J. Indian Inst. Sci., Mar.–Apr. 1999, 79, 99–111.

Rate control of a JPEG compression system

V. SELVARAJ AND K. R. RAMAKRISHNAN
Department of Electrical Engineering, Indian Institute of Science, Bangalore 560 012, India
email: krr@ee.iisc.ernet.in

AND

S. H. SRINIVASAN
Institute of Robotics and Intelligent Systems, Bangalore 560 001, India.

Abstract

In real-time systems, like remote-sensing systems onboard satellites or aircraft, where the image is generated line by line, the use of JPEG compression technique results in variable data rate. For transmission of this data over a constant bit rate (CBR) channel there is a need for buffering the output of the compression system. In this paper, two techniques have been proposed for controlling the rate while avoiding buffer overflows and underflows. Both the techniques have been simulated and the results are presented.

Keywords: Rate control, JPEG, image compression.

1. Introduction

Joint Photographic Experts Group (JPEG) still-picture compression standard[1] is widely used in many applications. The technique used in this standard is based on discrete consine transform (DCT) of the image pixels followed by quantisation and entropy-coding. The entropy-coding module is a variable length coder (Huffman/arithmetic coding) and hence the data rate of the output of JPEG compression system is not constant.

In real-time systems like the remote-sensing systems onboard satellites or aircraft, where the image is generated line by line, compressed using JPEG standard and transmitted over a CBR channel, the data should be buffered. This buffer should not be allowed to overflow as it would result in a loss of data. When the buffer underflows, that is, when the buffer becomes empty, dummy data should be introduced. The dummy data should be discarded at the receiver end. Buffer overflows and underflows can be avoided by suitably quantising the DCT coefficients. This is referred to as rate control.

In this paper, two rate-control techniques have been developed, which determine a suitable quantisation parameter *QP* (see Section 2) for every slice,[a] while avoiding buffer overflows and underflows. The first technique is called continuous quantisation technique (CONTQP), where the *QP* is computed based on the buffer depth by assigning a target bit rate for every slice. The second technique is called step quantisation technique (STEPQP), where *QP* is selected from a predefined set of values, based on the current buffer depth. The rest of the paper

[a] In this paper, a slice implies an eight-line segment of the full width of the image.

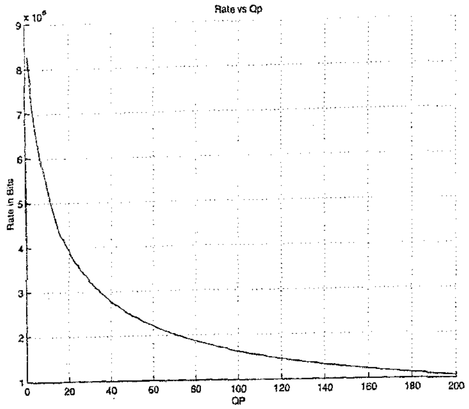

FIG. 1. Rate R vs QP.

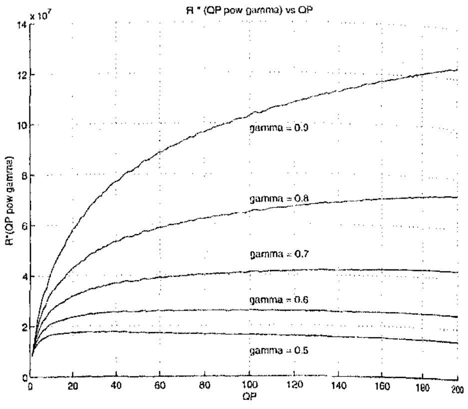

FIG. 2. $(R \times QP^{\gamma})$ vs QP.

is arranged as follows. In Section 2, JPEG-compressed data format is described followed by a discussion on ways of incorporation of rate-control information in it. In Section 3, the proposed rate-control techniques are explained. In Section 4, simulation results are discussed. Section 5 concludes the paper.

2. Description of the JPEG-compressed data format

In the JPEG standard, the bitstream syntax of the interchange format consists of marker segments and entropy-coded data segments. The marker segments consist of a marker followed by a sequence of related parameters. Markers serve to identify the various structural parts of the compressed data format. The first byte of a marker is always 0 × FF. The second byte, which is not equal to 0 × 00 or 0 × FF, is a value corresponding to the marker type. Entropy-coded data segment contains the output of an entropy-coding module which can be either Huffman or arithmetic. It contains an integer number of bytes. There is a frame between every start of image (SOI) and end of image (EOI) markers. Each frame contains an optional table specification or miscellaneous marker segment, a frame header followed by any number of scans. Each scan consists of an optional table specification or miscellaneous marker segment, followed by a scan header and any number of entropy-coded segments (ECS). Each ECS contains a sequence of minimum coded units (MCUs). The frame header contains a start of frame marker, the frame header length, sample precision in bits, number of lines in the source image, number of samples per line in the source image and number of source image components. For each source image component, horizontal sampling factor and vertical sampling factor in each MCU are specified. Besides, for each source image component, the quantisation table selector, T_{q_i}, is included. This specifies the quantisation table (out of the four possible quantisation tables that can be defined) that is used for quantisation. The scan header contains the start of scan (SOS) marker, the scan-header length, the number of source image components in the scan, the DC entropy-coding table selector, T_{d_j}, and the AC entropy-coding table selector, T_{a_j}, specify which of the four possible DC and AC entropy-coding tables is used.

