A Digital Video Quality (DVQ) apparatus and method that incorporate a model of human visual sensitivity to predict the visibility of artifacts. The DVQ method and apparatus are used for the evaluation of the visual quality of processed digital video sequences and for adaptively controlling the bit rate of the processed digital video sequences without compromising the visual quality. The DVQ apparatus minimizes the required amount of memory and computation. The input to the DVQ apparatus is a pair of color image sequences: an original (R) non-compressed sequence, and a processed (T) sequence. Both sequences (R) and (T) are sampled, cropped, and subjected to color transformations. The sequences are then subjected to blocking and discrete cosine transformation, and the results are transformed to local contrast. The next step is a time filtering operation which implements the human sensitivity to different time frequencies. The results are transformed to local contrast. The next step is a time filtering operation which implements the human sensitivity to different time frequencies. The results are transformed to local contrast. The next step is a time filtering operation which implements the human sensitivity to different time frequencies. The results are transformed to local contrast.
U.S. PATENT DOCUMENTS

5,621,660 A 4/1997 Chaddha et al. ............ 364/514
5,629,780 A 5/1997 Watson ................... 358/432
5,719,865 A 2/1998 Brill et al. ............... 382/260
5,818,520 A 10/1998 Janko et al. ............. 348/192
5,991,458 A * 11/1999 Kunitake et al. ........ 382/254
6,057,882 A * 5/2000 van den Branden Lambrecht et al. ............. 348/192
6,075,561 A * 6/2000 Janko ..................... 348/180
6,236,756 B1 * 5/2001 Kimura et al. ......... 382/239

OTHER PUBLICATIONS


* cited by examiner
FIG. 1

ORIGINAL (R)

VIDEO CODEC

PROCESSED (T)

DVQ

$E_{\omega, Q \Omega}$

FEEDBACK

$E_{\omega, Q \Omega}$

$20$

$25$

$10$
FIG. 2
FIG. 3
METHOD AND APPARATUS FOR EVALUATING THE VISUAL QUALITY OF PROCESSED DIGITAL VIDEO SEQUENCES

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application claims the priority of co-pending provisional patent application Ser. No. 60/077,862, filed on Mar. 13, 1998, which is incorporated herein in its entirety.

ORIGIN OF THE DISCLOSURE

The invention described herein was made by an employee of the National Aeronautics and Space Administration and may be manufactured and used by and for the United States Government for governmental purposes without the payment of royalties thereon or therefore.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a method and apparatus for the evaluation of the visual quality of processed digital video sequences. One common form of processing is compression to reduce the bit-rate of digital video. The invention can be used in various applications such as the automatic and continuous monitoring of processing of digital video sequences for transmission as High Definition Television (HDTV) or Direct Broadcast System (DBS) TV. More particularly, the present invention relates to a Digital Video Quality (DVQ) apparatus and method that incorporate a model of human visual sensitivity to predict the visibility of artifacts and the visual quality of processed video.

2. Description of Related Art

Considerable research has been conducted in the field of data compression, especially the compression of digital images. Digital images comprise a rapidly growing segment of the digital information stored and communicated by science, commerce, industry and government. Digital image transmission has gained significant importance in highly advanced television systems, such as high definition television using digital information. Because a relatively large number of digital bits are required to represent digital images, a difficult burden is placed on the infrastructure of the computer communication networks involved with the creation, transmission and re-creation of digital images. For this reason, there is a need to compress digital images to a smaller number of bits, by reducing redundancy and invisible image components of the images themselves.

A system that performs image compression is disclosed in U.S. Pat. No. 5,121,216 of Chen et al. and is incorporated herein by reference. The '216 patent describes a transform coding algorithm for a still image, wherein the image is divided into small blocks of pixels. For example, each block of pixels can be either an 8×8 or 16×16 block. Each block of pixels undergoes a two dimensional transform to produce a two dimensional array of transform coefficients. For still image coding applications, a Discrete Cosine Transform (DCT) is utilized to provide the transform.

In addition to the '216 patent, the DCT is also employed in a number of current and future international standards, concerned with digital image compression, commonly referred to as JPEG and MPEG, which are acronyms for Joint Photographic Experts Group and Moving Pictures Experts Group, respectively. After a block of pixels of the '216 patent undergoes a DCT, the resulting transform coefficients are subject to compression by thresholding and quantization operations. Thresholding involves setting all coefficients whose magnitude is smaller than a threshold value equal to zero, whereas quantization involves scaling a coefficient by step size and rounding off to the nearest integer.

Commonly, the quantization of each DCT coefficient is determined by an entry in a quantization matrix. It is this matrix that is primarily responsible for the perceived image quality and the bit rate of the transmission of the image. The perceived image quality is important because the human visual system can tolerate a certain amount of degradation of an image without being alerted to a noticeable error. Therefore, certain images can be transmitted at a low bit rate, whereas other images cannot tolerate degradation and should be transmitted at a higher bit rate in order to preserve their informational content.

The '216 patent discloses a method for the compression of image information based on human visual sensitivity to quantization errors. In the method of '216 patent, there is a quantization characteristic associated with block to block components of an image. This quantization characteristic is based on a busyness measurement of the image. The method of '216 patent does not compute a complete quantization matrix, but rather a single scaler quantizer.

Recent years have seen the introduction and widespread acceptance of several varieties of digital video. These include digital television broadcasts from satellites (DBS-TV), the US Advanced Television System (ATV), digital movies on a compact disk (DVD), and digital video cassette recorders (DV). Such a trend is expected to continue in the near future and to expand to widespread terrestrial broadcast and cable distribution of digital television systems.

