VARIABLE BIT RATE ENCODING USING JPEG2000

M. W. Marcellin and A. Bilgin

DTS, USA

ABSTRACT

A variable bit rate encoding method for JPEG2000 is presented. The method is suitable for encoding digital cinema content. All bit rate constraints described in the DCI Digital Cinema System Specification are satisfied. At the same time, average peak signal-to-noise ratio is maximized subject to these constraints. The encoder first creates high rate codestreams that satisfy the constraints. The high rate codestreams are subsequently analyzed and parsed to create final JPEG2000 codestreams at any desired average bit rate.

INTRODUCTION

In July 2005, Digital Cinema Initiatives published its Digital Cinema System Specification (1). In that specification, JPEG2000 was chosen for the distribution of digital cinema content. JPEG2000 is the latest international standard for image compression. JPEG2000 provides state of the art compression performance, as well as many advanced features and functionalities. Key among them is the ability to extract multiple resolution images from a single compressed codestream. This makes it possible to easily display a 2K (2048 × 1080) image or a 4K (4096 × 2160) image from a 4K compressed codestream.

Figure 1 illustrates a representative JPEG2000 encoder for digital cinema distribution. First, a component transform operates independently on each pixel of the three image color components using a 3×3 matrix to obtain three new color components having most of the information concentrated in one of the new components. The particular component transform used is the one normally used to transform RGB to YCbCr. For digital cinema, the transform is applied to X'Y'Z' input images. The transform used is solely for the purpose of compression and should not be thought of as a color space transform. No sub-sampling is allowed, i.e., the original X'Y'Z' components and the resulting transformed components are all 4:4:4. A wavelet transform is applied independently to each new component to produce a number of transform coefficients, organized into subbands. The transform coefficients for each subband are then partitioned into 32×32 blocks referred to as *codeblocks*. This is illustrated in Figure 2. Each codeblock is then encoded independently by a codeblock encoder.

For a given codeblock, encoding begins by quantizing the wavelet coefficients from the codeblock to obtain quantization indices. These quantization indices can be regarded as an array of signed integers. This array of signed integers can be represented using a sign array and a magnitude array. The sign array can be considered as a binary array where the value of the array at each point indicates whether the quantization index is positive or negative. The magnitude array can be divided into a series of binary arrays with each such binary array containing one bit from each quantization index. The first of these arrays corresponds to the Most Significant Bits (MSBs) of the quantized magnitudes, and the last one

