Appendix A: Derivation of the LOT

The derivation of the LOT by Malvar [62], and Cassereau et al. [61], is quite similar, except for the actual derivation of the basis functions. The derivation given here is basically that of Malvar [62], but with comments from [61], Malvar [63], and Le Gall [47] for further clarification.

Basic Properties of the LOT

To take the statistical interdependence of adjacent blocks into account, the transform needs to allow for the overlapping of basis functions between blocks. The length L, of the basis functions, must therefore be greater than that of the subblock size N, i.e. L>N. To avoid an increase in data rate the number of transform coefficients must be kept equal to the block size, which is the major difference between this technique and that of overlap- and add techniques.

The LOT is defined as a separable unitary transform for which the basis functions corresponding to adjacent data blocks overlap in the image domain [61].

Thus the LOT transform, for a segment consisting of M blocks of size N, can be numerically defined by [62]

$$T = \begin{bmatrix} P_1 & & & 0 \\ & P_0 & & \\ & & \ddots & \\ & & & P_0 \\ 0 & & & P_2 \end{bmatrix} \tag{1}$$

P_0 is an LxN matrix that contains the LOT basis functions for each block. The matrices P_1 and P_2 are introduced because the first and last blocks of a segment have only one neighbouring block, and thus the LOT for the first and last blocks needs to be defined in a slightly different way to compensate for boundaries.

The LOT of a single block is not invertible, since the matrix P_0 is not square. Nevertheless, in terms of reconstruction of the complete sequence, all that is required, is that the transform T, of the complete sequence be invertible. Orthogonality of T is also a desirable property, since it guarantees good numerical stability, and is also necessary for good energy compactation. In order for T to be orthogonal, the columns of P_0 must be orthogonal,

$$P_0^t P_0 = I\ , \tag{2}$$

With this condition satisfied, the transform process yields a non-redundant representation of the digitised image. The next condition for the orthogonality of T, requires that the overlapping basis functions of neighbouring blocks must also be orthogonal,

$$P_o^t W P_o = P_o^t W^t P_o = 0 \tag{3}$$

where I is the identity matrix, and the shift operator W is defined by

$$W = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \tag{4}$$

The shift operator simulates the overlapping of basis functions from adjacent blocks. The identity matrix in (4) is of order L-N.

For image coding it is important that the transform exhibit good energy compactation properties. The optimal transform, given second order statistics, is the Karhunen Loeve transform. Using this fact, and the modelling of the image as a first order Gauss-Markov process, it was found that the DCT is asymptotically equivalent to the KLT for correlation factors close to one. The 2NX2N correlation matrix is defined as

$$R_{xx} = \rho^{|k-l|} \qquad k, l = 1,2,\ldots,L \tag{5}$$

where ρ *is the correlation coefficient.*

Since R_{xx} is symmetric and Toeplitz, its eigenvectors (which define the KLT) are either symmetric or antisymmetric [62], i.e.,

$$R_{xx}\, y = \lambda\, y \;\Rightarrow\; J\, y = y \quad or \quad J\, y = -y \tag{6}$$

where J is the "counter-identity"

$$J = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & & \vdots & \vdots \\ \vdots & & \vdots & \vdots \\ \vdots & & \vdots & \vdots \\ 0 & 1 & & 0 \\ 1 & 0 & \cdots & 0 \end{bmatrix} \tag{7}$$

For the first order Markov process the basis functions of the KLT, consists of an equal number of symmetric and antisymmetric basis functions. From this point of view Malvar [62], decided that a "good" LOT matrix, should comprise of half odd and half even basis functions. To advocate the use of the DCT, Malvar noted that the DCT also has this even-odd symmetry.