Most of these systems depend upon lossy compression of the video stream. Lossy compression can introduce visible artifacts, and indeed there is an economic incentive to reduce bit rate to the point where artifacts are almost visible. Compounding the problem is the "bursty" nature of digital video, which requires adaptive bit allocation based on visual quality metrics, and the economic need to reduce bit rate to the lowest level that yields acceptable quality.

For this reason, there is an urgent need for a reliable means to automatically evaluate the visibility of compression artifacts, and more generally, the visual quality of processed digital video sequences. Such a means is essential for the evaluation of codecs, for monitoring broadcast transmissions, and for ensuring the most efficient compression of sources and utilization of communication bandwidths.

The following references that are incorporated herein by reference, describe visual quality metrics for evaluating, controlling, and optimizing the quality of compressed still images, and incorporate simplified models of human visual sensitivity to spatial and chromatic visual signals:


Video Quality

It would be desirable to extend the still image metrics described in the foregoing references to cover moving images. Most, if not all video quality metrics are inherently models of human vision. For example, if root-mean-squared-error (RMSE) is used as a quality metric, this amounts to the assumption that the human observer is sensitive to the summed squared deviations between reference and test sequences, and is insensitive to aspects such as the spatial frequency of the deviations, their temporal frequency, or their color. The DVQ metric is an attempt to incorporate many aspects of human visual sensitivity in a simple image processing algorithm. Simplicity is an important goal, since one would like the metric to run in real-time and require only modest computational resources.

A number of video quality metrics have been proposed in the following references:


Some of the video quality metrics described in the foregoing references cover spatial filtering operations employed to implement the multiple, bandpass, spatial filters that are characteristic of human vision. A shortcoming of these video quality metrics is that if the video quality metrics are not based closely enough upon human perception they can not accurately measure visual quality. Alternatively, if the video quality metrics are based closely upon human perception, they will require significant memory or computational resources that restrict the contexts in which they can be applied.

Therefore, there is still an unsatisfied need for a quality metric for digital video, which is reasonably accurate but computationally efficient.

SUMMARY OF THE INVENTION

A feature of the present invention is to provide a Digital Video Quality (DVQ) apparatus and method that incorporate a model of human visual sensitivity to predict the visibility of artifacts. The DVQ method and apparatus are used for the evaluation of the visual quality of processed or compressed digital video sequences, and for adaptively controlling the bit rate of the processed digital video sequences without compromising the visual quality. The DVQ apparatus minimizes the required amount of memory and computation.

The inventive Digital Video Quality (DVQ) apparatus can be widely used in various commercial applications including but not limited to satellite broadcasting of digital television (DBS-TV), movies on compact disc (DVD), high definition digital television (HDTV), digital video Camcorders (DV), Internet Video, digital terrestrial television broadcasting, and digital cable television distribution.

The present DVQ method offers significant advantages over conventional metrics in that the present DVQ method incorporates a reasonably accurate human vision model into a relatively simple processing architecture. A contributor to such architectural simplicity is the use of discrete cosine transforms (DCT) as a spatial filter bank, since the hardware and software to implement the DCT are widely available, due to its prevalence in most existing standards for video compression. Indeed, in some applications of the present DVQ method, the DCT may have already been computed as part of the digital video compression process.

Another contributor to the architectural simplicity of the present DVQ method is the use of Infinite Impulse Response (IIR) Filters in the temporal filtering stages. This reduces the amount of computation and memory required relative to other Finite Impulse Response (FIR) implementations.

The foregoing and other features and advantages of the present invention are achieved by a new DVQ apparatus and method. The input to the DVQ apparatus is a pair of color image sequences: the reference (R) or original non-compressed sequence, and the test (T) or processed sequence. Both sequences (R) and (T) are sampled, cropped, and subjected to color transformations. The sequences are then subjected to blocking and DCT transformation, and the results are transformed to local contrast. The next step is a time filtering operation which implements the human sensitivity to different time frequencies. The results are then converted to threshold units by dividing each DCT coefficient by its respective visual threshold. At the next stage the two sequences are subtracted to produce an error sequence. The error sequence is then subjected to a contrast masking operation, which also depends upon the reference sequence (R). The masked errors can be pooled in various ways to illustrate the perceptual error over various dimensions, and the pooled error can be converted to a visual quality (VQ) measure.

BRIEF DESCRIPTION OF THE DRAWINGS

The features of the present invention and the manner of attaining them, will become apparent, and the invention itself will be understood by reference to the following description and the accompanying drawings, with similar numerals referring to similar or identical elements, wherein:

FIG. 1 is a high level block diagram of video encoding system utilizing a DVQ apparatus according to the present invention;

FIG. 2 is a functional block diagram of the DVQ apparatus of FIG. 1 made according to the present invention;

FIG. 3 is a block diagram of a color transformer incorporated in the DVQ apparatus of FIG. 2, and illustrating a color transformation process according to the present invention:
FIG. 4 is a block diagram of a local contrast converter incorporated in the DVQ apparatus of FIG. 2, and illustrating the computational steps of the local contrast process according to the present invention:

FIG. 5 is a block diagram of a time filter incorporated in the DVQ apparatus of FIG. 2, and illustrating an exemplary temporal filtering process implemented by means of a second order IIR filter according to the present invention; and

FIG. 6 is a block diagram of a contrast masking processor incorporated in the DVQ apparatus of FIG. 2, and illustrating the computational steps of the contrast masking processor according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 illustrates a video encoding system 10 that incorporates a DVQ apparatus 20 according to the present invention. In operation, a sequence of original (R) digital video is fed to the input of a video codec 25, and are processed thereby. The video codec 25 is a well known device for coding and decoding video sequences. The sequence of original (R) video and the sequence of processed (T) video generated by the codec 25 are fed to the DVQ apparatus 20 for quality evaluation. The resultant error or quality control signal (E_k, Q_a) can be fed back to the codec 25 for regulating the compression bit rate to correspond to the desired image visual quality.

