Reduction of Blocking Effects for the JPEG Baseline Image Compression Standard

Gregory C. Zweigle
Microelectronics Research Center
Elec. and Comp. Eng. Dept.
University of New Mexico
Albuquerque, NM 87131-1356

Roberto H. Bamberger[1]
Imaging Research Laboratory
School of Elec. Eng. and Comp. Science
Washington State University

Abstract- **Transform coding has been chosen for still image compression in the JPEG [1] standard. Although transform coding performs superior to many other image compression methods [2] and has fast algorithms for implementation [3], it is limited by a blocking effect at low bit rates. The blocking effect is inherent in all nonoverlapping transforms. This paper presents a technique for reducing blocking while remaining compatible with the JPEG standard. Simulations show that the system results in subjective performance improvements, sacrificing only a marginal increase in bit rate.**

1   Introduction

Digital images demand large amounts of data to faithfully duplicate the analog scene. As a result, image coding for compression has been a major area of research since the earliest days of digital image processing. While memory capabilities for digital storage and channel bandwidth for digital transmission have increased in recent years, so have the applications for digital images and the need for compression remains. However, compressed data is not useful if it is unreadable by those who need it. An image compression standard allows images which have been compressed for storage or transmission to be easily decompressed and used.

A proven compression method is transform coding. This technique first uses a unitary transform to map image data into a space which allows more efficient representation. In the ideal case, the mapping results in data which is independent or uncorrelated. However, such transforms are difficult to implement. The discrete cosine transform (DCT) is a transform with a known fast algorithm that also performs close to the ideal for many images [3]. Subsequent to transformation the data is typically quantized and then entropy coded, taking advantage of the uncorrelated transformed data. Transform coding using the DCT has been chosen for the Joint Photographic Experts Group (JPEG) still image compression standard [1].

JPEG has been shown to perform well for greyscale compression ratios of five to fifteen. A major limitation to further compression is a blocking effect. This visually annoying effect has its roots in the method used for transform computation. In order to exploit local stationarity and reduce computational load, an image is first divided into nonoverlapping areas and then each area is acted upon individually. A JPEG standard codec divides the imaged into 8x8 blocks. If subsequent quantization is coarse, a noticeable discontinuity between neighboring regions is visible. This is especially noticeable to the viewer because the human visual

system is very sensitive to edges [4] and even more so to edges in the vertical and horizontal directions [5], which is the direction of the blocking effect edges in JPEG. Although in a mean-square error sense, the blocking effect may not contribute much to the overall error, research has shown that it can be up to ten times more objectionable to the human viewer than random noise distortion [6].

Several approaches to reducing the blocking effect have been researched. These include postprocessing with a smoothing function [7], doing quantization with a constraint on the amount of distortion that is allowed between neighboring blocks [8], using human visual system properties in coding [9], and adaptively changing block sizes [10]. Additionally, the discovery of useful overlapping transforms has provided another framework for reducing blocking effects. Lapped Orthogonal Transforms (LOT) [11], are a family of such transforms. Of these methods, only the approach using postprocessing can be utilized with a JPEG standard codec.

In this paper, a technique for reducing the blocking effect is introduced which uses an overlapping mean estimation operator. Compatibility with JPEG is maintained by using a pre processor before compression to make the estimate and a postprocessor after decompression to restore the estimate. The extra mean information is subtracted from the original image data before JPEG compression to allow JPEG to perform more efficiently and is then transmitted as a small amount of side information. It will be seen that the overall data rate remains approximately the same with this system, but the blocking effect is dramatically reduced.

Section 2 introduces and explains the Baseline sequential JPEG codec, which employs only the most used features of the full JPEG standard. The differences between these standards do not effect the theories of this paper and subsequent references to JPEG will imply the Baseline version. Section 3 presents the pre and postprocessors used to reduce the blocking effect. The performance of the new implementation will be compared to the performance of an unadulterated JPEG codec in Section 4. Section 5 concludes the paper.

2   The Baseline JPEG Standard

A Baseline JPEG standard image compression system begins by converting the input image data into a form suitable for processing, Fig. 1. This is necessary because JPEG is a compression standard and not a file format standard. Each pixel in the input image must, however, be eight bit unsigned for the Baseline lossy codec.