Quantisation tables, and DC and AC entropy-coding tables can be specified at the start of the frame or of the scan. However, in typical JPEG applications, these tables are defined once at the start of a frame and used for the entire image. But in applications where rate control is required, the quantisation table should be varied within the image at suitable intervals. One possibility to achieve this would be to use a new table every scan and specify the quantisation table in the scan header of every scan. But this would result in a considerable increase in data.

In the Independent JPEG Group (IJG) source code available in the public domain,[4] the quantisation table is generated using the default quantisation table for luminance component recommended in the standard.[1] The user specifies a quality factor QF which can range from 1 to 100. Higher value will result in better quality. The QF is then converted to a quantisation parameter QP as follows:

$$QP = \begin{cases} \frac{5000}{QF} & \text{if } QF < 50 \\ 200 - 2 \times QF & \text{otherwise} \end{cases} \tag{1}$$

The QP is multiplied with all the elements of the default quantisation table to get the quantisation table to be used.

Thus, in rate control, different QPs can be used to generate different quantisation tables for different portions of the image. The information required at the decoder is the QP used for each portion of the image. This information can be sent as an additional byte[b] in the higher transport layer of the bitstream. Alternatively, this information can be sent as an application data segment which can be incorporated in the beginning of the scan. In both the options, the syntax becomes application specific. However, since the rate control requirement is of application-specific nature, this will not be a limitation.

3. Proposed rate-control techniques

Two rate-control techniques discussed in this section use a slice-level selection of the QP which is used to scale the elements of the default quantisation table.

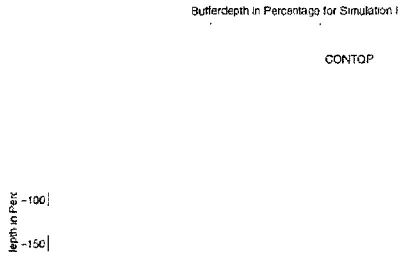

FIG. 3. $B_{\%}(i)$ for Simulation I.

---

[b] The value of QP if unrestricted will require two bytes to represent.

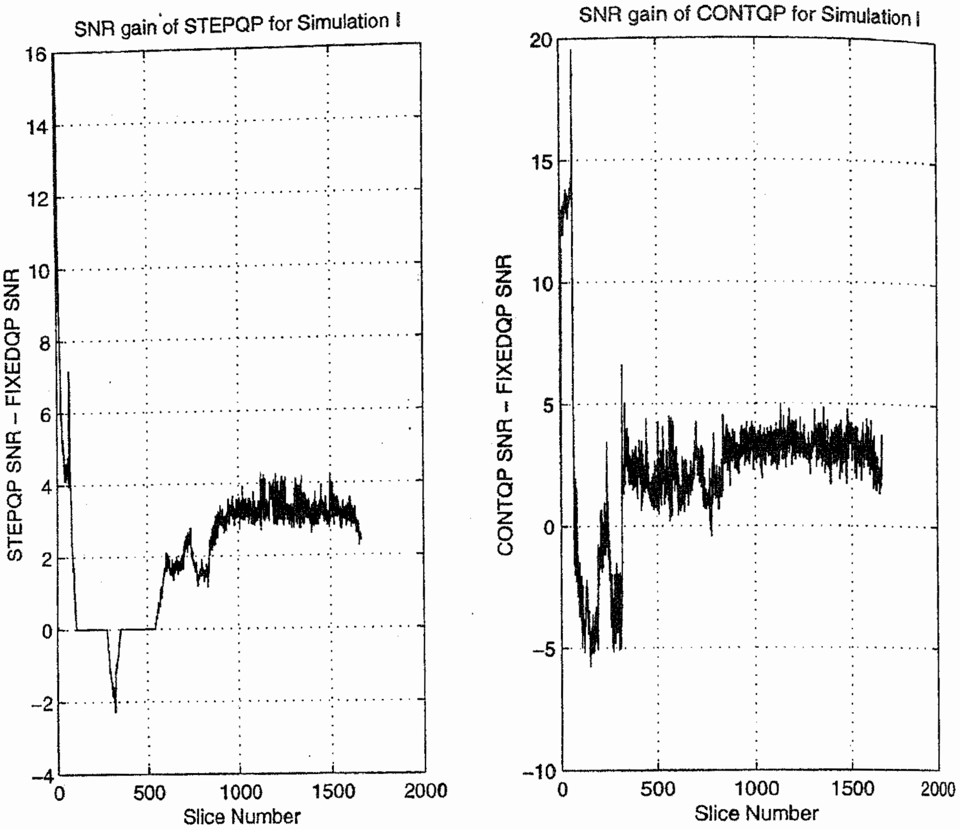

FIG. 4. SNR gain for Simulation I.