FIG. 2 provides an overview of the processing steps of the DVQ apparatus 20. These steps will be described later in greater detail. The input to the DVQ apparatus 20 is a pair of color video sequences: the original or reference (R) video sequences (indexed by s), and each color image includes a set of three images, one for each of three color channels (indexed by c). Each image includes a set of rows (indexed by y), and each row includes a set of pixels (indexed by x). The first of the two sequences (s=1) is the reference sequence (R), the second (s=2) is the test sequence (T). Typically, the test sequence (T) differs from the reference sequence (R) in the presence of compression or other artifacts. The input color space, indexed by c, is defined in sufficient detail that it can be transformed into CIE coordinates, for example by specifying the gamma and chromaticity coordinates of each color channel. The input is expressed as follows:

\[ d(s,i,c,y,x) \]

The sizes of the dimensions, i, y, and x depend upon the application. Also, since the DVQ metric can be computed continuously in a pipeline fashion upon a continuous stream of video images, the dimension indexed by i might not have a finite size. Associated with this input is a video image rate (w_i) expressed in Hertz (Hz), which specifies the time frequency of images in the input, and a display image rate (w_0), which specifies the time frequency of images on the display. Also associated with this input are various other display parameters, such as the color space, gamma, spatial resolution, and veiling light, all of which will be discussed below.

Up-Sampling of Color Components

In many digital color video formats, the three color channels such as Y, Cb, and Cr, are represented with different spatial resolutions. For example, the 4:2:2 variant of CCIR-601 standard digital video, is described in “Recommendation ITU-R BT.601-5, Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide Screen 16:9 Aspect Ratios,” (1995). The two color channels (Cb and Cr) in that standard are represented by 360 pixels/line, while the luminance channel (Y) is represented by 720 pixels/line.

According to the present invention, these channels (Y, Cb, and Cr) are converted to a common resolution before color conversion. In this example, the two color channels (Cb and Cr) are expanded horizontally by a factor of two through a process of up-sampling (US). Up-sampling is performed by the sampler 30 shown in FIG. 2. Although various up-sampling processes are possible, one example is pixel-replication, which is expressed by the following expression:

\[ d(s,i,c,y,x) = \text{US}(d(s,i,c,y,x)) \]

The up-sampling factors for each direction d (vertical=1, horizontal=2) and color channel c are specified by an array us(d,c). In the example above, this array would be \{[1,1], [1,2],[1,2]\}.
Region of Interest (ROI) Processing

The sampled sequences \(d_s\) can be spatially cropped by the ROI processor 32 (FIG. 2) to a ROI \(d_{s,i}\). This confines the computation of the DVQ apparatus 20 to that region, as expressed by the following equation:

\[
d_{s,i,c,y,x}=\text{ROI}[d_{s,i,c,y,x}]
\]

(3)

It is also convenient to make this region an integer multiple of 8x8 pixel blocks, or larger, if color down-sampling is used as described below. In an extension, regions-not-of-interest within the region-of-interest may be excluded by means of a binary mask. The region of interest can also be a continuously varying mask of weights, to emphasize some regions more than others.

Color Transformation Operation

The color transformation process shown in FIG. 3 is implemented by the color transformer 34 (FIG. 2), and will now be described in detail.

Transformation to R'G'B' Color Channels

The ROI sequences \(d_{s,i}\) are transformed from their native color space \(c_{in}\) to, for example, gamma-corrected color channels \(R', G', B'\) by a R'G'B' transformer 100. For example, if \(c_{in}\) corresponds to the YCbCr color space of CCIR-601 standard digital video, then the color channels \(R', G', B'\) are expressed by the following equation:

\[
\begin{align*}
R' &= \frac{c_{in,Y}}{65} \\
G' &= \frac{c_{in,Cb}}{37.5} - 128 \\
B' &= \frac{c_{in,Cr}}{37.5} - 128
\end{align*}
\]

(4)

The resulting color transformed sequences \(d_{s,i}\) can be expressed by the following equation:

\[
d_{s,i,c,y,x}=\text{R'G'G}[d_{s,i,c,y,x}]
\]

(5)

Transformation to RGB Color Channels

The R'G'B' color channels are converted to RGB color channels by a RGB transformer 102. This conversion is effected by dividing the R'G'B' color channels by 255, clipping to the range [0,1], and raising the result to an exponent \(y\). The clipping may be necessary because the range of \(c_{in}\) values, combined with the interpolation process, can produce excursions outside the permitted range. The resulting color transformed sequences \(d_{s,i}\) can be expressed by the following equation:

\[
d_{s,i,c,y,x}=\text{RGB}[d_{s,i,c,y,x}]
\]

(6)

Transformation to XYZ Coordinates

The RGB color channels \(d_{s,i}\) are then converted to the standard CIE color coordinates XYZ \(d_{s,i}\) by a XYZ transformer 104. This is accomplished through multiplication by a user-supplied matrix \(M_{\text{RGB,XYZ}}\) that describes the simulated display. In the following expression, the dot product is carried out over the index \(c_{\text{RGB}}\).

\[
d_{s,i,c,y,x}=\text{XYZ}[d_{s,i,c,y,x}]
\]