After conversion to a functional format, the image data is shifted to be centered around zero. A mapping is performed from $[0, 2^P - 1]$ to $[-2^{P-1}, 2^{P-1} - 1]$ by subtracting 2^{P-1}. For eight bit data, $P = 8$, and 128 is subtracted as shown in Fig. 1. This step can be considered as a simple mean subtraction.

The next step in the compression scheme is the DCT. Although an integer 8x8 DCT is specified in the JPEG standard, the implementation details are left to the user. This leaves room for improved algorithms to be utilized as they become available. The 8x8 forward DCT
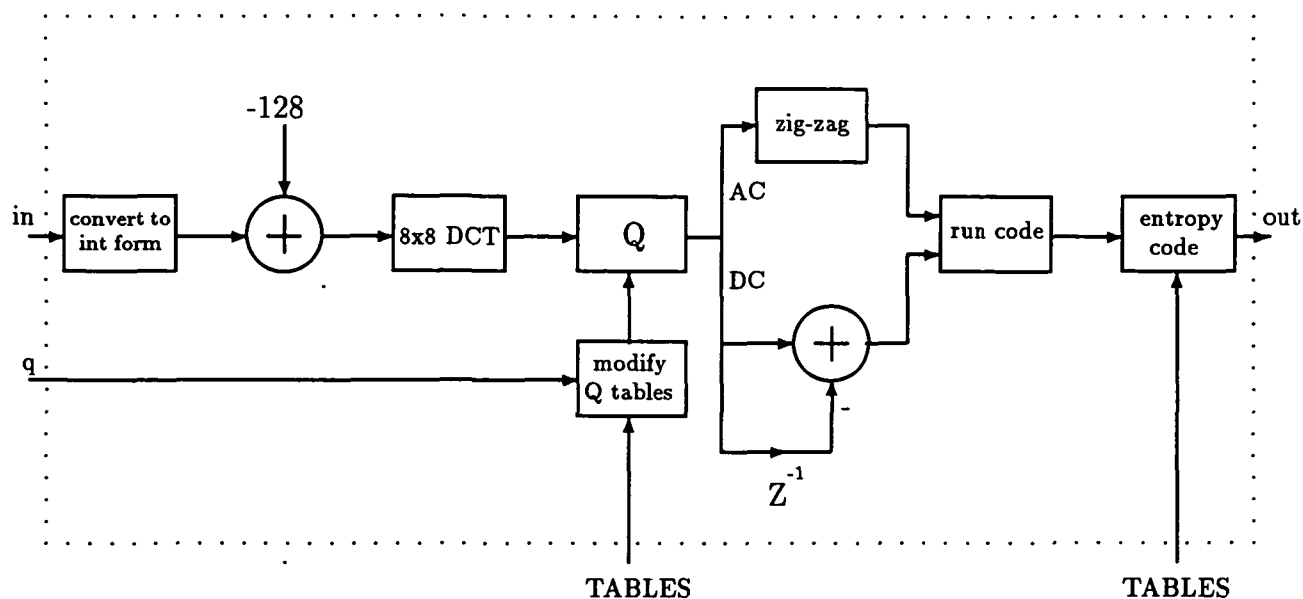
Figure 1: Block diagram of JPEG compression scheme

is defined according to

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)cos\left[\frac{(2x+1)u\pi}{16}\right]\left[\frac{(2y+1)v\pi}{16}\right].\qquad(1)$$

For decoding, the inverse DCT is defined as

$$f(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7}C(u)C(v)F(u,v)cos\left[\frac{(2x+1)u\pi}{16}\right]\left[\frac{(2y+1)v\pi}{16}\right].\qquad(2)$$

The values C(u) and C(v) are both defined according to

$$C(k) = \left\{\begin{array}{ll} \frac{1}{\sqrt{2}}, & k=0 \\ 1, & otherwise \end{array}\right\}.\qquad(3)$$

Extensive computer simulations have shown that the outputs from the DCT are integers in the range $[-2^{R-1}, 2^{R-1}-1]$ where R = P + 3 [1].

Each 8x8 block exiting from the DCT contains 64 frequency domain coefficients. These coefficients are then quantized using uniform threshold quantization in conjunction with a 64-element quantization table, Q(u,v), specified by the user. Quantization is the principle source of loss in the JPEG standard codec. The integer DCT also introduces a small amount of loss.