3.1. *CONTQP*

Let $QP(i+1)$ and $T(i+1)$ denote the quantisation parameter to be used for the $(i+1)$th slice and the target bit rate in bits for the $(i+1)$th slice, respectively. In this technique, $QP(i+1)$ is determined in two steps. In the first step, $T(i+1)$ is computed. In the second, a suitable $QP(i+1)$ is computed to achieve the target bit rate of $T(i+1)$.

3.1.1. *Target bit rate $T(i+1)$ computation*

$T(i+1)$ is computed based on $B(i)$, the current buffer depth in bits after encoding ith slice. Let the number of bits that will be read out from the buffer in a slice duration be denoted by *BR_slice*, which is given by:

$$BR\_slice = channel\_rate \times (8 \times line\_period)$$

where *channel_rate* is the channel rate in bits per second and *line_period* is the time in seconds in which each line of the image is generated. Let *B_Size* denote the total buffer size of the encoder in bits. In order to avoid a buffer overflow,

$$B(i) + T(i + 1) \leq B\_Size + BR\_slice.$$

Similarly, in order to avoid a buffer underflow,

$$B(i) + T(i + 1) \geq BR\_slice.$$

The operating level of the buffer depth (OL_buffer) given by $OL\_buffer = \alpha \times B\_Size$ is controlled by the parameter α in the range $0 < \alpha < 1$. In order to maintain the buffer depth at the prescribed operating level of OL_buffer, $T(i + 1)$ is assigned a value of $(OL\_buffer - B(i) + BR\_slice)$. Finally, $T(i + 1)$ is also ensured to be at least $0.9 \times BR\_slice$ and is not allowed to be higher than $1.5 \times BR\_slice$. This is done to avoid very less or very high values of $T(i + 1)$ which would result in large variations in quality.

3.1.2. *Computation of $QP(i + 1)$*

In this step, quantisation parameter $QP(i + 1)$ which is to be used for $(i + 1)$th slice is computed so that the resulting number of bits is $T(i + 1)$ bits. For this, a relationship between QP and the rate R, the number of bits generated, is required. Figure 1 shows the plot of R vs QP for an image of size 1024 × 1024, encoded using different QPs. From this we note that R shows a damped inverse relationship with QP. Hence, R could be expressed as:

$$R = \frac{X}{QP^{\gamma}} \tag{2}$$

where γ is a constant in the range $0 < \gamma \leq 1.0$ and X is a constant called complexity measure. From this, X can be expressed as

$$X = R \times QP^{\gamma}. \tag{3}$$

Figure 2 shows the plot of $R \times QP^{\gamma}$ vs QP, for various γ. For $\gamma = 0.6$, this plot is almost constant, except for very low values of QP. Hence, $\gamma = 0.6$ is chosen for the relationship in eqn 2. The value X is analogous to the global complexity measure defined in Test Model 5 of MPEG-2 Video.[3] After encoding the ith slice, $X(i)$, the complexity measure of ith slice can be computed as:

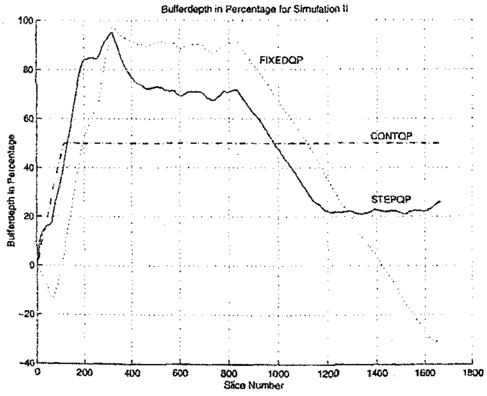

FIG. 5. $B_{\%}(i)$ for Simulation II.