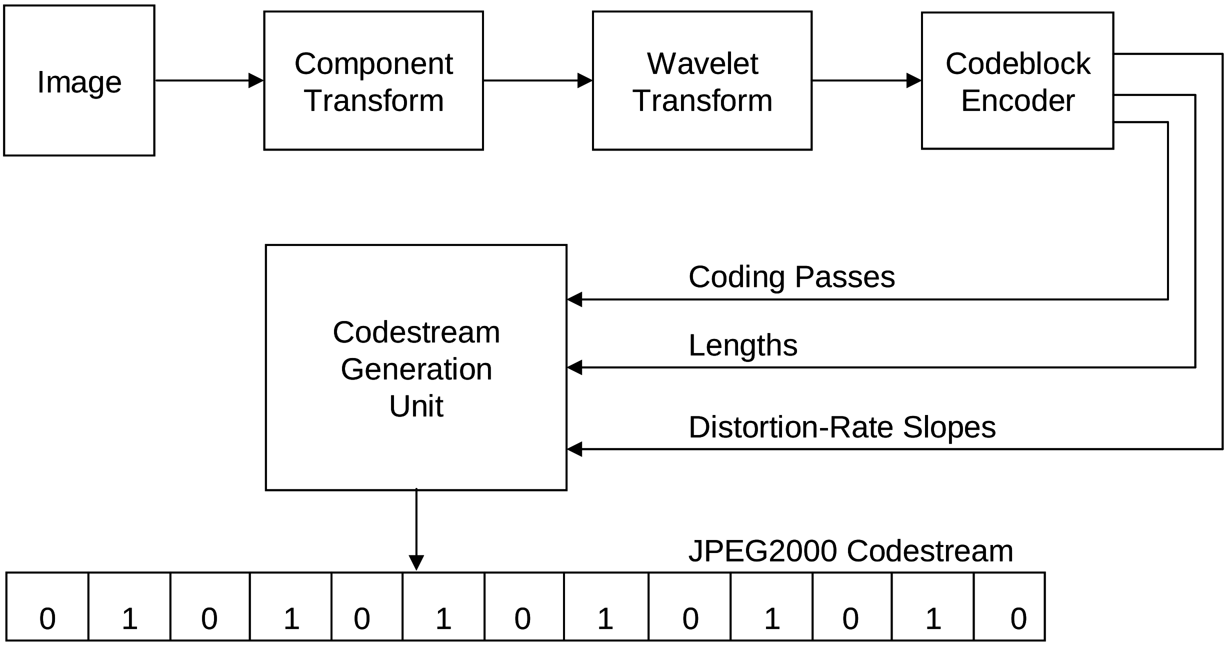

Figure 1 – Block diagram of JPEG2000 encoder

corresponds to the Least Significant Bits (LSBs). Each such array is referred to as a bitplane**.** Each bitplane of a codeblock is then coded using a bitplane coder. The bitplane coder used in JPEG2000 is a context-dependent, binary, arithmetic coder. The bitplane coder makes three passes over each bitplane of a codeblock. These passes are referred to as *coding passes*. Each bit in the bitplane is encoded in one of these coding passes. The resulting compressed data are referred to as compressed coding passes.

The codeblock encoder also computes the amount of distortion (typically weighted mean squared error) reduction provided by each compressed coding pass together with the length of the compressed coding pass. With this information, it is possible to define the ratio of the distortion reduction over the length of the compressed coding pass as the *distortion-rate slope* of the compressed coding pass. The distortion-rate slope of a compressed coding pass is the amount of distortion reduction per byte provided by the compressed coding pass. Thus, a compressed coding pass with a larger distortion-rate slope can be considered to be more important than one with a smaller distortion-rate slope. The codeblock encoder provides the compressed coding passes**,** together with their lengths and distortion-rate slopes, to a codestream generation unit that decides which compressed coding passes from each codeblock will be included in the codestream. Generally, the codestream generation unit includes the compressed coding passes with the largest distortion-rate slopes into the codestream until the byte budget is exhausted. For more details on JPEG2000, the interested reader is referred to Taubman and Marcellin (2).

As evident from (2, Section 8.2), it is sometimes desirable to disallow codeblock codestream termination between certain coding passes. Equivalently, it may be desirable to “group” two or more coding passes and treat them as essentially a single composite coding pass for the purpose of rate allocation. Such a composite coding pass has a single distortion-rate slope computed as the total distortion decrease of the group of coding passes divided by the total length of the group of coding passes. To simplify discussion, it should be assumed throughout the paper that this grouping is carried out when appropriate, and that then the term “coding pass” may refer to a composite coding pass.

