2 & -4 & -7 & -8 \\ 7 & 3 & -3 & -7 & -7 & -3 & 3 & 7 \\ 7 & -2 & -8 & -4 & 4 & 8 & 2 & -7 \\ 6 & -6 & -6 & 6 & 6 & -6 & -6 & 6 \\ 4 & -8 & 2 & 7 & -7 & -2 & 8 & -4 \\ 3 & -7 & 7 & -3 & -3 & 7 & -7 & 3 \\ 2 & -4 & 7 & -8 & 8 & -7 & 4 & -2 \end{pmatrix}$$

with scale factor equal to 16.

FIGURE 3 illustrates a second implementation of a method in accordance with the
 15 present principles for creating a bit-accurate pattern of film grain. The method of FIG. 3
 commences upon execution of the start step 200 during which initialization occurs, although
 such initialization need not necessarily happen. Next step 201 occurs, initiating entry into a
 loop that repeats for all possible film grain sizes and shapes. Upon entry into the loop, step
 202 executes to create a set or image of random Gaussian values. In practice, the Gaussian
 20 random image has a size of N x N values. Creation of the image of Gaussian random values
 occurs during execution of the method of FIG. 3 by reading a set of Gaussian random values
 from a Gaussian random number look-up table (LUT) 203.

The image of Gaussian random values obtained during step 202 then undergoes a bit-
 accurate transform, typically by way of an integer Discrete Cosine Transform (DCT)
 25 operation during step 204, although other bit-accurate transform techniques exist. The integer
 transformed coefficients established during step 202 then undergo frequency filtering during

-10-

step 206. Typically, the frequency filter occurs using a predefined set of cut frequencies f_{HL} , f_{VL} , f_{HH} , and f_{VH} that represent cut-off frequencies (in two dimensions) of a filter that characterizes the desired film grain pattern. Next step 208 occurs during which the frequency filtered block of transformed coefficients undergoes an inverse bit-accurate transformation, typically, although not necessarily, an integer Inverse Discrete Cosine Transformation (IDCT), to yield a bit-accurate film grain pattern. Under some circumstances, scaling of the frequency filtered, inverse transformed block of coefficients will prove useful.

Thereafter, step 210 occurs and the film grain pattern resulting from the integer inverse transformation (and scaling if performed) undergoes storage in a database 211. The steps 202-210 within the loop initiated during step 201 undergo repeating for all possible film grain sizes and shapes, whereupon loop execution ends during step 212, after which the method ends during step 214. In this way, the database 211 stores a plurality of film grain patterns for future use in simulating film grain in a video signal.

As compared to the method of FIG. 2, the film grain simulation method of FIG. 3 obviates the use of a Gaussian random number generator. Instead, the method of FIG. 3 makes use of the look-up table 203 that contains pre-computed Gaussian random values. The approach simplifies the required hardware needed for implementation.

FIGURE 4 illustrates a third implementation of a method in accordance with the present principles for creating a bit-accurate pattern of film grain. The method of FIG. 4 commences upon execution of the start step 300 during which initialization occurs, although such initialization need not necessarily happen. Next, step 302 occurs to create a set or image of random Gaussian values. In practice, the Gaussian random image has a size of $N \times N$ values. Creation of the image of Gaussian random values occurs in the method of FIG. 4, by reading a set of Gaussian random values from a Gaussian random number look up table (LUT) 303. The method used to create the Gaussian random number LUT 303 does not require bit accuracy. As long as all implementations of the film grain pattern database creation use the same LUT, the result assures bit accuracy. However, when using a bit-accurate method to create such a LUT, then LUT generation can occur at the beginning of the film grain pattern database creation process, avoiding the need for permanent storage of the LUT. Given that a single transform (step 304) needs to be computed for the generation of the entire film grain pattern database, the randomness of the film grain patterns stored in the database is reduced when compared to the result obtained with previously illustrated

-11-

implementations (FIGS. 2 and 4). However, the third implementation illustrated in FIG. 4 also has lower computational needs, which proves useful for hardware and/or real-time implementations. The image of Gaussian random values created during step 302 then undergoes a bit-accurate transform, typically by way of an integer Discrete Cosine Transform (DCT) operation during step 304; although other bit-accurate transform techniques exist. Following step 304, step 305 occurs which initiates entry into a loop that repeats for all possible film grain sizes and shapes. Step 306, the first step within the loop, initiates frequency filtering of the integer DCT coefficients established during step 304. Typically, the frequency filter occurs using a predefined set of cut frequencies f_{HL} , f_{VL} , f_{HH} , and f_{VH} that represent cut-off frequencies (in two dimensions) of a filter that characterizes the desired film grain pattern. Following step 306, the frequency filtered integer DCT coefficients undergoes an inverse bit-accurate transformation, typically, although not necessarily, an integer Inverse Discrete Cosine Transformation (IDCT), during step 308 to yield a bit-accurate film grain pattern. Under some circumstances, scaling of film grain pattern created from the inversely transformed coefficients following step 308 will prove useful. Thereafter, step 310 occurs and the film grain pattern undergoes storage in a database 311.

