Minor Errors in ‘Data Compression, the Complete Reference’

On page 303 there are reproduced Tables 4.23 and 4.24 each of which is supposed to show the effect of applying the DCT of equations 4.15 and 4.16 to an 8x8 matrix of data values. Table 4.23 does this faultlessly but there are a number of errors in Table 4.24. Three of these are obvious in that the quantization of the DCT coefficients (b) has produced a -10 in row 3 and a -20 and -10 in row 4 where the correct values should be -1, -2 and -1, respectively. However, checking all of the DCT coefficients (b) shows that they do not correspond to the original data (a). (The same checking program did produce a complete verification for Table 4.23). The corrected Table 4.24.is reproduced below together with a printout of the differences between the recovered and the original data that do not seem to support the suggestion that the ‘Correlated Values’ are subject to smaller errors than the ‘Random Values’ as the square root of the sum of the squares of the errors is very similar and the largest individual error occurs in the first set.

$$D = \begin{pmatrix} 8 & 10 & 9 & 11 & 11 & 9 & 9 & 12 \\ 11 & 8 & 12 & 8 & 11 & 10 & 11 & 10 \\ 9 & 11 & 9 & 10 & 12 & 9 & 9 & 8 \\ 9 & 12 & 10 & 8 & 8 & 9 & 8 & 9 \\ 12 & 8 & 9 & 9 & 12 & 10 & 8 & 11 \\ 8 & 11 & 10 & 12 & 9 & 12 & 12 & 10 \\ 10 & 10 & 12 & 10 & 12 & 10 & 10 & 12 \\ 12 & 9 & 11 & 11 & 9 & 8 & 8 & 12 \end{pmatrix} \qquad C = \begin{pmatrix} 79.87 & 0.04 & -0.34 & -1.25 & 1.63 & -1.75 & 0.82 & 1.04 \\ -1.8 & -0.44 & -0.9 & 1.06 & -0.76 & -1.09 & -1.31 & 0.93 \\ 1.55 & -1.02 & 0.25 & -2.36 & 1.88 & 0.92 & 0.68 & 0.73 \\ 1.83 & -3.18 & -0.84 & 0.03 & -0.18 & 0.34 & -0.21 & -1.26 \\ -2.12 & 1.2 & 1.89 & -0.89 & 2.62 & -1.01 & 0.98 & -1.19 \\ -0.35 & 0.22 & -0.72 & -1.75 & -1.59 & -2.12 & -1.97 & -1.48 \\ -0.51 & -0.11 & -1.57 & -0.5 & -0.37 & 0.35 & -2.75 & -3.24 \\ 0.47 & -0.16 & -1.1 & 1.38 & 3.65 & -1.4 & 0.07 & 1.53 \end{pmatrix}$$

$$B = \begin{pmatrix} 80 & 0 & 0 & -1 & 2 & -2 & 1 & 1 \\ -2 & 0 & -1 & 1 & -1 & -1 & -1 & 1 \\ 2 & -1 & 0 & -2 & 2 & 1 & 1 & 1 \\ 2 & -3 & -1 & 0 & 0 & 0 & 0 & -1 \\ -2 & 1 & 2 & -1 & 3 & -1 & 1 & -1 \\ 0 & 0 & -1 & -2 & -2 & -2 & -2 & -1 \\ -1 & 0 & -2 & 0 & 0 & 0 & -3 & -3 \\ 0 & 0 & -1 & 1 & 4 & -1 & 0 & 2 \end{pmatrix} \qquad d = \begin{pmatrix} 8.32 & 9.74 & 9.4 & 10.85 & 11.56 & 9.02 & 8.78 & 11.97 \\ 11.36 & 8.24 & 12 & 8 & 10.57 & 10.46 & 10.78 & 10.08 \\ 8.96 & 10.81 & 8.72 & 9.83 & 11.88 & 8.62 & 8.84 & 7.61 \\ 8.86 & 12.26 & 10.02 & 7.97 & 7.83 & 9.26 & 8.26 & 9.35 \\ 12.2 & 7.86 & 8.74 & 8.95 & 11.77 & 9.43 & 7.72 & 11.4 \\ 8 & 11.25 & 9.57 & 11.82 & 9.46 & 12.27 & 11.61 & 9.97 \\ 10.07 & 9.98 & 12.39 & 10.05 & 12.01 & 10.05 & 10.41 & 12.34 \\ 12.31 & 8.82 & 10.47 & 11.04 & 9.34 & 8.19 & 8.03 & 12.29 \end{pmatrix}$$