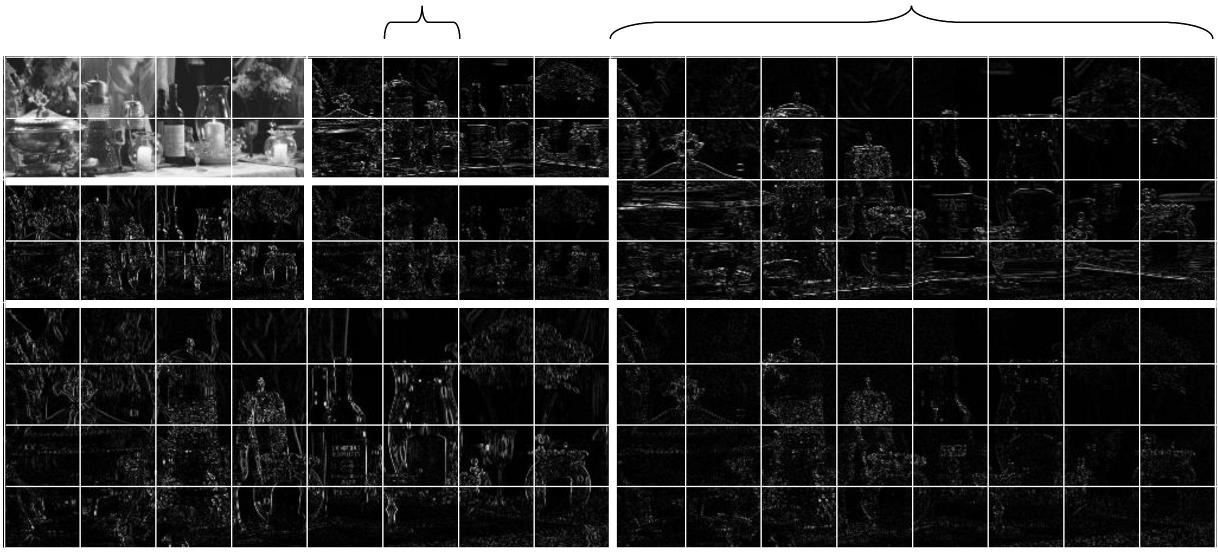

Figure 2 – Wavelet transform coefficients organized into subbands and codeblocks

The encoder operation described above defines how a single image can be encoded using JPEG2000. However, JPEG2000 can be used to encode images that make up an image sequence, e.g., a motion picture. When JPEG2000 is used to compress a sequence of images, there are only a few methods previously known for determining what rate to use for each image in the sequence. One possibility is to select a fixed rate (i.e., fixed number of bytes) to encode each image in the sequence. While this method is simple and allows easy implementation, it does not yield adequate performance in some applications. In many image sequences, the characteristics of the images in the sequence vary immensely. Since this method assigns a fixed number of bytes to each image, the resulting decompressed image sequence exhibits large variations in quality among images.

In, Tzannes et al (3) adaptive selection of compression parameters is used to achieve some performance improvement when images are encoded in succession. The adaptation is performed for the current image using information gathered from only the previous images in the sequence: subsequent images are not considered when allocating rate for the current image. Furthermore, if two consecutive images in the sequence are not highly correlated (such as the case during a scene change), the adaptation falters. Another alternative to fixed rate coding was presented by Dagher et al (4). In this method, compressed images are placed in a buffer. Compressed data are pulled out of the buffer at a constant rate. New compressed images are added to the buffer when they become available. If the buffer is full when a new compressed image is to be added, the new compressed image (as well as the other images already in the buffer) is truncated so that all compressed data fit into the buffer. The resulting images have relatively low quality variation within a “sliding time window” corresponding to the length of the buffer employed. However, quality can vary widely over time-frames larger than the length of the sliding window.

In Smith and Villasenor (5), images are coded to a fixed Peak-Signal-to-Noise Ratio (PSNR) using the method mentioned in the first paragraph of (2, Section 8.2.1). This method yields constant quality as defined by PSNR, but control of average rate is not possible. The resulting rate (or equivalently, the total size of the resulting digital cinema package) is unpredictable.

None of the methods described above discuss any facility to place constraints on subsets of image data, such as individual images, individual components, etc. Such a facility is important for application to digital cinema. Indeed, the Digital Cinema Initiatives Specification (1), places a strict limit on the size of each individual image. In particular, no compressed image shall exceed 1,302,083 bytes in length.[1] Additionally, no compressed color component from a 2K image shall exceed 1,041,666 bytes in length. The 2K portion of a 4K image must also satisfy this latter constraint.

SATISFYING THE CONSTRAINTS

In this section, we describe a rate control method that maximizes quality while satisfying all required constraints.

The method begins by defining subsets of coding passes. A subset defined on an individual image is referred to as an image-wise subset. Image-wise subsets are useful in defining constraints on individual images. For example, in the 2K case, compressed size limits are placed on each component of an image, as well as on the entire image. Thus, the coding passes of an image would be grouped into three subsets corresponding to the three image components, together with a forth subset comprising of all coding passes in the image.

For each image, the proposed rate control method selects the coding passes having largest distortion-rate slopes from each image-wise subset such that the total size of the selected coding passes from that subset satisfies any image-wise constraints for that subset. The selected coding passes, distortion-rate slopes and lengths are saved for each image. Non-selected coding passes are discarded. Once all of the images are so processed, the method selects the coding passes having the largest distortion-rate slopes from those remaining in the aggregate of all images. Such coding passes are selected so that the total size satisfies the total desired compressed size of the image sequence. The selected coding passes are used to form the final JPEG2000 codestreams.

More concretely, assume a 2K image is to be encoded. Coding passes from the first (transformed) component are selected until the maximum allowable total length of 1,041,666 bytes (including all header information) is reached. The coding passes so selected are those having the largest distortion-rate slopes among those in the first component. Coding passes from the first component that are not selected are discarded. This process is repeated for the second and third (transformed) components. From the aggregate (over all three components) of all remaining coding passes, those having largest distortion-rate slopes are selected until the maximum allowable total length of 1,302,083 bytes (including all relevant header information) is reached.

At this point, the coding passes with largest slopes have been chosen. Thus, the quality of the image is maximized, as allowed by the required constraints. Extension to 4K is similar, but the component constraint of 1,041,666 bytes is only applied to the 2K portion of each component.