The quantization is done according to

$$F_Q(u,v) = Integer\ Round\ \{\ \tfrac{F(u,v)}{Q(u,v)}\ \}.\qquad(4)$$

It can be seen that as the value of Q(u,v) approaches unity, the quantization step goes away. Note also that there is no saturation point in the quantizer. Saturation is not necessary

F(0,0)   F(0,7)
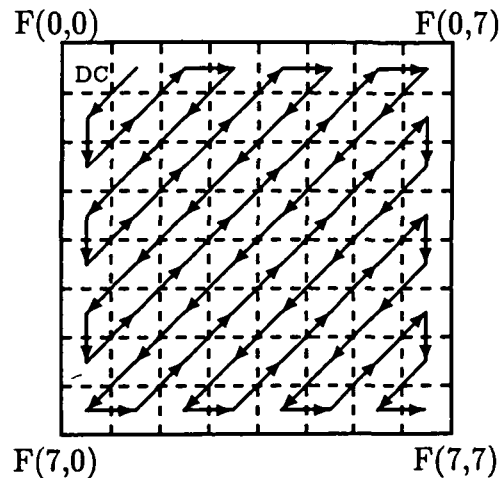
F(7,0)   F(7,7)

Figure 2: Zig-zag sequence

because the input dynamic range is known apriori. Variable length entropy coding is sufficient to deal with low probability, high valued inputs.

For this JPEG implementation, the user has control over a free parameter, q, at the time the image is compressed. This "q-factor" is used to modify the quantization tables before quantization. The initial values in the quantization table are in the range [1,255]. A q-factor of 100 sets all the quantization table values to one. As the q-factor is decreased, the quantization table values are multiplied by an at first linearly increasing and subsequently exponentially increasing factor. When q is zero, the large quantization values basically set all DCT coefficients to zero.

After quantization the F(0,0), or dc, coefficient of each DCT block is separated from the other coefficients. The dc coefficient is treated differently because it represents a local mean of image intensities on a block by block basis. Most images can be modeled as a source whose power spectral density is concentrated in the low frequencies [12] and so the dc coefficient is known to change slowly as the image is traversed. This correlation between blocks is exploited by encoding only the difference between dc components.

Meanwhile, the other DCT coefficients (called the ac coefficients) are arranged into a vector according to a zig-zag pattern, Fig. 2. This arrangements facilitates run-length coding by placing the higher frequency coefficients, which are likely to be zero, after the lower frequency coefficients which typically have more energy.

Entropy coding for the JPEG Baseline standard is tackled in two steps. The first is an intermediate symbol coding step which does a type of run-length coding and outputs a pair of symbols associated with each run. The second step does entropy coding on these symbols. In Fig. 1 the first step is denoted "run code" and the second "entropy code".

For run-length coding, only runs of zeros are counted. Two symbols are used for representation. The first symbol contains the number of zero valued coefficients that preceded the nonzero valued coefficient which terminated the run of zeros. Symbol one also contains the size in bits of the variable length integer (VLI) code that will be used to represent the amplitude of the nonzero value in the entropy coder. Symbol two is the actual amplitude of the nonzero coefficient.
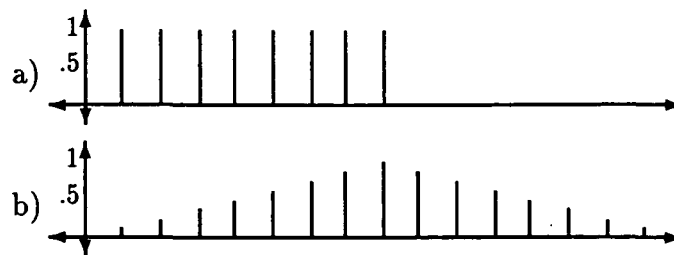
Figure 3: First and second order interpolators

The dc coefficients are also placed in a symbol one, symbol two pair by the "run coder", but no run-length coding is done. Symbol one is only the size in bits of the VLI code needed to represent the amplitude of the dc coefficient which is stored in symbol two.

The entropy coding treats symbols one and two from the "run coder" separately. Symbol one is encoded using a Huffman code. The Huffman coder requires the use of table sets which are supplied by the user. Each set consists of a table for the ac coefficients and a table for the dc coefficients. Symbol two is encoded using a variable length integer coder. Although marginally less efficient than the Huffman code, the VLI code has the advantage of being hardwired into the codec resulting in faster computation speeds and simpler implementation.