$$D - d = \begin{pmatrix} -0.32 & 0.26 & -0.4 & 0.15 & -0.56 & -0.02 & 0.22 & 0.03 \\ -0.36 & -0.24 & 0 & 0 & 0.43 & -0.46 & 0.22 & -0.08 \\ 0.04 & 0.19 & 0.28 & 0.17 & 0.12 & 0.38 & 0.16 & 0.39 \\ 0.14 & -0.26 & -0.02 & 0.03 & 0.17 & -0.26 & -0.26 & -0.35 \\ -0.2 & 0.14 & 0.26 & 0.05 & 0.23 & 0.57 & 0.28 & -0.4 \\ 0 & -0.25 & 0.43 & 0.18 & -0.46 & -0.27 & 0.39 & 0.03 \\ -0.07 & 0.02 & -0.39 & -0.05 & -0.01 & -0.05 & -0.41 & -0.34 \\ -0.31 & 0.18 & 0.53 & -0.04 & -0.34 & -0.19 & -0.03 & -0.29 \end{pmatrix}$$

The errors detected in Table 4.24 may be due to no more than typographical errors in the original data matrix but there is no way of identifying this a posteriori.

On page 535 there are typographical errors in equations (5.1) and (5.2). The factor $2\pi/n$ has been omitted from the argument of the exponential in the second line of both equations. In the second line of equation (5.2) the initial factor 1/n has been omitted and the summation index has been incorrectly identified as *t* rather than *f*.

These errors are merely typographical but when they are corrected the resulting equations are fundamentally flawed in that they do not meet the Nyquist rate requirement (discussed in the book on page 541) that there be at least two sample values within each frequency cycle. This is a long standing error in much of the IT literature but it is a great shame if it is thus perpetuated in a book that is otherwise of such an outstandingly high quality. A correct form for these equations can be analytically derived as is shown in my brief paper at the Fourth Biennial Engineering and Mathematics Conference held at RMIT University, Melbourne, Australia in September 2000. In the nomenclature of equations (5.1) and (5.2) this gives:

$$G(f) = \sum_{t=0}^{n-1} g(t) exp\left(\frac{-2\pi ift}{n}\right) \; ; \qquad -\frac{n}{2} \leq f \leq \frac{n}{2}$$

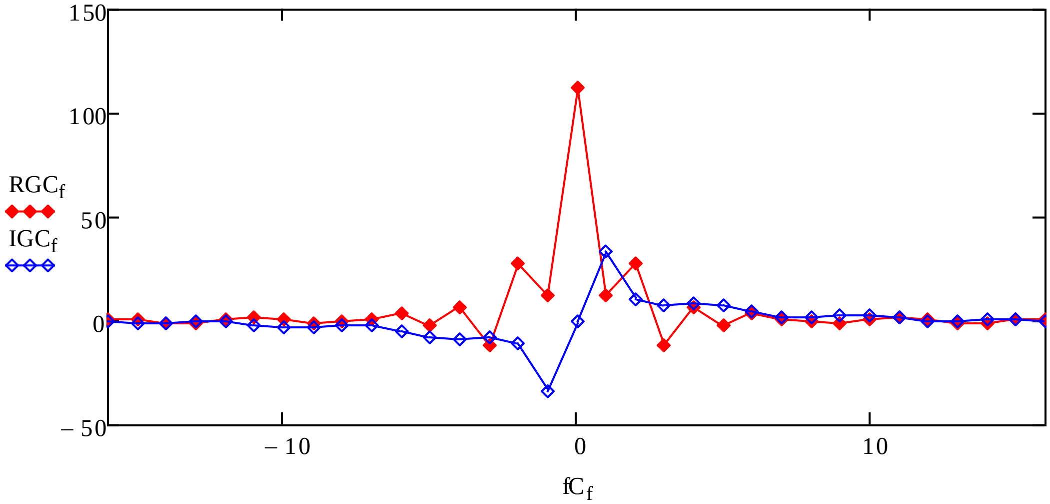
$$g(t) = \frac{1}{n} \sum_{f=-n/2}^{n/2} \ddot{G}(f) exp\left(\frac{2\pi ift}{n}\right) \; ; \qquad 0 \leq t \leq n-1$$