The process described above is repeated for each image in the sequence. Since no dependencies between images exist, multiple images can be processed in parallel. It is worth noting that if fixed rate encoding is desired, the process can end here. The images

---

[1] For simplicity, we state here only the constraints for the case of 24 frames (images) per second.

obtained will be encoded at 250 Mbit/s[2] and will satisfy all constraints. If a lower (fixed) rate is desired, a suitable value can be substituted for 1,302,083 in the description above.

Once the constraints on individual images are satisfied as described above, additional rate control can be performed over the entire sequence in an optimal fashion. Suppose a desired total size for the compressed sequence is known. A desired average encoding rate is easily converted into a total size. For example, 30 minutes of content at 125 Mbit/s yields a compressed size of 2.8125×10^9 bytes. From the aggregate of all coding passes of all images, as selected above, coding passes are further selected until the total desired size is achieved. The selection criteria is again, to select the coding passes having largest distortion-rate slopes. Non-selected coding passes are discarded. The final image codestreams are formed from the selected coding passes.

The proposed approach maximizes reconstructed image quality within the relevant subsets (as feasible within the constraints). Specifically, the algorithm maximizes quality for individual images by selecting coding passes having highest distortion-rate slopes until maximum allowable sizes are reached. The algorithm further maximizes average quality for the entire sequence by subsequently selecting the coding passes having highest distortion-rate slopes until the desired size for the entire compressed sequence is reached.

IMPLEMENTATION

The rate control algorithm described above employs the process of selecting coding passes having largest distortion-rate slopes within a subset. This process can be accomplished in many ways. One method sorts coding passes in descending order of their distortion-rate slopes and then selects from the top of the list. Alternately, a significance threshold is set for a subset with the idea that all coding passes with distortion-rate slopes above the threshold will be selected. This can be seen as equivalent to selecting a number of coding passes from the top of an ordered list of the subset by considering Figure 3.

In Figure 3 each solid horizontal line represents a coding pass and each stack of horizontal lines represents a codeblock. The collection of all coding passes from these codeblocks represents a subset**.** In the figure, distortion-rate slopes are indicated next to each coding pass. It is clear that selecting all coding passes of each codeblock having a significance value above a threshold set at nineteen, (as indicated by the dashed horizontal lines) is equivalent to sorting all coding passes in the subset and then selecting fifteen coding passes from the top of the list. If the total length of the coding passes thus selected is not as desired, the threshold can be adjusted and the process repeated until the total length is as desired (within some tolerance). It is important to note that no recoding of data is required. The iteration is carried out using distortion-rate slopes and lengths previously computed. This approach can have computational and memory advantages over actually sorting and listing distortion-rate slopes or coding passes. Increasing the significance threshold results in a smaller total size of selected coding passes and decreasing the significance threshold results in a larger one. The significance threshold may be varied using a number of techniques including trial and error, bisection, gradient descent or any other 1-D numerical search technique. Rather than iterating it is also possible to test multiple thresholds in parallel.