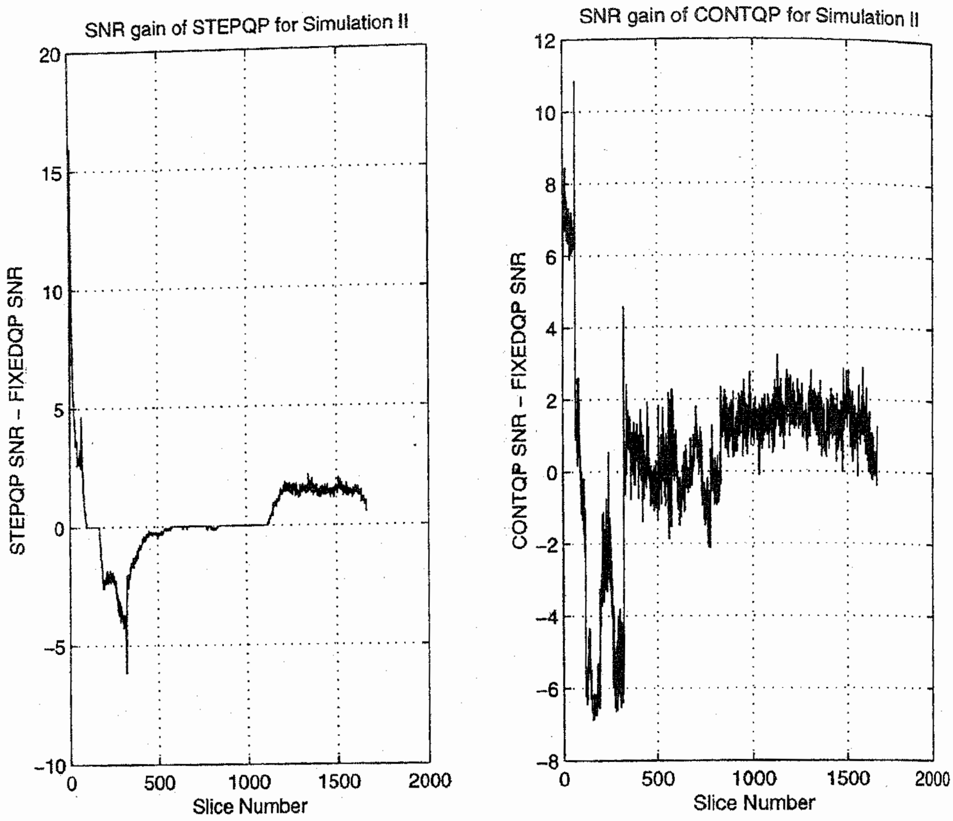

FIG. 6. SNR gain for Simulation II.

$$X(i) = S(i) \times (QP(i))^{\gamma} \tag{4}$$

where $S(i)$ is the number of bits generated for the ith slice. $QP(i+1)$ can be computed using $X(i)$ and $T(i+1)$ as:

$$QP(i+1) = \left(\frac{X(i)}{T(i+1)} \right)^{\frac{1}{\gamma}} \tag{5}$$

where $X(i)$ has been used instead of $X(i+1)$ assuming the change in X between successive slices to be less. Finally, $QP(i+1)$ is clipped to allowed limits of 1 and $3 \times QP_{des}$ in case it exceeds these limits. QP_{des} is the desired QP which is computed from QF_{des}, the desired QF.

3.2. *STEPQP*

In this technique, $QP(i+1)$, the QP to be used for the $(i+1)$th slice is selected from a predefined set of values. QP is allowed to vary from 1 to $2 \times QP_{des}$, the number of different values

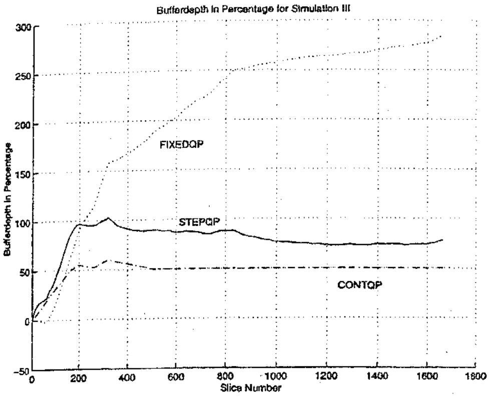

FIG. 7. $B_{\%}(i)$ for Simulation III .

that QP can assume being $N_{qp} = 2 \times QP_{des}$. $QP(i+1)$ is selected based on the current buffer depth percentage $B_{\%}(i)$ which is expressed as