(7)

Transformation to YOZ Coordinates

The XYZ color coordinates \(d_{s,i}\) are converted to color coordinates YOZ \(d_{s,i}\) by a YOZ transformer 106. This transformation is described in H. Peterson, A. J. Ahumada, Jr. and A. Watson, "An Improved Detection Model for DCT Coefficient Quantization," SPIE Proceedings, 1913, 191-201 (1993), which is incorporated by reference, in modeling perceptual errors in still image compression. In the present invention, the transformation is accomplished through multiplication by a matrix \(M_{\text{YOZ}}\). In the following expression, the dot product is carried out over the index \(c_{\text{YOZ}}\).

\[
d_{s,i,c,y,x}=\text{YOZ}[d_{s,i,c,y,x}]
\]

(8)

The transformations to XYZ and to YOZ can be concatenated into a single matrix multiplication.

De-Interlacing

If the input sequence \(R\) or \(T\) contains interlaced video fields, then the index \(i\) specifies fields, and odd numbered fields contain odd (or even) numbered video lines, and even fields contain even (or odd) video lines. In this case, the first step includes converting the interlaced fields to a progressive sequence \(d_{s,i}\) by means of a de-interlacer 110. The de-interlacing process can be implemented, for example, by one of three methods, depending upon the system requirements. Each of these three de-interlacing methods will now be described in detail.

1. De-interlacing by Inserting Blank Lines.
   - In this method, each field is converted to an image by inserting blank lines into even numbered lines in odd (or even) fields, and odd numbered lines in even (or odd) fields. This method doubles the total number of pixels to be processed. The advantage of this method is that it correctly represents the spatial and time relationship of all video lines.
   - In this method, the display image rate is specified as equal to the video image rate \(w_{\text{disp}}\).

2. De-interlacing by Treating One Field as One Image.
   - In this method, each field is treated as an image. This method is more efficient than the method of inserting blank lines, since the number of pixels is not increased. However, this method does not completely accurately represent the spatial relationship of lines in odd and even fields. For example, the first lines of odd and even fields are treated as superimposed rather than offset by one line. In this method, the display image rate is specified as equal to the video image rate \(w_{\text{disp}}\).

3. De-interlacing by Treating Two Fields as One Image.
   - In this method, each pair of odd and even fields are combined into one image; the odd field contributes the odd (or even) lines, and the even field contributes the even (or odd) lines. This method is as efficient as the method above at treating each field as an image, since the number of pixels is not increased (the number of images is halved, but the number of lines/image is doubled). However, this method does not completely correctly represent the temporal relationship of lines in odd and even fields. For example, the odd and even fields are treated as occurring in the same field rather than offset by one field time. In this method, the display image rate is half the video image rate \(w_{\text{disp}}/2\).
   - The application of the de-interlace operation can be expressed by the following equation:

\[
d_{s,i,c,y,x}=\text{CSV}[d_{s,i,c,y,x}]
\]

(9)

It should also be noted that this operation can change the size of dimensions \(i\) or \(y\), depending on which method is selected. If the input is progressive video, then the de-interlace operation is omitted.

Veiling Light

The next step is the addition of a veiling light to both processed sequences \(R\) and \(T\) by a veiling light combiner.
Local Contrast

expressed by the following equation:

where the result vector is understood to be added to each color pixel.

Down-Sampling of Color Components

Since visual acuity for color signals is much lower than that for luminance, it is often possible to lower the resolution of the two color channels O and Z. To achieve this, the color channels O and Z are down-sampled by factors of dslc,dz, where c is color (Y,O, or Z), and d is direction (vertical or horizontal), by means of a down sampler 114. This down-sampling process can be accomplished by any number of well known or available filtering and sampling procedures. Such as block-averaging. The color down-sampling step can be expressed by the following equation:

Blocked DCT

Referring back to FIG. 2, the color transformed sequences (d_4) in each channel is each color channel is divided into 8×8 pixel blocks by the block constructor 36, and a DCT is applied to each block by the DCT transformer 38. This operation is referred to as blocked DCT (BDCT). The input will typically have been cropped to an integer number of blocks in the vertical and horizontal directions. The dimensions of the result are (s, t, c, by, bx, v, u), where by and bx are the number of blocks in vertical and horizontal directions, respectively, and where v and u are the DCT frequencies that are integers between 0 and 7. The BDCT is expressed by the following equation:

Local Contrast

FIG. 4 represents a functional block diagram of an exemplary local contrast converter 40. The local contrast converter 40 converts the DCT coefficients (d_4) to units of local contrast (d_3). First, the DCT coefficients (d_4) are adjusted by the relative magnitudes of their coefficients corresponding to a unit contrast basis function (A) 206, as illustrated in the following Table 1, and as expressed by the following equation:

