

# A Novel, Automatic Quality Control Scheme for Real Time Image Transmission

S. RAMACHANDRAN\* and S. SRINIVASAN

*Department of Electrical Engineering, Indian Institute of Technology-Madras, Chennai 600 036, TN, India*

*(Received 1 May 2000; Revised 5 December 2000)*

A novel scheme to compute energy on-the-fly and thereby control the quality of the image frames dynamically is presented along with its FPGA implementation. This scheme is suitable for incorporation in image compression systems such as video encoders. In this new scheme, processing is automatically stopped when the desired quality is achieved for the image being processed by using a concept called pruning. Pruning also increases the processing speed by a factor of more than two when compared to the conventional method of processing without pruning. An MPEG-2 encoder implemented using this scheme is capable of processing good quality monochrome and color images of sizes up to  $1024 \times 768$  pixels at the rate of 42 and 28 frames per second, respectively, with a compression ratio of over 17:1. The encoder is also capable of working in the fixed pruning level mode with user programmable features.

**Keywords:** Image quality; Pruning level; Image compression; Discrete cosine transform; Quantization; Variable length code

## INTRODUCTION

Image processing applications such as high definition television, video conferencing, computer communication, etc. require large storage and high speed channels for handling huge volumes of image data. In order to reduce the storage and communication channel bandwidth requirements to manageable levels, data compression techniques are imperative. It is of paramount importance that systems designed for these applications communicate with one another effectively and also offer connectivity and compatibility among different services. This requirement is met if these systems are designed to conform to international standards such as JPEG, H.261, HDTV, and MPEG. The basic operations that bring about image compression, namely, the Discrete Cosine Transform (DCT), Quantization (Q) and Variable Length Coding (VLC) are, however, common to all these standards. DCT and Q aid in making most of the picture energy to be concentrated in the first few low frequency coefficients, preparing the ground for effective compression. The resulting quantized DCT coefficients are Huffman coded

in the VLC coder, thus effecting further compression. The serial bit stream output is then sent out to the channel. DCT applied on an  $8 \times 8$  pixel block of image results in the generation of 64 coefficients in the raster scan order as depicted by 1,2,... up to 64 in Fig. 1. The diagonals, numbered from 0 to 14, are referred to as pruning levels (PL) and indicate the stage at which computation of DCT coefficients is stopped owing to insignificant contribution of subsequent coefficients towards the quality of the image.

In the recent years, VLSI and FPGA implementations have attracted considerable interest for applications in high speed image communication [1–7]. In order to reduce the processing time of still pictures, a modified adaptive pruning technique [8] was introduced at the DCT stage itself. This method, however, is not suitable for implementation on FPGAs both from the quality and exacting speed considerations encountered in motion pictures. In an earlier work [7], a user programmable, fixed pruning level-based control for video encoder has been implemented. Although good processing speed was obtained in that work, image quality was not satisfactory

\*Corresponding author. E-mail: ram@ee.iitm.ernet.in.

Portions reprinted, with permission, from ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28–31, 2000 Geneva, Switzerland. © 2000 IEEE

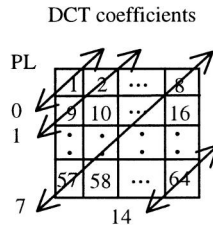


FIGURE 1 Pruning levels in an image block.

for some types of images for pruning level of about three, for which maximum processing speed is likely to be attained without sacrificing the quality. Further, block artifacts were visible appreciably for some of the images processed. The present work considerably reduces these limitations by evaluating the image quality dynamically as DCT coefficients are being computed. The processing for the current image block is immediately stopped when the desired quality is met, thus speeding up the system considerably while retaining the desired quality of the picture.

The present design implemented on FPGAs is capable of throughputs of 50 Mbps meeting MPEG 2 standards [9] for processing I frames including the header and color information. Additionally, it has the following advantageous feature: while computing the  $(n + 1)$ th block of DCTQ by the DCT and Quantization processor, the  $n$ th block of VLC can also be executed in parallel since the  $n$ th block of DCTQ is already available. Thus, the two processes, namely, the DCTQ and the VLC can be pipelined. Pruning level-based control is incorporated in each of these two processes.