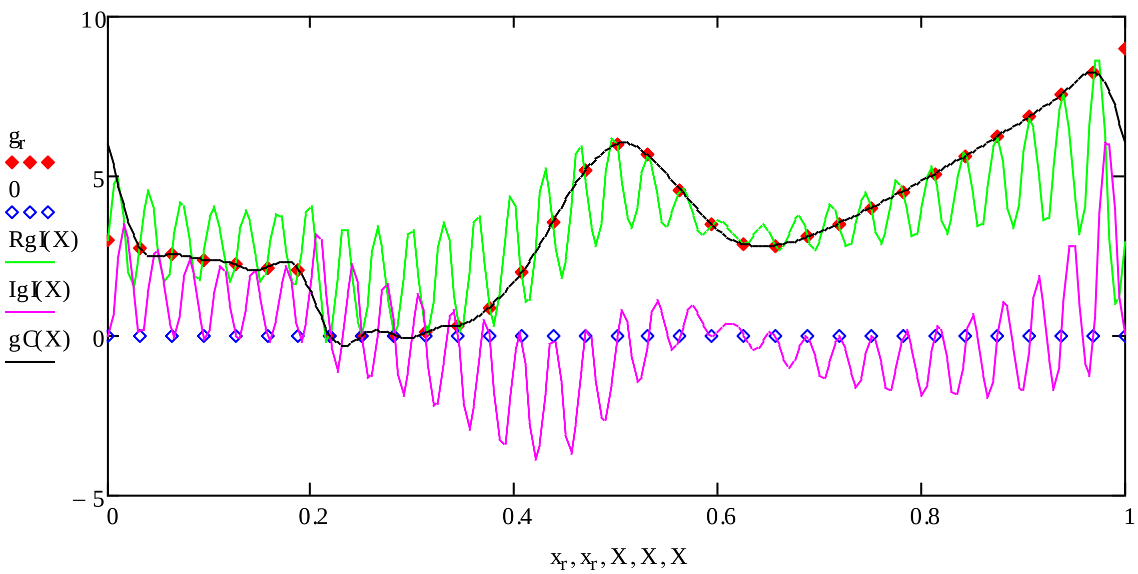
Where the double prime on *G* indicates that a half value is used for $f = \pm n/2$.

These equations are quoted in early IT literature (e.g. equation 124.19 in “The Engineering Handbook” Ed. R.C.Dorf, IEEE Press, 1996) but seem to have been sadly neglected. They of course meet the Nyquist rate requirement and it is only necessary to apply them to a data set obtained from any simple curve to see that they are markedly superior to equations (5.1) and (5.2) in terms of generating a compact set of frequency coefficients that provide a vastly improved approximation to the true Fourier series coefficients and, when the resulting curve is plotted for non-integral values of *t*, a greatly superior interpolated form.