### Table 1

<p>| Relative magnitudes of unit contrast DCT Basis Functions, A (c, u) |
|-----------------|--------|--------|--------|--------|--------|--------|--------|</p>
<table>
<thead>
<tr>
<th>c</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>0</td>
<td>1.38704</td>
<td>1.30656</td>
<td>1.38704</td>
<td>1.38704</td>
<td>1.30656</td>
<td>1.38704</td>
<td>1.38704</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1.38704</td>
<td>1.92388</td>
<td>1.81225</td>
<td>1.92388</td>
<td>1.81225</td>
<td>1.92388</td>
<td>1.92388</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.30656</td>
<td>1.81225</td>
<td>1.7071</td>
<td>1.81225</td>
<td>1.7071</td>
<td>1.81225</td>
<td>1.92388</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.38704</td>
<td>1.92388</td>
<td>1.81225</td>
<td>1.92388</td>
<td>1.81225</td>
<td>1.92388</td>
<td>1.92388</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.38704</td>
<td>1.92388</td>
<td>1.81225</td>
<td>1.92388</td>
<td>1.81225</td>
<td>1.92388</td>
<td>1.92388</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.30656</td>
<td>1.81225</td>
<td>1.7071</td>
<td>1.81225</td>
<td>1.7071</td>
<td>1.81225</td>
<td>1.92388</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1.38704</td>
<td>1.92388</td>
<td>1.81225</td>
<td>1.92388</td>
<td>1.81225</td>
<td>1.92388</td>
<td>1.92388</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1.38704</td>
<td>1.92388</td>
<td>1.81225</td>
<td>1.92388</td>
<td>1.81225</td>
<td>1.92388</td>
<td>1.92388</td>
</tr>
</tbody>
</table>

DC coefficients (d_3) are then extracted from all the blocks (d_4) by a DC extractor (DC) 202, as expressed by the following equation:

The DC coefficients (d_3) are then time filtered by a time filter (TF) 204, using a first-order, low-pass, IIR filter with unit DC gain this can be done by specifying a time constant T in seconds and a display image-rate w in Hz. The filter parameters a_1 and b_1 can then be expressed by the following equations:

Table 1
The mean filtered DC coefficients \(d_{10}\) are computed by another averager 212, as follows:

\[
d_{10}(t, i, c, b, v, u) = \frac{1}{N_b N_v} \sum_{b=1}^{N_b} \sum_{v=1}^{N_v} d_{10}(t, i, c, b, v, u)
\]

(20)

The difference between the DC coefficients \(d_{12}\) and their mean \(d_{10}\) is divided by the mean filtered DC coefficients \(d_{10}\), again using the Y component for the denominator of the O component, to generate the DC contrast coefficients \(d_{17\alpha}\), as follows:

\[
d_{17\alpha}(t, i, c, b, v, u) = \frac{d_{12}(t, i, c, b, v, u, 0, 0) - d_{10}(t, i, c)}{d_{10}(t, i, c)}
\]

(22)

The insertion of the processed DC contrast coefficients \(d_{17\alpha}\) is labeled DC\(^{-1}\). These operations convert each DCT coefficient \(d_{10}\) to a number that expresses the amplitude of the corresponding basis function as a fraction of the time-filtered average luminance in the corresponding block. For the DC coefficients \(d_{12}\), it expresses their difference from the mean DC coefficient \(d_{10}\) as a fraction of the mean filtered DC coefficients \(d_{10}\).

The final local contrast signal \(d_{12}\) is composed of AC coefficients \(d_{18}\), combined with DC coefficients \(d_{17\alpha}\), by means of a DC insertion process \(\text{DC}^{-1}\) 208.

Temporal Filtering

With further reference to FIG. 5, the local contrast signals \(d_{12}\) are subjected to temporal filtering. In a preferred embodiment, the time filter 42 is a second-order IIR filter. Parameters of the time filter 42 are estimated from calibration data. The filtered coefficients \(d_{18}\) resulting from the time filter 42 can be expressed as follows:

\[
d_{18}(t, i, c, b, v, u) = b_2(c, v, u) d_{18}(t, i, c, b, v, u, 0) + \sum_{k=1}^{2} a_{2k}(c, v, u) d_{18}(t, i-k, c, b, v, u, u)
\]

(23)

where \(b_2\) and \(a_{2k}\) are arrays of filter coefficients. These arrays \(b_2\) and \(a_{2k}\) allow different temporal filtering for each DCT frequency and each color. For simplicity, these arrays \(b_2\) and \(a_{2k}\) can be made constant, independent of \(c, v, u\), or can be made to depend only on \(c\) or on both \(c\) and a simple function of \(v, u\).

Since the image rate of the digital video can vary from application to application, it is necessary to define the filter arrays \(b_2\) and \(a_{2k}\) in a way that is independent of image-rate. The present method specifies the time filter 42 in terms of a center frequency \(w_c\) and a tuning factor \(q\). If the time filter 42 were constrained to have a magnitude of one at the center frequency \(w_c\), and if the display image rate were \(w_d\), then the filter arrays \(b_2\) and \(a_{2k}\) are expressed by the following equations:

\[
d_{18}(t, i, c, b, v, u) = b_2(c, v, u) d_{18}(t, i, c, b, v, u, 0) + \sum_{k=1}^{2} a_{2k}(c, v, u) d_{18}(t, i-k, c, b, v, u, u)
\]

(23)

Subtraction of Test and Reference

After conversion to threshold units \(d_{10}\), the units corresponding to the two sequences \((R)\) and \((T)\), are subtracted by a subtractor 46 to produce an error sequence \(d_{20}\), as follows:

\[
d_{20}(t, i, c, b, v, u) = d_{18}(t, i, c, b, v, u, 0) - d_{18}(t, i, c, b, v, u, 0)
\]

(33)
Contrast Masking

With further reference to FIG. 6, contrast masking is accomplished by first constructing a masking sequence \((d_{20})\). The threshold units \((d_{10})\) corresponding to the reference sequence \((R)\) are rectified by a rectifier \(215\), as follows:

\[
d_{21}(i,c,b,v,u) = \frac{d_{10}(i,c,b,v,u)}{1 + a_{1}}
\]

and are then time-filtered by a first-order, low-pass, discrete filter \(217\), with parameters \(a_1\) and \(b_1\) to generate a filtered masking sequence \((d_{21})\). Parameter \(b_1\) can be derived from a contrast masking gain \(g\), a time constant \(t_1\) and the display image-rate \(w_d\), as follows:

\[
b_1 = g(1 + a_1)
\]