The next section presents a novel method for the computation of quality of the image processed dynamically. The third section describes the video encoder as implemented on FPGAs, whereas the fourth section deals with the architecture of the automatic quality based controller. Results and discussions are presented in the fifth section followed by conclusions.

## A NEW ALGORITHM FOR ASSESSING IMAGE QUALITY DYNAMICALLY

In transform coding, a signal is mapped from one domain, usually spatial or temporal, into the transform domain. The signal can be one-dimensional or multi-dimensional. DCT is an orthogonal transform consisting of a set of vectors that are sampled cosine functions. The mapping is therefore unique and reversible. In this case, the energy is preserved in the transform domain, and the signal can be recovered completely by the inverse transform. Since the DCT transform is orthogonal, implementing the inverse transform is essentially the same as implementing the

forward transform. Therefore, the properties such as fast algorithm, recursive structure are preserved in the inverse transform. In fact, the hardware, e.g. a VLSI chip, designed for the forward transform can be used with minor modifications for implementing the inverse transform. In image and video coding standards such as JPEG, MPEG, H.263, etc., two-dimensional-DCT is the primary factor in achieving compression.

Two-dimensional DCT of a block of size  $8 \times 8$  pixels is defined as

$$Z = CXC^T, \quad (2.1)$$

where  $X$  is the input image matrix,  $C$  the cosine coefficient matrix and  $C^T$ , its transpose. In order to achieve a regular and efficient method of implementation, a parallel matrix multiplication algorithm has been proposed in Ref. [3]. The two-stage matrix multiplication of Eq. (2.1) can be implemented by a parallel architecture wherein eight partial products, which are the row vectors of  $CX$  generated in the first stage, are fed to the second stage. Subsequently, multiplying the row vector of  $CX$  by the  $C^T$  matrix generates eight DCT coefficients, corresponding to a row of  $CXC^T$ . While computing the  $(i + 1)$ th partial products of  $CX$ , the  $i$ th row (of  $CXC^T$ ) DCT coefficients can also be computed simultaneously since the  $i$ th partial products of  $CX$  are already available. Quantized outputs can be obtained by dividing each of the 64 DCT coefficients by the corresponding quantization table values. These stages can be pipelined in such a way that all the 64 quantized DCT coefficients are generated in 83 clock cycles.

In general from a visual perception viewpoint, the low frequency coefficients are much more sensitive than the high frequency coefficients. The energy is invariant to orthogonal transformation. The sum of the squares of all the DCT coefficients or the spatial data values is the energy of the block, i.e.

$$\sum_{u=0}^7 \sum_{v=0}^7 (\text{DCT}_{u,v})^2 = \sum_{n=0}^7 \sum_{m=0}^7 (X_{n,m})^2, \quad (2.2)$$

$$u, v, n, m = 0-7$$

Quantized DCT coefficients beyond pruning levels of about two are zero for most of the image blocks as shown in Fig. 2 for a sample  $8 \times 8$  block of DCT coefficients. Therefore, computation time is wasted in processing beyond these pruning levels which is the case in all the earlier methods cited [1–6]. This problem is solved in the present work by computing the sum of energy of AC coefficients lying on every diagonal commencing from PL1 up to PL14. PL0 is not taken into account for the energy computation since considerable amount of energy is packed in the DC coefficient, which needs to be processed. At each step, the computed energy is compared with a threshold energy,  $e_{\text{THR}}$ . If it is less than  $e_{\text{THR}}$ , the computation for the current image block is immediately