The last point given for the use of the DCT, is that the basis functions of the LOT should be as smooth as possible, i.e. sampled sinusoid. The reason for the required smoothness is given, for energy compactation by Malvar [62], and because the mse by itself is not a sufficient goodness criteria for the evaluation of image coding techniques by Le Gall [47]. Intuitively, any discontinuities in the basis functions would give rise to blocking effects. The KLT, for first order Markov processes,

and the DCT are members of the sinusoidal family of unitary transforms [45].

The transform length in [61] and [62] is chosen L=2N for the reason that shorter lengths may cause discontinuities at two positions, rather than at one, and also for convenience in introducing the DCT based LOT. A further desirable property of the LOT is that it should be able to represent the DC value of a flat field with only one coefficient. This can only be achieved if the overlapped lowest basis function has a constant DC value. If L is chosen any other value than multiples of N, discontinuities will be introduced into this basis function which is undesirable. Although L could be any multiple of N, to minimise boundaries, an overlap of two is chosen to minimise processing requirements.

The properties of the desired LOT matrix P_0 are, equal even and odd basis functions, basis functions which are smooth and decay toward zero at the boundaries, and an overlapping of N samples. The shifted orthogonality in (3) forces the decay toward zero of the basis functions.

Optimal LOT in terms of Energy Compactation

The optimal LOT is that transform which maximises the Energy Compactation, since this minimises the bit rate. Assuming that the Markov model is applicable, the energy compactation measure normally used is given by the transform coding gain

Instead of using an iterative technique that searches for the maximum

$$G_{TC} = \frac{\frac{1}{N}\sum_{i-1}^{N}\sigma_i^2}{\left(\prod_{i-1}^{N}\sigma_i^2\right)^{\frac{1}{N}}} \tag{8}$$

where σ_i^2 *is the variance from the autocorrelation matrix,*

$$R_0 = P_0^t R_{xx} P_0 . \tag{9}$$

coding gain as in the technique by Casserau [61], Malvar [62] uses a direct method that, although less sensitive to numerical errors, is less likely to converge to a local minima than that of Casserau. The optimal P_0, as generated by Casserau's method, may not be easily factorable so that a fast algorithm may not exist. The lack of a fast algorithm and sensitivity to numerical errors of Casserau's method made the method of Malvar a better choice. However, it is important to note that Malvar's method doesn't give a *global* optimal LOT, since his approach leads to an optimal LOT that is tied to the choice of P. The choice of P given by Malvar seems to be good since they achieved a slightly higher coding gain than Casserau [62].

Malvar starts by defining a feasible LOT matrix P that may not be optimal. This matrix is then optimised to fulfil the desired features of an optimal LOT as discussed. The optimisation is done by modifying the P matrix with an optimising matrix Z, such that

$$P_0 = PZ \tag{10}$$

is an optimal LOT matrix. P_0 is a feasible LOT matrix for any

orthogonal Z, since it still conforms to the orthogonality definitions in (2) and (3), i.e.

$$P_0^t P_0 = Z^t P^t P Z = Z^t Z = I \tag{11}$$

and

$$P_0^t W P_0 = Z^t P^t W P Z = 0 \tag{12}$$

To compute the Z that will optimise P_0 the transform domain correlation matrix is computed, i.e. substitute (10) into (9)

$$R_0 = Z^t P^t R_{xx} P Z \tag{13}$$

where R_{xx} is the correlation matrix for the first order Markovian process, defined in (7). With R_{xx} and P fixed the coding gain G_{TC} will be maximised when R_o is diagonal, i.e. when Z are the eigenvectors of $P_t R_{xx} P$. Given such a Z the LOT matrix P_0 is optimal.