In an alternative embodiment, both \(t_1\) and \(g\) can be functions of \(c,v,u\). The filtered sequence \((d_{21})\) is then obtained by the following expression:

\[
d_{22}(i,c,b,v,u) = a_{1}d_{21}(i,c,b,v,u) + \frac{d_{21}(i,c,b,v,u)}{1 + a_{1}}
\]

In an alternative embodiment, the values of the filtered masking sequence \((d_{21})\) can be blurred, within each block to implement a form of cross-channel masking, as explained in A. B. Watson and J. A. Solomon, “A Model of Visual Contrast Gain Control and Pattern Masking,” Journal of the Optical Society A, 14, 2378–2390 (1997) which is incorporated herein by reference.

The values of the filtered masking sequence \((d_{21})\) are then raised to a power \(m\) by an expotentiator \(219\), wherein any values less than 1 are replaced by 1. The resulting values \((d_{22})\) are used to divide the difference sequence \((d_{20})\), as expressed below for generating a masked error sequence \((d_{24})\):

\[
d_{24}(i,c,b,v,u) = \frac{d_{22}(i,c,b,v,u)}{d_{20}(i,c,b,v,u) + \varepsilon}
\]

This process resembles the traditional contrast masking result in which contrasts below threshold have no masking effect, and for contrasts above threshold the effect rises as the \(m\)th power of mask contrast in the threshold units \((d_{10})\), as explained in G. E. Legge and J. M. Foley, “Contrast Masking in Human Vision,” Journal of the Optical Society of America, 70(12), 1458–1471 (1980), which is incorporated herein by reference.

Error Pooling

Referring back to FIG. 2, the dimensions of the resulting sequence \((d_{24})\) are \([i, c, b, v, u]\), where \(i\) is images, \(c\) is color channels, and \(b, v\) and \(u\) are the number of blocks in vertical and horizontal directions, and \(v, u\) are the vertical and horizontal DCT frequencies. These elementary errors can then be combined over a subset of dimensions \(\Omega\), or all dimensions, to yield summary measures of visual error distributed over the complementary dimensions \(\Omega^c\). In a preferred embodiment, this summation is implemented using a Minkowski metric as follows:

\[
E_\Omega^c(i,c,b,v,u) = \left[ \sum_{i,c,b,v,u} d_{24}(i,c,b,v,u) \right]^{\frac{1}{d}}
\]

Different applications can require summation over different subsets of dimensions. For example, summation over all dimensions except \(i\) would provide a continuous time record of overall quality, while pooling over all dimensions except \(v\) and \(u\) (over some number of images) would indicate visual error as function of DCT frequency.

Output

The output of the DVQ apparatus \(20\) can be either the perceptual error \((E_{Q1})\) or a quality measure \((Q_{Q1})\) outputted by the visual quality converter \(52\). The quality measure \((Q_{Q1})\) can be computed as follows:

\[
Q_{Q1} = \frac{2}{1 + E_{Q1}}
\]

This quality measure \((Q_{Q1})\) has a maximum value of 2, which is reached when the perceptual error \((E_{Q1})\) is zero, and has a value of 1 when the perceptual error \((E_{Q1})\) is at threshold (a value of 1). Other monotonic transforms of the perceptual error \((E_{Q1})\) can alternatively be employed.

Exemplary Parameters

The following Table 2 provides exemplary parameters used in the DVQ apparatus \(20\). Alternative embodiments can use different values of these parameters.

**TABLE 2**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Example value</th>
<th>Definition</th>
<th>Unit</th>
<th>Equ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(y_{\text{CMYK}})</td>
<td>1, -0.002463, 1, 3.6558</td>
<td>Color transform matrix</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>(w_{\text{CMYK}})</td>
<td>1, -0.33356, -0.69982</td>
<td>Color transform matrix</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>(w_{\text{RGB}})</td>
<td>40.85, 32.13, 18.95</td>
<td>Color transform matrix</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>(w_{\text{XYZ}})</td>
<td>23.20, 67.62, 7.90</td>
<td>Color transform matrix</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>(w_{\text{Y}0Z})</td>
<td>2.09, 12.20, 104.75</td>
<td>Color transform matrix</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>(T_{0}(c))</td>
<td>0.47, -0.37, 0</td>
<td>Global thresholds ([Y, O, Z])</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>(T_{1}(c))</td>
<td>1/0.843, 1/231.09, 1/27.7</td>
<td>Veiling Light</td>
<td>CIE XYZ</td>
<td>10</td>
</tr>
<tr>
<td>(V)</td>
<td>1, 1, 1</td>
<td>Video image rate</td>
<td>Hz</td>
<td>24</td>
</tr>
<tr>
<td>(W)</td>
<td>60</td>
<td>Display image rate</td>
<td>Hz</td>
<td>29</td>
</tr>
<tr>
<td>(f_{s}(\theta))</td>
<td>19.78. 4.85, 4.85</td>
<td>Spatial corner frequency ([Y, O, Z])</td>
<td>cycles/degree</td>
<td>30, 39</td>
</tr>
<tr>
<td>(\beta)</td>
<td>4</td>
<td>Pooling exponent</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>(r)</td>
<td>0.167</td>
<td>Oblique effect parameter</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>
TABLE 2-continued