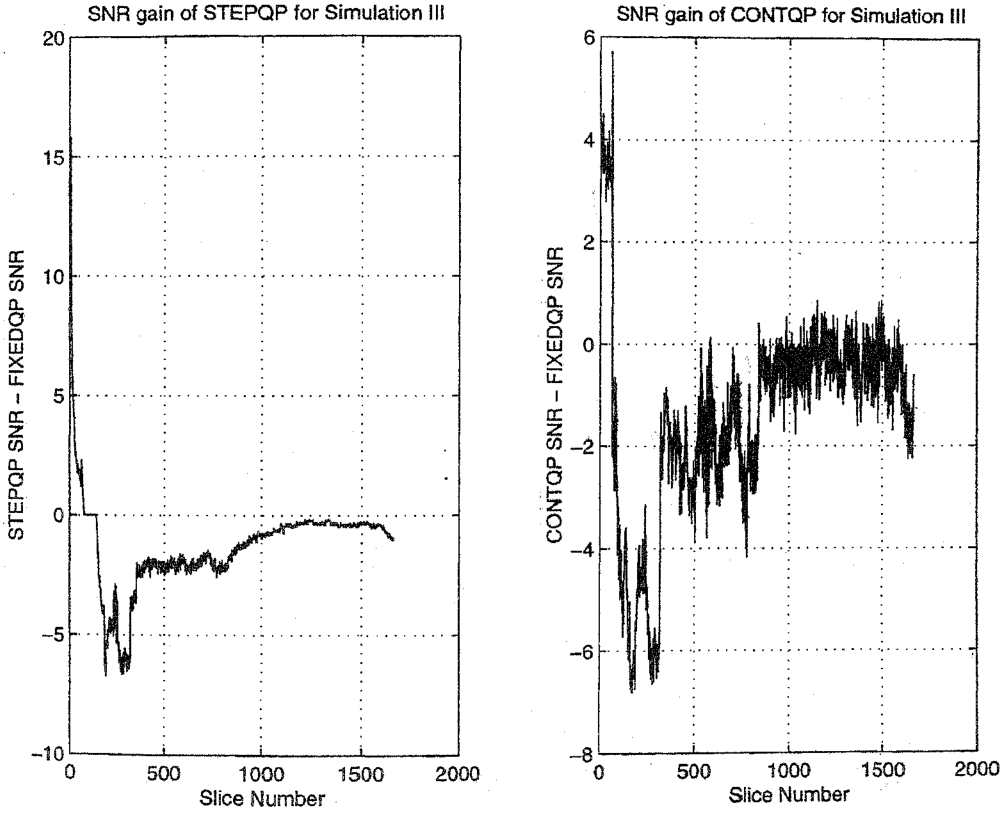

FIG. 8. SNR gain for Simulation III.

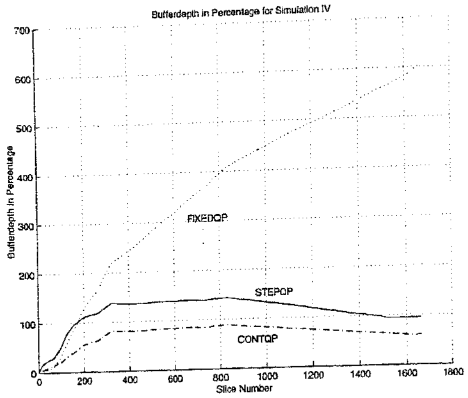

FIG. 9. $B_{\%}(i)$ for Simulation IV.

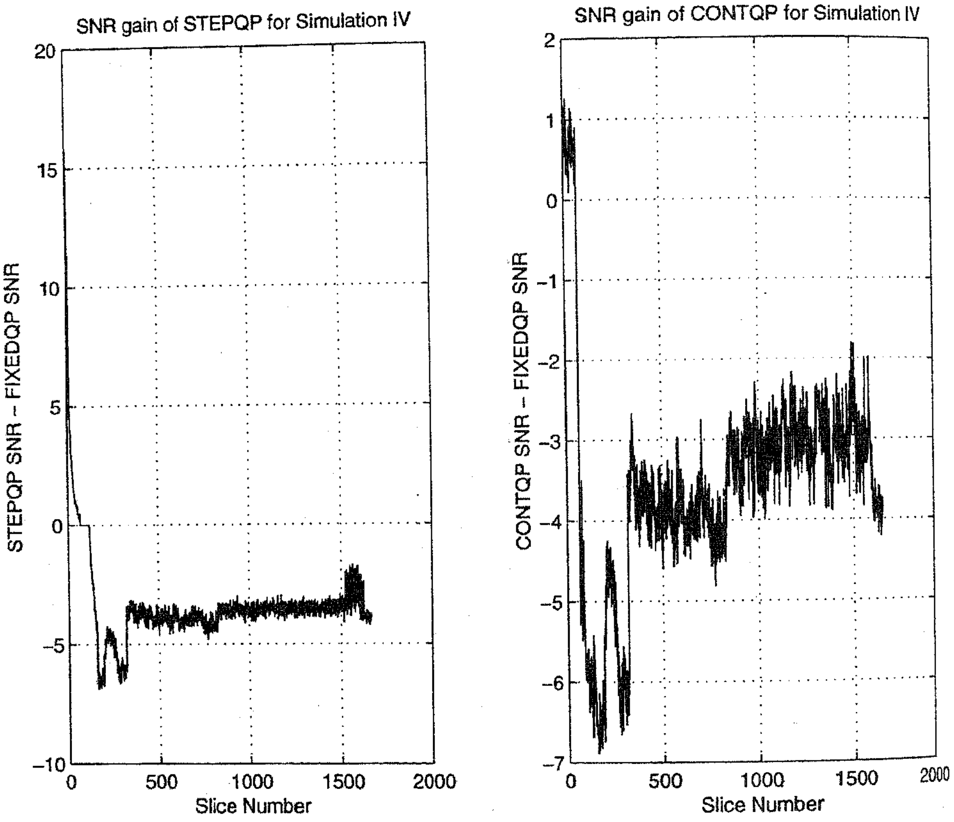

FIG. 10. SNR gain for Simulation IV.