Malvar [62] defined the feasible LOT as

$$P = \frac{1}{2}\begin{bmatrix} D_e - D_o & D_e - D_o \\ J(D_e - D_o) & -J(D_e - D_o) \end{bmatrix} \tag{14}$$

where D_e and D_o are the NxN/2 matrixes containing the even and odd DCT functions, defined by

$$[D_e]_{nk} = c(k)\sqrt{\frac{2}{N}}\cos\left(\frac{\pi}{N}2k\left(n+\frac{1}{2}\right)\right) \tag{15}$$

and

$$[D_o]_{nk} = \sqrt{\frac{2}{N}}\cos\left(\frac{\pi}{N}(2k+1)\left(n+\frac{1}{2}\right)\right)$$

$$for\ n=0,1,\ldots,N-1,\ k=0,1,\ldots,N/2-1, \tag{16}$$

$$where\ \ c(k) = \begin{cases} 1/\sqrt{2}, & k=0 \\ 1, & otherwise. \end{cases}$$

The basis functions of the feasible LOT P, for a Markov model with a correlation coefficient of 0.95 and n=8, are shown in figure A1, and that of the optimised LOT P_0 in figure A2. Notice the discontinuities in the odd basis functions of the non-optimised LOT.

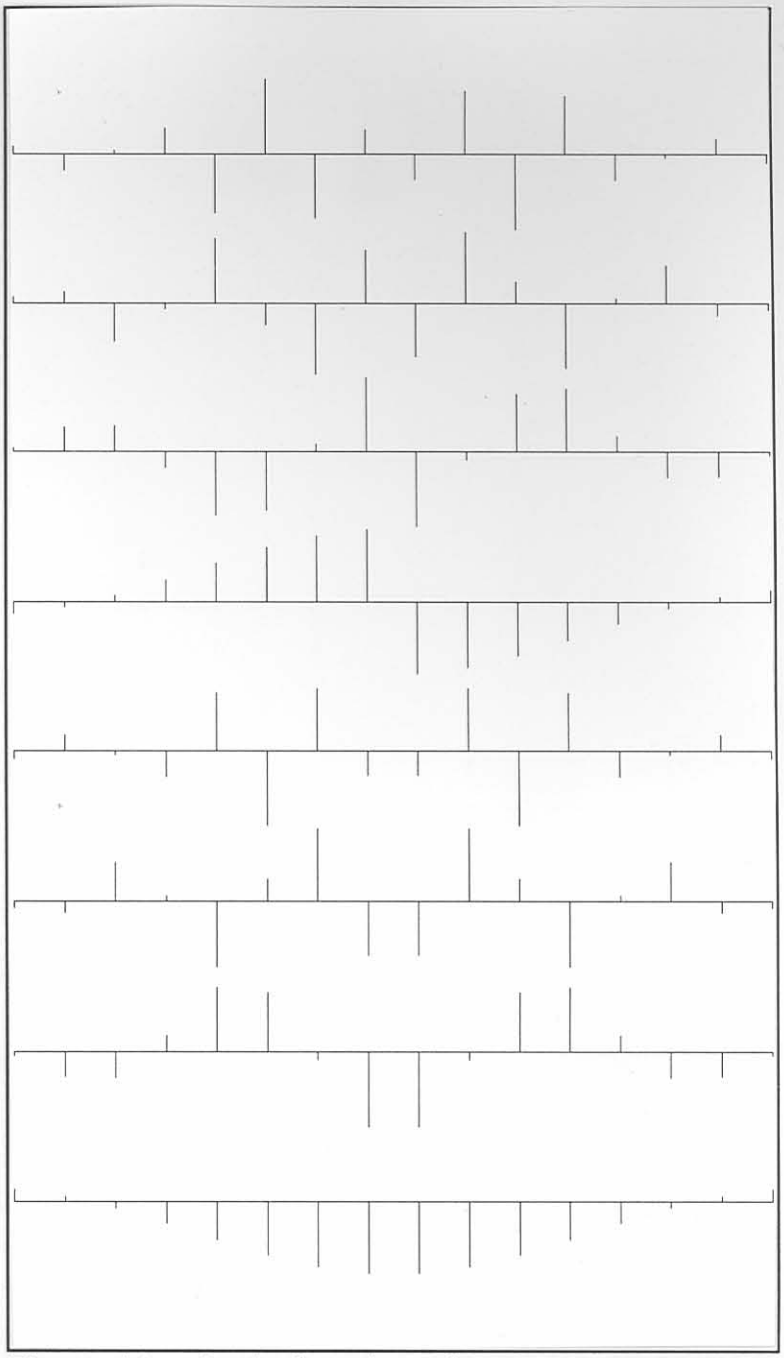

Figure A1 Basis functions for the non-optimised LOT for n=8.