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Example value</th>
<th>Definition</th>
<th>Unit</th>
<th>Equ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>vr (c, d)</td>
<td>[{32, 32}, {32, 16}]</td>
<td>Video resolution</td>
<td>pixels/degree</td>
<td>27</td>
</tr>
<tr>
<td>ws (c, d)</td>
<td>[{11, 1}, {1, 2}, {1, 2}]</td>
<td>Up-sampling factors</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>ds (c, d)</td>
<td>[{11, 1}, {2, 2}, {2, 2}]</td>
<td>Down-sampling factors</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>cr (c, d)</td>
<td>[{32, 32}, {16, 16}, {16, 16}]</td>
<td>Calibration resolutions</td>
<td>pixels/degree</td>
<td>31</td>
</tr>
<tr>
<td>t_l</td>
<td>0.04</td>
<td>Light adaptation time constant</td>
<td>seconds</td>
<td>16</td>
</tr>
<tr>
<td>t_s</td>
<td>0.04</td>
<td>Contrast masking time constant</td>
<td>seconds</td>
<td>15</td>
</tr>
<tr>
<td>g</td>
<td>3</td>
<td>Contrast masking gain</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>m</td>
<td>0.9</td>
<td>Contrast masking exponent</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>S</td>
<td>3.7</td>
<td>Summation factor</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>q (c)</td>
<td>{1.3, 1.3, 1.3}</td>
<td>Temporal filter Q factor, [V, O, Z]</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>w (c)</td>
<td>{7.31, 7.31, 7.31}</td>
<td>Temporal filter center frequency</td>
<td>Hz</td>
<td>15</td>
</tr>
</tbody>
</table>

It should be clear that alternative embodiments of the DVQ apparatus are possible within the scope of the present invention. In one embodiment, and if additional processing speed were desirable, and if the input were in an appropriate format such as RGB color space, the color transforms and gamma conversion can be omitted.

In an alternative embodiment, and if additional processing speed were desirable, and if the input were in an appropriate format such as the blocked DCT of YCbCr color channels, then the DCT transform, as well as the color transforms and gamma conversion can be omitted.

In another alternative embodiment, the subtraction of the test (T) and reference (R) sequences by the subtractor 46 can be postponed until after the contrast masking is implemented by the contrast masking processor 48. The contrast masking process can be combined with an alternate masking formulation in which each DCT coefficient is divided by a rectified, filtered set of neighboring coefficients, with a small added constant.

In still another embodiment, rather than using a single set of time filter coefficients (b, a1, a2, b), a matrix with one entry for each color and DCT frequency can be used. This does not substantially increase the computational complexity, but improves the accuracy of the temporal filtering model.

One skilled in the art will appreciate that the present invention can be practiced by other than the described embodiments or values, which are presented for purposes of illustration and not of limitation. For example, while the DVQ apparatus and method are described in terms of discrete components, it should be clear that the function of these components can be implemented by means of a software program.

What is claimed is:

1. A digital video quality method for evaluating the visual quality of a processed video sequence relative to an original video sequence, the method comprising:

- sampling the original and processed video sequences to generate sampled sequences (d1) therefrom;
- limiting the processing of said sampled sequences (d1) to a region of interest and generating region of interest sequences (d2) therefrom;
- transforming said region of interest sequences (d2) to local contrast coefficients (d3);
- filtering said local contrast coefficients (d3) to generate filtered components (d4) therefrom;
- converting said filtered components (d4) to threshold units (d5);
- substracting said threshold units (d5) corresponding to the original (R) and processed (T) sequences to obtain an error sequence (d6);

subjecting said error sequence (d6) to a contrast masking operation to generate a masked error sequence (d7) therefrom; and

pooling said masked error sequence (d7) to generate a temporal error (d8).

2. A method according to claim 1, further including converting said perceptual error (d8) to a visual quality measure (Qo), to provide an output in terms of quality.

3. A method according to claim 1, further including feeding back said perceptual error (d8) to a codec, for regulating a compression bit rate to correspond to a desired image visual quality.

4. A method according to claim 2, further including feeding back said visual quality measure (Qo) to a codec, for regulating a compression bit rate to correspond to a desired image visual quality.

5. A method according to claim 1, wherein each of said processed (T) video sequence and said original (R) video sequence includes color channels:

subjecting said color channels to a color transformer to a gamma corrected color channels R', G', B' by means of a discrete cosine transformer.

6. A method according to claim 5, further including subjecting said color transformed sequences (d9) to blocking to generate blocks.

7. A method according to claim 6, further including converting said blocks to a block of frequency coefficients (d10) by means of a discrete cosine transformer.

8. A method according to claim 7, wherein said block of frequency coefficients (d10) are converted to a local contrast signals (d11) by means of a local contrast converter; and

wherein said local contrast signal (d11) includes a combination of AC coefficients (d11a) and DC coefficients (d11b).

9. A method according to claim 8, wherein contrast masking is accomplished by rectifying said threshold units (d5).

10. A method according to claim 9, further including converting said RGB color channels R', G', B' to RGB color channels by a RGB transformer.

11. A method according to claim 10, further including converting said RGB color channels R', G', B' to XYZ color channels by a RGB to XYZ transformer.

12. A method according to claim 11, further including converting said RGB color channels to XYZ color coordinates by a XYZ transformer.