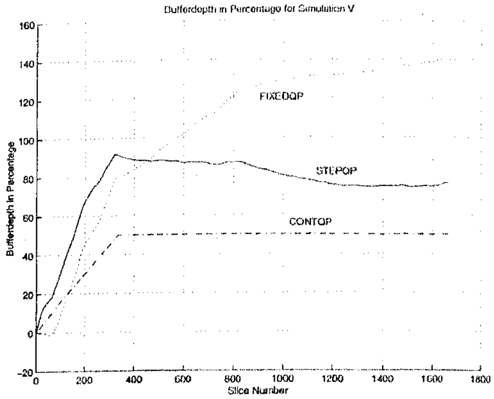

FIG. 11. $B_{\%}(i)$ for Simulation V.

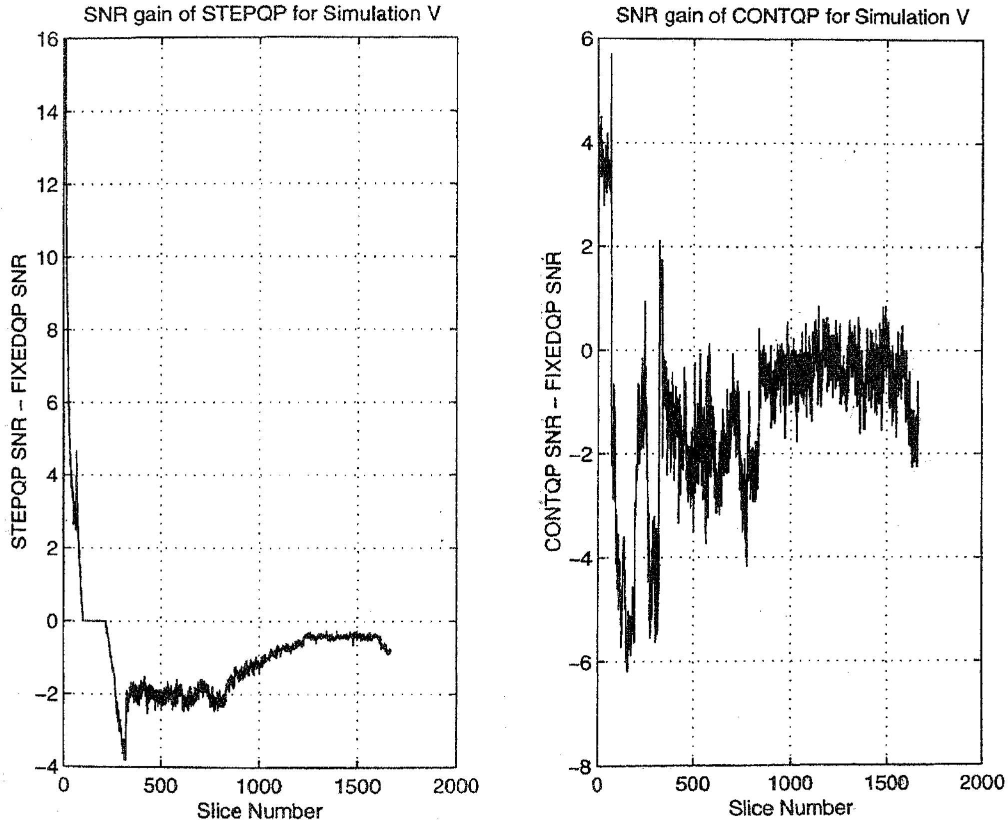

FIG. 12. SNR gain for Simulation V.

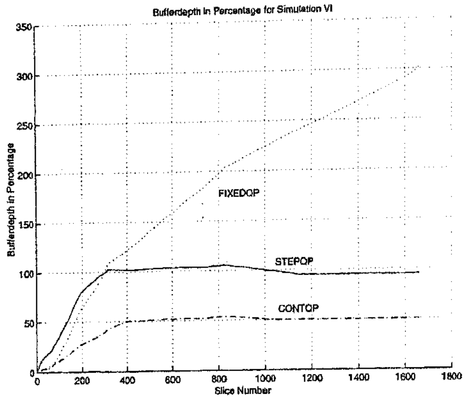

FIG. 13. $B_{\%}(i)$ for Simulation VI.