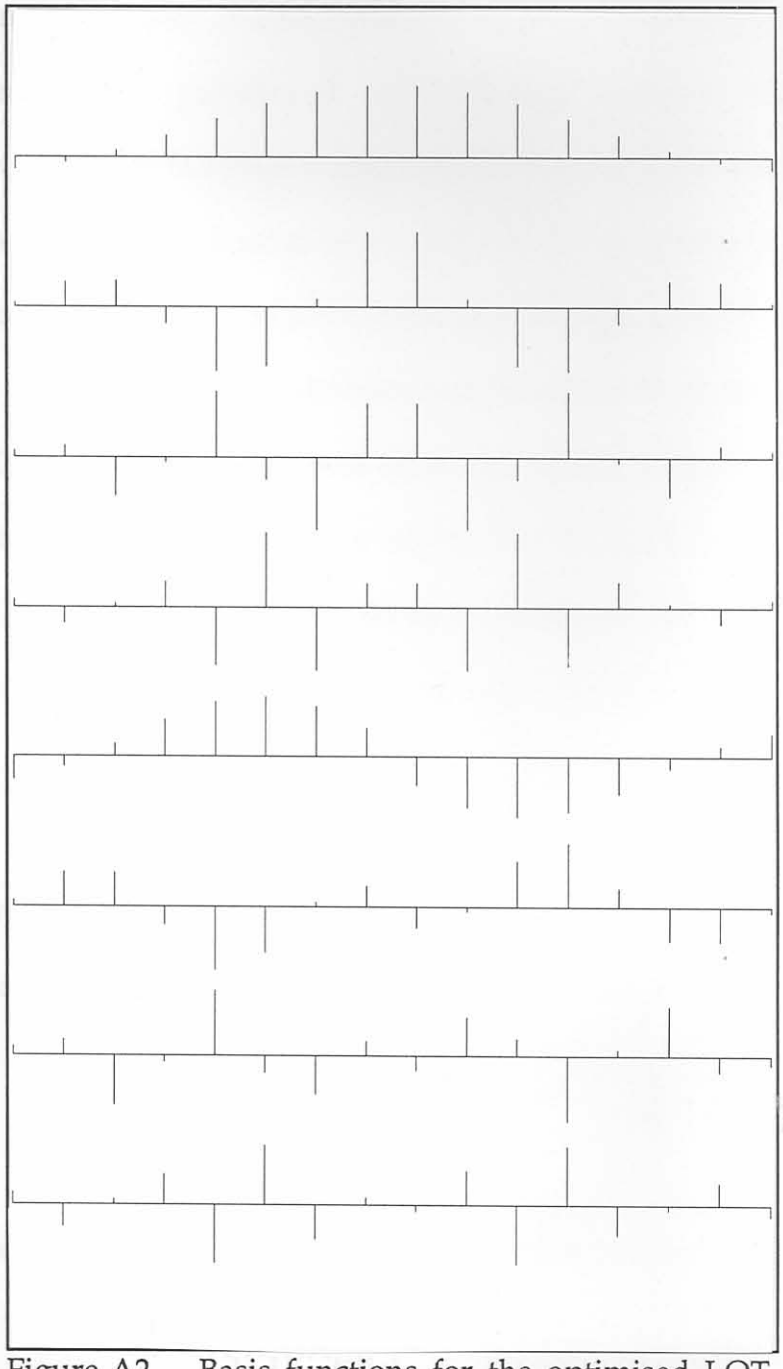

Figure A2 Basis functions for the optimised LOT, correlation coefficient = 0.95, n=8.