13. A method according to claim 12, further including converting said XYZ color coordinates to YOZ color coordinates by a YOZ transformer.
14. A method according to claim 13, wherein if any of the
processed (T) video sequence or the original (R) video
sequence contains interlaced video fields, then
de-interlacing said interlaced fields to a progressive
sequence (d_7) by means of a de-interlacer.
15. A method according to claim 14, wherein
de-interlacing is implemented by inserting blank lines into
even numbered lines in odd fields, and odd numbered lines
in even fields.
16. A method according to claim 14, wherein
de-interlacing is implemented by inserting blank lines into
even numbered lines in even fields, and odd numbered lines
in odd fields.
17. A method according to claim 14, wherein
de-interlacing is implemented by each pair of odd and even
video fields as an image.
18. A method according to claim 14, further including
adding a veiling light to said progressive sequence (d_7) by
means of a veiling light combiner.
19. A method according to claim 1, wherein sampling
includes pixel-replication.
20. A digital video quality apparatus with an original (R)
video sequence and a processed (T) video sequence being
fed thereto, the apparatus comprising:
a sampler for sampling the original and processed video
sequences to generate sampled sequences (d_j) therefrom;
a region-of-interest processor for limiting the processing
of said sampled sequences (d_j) to a region of interest
and for generating region of interest sequences (d_j)
therefrom;
a local contrast converter for transforming said region of
interest sequences (d_j) to local contrast coefficients
(d_jT);
a time filter for filtering said local contrast coefficients
(d_jT) and for generating filtered components (d_jT)
therefrom;
a threshold scaler for converting said filtered components
(d_jT) to threshold units (d_jT);
a subtractor for subtracting said threshold units (d_jT)
corresponding to the original (R) and processed (T)
sequences to obtain an error sequence (d_j0);
a contrast masking processor for subjecting said error
sequence (d_j0) to a contrast masking operation and for
generating a masked error sequence (d_j2T) therefrom;
and
a pooling processor for pooling said masked error
sequence (d_j2T) to generate a perceptual error (E_jT).
21. An apparatus according to claim 20, further including
a visual quality converter that converts said perceptual error
(E_jT) to a visual quality measure (Q_jT), for providing an
output in terms of quality.
22. An apparatus according to claim 21, further including
a codec to which said perceptual error (E_jT) is fed back for
regulating a compression bit rate to correspond to a desired
image visual quality.
23. An apparatus according to claim 21, further including
a codec to which said visual quality measure (Q_jT) is fed
back for regulating a compression bit rate to correspond to
a desired image visual quality.
24. An apparatus according to claim 20, wherein each of
said processed (T) video sequence and said original (R)
video sequence includes color channels; and
a color transformer that converts said color channels to a
perceptually relevant color space, for generating color
transformed sequences (d_j) from said region of interest
sequences (d_jT).
25. An apparatus according to claim 24, further including
a block constructor that subjects said color transformed
sequences (d_jT) to blocking in order to generate blocks.
26. An apparatus according to claim 25, further including
a discrete cosine transformer for converting said blocks to a
block of frequency coefficients (d_j10).
27. An apparatus according to claim 26, wherein if any of
the processed (T) video sequence or the original (R) video
sequence contains interlaced video fields, then
de-interlacing said interlaced fields to a progressive
sequence (d_j) by means of a de-interlacer.
28. An apparatus according to claim 27, further including
a veiling light combiner for adding a veiling light to said
progressive sequence (d_j).
29. An apparatus according to claim 28, further including
a local contrast converter for converting said block of
frequency coefficients (d_j10) to a local contrast signal (d_jT);
and
wherein said local contrast signal (d_jT) includes a
combination of AC coefficients (d_j1T) and DC coefficients
(d_j0T).
30. A digital video quality apparatus with original (R)
video sequence and a processed (T) video signal being
fed thereto, the apparatus comprising:
a sampler for sampling the original and processed video
sequences and for generating sampled sequences (d_j)
therefrom;
a region-of-interest processor for limiting the processing
of said sampled sequences (d_j) to a region of interest
and for generating region of interest sequences (d_j)
therefrom;
a local contrast converter for transforming said region of
interest sequences (d_j) to local contrast coefficients
(d_jT);
a time filter for filtering said local contrast coefficients
(d_jT) and for generating filtered components (d_jT)
therefrom;
a threshold scaler for converting said filtered components
(d_jT) to threshold units (d_jT);
a subtractor for subtracting said threshold units (d_jT)
corresponding to the original (R) and processed (T)
sequences to obtain an error sequence (d_j0);
a contrast masking processor for subjecting said error
sequence (d_j0) to a contrast masking operation and for
generating a masked error sequence (d_j2T) therefrom;
and
a pooling processor for pooling said masked error
sequence (d_j2T) to generate a perceptual error (E_j).
31. An apparatus according to claim 30, further including
a visual quality converter that converts said perceptual error
(E_j) to a visual quality measure (Q_j), for providing an
output in terms of quality.
32. A digital quality method for evaluating the visual
quality of a processed (T) video sequence relative to an
original (R) video sequence, the method comprising:
sampling the original and processed video sequences
and for generating sampled sequences (d_j) therefrom;
limiting the processing of said sampled sequences (d_j)
to a region of interest and for generating region of interest
sequences (d_jT) therefrom:
transferring said region of interest sequences \( d_1 \) to local contrast coefficients \( d_{1,7} \);
filtering said local contrast coefficients \( d_{1,7} \) and for generating components \( d_{1,8} \) therefrom;
subtracting said threshold units \( d_{1,0} \) to obtain an error sequence \( d_{2,0} \);
subtracting said threshold units \( d_{1,0} \) to obtain an error sequence \( d_{2,0} \);

subjecting said error sequence \( d_{2,0} \) to a contrast masking operation to obtain a masked error sequence \( d_{2,4} \); and pooling said error sequence \( d_{2,0} \) to generate a perceptual \( E_{4,4} \).