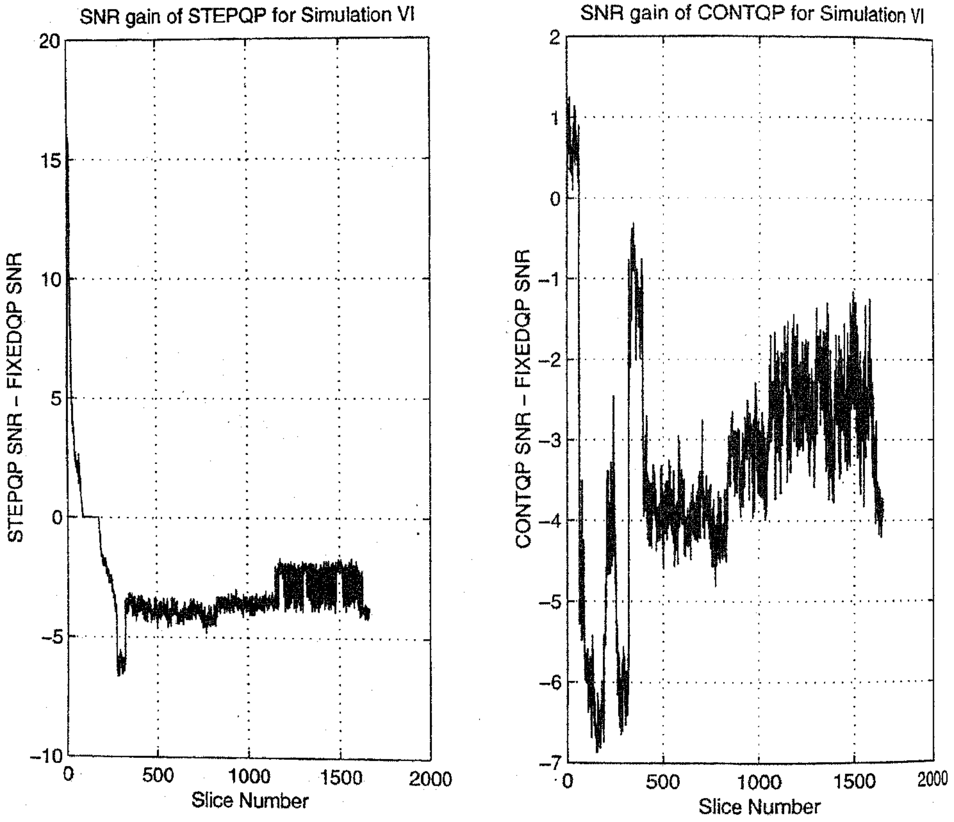

FIG. 14. SNR gain for Simulation VI.

$$B_{\%}(i) = \frac{B(i)\times 100}{B\_Size}.$$

When $B_{\%}(i)$ is in $(50 - \delta) \le B_{\%}(i) \le (50 + \delta)$, QP_{des} is used. Here, δ is used to control the range of $B_{\%}(i)$ in which a constant desired quantisation parameter QP_{des} is to be used. When $B_{\%}(i)$ moves away from 50%, an incremented or decremented value of QP_{des} is suitably used. This is summarised below:

$$QP(i+1) = QP_{des} + \Delta QP(i+1) \tag{6}$$

where, $\Delta QP(i+1)$

$$\begin{array}{ll}
1 - QP_{des} & B_{\%}(i) < 5 \\
-1 \times \frac{(50-\delta)-B_{\%}(i)}{\frac{50-\delta}{0.5\times N_{qp}}} & 5 \le B_{\%}(i) < (50-\delta) \\
0 & (50-\delta) \le B_{\%}(i) \le (50+\delta) \\
\frac{B_{\%}(i)-(50-\delta)}{N_{qp}} & (50+\delta) < B_{\%}(i) \le 95 \\
2 \times QP_{des} & B_{\%}(i) > 95
\end{array}$$

4. Simulation results

Both these techniques of rate control were implemented using the library functions of the IJG software and simulated on remote-sensing images with one component per pixel and 8 bits per component. Baseline JPEG mode with floating point DCT was used. A single image with a width of 512 pixels and a height of 13312 pixels was created by concatenating remote-sensing images. A *line_period* of 250μs, a QF_{des} of 85 and a corresponding QP_{des} of 30 have been used. Six simulations were carried out (Table I). $\alpha = 0.5$ was used for CONTQP and $\delta = 20$ for STEPQP in all the simulations. In each simulation, initially the image was compressed and decompressed without rate control using a fixed quantisation parameter of QP_{des}. This mode is referred to as FIXEDQP mode. Subsequently, the image was compressed and decompressed using STEPQP and CONTQP techniques.

Figures 3, 5, 7, 9, 11 and 13 show the buffer depth percentage for the six simulations and Figs 4, 6, 8, 10, 12 and 14 the SNR gain for STEPQP and CONTQP techniques over FIXEDQP mode. Figure 15 shows the buffer depth percentage for CONTQP technique for various values of α. Figure 16 shows the SNR gain of STEPQP over FIXEDQP for $\delta = 0$ and 30. From these results it is noted that CONTQP achieves constant buffer depth percentage and STEPQP allows variations in buffer depth to give a better SNR performance compared to

Table I
Parameters used

Simulation	I	II	III	IV	V	VI
channel_rate (Mbps)	5	4	3	2	3	2
B_Size (Mbits)	1	1	1	1	2	2

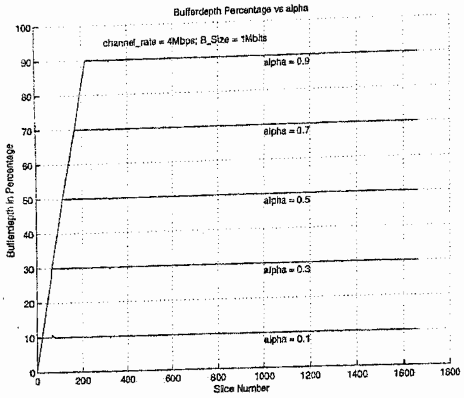

FIG. 15. $B_{\overline{\alpha}}(i)$ of CONTQP for various values of α.