Steps 306-310 within the loop initiated during step 305 undergo for all possible film grain sizes and shapes, whereupon loop execution ends during step 312, after which the method ends during step 314. In this way, the database 311 stores film grain patterns for all sizes and shapes of film grain.

FIGURE 5 illustrates a fourth implementation of a method in accordance with the present principles for creating a bit-accurate pattern of film grain. The method of FIG. 5 commences upon execution of the start step 400 during which initialization occurs, although such initialization need not necessarily happen. Next, step 402 occurs, during which an image or set of DCT coefficients is read from a look-up table 403 created from an image of Gaussian random values. In practice, the image of DCT coefficients read from the LUT 403 has a size of $N \times N$. Thus, unlike the previously described implementations of FIGS 2-4, the film grain simulation method of FIG. 5 obviates the need to perform a separate integer DCT operation since the values read from the LUT 403 have already undergone such a transformation prior to loading into the LUT.

Following step 402, step 405 occurs which initiates entry into a loop that repeats for all possible film grain sizes and shapes. Step 406, the first step within the loop, initiates

-12-

frequency filtering of the integer DCT coefficients obtained during step 402. Typically, the frequency filter occurs using a predefined set of cut frequencies f_{HL} , f_{VL} , f_{HH} , and f_{VH} that represent cut-off frequencies (in two dimensions) of a filter that characterizes the desired film grain pattern. Following step 406, the frequency filtered integer DCT coefficients undergo an
5 inverse bit-accurate transformation, typically, although not necessarily, an integer Inverse Discrete Cosine Transformation (IDCT), during step 408 to yield a pattern of film grain. Under some circumstances, scaling of the inversely transformed block of coefficients following step 408 will prove useful. Thereafter, step 410 occurs and the film grain pattern resulting from the integer IDCT transformation (and scaling if performed) undergoes storage
10 in a database 411.

Steps 406-410 within the loop initiated during step 405 undergo repeating for all possible film grain sizes and shapes, whereupon loop execution ends during step 412, after which the method ends during step 414. In this way, the database 411 stores film grain patterns for all sizes and shapes of film grain.

15 The method of FIG. 5 makes use of a single pre-computed block of transformed coefficients to generate the film grain patterns that populate the database 411. Compared to the previous implementations described in connection with FIGS. 2-4, implementation of FIG. 5 eliminates the step of performing an integer DCT transform, thus reducing computational cost.

20 FIGURE 6 illustrates a fifth implementation of a method in accordance with the present principles for creating a bit-accurate pattern of film grain. The method of FIG. 6 commences upon execution of the start step 500 during which initialization occurs, although such initialization need not necessarily happen. Next, step 501 occurs, initiating entry into a loop that repeats for all possible film grain sizes and shapes. Step 502, the first step in the
25 loop initiates reading of an image of set of integer DCT coefficients from a look up table (LUT) 503. In practice, the image of DCT coefficients has a size of $N \times N$. Like the implementation of FIG. 5, the implementation of FIG. 6 establishes an image of integer transformed coefficients image by obtaining a set of integer DCT coefficients from the LUT 503. In practice, the values within the LUT 503 originate from an image of Gaussian random
30 values that subsequently undergo an integer DCT. Thus, unlike the previously described implementations of FIGS 2-4, the implementation of FIG. 6, like that of FIG. 5, obviates the

-13-

need to perform a separate integer DCT since the values read from the look-up table 503 have already undergone such a transformation prior to loading in the table.

Following step 502, step 506 occurs during which the integer DCT coefficients undergo frequency filtering. Typically, the frequency filter occurs using a predefined set of cut frequencies f_{HL} , f_{VL} , f_{HH} , and f_{VH} that represent cut-off frequencies (in two dimensions) of a filter that characterizes the desired film grain pattern. Following step 506, the frequency filtered integer DCT coefficients undergo an inverse bit-accurate transformation, typically, although not necessarily, an integer Inverse Discrete Cosine Transformation (IDCT), to yield a block of film grain during step 508. Under some circumstances, scaling of the inversely transformed block of coefficient following step 508 will prove useful. Thereafter, step 510 occurs and the block of film grain that results from the integer IDCT transformation (and scaling if performed) undergoes storage in a database 511.

Steps 502-510 within the loop initiated during step 501 undergoes repeating for all possible film grain sizes and shapes, whereupon loop execution ends during step 512, after which the method ends during step 514. In this way, the database 511 stores film grain patterns for all sizes and shapes of film grain.