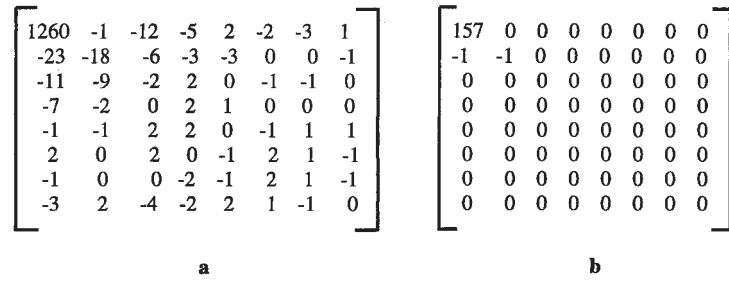


FIGURE 2 A sample image block. (a) DCT coefficients. (b) Quantized DCT coefficients.

terminated, and the processing for the next block commences, thus speeding the processing by over two times when compared to conventional approaches of computing up to PL14. The algorithm for this method is shown in Fig. 3. Applying this method for the DCT coefficients cited in the example, it is seen that the energies of the DCT coefficients for PL1–PL3 are 530, 589 and 191 units, respectively, while those for PL4–PL14 are each less than 23 units. By applying the energy threshold of 200 units, the final pruning level for this image block is evident to be two, as can be readily seen from Fig. 2b. With a few exceptions, exactly similar results were obtained for a large number of image blocks for a number of images that have been experimented. Therefore, the energy threshold of 200 units has been made as the default value in the implementation, although the user can reprogram the same. The image quality obtained by this method is quite good and is very close to that obtained by processing up to the full quality level of PL14 as presented in Results and discussion. This method of applying threshold energy has been arrived at after conducting exhaustive experiments with a number of images.

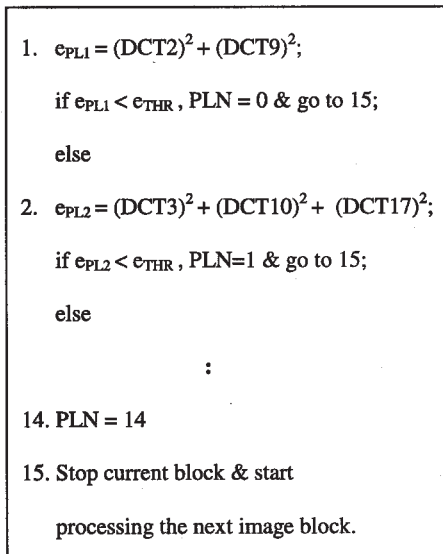


FIGURE 3 Algorithm for the computation of energy (quality) of an image.

## ARCHITECTURE OF THE VIDEO ENCODER

Figure 4 depicts the basic architecture of the implemented encoder. The image to be processed is input block by block, by a host computer such as Pentium, into the DCTQ processor [3], where the DCT is performed followed by quantization. The optimum pruning level, PLN, up to which the DCTQ is to be computed, is processed in the automatic quality controller circuit. The computed PLN, which is communicated to both DCTQ and VLC processors, changes dynamically from block to block depending upon the picture content and energy (which has a direct bearing on quality) computed. The resulting quantized coefficients are applied to the next stage, VLC [6], where they are assigned variable length codes and buffered by FIFO before they are sent out onto a serial channel as a bit stream. The color information, Y, Cb and Cr, are input once per macroblock. The energy threshold,  $e_{THR}$ , which is a measure of image quality, is user programmable. The encoder is also capable of processing up to a fixed pruning level.

## ARCHITECTURE OF THE AUTOMATIC QUALITY CONTROLLER

The automatic quality controller is shown in Fig. 5. It basically consists of a squaring circuit to evaluate  $(DCT)^2$ , adders/registers to accumulate 14 energy levels,  $e_{PL1}$  through  $e_{PL14}$ , registered comparators and a controller to

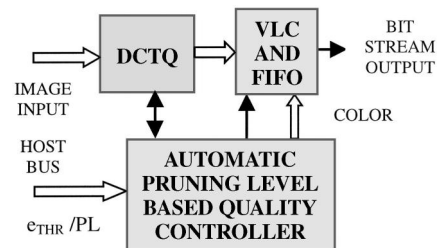


FIGURE 4 The basic architecture of the video encoder.