---

[2] 1,302,083 bytes/frame × 24 frames/sec × 8 bits/byte = 250 Mbit/s

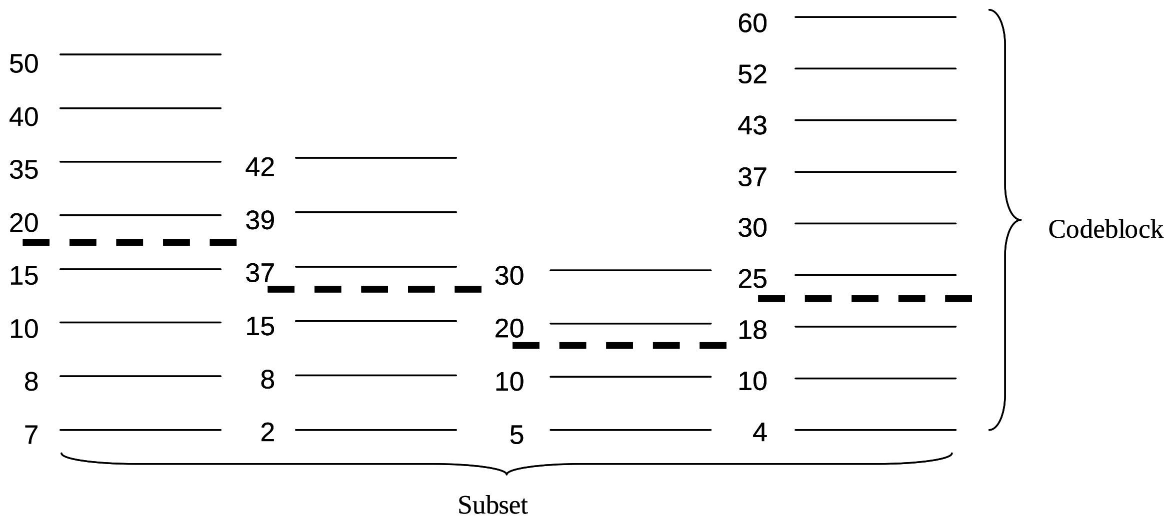

Figure 3 - Coding passes and distortion-rate slopes

As mentioned previously, it is possible to carry out the initial coding (generation of coding passes and satisfying constraints on individual images) in parallel. It is also possible to parallelize much of the work required to select coding passes to satisfy the total desired sequence length. Assume each processor has the coding passes, distortion-rate slopes, and coding pass lengths for one or more images. A control processor can broadcast a significance threshold for each subset to all processors. Each processor can return the total size for its images corresponding to the broadcast threshold. The control processor can sum these sizes and broadcast new thresholds, iterating until the desired total sizes (sums) are achieved. The control processor can then request each processor to create and output the final codestreams for its images from the coding passes having distortion-rate slopes above the final threshold.

From the discussion above, the final codestreams contain all coding passes of all images within the sequence having distortion-rate slopes above a significance threshold less those discarded due to any relevant image-wise constraint. Thus, if no image-wise constraint is in effect, high rate quantization theory can be used to show that the quality will be (roughly) constant from image-to-image. When image-wise constraints come in effect, significant quality deviations may occur when a particular image is very difficult to encode as compared to others in the sequence. For such an image, coding passes are discarded according to any image-wise requirement and quality may fall for that image. The ability to achieve constant quality is thus governed by the relative strictness of the image-wise constraints, compared to the desired average encoding rates. It should also be noted that when an image is extremely easy to code compared to others in the sequence, quality may be higher than that of the other images, even though its compressed size might be very low since in these cases, high rate theory does not apply.