The optimising matrix Z may not be factorable in Nlog(N) butterfly stages. However, Malvar introduced two different methods to approximate this matrix for fast implementations. In the first approach the eigenvalue matrix is approximated by a number of matrix rotations. This technique is approximately 30% more computationally intensive than the DCT by itself, but the approximation only keeps for N smaller or equal to 16. The second method based on a sine/cosine transform approximation is valid for larger sizes of N, but is 100% more computationally intensive. The next section describes the two approximations.

Fast Algorithm

A brief summary of the approximation algorithms is given in this section. In both cases the optimising Z can be approximated by

$$Z = \begin{bmatrix} I & 0 \\ 0 & B \end{bmatrix} \tag{17}$$

For the first approximation B is a cascade of N/2-1 plane rotations,

$$B = A_1 A_2 \ldots A_{M/2-1} \tag{18}$$

the plane rotations is defined by

$$A_i = \begin{bmatrix} I^{i-1} & & 0 \\ & Y(\theta_i) & \\ 0 & & I^{\frac{N}{2}-2-(i-1)} \end{bmatrix} \tag{19}$$

$$Y(\theta_i) = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \tag{20}$$

where θ_i is the rotation angle.

where I^j is an identity matrix of order j. The rotation angles has been computed by Malvar [62] for n=8 and n=16 and are given in Table A1.

Rotation-angle / π	1	2	3	4	5	6	7
n=8	0.13	0.16	0.13				
n=16	0.42	0.53	0.5	0.44	0.35	0.23	0.11

Table A1 Rotation angles for fast LOT, correlation coefficient = 0.95

The loss in coding gain due to the approximation were computed to be 0.08 dB. The gain over the DCT for this algorithm was 0.32dB for the Markov process.

The approximation to B for any length given by Malvar [63],

$$B = C^{II}_{N/2} S^{IV}_{N/2} \tag{21}$$

where C and S are the DCT type II and DST type IV matrixes, defined by

$$[C^{II}_K]_{kr} = c(k)\sqrt{\frac{2}{K}} \cos\left(\frac{\pi}{K}k\left(r+\frac{1}{2}\right)\right) \tag{22}$$

with c(k) as in (15), and

$$[S^{IV}_K]_{kr} = \sqrt{\frac{2}{K}} \sin\left(\frac{\pi}{K}\left(k+\frac{1}{2}\right)\left(r+\frac{1}{2}\right)\right) \tag{23}$$

The LOT basis functions generated with this approximation satisfies the following symmetry condition:

$$[P_o]_{nk} = (-1)^k [P_o]_{2n-1-n,k} \tag{24}$$

which guarantees that all the filters in the analysis and synthesis filter banks have linear phase, which is an advantageous property for subband coding.

The P_0 matrix can be restuctured to isolate the DCT section from the LOT modification and optimation sections, i.e.

$$P_o = \frac{1}{2}\begin{bmatrix} D_e & D_o & 0 & 0 \\ 0 & 0 & D_e & D_o \end{bmatrix} \begin{bmatrix} I & I & 0 & 0 \\ I & -I & 0 & 0 \\ 0 & 0 & I & I \\ 0 & 0 & I & -I \end{bmatrix} \begin{bmatrix} 0 & 0 \\ I & I \\ I & -I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & B \end{bmatrix} \tag{25}$$

Most of the operations after the DCT consists of additions and

subtractions, as can be seen from the matrix contents of (25). Only one DCT need to be computed per LOT as can be seen in the flowgraph shown in figure A3. P_1 and P_2 is computed by reflecting the data at the boundaries. This concludes the derivation of the LOT algorithm.