A decoder based on the JPEG standard is basically just the inverse of the encoder. For the Baseline codec, only two sets of Huffman tables can be used by the decoder at a time. This limits what can be attempted in the coder. The inverse quantization is accomplished according to

$$F'(u,v) = F_Q(u,v)Q(u,v). \tag{5}$$

At the close of decoding, the offset is added back to the data.

3 Blocking Effect Reductions

Because images tend to be low frequency in nature and the human visual system has a lowpass spatial frequency response [13], the DCT dc coefficient conveys much of the image information to the human viewer. Quantization of this coefficient without regard to neighboring regions is a major contributor to the blocking effect. If the dc information can be represented with an operator that has block overlap then the effect of the quantization will be smoothed between regions and the blocking effect reduced.

Figure 3a shows the basis function that the DCT uses to calculate the dc coefficient. This boxcar shaped function is of length eight, the same size as the transform. Because of the size, discontinuities between blocks cannot be smoothly interpolated. Figure 3b shows an alternate basis function, the second order interpolator, which has similar frequency content as the DCT dc basis function. This function is able to represent the dc level of the image more smoothly because the function of Fig. 3b overlaps the block size. But the triangular shaped basis function cannot be directly substituted for the boxcar shaped DCT basis function because of the size difference and the desire for an orthogonal transform. In order to use the new function, a preprocessor is utilized. This is shown in Fig. 4.
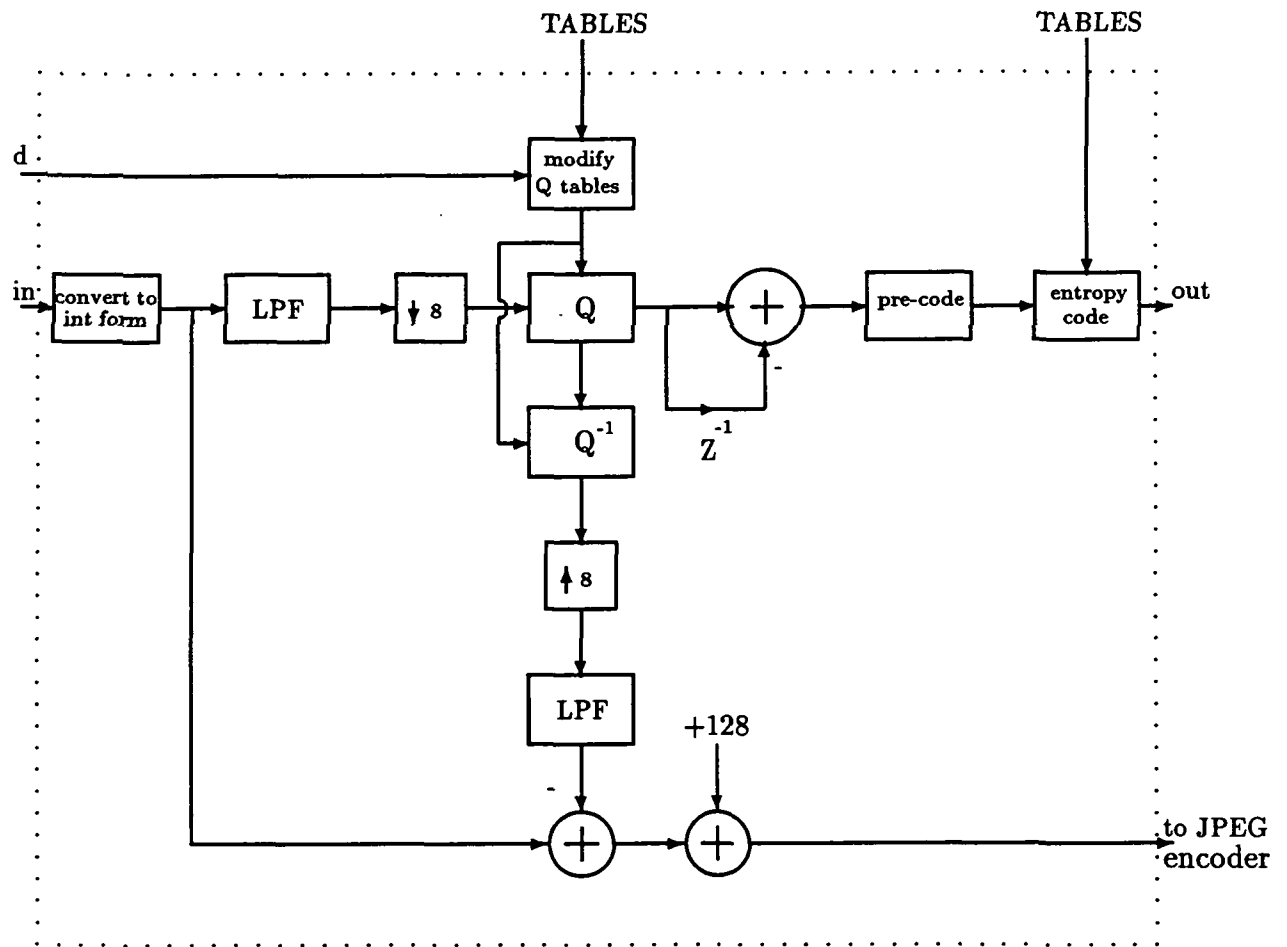
TABLES   TABLES