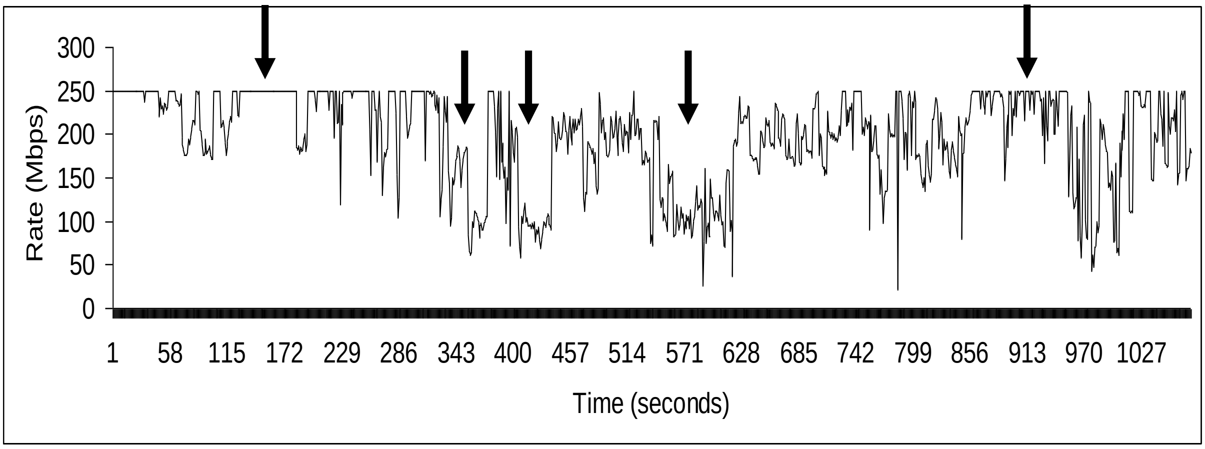

Figure 44 - Bit rate vs. time for 1 reel of digital cinema content

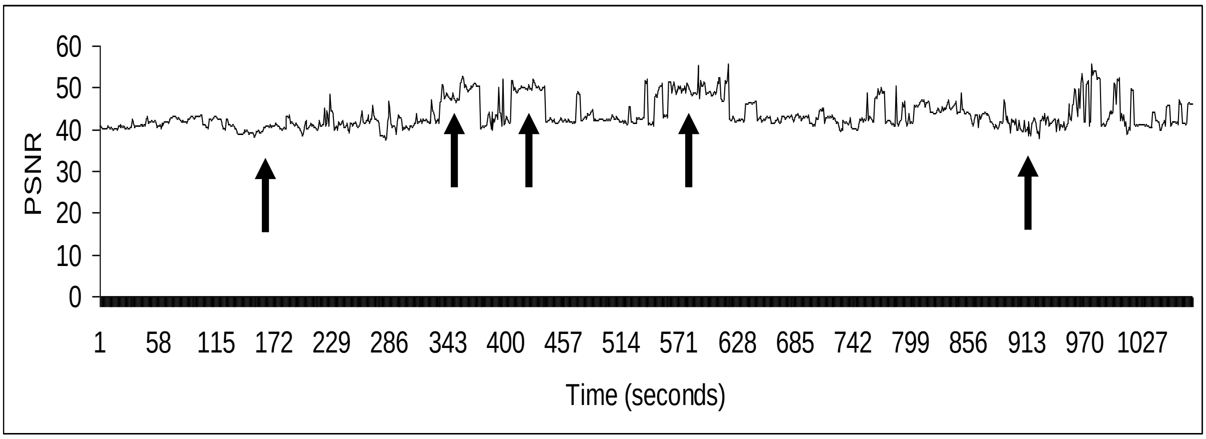

Figure 5 - PSNR vs. time for 1 reel of digital cinema content

RESULTS

Figure 4 and Figure 5 show results obtained by the DTS Digital Cinema Encoder™ as applied to one reel of a 4K motion picture. For the example depicted in these figures, the encoder employed the rate allocation algorithm described in this paper. The bit rates achieved are shown in Figure 4. This figure depicts encoding rate in Mbit/s vs. elapsed time for an average of 195 Mbit/s. It can be seen that the encoding rate varies widely for this example. In particular, the bit rate in Figure 4 ranges from a low of about 25 Mbit/s to a high of 250 Mbit/s (the maximum allowed by the DCI specification). Results for other content are similar. Near the center of the graph in Figure 4 are three arrows indicating long periods of low rates. These periods correspond to image frames that are “easy” to compress. Near the left and right ends of Figure 4 are two arrows indicating long periods of high rates. Indeed, the left arrow indicates a sustained period where the bit rate is at the maximum allowed value of 250 Mbit/s. These two periods correspond to images that are “difficult” to compress. In effect, the algorithm is “taking” bits from the easy images and “giving” them to the difficult images, while maintaining the average rate for the whole reel at precisely the pre-selected value of 195 Mbit/s.

Figure 5 shows the peak signal-to-noise ratio (PSNR) achieved for the rate allocation shown in Figure 4. It is interesting to note that the lowest PSNR results occur where the rate is highest. Periods where this occurs are indicated by the arrows at the left and right of the figure, and correspond to the identically placed arrows in Figure 4. As discussed previously, the image-wise constraint comes into play during these periods and limits the bits allocated to these difficult images, thereby limiting their PSNR values. It is also worth noting that the largest PSNR values occur where rates are lowest. Examples of this are indicated by the three arrows near the center of Figure 5, corresponding to the identically placed arrows in Figure 4. This indicates that although bits are taken from the easy frames, the resulting quality is still high. In particular, the quality of the easy frames is maintained at or above the levels of other image frames.

SUMMARY

We have presented a rate allocation scheme for JPEG2000 encoding of digital cinema image frames. The encoder is variable bit rate (VBR) and assigns more bits to difficult images and fewer bits to easy images. It does so in a way that maximizes the average quality over all images in a reel, or even over an entire motion picture. At the same time, all bit rate constraints as specified in (1) are guaranteed to be satisfied.

REFERENCES

1. Digital Cinema Initiatives, LLC, 2005. Digital cinema system specification V1.0. www.dcimovies.com, July 2005.
2. Taubman, D. S. and Marcellin, M. W., 2002. JPEG2000: Image compression fundamentals, standards, and practice. Kluwer Academic Publishers, Boston, 2002.
3. Tzannes, A. and [redacted].Gut R. US Patent Application US 2004/0047511 A1.
4. Dagher, J. C., Bilgin, A. and Marcellin, M. W., 2003. Resource constrained rate control for motion JPEG2000. IEEE Transactions on Image Processing. December, 2003. pp. 1522 to 1529.
5. Smith, M. D. and Villasenor, J., 2006. JPEG2000 rate control for digital cinema. Proceedings of SMPTE Advanced Motion Imaging and VSF VidTrans 2006. January, 2006.