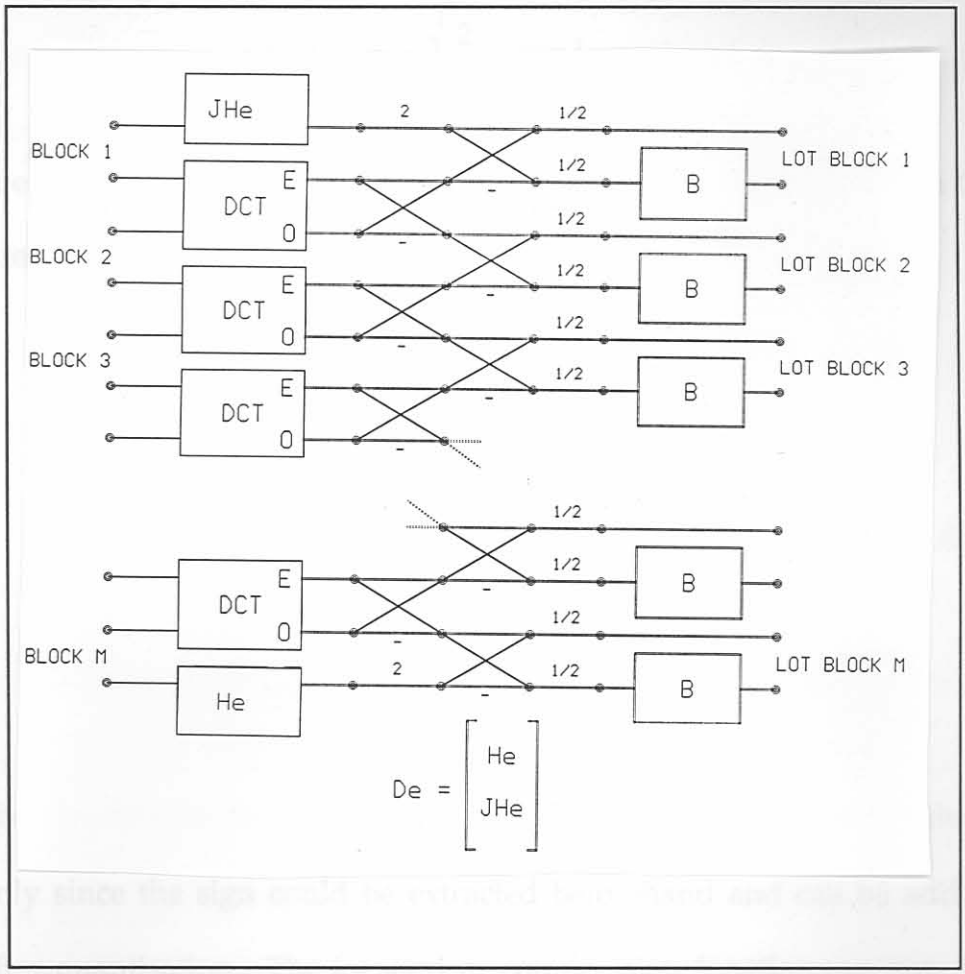

Figure A3 Flowgraph of the fast LOT for a data segment composed of M blocks of size N. Each line consists of N/2 elements. In the figure "E" denotes even and "O" denotes odd.

Appendix B Laplacian Compander

The Laplacian probability density function is given by

$$p(x) = \frac{\alpha}{2}\exp(-\alpha\ |x|) \tag{1}$$

$$where \quad \alpha = \frac{\sqrt{2}}{\sigma}, \quad \sigma^2 = variance$$

The distribution function can be obtained from the density function by integrating from minus infinity to x, i.e.

$$\begin{aligned} F(x) &= \int_{-\infty}^{x} p(x)dx \\ &= \frac{1}{2} + \frac{\alpha}{2}\int_{0}^{x}\exp(-\alpha\ x)dx \\ &= \frac{1}{2} + \left(\frac{-\exp(-\alpha\ x)}{2}\right)\Bigg|_{0}^{x} \\ &= \frac{1}{2}\left[2-\exp(-\alpha\ x)\right], \quad for \quad x \geq 0. \end{aligned} \tag{2}$$