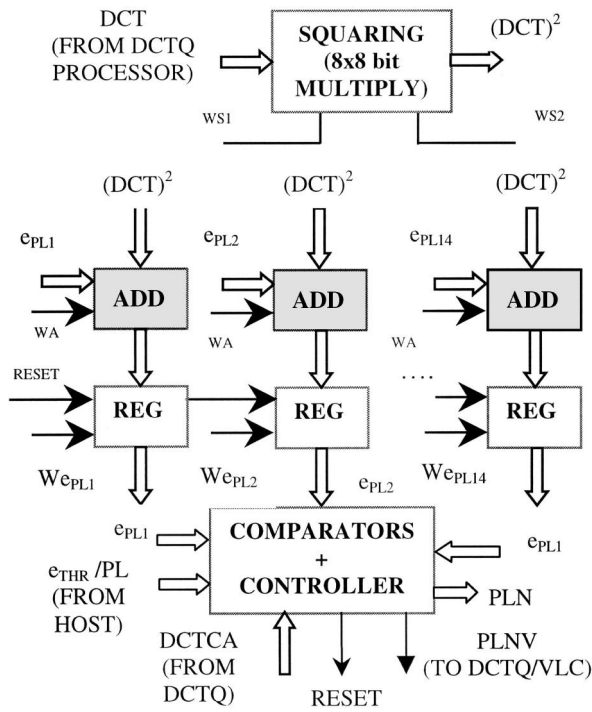


FIGURE 5 The architecture of automatic quality controller.

evaluate different steps of the algorithm given in Fig. 3. There are two modes of operations possible: fixed pruning level up to which the processing is required and automatic control of pruning level in order to get the desired quality level. The DCT coefficients generated in a raster scan order together with its address, DCTCA, are input to the controller one by one. Since squaring and additions are time consuming operations, they are pipelined using signals, WS1, WS2 and WA, derived from the DCTQ processor. When the DC coefficient is processed, the controller issues RESET signal to clear all the fourteen  $e_{PL}$  registers. While the subsequent AC coefficients are processed, the DCTQ processor generates write signals,  $We_{PL1}$  through  $We_{PL14}$ , as is appropriate. The comparator compares the accumulated energy for every pruning level with the threshold energy programmed and if the accumulated energy is less, then the controller outputs the pruning level number, PLN. Both DCTQ and VLC processors process the quantized DCT coefficients only up to this PLN level and not beyond, thus speeding up the entire system. The pulse, PLNV, signals the validity of the PLN.

## RESULTS AND DISCUSSIONS

The automatic, quality controlled encoder has been implemented using two pieces of Altera's EPLDs, 10K100ARC-1, with a chip set complexity of about 1,90,000 logic gates. The design has been realized by using both circuit diagrams and VHDL codes. Simulation

tests were carried out successfully at 50 MHz clock operation and verified by C and Matlab programs.

Table I shows the processing time of DCTQ for various pruning levels as implemented in the present design. DCTQ coefficients are processed in a raster scan order, left to right and top to bottom, assuming that they are arranged as an  $8 \times 8$  matrix. The number of coefficients processed for the pruning levels from 0 to 14 are as shown in Table I and are easily inferred from Fig. 1. The processing time of DCTQ is 800 ns for PL0 because the first DC coefficient is issued at the twentieth clock cycle owing to high pipelining inherent in the design. The execution times for subsequent pruning levels increase by the expression:

(Difference of number of DCTQ coefficients between two subsequent pruning levels)  $\times$  40 ns for every additional increase of pruning level since each coefficient is generated every clock cycle of DCTQ with a time period of 40 ns. From the table, it is clear that the execution times for DCTQ increases steeply for every additional pruning level up to level 7 and thereafter only gradually. Further, there is no appreciable improvement of visual quality for pruning level beyond 3 or 4 for most of the images. Therefore, by terminating the computation of DCTQ at that level where good quality of image is already met, the processing speed can be stepped up by about a factor of two or more when compared to the full pruning level of 14. The savings in DCTQ computation, thus obtained, are also extended to VLC processing.