d

modify
Q tables

in convert to   LPF   ↓ 8   Q   ⊕   pre-code   entropy   out
    int form   code

                              Q⁻¹   Z⁻¹

                              ↑ 8

                              LPF   +128

                              ⊕   ⊕   to JPEG
                                         encoder

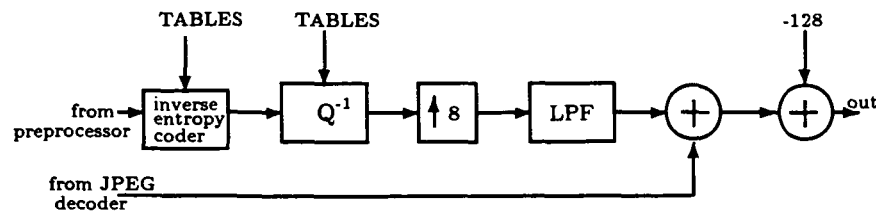Figure 4: Block diagram of preprocessor
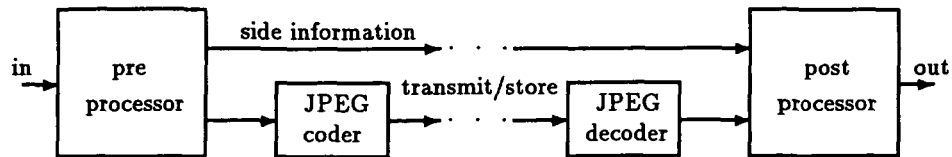
Figure 5: Block diagram of postprocessor

Figure 6: Block diagram of JPEG with pre and postprocessor

The preprocessor operates similar to a laplacian pyramid [14]. After conversion to a useable internal format, image data is lowpass filtered ("LPF" block) by circular convolution with the second order interpolator of Fig. 3b. Circular convolution is performed instead of linear convolution to avoid data expansion, although some picture edge effects result. The lowpass filtering results in an estimate of the mean of the image with the new basis function. The image is then decimated by eight to take advantage of the now oversampled image spectrum. The quantization and entropy coding steps are performed with the same tables as the ones the JPEG codec will use for its dc coefficient. However, a new parameter, 'd', is used to adjust the quantization table of the preprocessor. When 'd' is large, the effect of quantization is reduced and the mean estimate is smooth between neighboring regions. As 'd' approaches zero, the quantization becomes more coarse and the error more blocky. When d = 0, the preprocessor goes away.

The inverse quantization, upsampling, filtering, and subtraction of the new mean estimate from the original signal allows the JPEG coder to use less information for its dc estimate which reduces the JPEG bit rate. In the ideal case, the JPEG bit rate decreases by the same amount as the extra bit rate needed for side information. The addition of 128 before JPEG processing is necessary to cancel the effect of JPEG's initial subtraction of 128.

The postprocessor used after JPEG decompression to reconstruct the mean estimate is shown in Fig. 5. All LPF blocks in Figs. 4 and 5 represent circular convolution with the function of Fig. 3b. The JPEG input to the postprocessor is the output of a JPEG decoder.

The postprocessor simply puts back the information that was subtracted from the original input. The complete system is shown in Fig. 6.