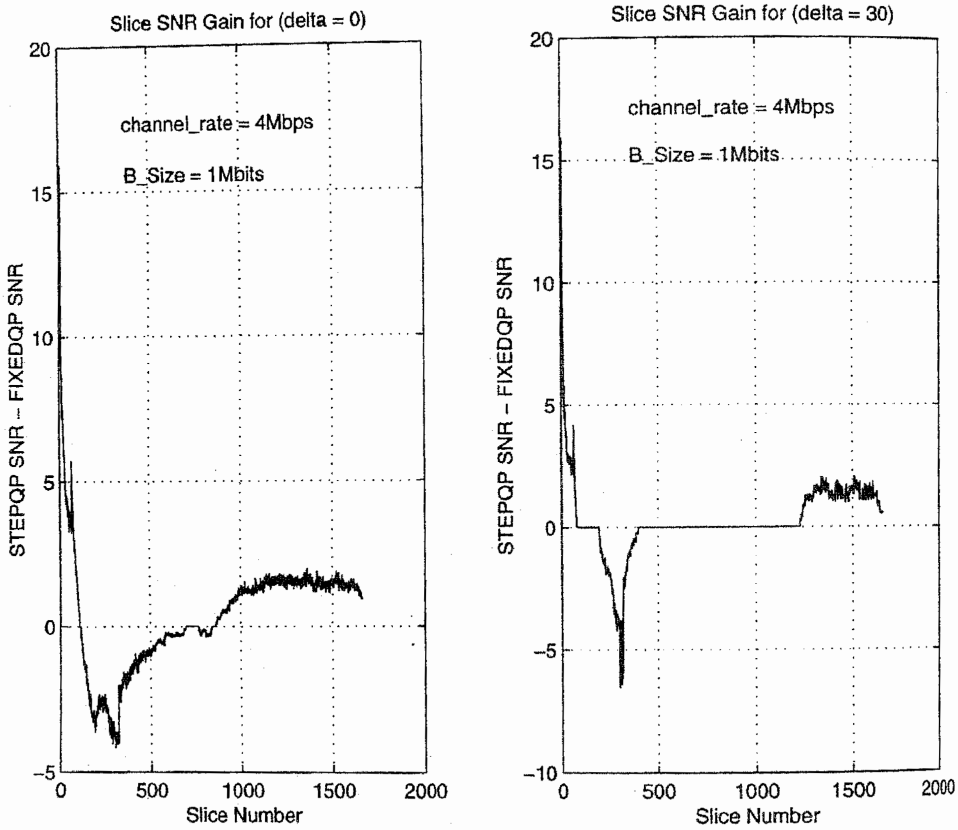

FIG. 16. SNR gain of STEPQP for various values of δ.

CONTQP. In STEPQP, by choosing a large value of δ, the desired quality factor QF_{des} can be achieved for a larger portion of the image. However, sharp changes in SNR could result in some portions as is evident from the figure. Simulation IV uses a very low data rate of 2 Mbps and it is found that STEPQP has buffer overflows. CONTQP also has a variation in buffer depth percentage though there are no overflows. However, in Simulation VI, for the same data rate with increased *B_Size* the performance of both the techniques has improved. Thus, for a given class of images and a desired quality factor QF_{des}, the *channel_rate* and *B_Size* should be chosen by simulations. When the *channel_rate* is a constraint, *B_Size* should be suitably sized and when it becomes a constraint the *channel_rate* should be suitably chosen.

6. Conclusions

Two rate-control techniques have been proposed for real-time systems like remote-sensing systems onboard satellites or aircraft, using JPEG-based compression. Their SNR performance has been compared to that of the uncontrolled FIXEDQP mode. STEPQP attempts to achieve the desired quality by utilising the available buffer depth within the constraints of *B_Size* and *channel_rate*. Whenever the buffer depth percentage is less than 50%, the performance is better than the desired level. Unlike in video transmission, buffer delays are not a major concern in applications where online decoding of the compressed images is not a requirement. STEPQP would be preferred in such applications. CONTQP maintains the buffer depth at a desired level. This technique could be used for constant delay applications. Both the techniques require only simple calculations once in every slice and would not have any impact on operational speed. The simulation software developed for this work using the IJG library functions can also be used for choosing the *channel_rate* and *B_Size* for different classes of images.

References

1. Digital compression and coding of continuous-tone still images, Part 1, requirements and guidelines, *ISO/IEC DIS* 10918-1, Nov. 1991.

2. WALLACE, G. K. The JPEG still picture compression standard, *Commun. ACM*, 1991, **34**, 30–44.

3. *ISO/IEC (MPEG-2)*, Test Model 5, April, 1993.

4. Independent JPEG Group's (IJG) Software, Release 6a, February 1996, *ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6a.tar.gz*.