A minor variation that also introduces *g(n)* provides a simple direct stepping stone to the DCT obviating the need for the elaborate explanations in the book and, by clarifying the reason ***why*** the DCT provides improved performance in data compression applications, points the way towards further very significant gains that can be made. To illustrate the differences that these changes make the incorrect DFT (IDFT), its correct form (incorporating the minor change mentioned in the previous sentence CDFT) the resulting DCT and a subsequent enhancement (EDFT) have each been applied to a data set obtained from a specimen curve. The frequency coefficients for the CDFT are shown on the first figure where it is seen that the real parts of the frequency coefficients are symmetrical with respect to the zero frequency and the imaginary parts are antisymmetrical. This results in the recovered data values being simply real.

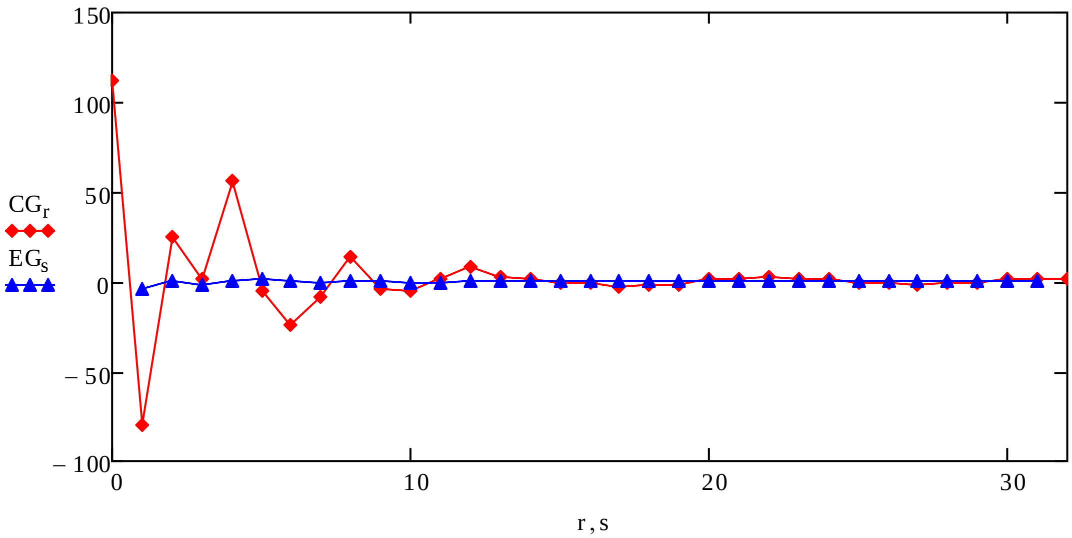
The frequency coefficients for the IDFT can be obtained from those shown on the figure by simply shifting the section for –N/2<fC<0 to occupy the frequency range from N/2 to N. This results in large frequency coefficients at both the low and the high end of the frequency range, which is atypical of Fourier series and results in there being both a real and an imaginary contribution to the recovered data curve as shown on the next figure. It is true that the imaginary parts are zero and the real parts equal to the data values at the locations of the latter but the intermediate values are far from lying on a smooth curve whereas for the CDFT (also shown on the figure) they do.





As the CDFT requires, if anything, less computation than the IDFT and provides a simplified view of how well the approximate Fourier series is fitting the data, there would seem to be good reasons to opt for this form. It is seen from the above figure that the one region that stands out as providing a poor fit is the beginning/end point. Here the CDFT, in common with the true Fourier series, passes through the average value, which can impose a requirement for large values of some of the high frequency coefficients. The DCT overcomes this problem by applying the CDFT method to the given data extended by adding the same data set in reverse order. The

EDFT further improves the smoothness of the data. The effect of these enhancements in decreasing the magnitude of the frequency coefficients is shown on the next figure.



It is seen that the decay of the frequency coefficients is rapid for the DCT (i.e. CG_r) but it is far outperformed by the EDFT.