The VLC is processed in a zig-zag order, diagonal-wise from top left hand corner. Hence the number of DCTQ coefficients lying on each diagonal increases by one up to PL7 and decreases by one thereafter. Also in practice, all the DCTQ coefficients for pruning levels greater than about six will be zero. Therefore, the number of cumulative coefficients processed for various pruning levels from 0 to 6 will be as shown in Table II. The computation of VLC for the quantized DCT coefficients depends on the picture content and, therefore, only a representative example is taken for the present treatment. In this example, the value of first DC coefficient is 1 for which the VLC size is 5 bits including 2 bits for end of

TABLE I Processing time of DCTQ for various pruning levels

Pruning level	No. of DCTQ coefficients processed	DCTQ processing time in ns
0	1	800
1	9	1120
2	17	1440
3	25	1760
4	33	2080
5	41	2400
6	49	2720
7	57	3040
8	58	3080
9	59	3120
10	60	3160
11	61	3200
12	62	3240
13	63	3280
14	64	3320



TABLE II Processing time of VLC for various pruning levels

Pruning level	No. of DCTQ coefficients processed	VLC size in bits	VLC processing time in ns
0	1	5	160
1	3	7	340
2	6	8	680
3	10	6	1220
4	15	20	1320
5	21	3	1440
6	28	28	2140

block and requires eight clock cycles or 160 ns for execution in the present pipelined design. The VLC size and the cumulative execution times for other pruning levels are computed likewise and as detailed in Ref. [6]. In VLC scheme, the zero-valued AC coefficients are skipped, as they are not coded, thus effecting image compression. The number of bits for this example adds up to 77 while the original picture block size is  $64 \times 8$  bits, thus effecting a compression of 6.6 for this example. The present design is capable of throughputs of 50 Mbps with a 50 MHz, single-phase clock and over 6:1 compression ratio.

As explained earlier, the DCTQ and VLC processing take place concurrently. The DCTQ computation determines the overall speed of the system since VLC processing is much faster as can be seen from Tables I and II for all pruning levels. The DCTQ computation for an  $8 \times 8$  pixel block of image requires 800 ns for a PL of

0. Hence, it is capable of processing monochrome images of size  $1600 \times 1200$  pixels at the rate of 40 frames per second. However, the picture quality for PL0 will be the lowest and can be used by the user only to browse through the images fast in areas where he is not interested. By conducting experiments on various images, it is found that image quality is acceptable for a PL of 3 or 4, for which image sizes up to  $1200 \times 1024$  pixels for monochrome and  $1024 \times 768$  pixels for color can be processed at 25 frames per second. For color images, four blocks of Y and two blocks of Cb and Cr will have to be processed for every macro block of the picture. This is 50% more execution time than that for the monochrome pictures, which needs processing of only four blocks per macro block. Therefore, the maximum size of color image that can be processed will be only 67% of the size of the monochrome picture. Normally, only 129 bits of header information is required to be processed per frame, which requires 2920 ns for execution in the present design. This is a small fraction when compared to the execution speed of 40 ms for a full frame and, therefore, does not affect the overall execution time. The above results are applicable for user programmed fixed pruning levels.

The results for automatic pruning level control are portrayed graphically in Fig. 6. In this figure, the picture quality obtained as PSNR, the compression effected by the scheme in terms of bits per pixel (bpp), the average pruning level attained and finally the execution time achieved are presented for various frames for two image sequences, Car and Rugby. Similar results were obtained for other images such as table tennis, Susie etc. though not presented here. From these graphs, the following points may be inferred: Picture quality is good around 30 dB with a maximum variation of only 2 dB. Compression is fairly constant around 0.45 bits/pixel giving a compression ratio of 17.8:1. Average PL per frame is between 3 and 5 for all the images tested even though quality achieved is quite close to the full quality level of PL14. Average PL and execution time graph patterns are almost identical since they are mutually proportional. Energy threshold is user programmable in the range between 0 and 4095 since it determines the end results. For example, the threshold zero gives full quality at PL14 for each of the blocks, but execution time doubles when compared to the automatic PL control at 200 units of energy threshold. This figure can be used as a default value for most of the pictures, in general, to yield good quality. Table III presents the