4 Performance Comparisons

Test results show that the performance of the system is dependent on the choice of 'd'. For large 'd', the total bit rate is marginally increased due to the side information that must be transmitted. However, subjective analysis shows the blocking effect to be significantly reduced.
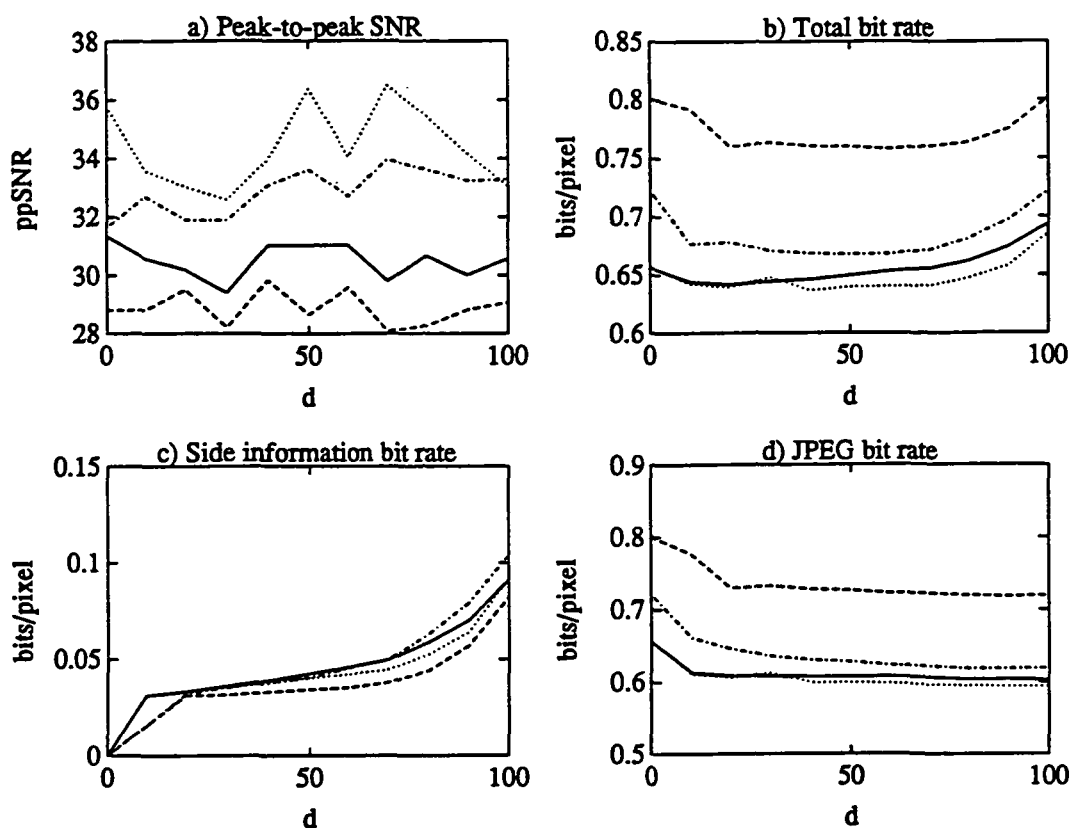
Figure 7: Bit and distortion rates, q = 15, d increases from 0 to 100

Figure 7 shows test results for four different images all compressed and decompressed with the system of Fig. 6 and a JPEG q-factor of 15. This results in greyscale compression ratios of 10 to 12, depending on the images. The data for d = 0 represents compression by JPEG without the extra processors. The solid line represents data for the building image, long dashed line for kboat, short dashed line for cameraman, and short/long dashed line for lenna. These are standard comparison images and will be shown at the symposium. Along the horizontal axis of all four plots is the preprocessor parameter 'd'. Figure 7a shows the bit rate associated with the complete system. For d = 0, the plain JPEG codec has the lowest bit rate. The images however, are blocky. As d increases, the images become less blocky but the bit rate is slightly increased. Figure 7b shows the bit rate associated with just the JPEG portion of the system of Fig. 6. It is apparent that the subtraction of the mean information from the original image before JPEG compression helps reduce JPEG's data rate.