One significant advantage of both the CDFT and the EDFT is that the Fast Fourier Transform can be applied to the evaluation of the frequency coefficients and to the recovery of the data vector from a cropped set of the latter. This would only confer a small advantage when small sets are used (such as the 8x8 sets in the JPEG algorithm) but would become increasingly beneficial with increases of set size.

One remaining area of concern in considering the fit of the discrete functions to this data set is the oscillations about the original curve, which was smooth except for the step in the data values at x=0.2. Such steps are clearly common when pictures are being compressed. They will tend to require large amplitudes of the frequency coefficients to correctly represent them. The way that was used to overcome this problem, which resulted in the DCT, is not generally available because the locations of the steps are not known in advance. However a technique based upon the use of the Data Smoothing Sampling Theorem can achieve much the same results.

My next comments are in relation to figure 5.5 on page 537. It is suggested on the following page that part (b) shows the rapid approach to the square wave, used to generate the Fourier series, as more terms are used. However it is clear to any tyro that something else is going on. This is of course the Gibb's phenomenon, which shows that the limiting form is the square wave plus a zero width 'spike' extending 18% of the height of the step beyond the upper and lower bounds of the latter at both its beginning and end locations. Its importance to practicians (in addition to signaling that even the fundamental Fourier theory cannot exactly reproduce all shapes that would seem to lie within its scope) is that it signals a region where the Fourier series will converge only very slowly and that will read over to the discrete forms available to them. Further it provides the rationale for the construction of a DCT that counters the possible adventitious occurrence of such steps between the beginning and end values of an excised data set. These values are of course at the same location in a Fourier series representation. Some comments covering such matters would add value and provide future readers, having some knowledge of Fourier series and the discrete form, with a simple rationale for and direct route to the DCT. Section 4.6 would then possibly provide some broadening of this understanding.

David,

Upon re-reading my comments in the above note I recognized that its conciseness might make it difficult to follow in parts. In particular the simple derivation of the DCT from the correct form for the DFT probably needs amplification. The analytical derivation of the latter from the Fourier Integral Transform is covered in a paper that I wrote for the benefit of my students before I retired (some 16 years ago) entitled "Complex Variable and Fourier Analysis for Engineering Science Applications" An extract from this is:

"2.2.5 $u(\theta) = \frac{1}{2N} \sum_{s=-N}^{N}{}'' u_s \sum_{r=-N}^{N}{}'' \left(\frac{E}{E_s} \right)^r$

This form provides a close simulation of the Fourier series when the latter is applied to a function that has a discontinuity of value between its ends at $\theta = \pm\pi$ and hence exhibits the Gibbs phenomenon (see Fig A2.2 (i) in Appendix 2)."

The E in this equation is simply $exp(i\theta)$ and the E_s is $exp(i\theta_s)$, where $\theta_s = s\pi/N$. The double prime on the summation simply means that at the upper and lower limits, half of the value shown is used.

It is seen that the argument in the summation over r is simply $exp[ir(\theta - \theta_s)]$. It is easily seen that if a set of points u_s for $-N \leq s \leq N$ is considered then a new double length set of points can be readily constructed in which the above set extend from $-2N \leq s \leq 0$ and the new set are the same values in reverse order extending from $0 \leq s \leq 2N$. The two half values at $s = 0$ then add to give the correct value at this point and the addition of the exponentials for the corresponding $\pm r$ and the u_s at $\pm s$ gives $2cos[r(\theta - \theta_s)] + 2cos[r(\theta + \theta_s)] = 4cos(r\theta)cos(r\theta_s)$; hey presto, the DCT.

If you would like to have further elucidation I can send you a copy of the above paper but I should warn you that it does not contain details of the genesis of the Enhanced DFT or of a more recent exciting Data Compression discovery 'The Data Smoothing Sampling Theorem'. I am reserving my knowledge of these until I can make contact with a large company that can exploit them on a profit sharing basis.

Richard