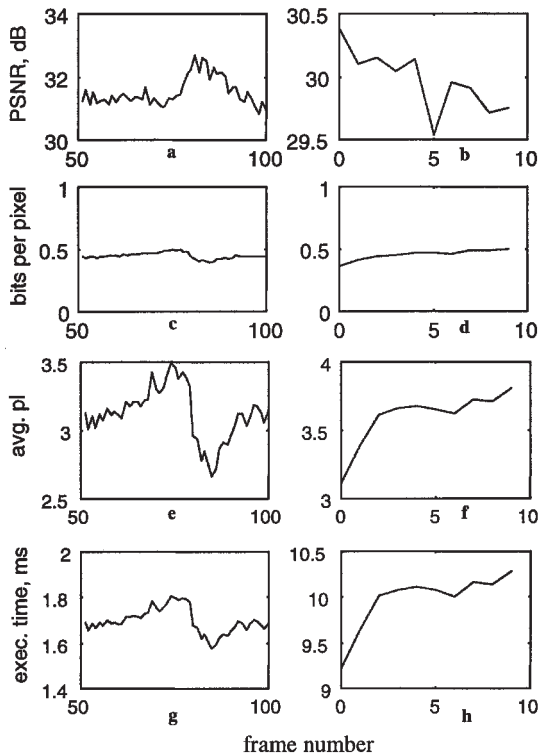


FIGURE 6 (a/b) PSNR, (c/d) bits per pixel, (e/f) average PL, (g/h) execution time plots for Car and Rugby image sequences respectively using automatic pruning level based quality control.

TABLE III Average quality, compression, pruning level, execution times and speed up ratios obtained for various images using fixed as well as automatic PL based quality controls

Per frame	Car 51–100 ( $256 \times 256$ pixels)			Rugby 0–9 ( $688 \times 480$ pixels)			TT 0–15 ( $720 \times 480$ pixels)		
	Fixed PL	Auto PL		Fixed PL	Auto PL		Fixed PL	Auto PL	
Avg. PSNR in dB	27.4	32.1	31.5	27.3	33.1	30.0	26.9	29.0	28.1
Avg. bits per pixel	0.33	0.48	0.45	0.39	0.53	0.46	0.4	0.55	0.46
Avg. PLN	3	14	3.12	4	14	3.60	5	14	4.70
Avg. exec. time, ms	1.80	3.40	1.70	10.7	17.1	9.97	13.0	17.9	10.6
Speed up ratio	1.9	1	2	1.6	1	1.7	1.4	1	1.7

overall average for all the frames for images with three different picture sizes. It may be noted that picture quality attained for auto PL control is closer to full quality level of PL14 than that obtained for the fixed PL for all the images. Further, the execution speed attained using auto PL is about two times faster than that for the fixed PL of 14. Extrapolating the average execution times to a standard

picture size, we can conclude that the auto PL controlled encoder is capable of processing  $1024 \times 768$  pixels size of images at 42 frames/s for monochrome and 28 frames/s for color on the average. Figure 7 shows a sample of the original and reconstructed frames obtained for two images using both fixed and automatic PL control modes. Similar results were obtained for various other images as well.