In order to make better comparisons as to the amount image blockiness is reduced with the use of the extra processors, the JPEG dc coefficient quantization table value is altered for further tests to reduce blocking effects associated with JPEG alone. The quantization table value for the dc coefficient is set to 1 for the JPEG codec used for d = 0 data. For the JPEG codec used in conjunction with the pre and postprocessor the quantization table value is not changed. These test results are shown in Fig. 8. From Fig. 8a, the objective distortion measure, ppSNR, is seen to show no significant change as 'd' changes. The subjective distortion is reduced due to a reduction in the blocking effect. This can be seen by looking at the images. Figure 8b shows the total bit rate of the system. It can be
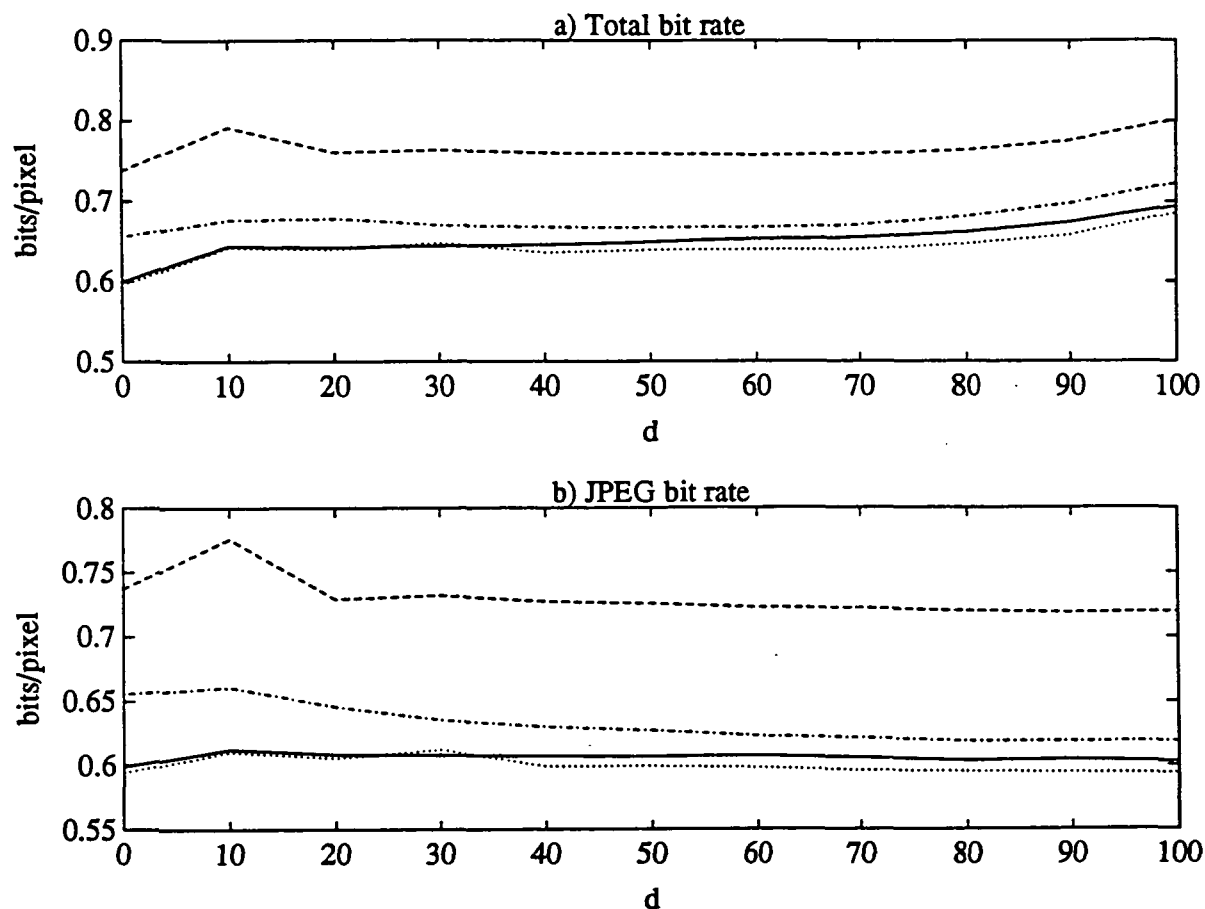
Figure 8: Bit and distortion rates, different Q table for d = 0

seen that the total bit rate is lowest for intermediate values of 'd'. For d = 0, the JPEG codec has a higher bit rate because of reduced dc coefficient quantization. As 'd' increases, the JPEG rate (shown in Fig. 8d) decreases more than the side information rate increases, (side rate shown in Fig. 8c) resulting in the dip in Fig. 8b. Notice from Fig. 8c, that the side bit rate associated with the pre and postprocessor is less than a tenth of a bit for all four of the images and all values of 'd'. This side rate is independent of the JPEG q-factor. So Fig. 8c gives an indication of the side rate regardless of what q-factor is chosen. Because of image degradation that can occur in printing the symposium proceedings, the images will be presented at the symposium for subjective comparisons.