The implementation of film grain pattern creation described with respect to FIG. 6, like the implementation of FIGS. 2 and 3, makes use of different sets of integer DCT coefficients. Using different sets of integer DCT coefficients allows for creation of a richer database of film blocks patterns as compared to the implementation of FIGS. 1, 4 and 5, which utilizes a single block of integer transformed coefficients. While the implementation described with respect to FIGS 1, 4 and 5 affords reduced storage and computational requirements, such implementations lower afford performance because the same noise pattern serves as the basis for generating all film grain patterns.

The foregoing describes several different implementations of a technique for creating at least one, and preferably a plurality of film grain patterns for simulating film grain in a video signal.

CLAIMS

(41)

1 1. A method of creating a pattern of film grain, comprising the steps of:
2 (a) establishing a set of bit-accurate transformed coefficients;
3 (b) frequency filtering set of bit-accurate transformed coefficients; and
4 (c) performing a bit-accurate inverse transformation on the frequency filtered
5 transformed coefficients to yield a film grain pattern.

1 2. The method according to claim 1 wherein steps (a) - (c) are repeated for all
2 possible sizes and shapes of film grain.

1 3. The method according to claim 1 further comprising step of (d) storing the film
2 grain pattern in a database.

1 4. The method according to claim 3 wherein steps (a) - (d) are repeated for all
2 possible sizes and shapes of film grain.

1 5. The method according to claim 1 further comprising the step of scaling the film
2 grain pattern.

1 6. The method according to 1 claim wherein the step of establishing the set of bit-
2 accurate transformed coefficients comprises the step of reading a set of integer Discrete
3 Cosine Transformed coefficients from a database.

1 7. The method according to claim 1 wherein the step of establishing the set of bit-
2 accurate transformed coefficients comprises the step reading a set of integer Discrete Cosine
3 Transformed coefficients from a database for all possible film grain sizes and shapes.

1 8. The method according to claim 1 wherein the step of establishing the set of bit-
2 accurate transformed coefficients comprises the steps of:
3 generating an image of Gaussian random number; and

-15-

4 performing an integer Discrete Cosine Transformation on the image of Gaussian
5 random numbers.

1 9. The method according to claim 1 wherein the step of establishing the set of bit-
2 accurate transformed coefficients comprises the steps of:

3 (i) generating an image of Gaussian random number;

4 (ii) performing an integer Discrete Cosine Transformation on the image of Gaussian
5 random numbers; and

6 (iii) repeating steps (i) and (ii) for all possible film grain sizes and shapes.

1 10. The method according to claim 1 wherein the frequency filtering step makes
2 use of a predefined set of cut frequencies f_{HL} , f_{VL} , f_{HH} , and f_{VH} that represent cut-off
3 frequencies, in two dimensions of a filter that characterizes the desired film grain pattern.

1 11. The method according to claim 1 wherein the step of performing a bit accurate
2 inverse transformation on the frequency filtered transformed coefficients comprises the step of
3 performing an integer inverse Discrete Cosine Transform.

1 12. Apparatus comprising one of a processor and dedicated logic circuit for
2 creating a pattern of film grain by (a) establishing a set of bit-accurate transformed
3 coefficients; (b) frequency filtering set of bit-accurate transformed coefficients; and (c)
4 performing a bit-accurate transformation on the frequency filtered transformed coefficients to
5 yield a film grain pattern.

1 13. The apparatus according to claim 12 further comprising a first memory for
2 storing the film grain pattern.

1 14. The apparatus according to claim 12 further comprising a second memory for
2 storing at least one set of integer Discrete Cosine Transformed coefficients.

1 15. The apparatus according to claim 13 further comprising a second memory for
2 storing a plurality of sets of integer Discrete Cosine Transformed coefficients.

SUBSTITUTE SHEET

AMENDED SHEET

01/06/2006

-16-

1 16. The apparatus according to claim 12 further comprising:
2 a Gaussian random generator for generating a Gaussian random number image;
3 means for performing an integer Discrete Cosine transformation on the Gaussian
4 random number image to yield the set of set of bit-accurate transformed coefficients.

1 17. The apparatus according to claim 12 further comprising:
2 a second memory for storing an image of Gaussian random numbers; and;
3 means for performing an integer Discrete Cosine transformation on the Gaussian
4 random number image to yield the set of set of bit-accurate transformed coefficients.

1 18. Apparatus for creating a pattern of film grain, comprising:
2 means for establishing a set of bit-accurate transformed coefficients;
3 means for frequency filtering set of bit-accurate transformed coefficients; and
4 means for performing a bit-accurate transformation on the frequency filtered
5 transformed coefficients to yield a film grain pattern.

SUBSTITUTE SHEET

AMENDED SHEET

01/06/2006