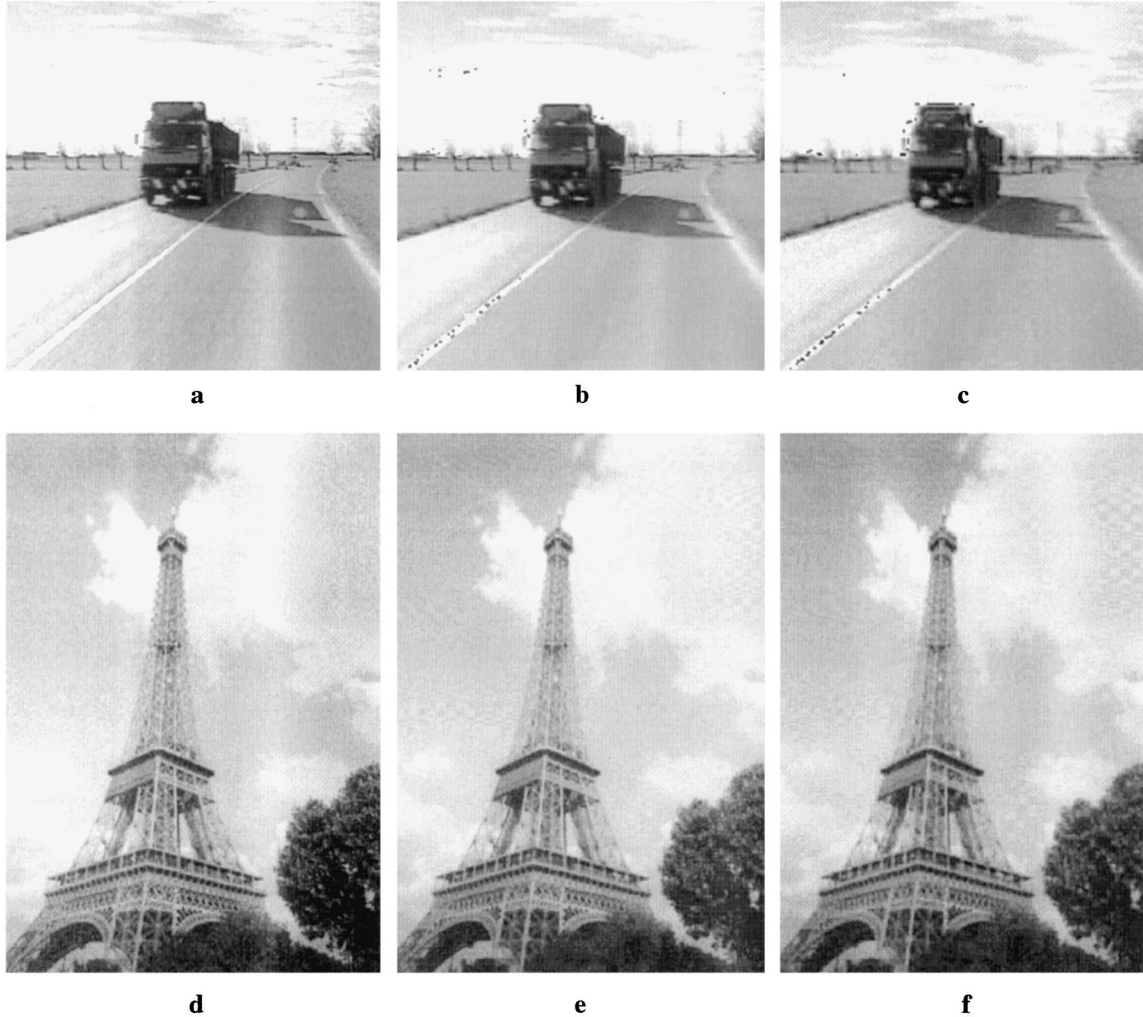


FIGURE 7 (a) Original Car 70 image, size:  $256 \times 256$  pixels. (b) Reconstructed Car 70 image by automatic PL based quality control method for  $e_{\text{THR}} = 200$  units, PSNR = 31.3 dB, Average PLN = 3.30, execution time = 1.75 ms/F and bpp = 0.475, Speed up factor = 1.94. (c) Reconstructed Car 70 image by fixed PL based quality control method for FPLN = 3, PSNR = 27.3 dB and bpp = 0.346. (d) Original Eiffelt0 image, size:  $520 \times 732$  pixels. (e) Reconstructed Eiffelt0 image by automatic PL based quality control method for  $e_{\text{THR}} = 200$  units. (f) Reconstructed Eiffelt0 image by fixed PL based quality control method for FPLN = 2, PSNR = 32.5 dB and bpp = 0.223.