5   Conclusion and Future Work

A JPEG compatible codec has been described which reduces the blocking effect for low bit rates. The codec uses an overlapping basis function for average pixel intensity estimation. The mean estimate is transmitted as a small amount of side information and subtracted from the original image before JPEG compression. Experimental results show that the blocking effect is reduced.

There are some areas of work that could lead to better results for this system. The

second order interpolator was used for computational simplicity, but it is not orthogonal to the other DCT basis functions. Use of a LOT basis function could possibly give better results. Another area that should be pursued is the reduction of picture edge effects that result from circular convolution. This area of research has been well studied for subband coding systems [15] and these results should be extended to this system. Also, there is still a slight blocking effect in the neighborhood of edges for very low bit rates. This is a result of edge information being contained in ac coefficients which are not dealt with in the extra processors. The possible reduction of this effect needs to be addressed.

References

[1] Gregory K. Wallace, "The JPEG Still Picture Compression Standard," *Comm. of the ACM*, vol. 34, pp. 31-44, April 1991.

[2] Gregory K. Wallace, R. Vivian, and H. Poulsen, "Subjective Testing Results for Still Picture Compression Algorithms for International Standardization," *Proceedings of the IEEE Global Telecommunications Conference*, pp. 1022-1027, Nov. 1988.

[3] N. Ahmed, T. Natarajan, and K. Rao, "Discrete Cosine Transform," *IEEE Trans. Comput.*, vol. 23, pp. 90-93, Jan. 1974.

[4] Murat Kunt, A. Ikonomopoulos, and Michel Kocher, "Second-Generation Image-Coding Techniques," *Proc. IEEE*, vol. 73, pp. 549-574, April 1985.

[5] F.W. Campell and J.J. Kulikowski, "Orientational Selectivity of the Visual System," *J. Physiol.*, vol. 187, pp. 437-445, 1966.

[6] Makoto Miyahara and Kazunori Kotani, "Block Distortion in Orthogonal Transform Coding - Analysis, Minimization, and Distortion Measure," *IEEE Trans. on Comm.*, vol. 33, pp. 90-96, Jan. 1985.

[7] Bhaskar Ramamurthi and Allen Gersho, "Nonlinear Space-Variant Postprocessing of Block Coded Images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol.34, pp. 1258-1267, Oct. 1986.

[8] William A. Pearlman, "Adaptive Cosine Transform Image Coding with Constant Block Distortion," *IEEE Trans. on Comm.*, vol. 38, pp. 698-703, May 1990.

[9] King N. Ngan, Kin S. Leong, and H. Singh, "Adaptive Cosine Transform Coding of Images in Perceptual Domain," *IEEE Trans. Circuits and Systems*, vol. 37, pp. 1743-1749, Nov. 1989.

[10] Its'Hak Dinstein, Kenneth Rose, Arie Heiman, "Variable Block-Size Transform Image Coder," *IEEE Trans. on Comm.*, vol. 38, pp. 2073-2078, Nov. 1990.

[11] Henrique S. Malvar and David H. Staelin, "The LOT: Transform Coding Without Blocking Effects," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp.553-559, April 1989.

[12] N.S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[13] J.G. Robson, "Spatial and Temporal Contrast- Sensitivity Functions of the Visual System," *J. Opt. Soc. Amer.*, vol. 56, pp.1141-1142, 1966.

[14] Peter J. Burt and Edward H. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. on Comm.*, vol. 31, pp. 532-540, April 1983.

[15] Roberto H. Bamberger, Steven L. Eddins, and Veyis Nuri, "Generalizing Symmetric Extension: Multiple Nonuniform Channels and Multidimensional Nonseparable IIR Filter Banks", *IEEE Intl. Symp. on Circuits and Systems*, San Diego, CA, May, 1992.