The distribution function has been computed for the positive values only since the sign could be extracted beforehand and can be added after quantisation. The forward or compression function can now be computed by using (Eq.47 pp.47). The output is scaled to be uniformly distributed between plus and minus one half. The companding function is given by

$$g(x) = (y_{max} - y_{min})F[x] + y_{min}$$
$$= 1\left(\frac{1}{2}(2 - \exp(-\alpha\ x))\right) + \frac{1}{2} \qquad (3)$$
$$= \frac{1}{2}(1 - \exp(-\alpha\ x)) \quad for \quad x \geq 0$$

The inverse or expanding function can now be computed by solving for x in (3), i.e.

$$h(g(x)) = -\frac{1}{\alpha}\ln(1 - 2\,g(x)) \quad for \quad g(x) \geq 0 \qquad (4)$$

$g(x)$ is uniformly quantised according to the number of number of bits assigned to the coefficient. If the number of bits is b, the number of levels is 2^b and the compander can be implemented by

$$\hat{x} = h(\,Trunc(g(x)2^b + 2^{-2b})\,) \qquad (5)$$

where the 2^{-2b} is added to obtain the optimum reconstruction levels for the uniformly distributed variable. That is, the Lloyd Max quantiser is applied to the uniformly distributed variable, from which it follows that the optimum reconstruction level is halfway between the two decision levels.

Appendix C Source Coding Quantiser Stepsize

In order to achieve the desired bit rate by using source coding the source have to be quantised. The optimal quantiser for the Laplacian distribution is an uniform quantiser Berger [53]. The problem is to find the stepsize for the uniform quantiser that would achieve this rate. The entropy of a continuously distributed source is given by

$$H(x) = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$

$$\text{where} \quad p_i = \int_{x_i}^{x_{i+1}} p(x)\,dx \qquad (1)$$

$$\text{and} \quad p(x) = \frac{\alpha}{2} e^{-\alpha |x|}$$

For the Laplacian distribution the stepsize can be computed numerically by increasing or decreasing the number of levels N, in (1). However, this would be computationally intensive for large N. For the simulation in this thesis (1) was solved numerically and the results saved in a file. The file was loaded into a lookup table during system initialisation so that no stepsize computation was necessary during coding simulations.

An approximate solution to the stepsize was given by Eggerton [56] and is derived as follows: Since most common probability density functions is monotonously decreasing from zero (for zero average values), there exists a value $\alpha \in [x_i..x_{i+1}]$ for which

$$\int_{x_i}^{x_{i+1}} p(x)\,dx = p(\alpha)\,\Delta \qquad (2)$$

Equation (1) can now be written in terms of (2) as

$$\begin{aligned} H(x) &= -\sum_i \Delta\, p(\alpha_i) \log_2 (\Delta\, p(\alpha_i)) \\ &= -\sum_i \Delta\, p(\alpha_i) \log_2 p(\alpha_i) - \log_2 \Delta \end{aligned} \tag{3}$$

The first term on the right is now intuitively approximated by the entropy of continuous data $H_c(x)$ and a constant ß times the step size

$$\begin{aligned} -\sum_i p(\alpha_i) \log_2 p(\alpha_i)\, \Delta &\approx H(x) + \beta\, \Delta \\ \textit{where} \quad H(x) &= -\int_{-\infty}^{\infty} p(x) \log_2 p(x)\, dx \end{aligned} \tag{4}$$

The value of ß was determined as the one that provides a least squares fit to approximation to

$$H(x) \approx H_c(x) + \beta\, \Delta - \log_2 \Delta \tag{5}$$

and is given by

$$\beta = \frac{\boldsymbol{D}^T \cdot \boldsymbol{Y}}{\boldsymbol{D}^T \cdot \boldsymbol{D}} \tag{6}$$

where

$$\boldsymbol{Y} = \begin{bmatrix} Y(\Delta_1) \\ \vdots \\ Y(\Delta_N) \end{bmatrix} \quad \textit{and} \quad \boldsymbol{D} = \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_N \end{bmatrix} \tag{7}$$