## CONCLUSIONS

A novel, automatic, pruning level-based quality control scheme is proposed in this paper. An implementation of a video encoder incorporating this scheme for effective image compression is also presented. By incorporating dynamic control based on assessing the quality of a picture on-the-fly, more than twofold speed advantage is demonstrated over the conventional approach of processing without pruning without sacrificing the quality. The design is cost effective, flexible and lends itself to quick changes since the implementation is on FPGAs. The processing power of the video encoder can be further enhanced, at the cost of additional hardware, by designing a reconfigurable video encoder system that caters to a wide variety of applications conforming to JPEG, MPEG, H.263 standards, etc. This work is in progress. The enhanced system can also be implemented on ASIC.

## References

- [1] Wu, C.M. and Chiou, A. (1993) "A SIMD systolic architecture and VLSI chip for the two-dimensional DCT and IDCT", *IEEE Trans. Cons. Electronics* **36**(4), 859–869.
- [2] Chou, N.I. and Lee, S.U. (1995) "Fast algorithm and implementation of 2D-discrete cosine transform", *IEEE Trans. Circuits Syst. CAS* **38**, 297–305.
- [3] Ramachandran, S., Srinivasan, S. and Chen, R. (1999) "EPLD-based architecture of real time 2D-discrete cosine transform and quantization for image compression," *IEEE International Symposium on Circuits and Systems*, Orlando, Florida, May–June, iii375–iii378.
- [4] Park, H. and Prasanna, V.K. (1993) "Area efficient VLSI architectures for Huffman coding", *IEEE Trans. Circuits Syst.* **40**(9), 568–575.
- [5] Chang, H.C., Chen, L.G., Chang, Y.C. and Huang, S.C. (1999) "A VLSI architecture design of VLC encoder for high data rate video/image coding," *International Symposium on Circuits and Systems*, Orlando, Florida, May–June, iv398–iv401.
- [6] Ramachandran, S. and Srinivasan, S. (2000) "Design and implementation of an EPLD-based variable length coder for real time image compression applications," *The IEEE International Symposium on Circuits and Systems (ISCAS)*, Geneva, Switzerland, May 28–31, I607–I610.
- [7] Ramachandran, S. and Srinivasan, S. (2000) "A programmable pruning level control for MPEG video encoders," *The IEEE International Symposium on Circuits and Systems (ISCAS)*, Geneva, Switzerland, May 28–31, I571–I574.
- [8] Rangarajan, S.R., Srinivasan, S., Fast image compression through pruned DCT, Thesis for M. Tech., Department of Electrical Engineering, Indian Institute of Technology (Madras).
- [9] "ISO-IEC MPEG 2 Standards for generic coding of moving pictures": Part 2, Video, 1998.

## Authors' Biographies

**S. Ramachandran** is a research scholar at the Indian Institute of Technology, Madras. He has a wide industrial experience, having worked in the design of microprocessor based systems. His research interests include parallel algorithms, video processing, DSP, reconfigurable computing using FPGA and ASIC designs. He is the recipient of the Best Design Award at VLSI DESIGN 2000 International Conference. He is a member of IEEE.

**Sambasiva Srinivasan** is a professor in the Electrical Engineering Department at the Indian Institute of Technology, Madras. His teaching and research interests are in the areas of Digital Design, DSP Architectures and Applications and VLSI Design. He has taught as a Visiting Faculty at the University of California, Davis and at the California State University, San Jose. He is also the recipient of German Academic Exchange Fellowship and the Alexander von Humboldt Fellowship. He is currently coordinating the VLSI activity at the Indian Institute of Technology, Madras. Dr Srinivasan is a Senior Member of IEEE, a Fellow of the IETE (India) and a member of the VLSI Society of India.