Computer simulations were done by Eggerton [56] in which he found that ß was essentially equal to a constant that depended on the distribution and variance of the density function. He determined ß as a function of the deviation σ as follows

Gaussian	$ß=0.082/\sigma$
Laplacian	$ß=0.096/\sigma$
Cauchy	$ß=0.060/\sigma$

With these values of ß the step sizes of the uniform quantiser can be easily computed.

Appendix D Rate Distortion Simulation Parameters

The first order Markov model was used as model for the class of images for which the rate distortion simulation was done. This model was chosen for the lack of a large enough database of images that would be representative of the images the encoder could encounter. The model as defined previously is given by

$$R(x) = \sigma^2 e^{-\alpha |x|} = \sigma^2 \rho^{|x|}$$
$$\textit{where} \quad \rho = e^{-\alpha} \quad \textit{or} \quad \alpha = -\ln \rho \,, \qquad (1)$$
$$\textit{and} \quad \sigma^2 \quad \textit{is the image variance.}$$

The power spectral density can be computed by taking the Fourier transform of the correlation function -

$$S(\omega) = \int_{-\infty}^{\infty} R(x)\, e^{-j\omega x}\, dx \qquad (2)$$

By taking advantage of the odd/even symmetries of the cosine and sine functions, as well as the symmetry of the correlation function, (2) can be written as

$$S(\omega) = 2 \int_{0}^{\infty} R(x) \cos\omega x \; dx \qquad (3)$$

The power spectral density can easily be computed by replacing (1) in (3), i.e.

$$\begin{aligned} S(\omega) &= 2\sigma^2 \int_0^\infty e^{-\alpha x} \cos \omega x \, dx \\ &= 2\sigma^2 \left[\frac{e^{-\alpha x}\left(\cos \omega x + \omega \sin \omega x\right)}{\alpha^2 + \omega^2} \right] \Bigg|_0^\infty \\ &= 2\sigma^2 \left(0 - \frac{1(-\alpha + 0)}{\alpha^2 + \omega^2} \right) \\ &= 2\sigma^2 \frac{\alpha}{\alpha^2 + \omega^2} \end{aligned} \tag{4}$$

For a discrete system the radial frequency can be written in terms of an index

$$\omega = \frac{2\pi}{N} x \qquad x = 0,1,2,\ldots,\frac{N}{2}-1. \tag{5}$$

where N is the image dimension. The frequency weighing function of the human visual system as derived by Mannos and Sakrison [16] is

of the form

$$A(f_r) \approx 2.6\left(0.0192 + 0.114 f_r\right) e^{-(0.114 f_r)^{1.1}} \tag{6}$$

The nonlinear function g is of the form

$$g(u) = u^{0.33} \tag{7}$$

To do the rate distortion simulation the threshold μ has to be calculated for the desired rate R to satisfy

$$R(\mu) = \pi \int_{S_{HVS}(f_r) > \mu} \log_2 \left[\frac{S_{HVS}(f_r)}{\mu} \right] f_r \, df_r \tag{8}$$

where the power spectral density of the human visual system (HVS) is the

weighted power spectral density of the Markov model given by (4), i.e.

$$S_{HVS}(f_r) = \left|A(f_r)\right|^2 S(f_r) \tag{9}$$

To implement the HVS, it has been shown that for the given image the maximum visual frequency is equal to 6.18 cycles/degree (Table I). The radial frequency f_r can be written in terms of the discrete grid as follows

$$f_r = 6.18\sqrt{(x/N)^2+(y/N)^2} \qquad x,y = 0,1,2,\ldots,N-1 \tag{10}$$