

SIGNAL COMPRESSION

8. Lossy image compression: Principle of embedding

8.1 Lossy compression

8.2 Embedded Zerotree Coder

161

8.1 Lossy compression - many degrees of freedom and many viewpoints

- The fundamental trade-off is between the bitrate and the distortion. The larger the bitrate, the less the distortions.
- Setting the requirements is in itself a challenge. Sometimes the applications know exactly what is the allowed bitrate they can afford (restricted by the speed of the communication link, real time requirements etc.)
- Negotiating between encoder and decoder - the best alternative is embedding: the encoder produces a bitstream intended for virtually all kinds of decoders (from very low bit rate to lossless).
- Given a bitrate, what is the best achievable quality? One has to split the bit budget in between different blocks, subbands, quantizers, or contexts.
- Bit allocation is difficult even for the simplest predictive coding (1 predictor, 1 quantizer) : fix the number of quantization levels. Compute the prediction errors. Encode them entropically with arithmetic coding. What you get is a better bitrate. Change the number of levels etc.
- Perceptual effects are difficult to be exploited (masking effects are not well described quantitatively for nontrivial signals).
- Low bitrates. At 1 bit/pixel, you should transmit only a black and white image (0/1) decisions.

162

8.2 Embedded Zerotree Coder

- Introduced by Jerome Shapiro (from David Sarnoff Research Center, Princeton).
- We follow the paper and discuss the figures from J. Shapiro "Embedded image coding using zerotrees of wavelet coefficients", IEEE Trans. on Signal Processing, 41:12, 3445-3462, 1993.
- The bits in the bitstream are generated in order of importance, yielding a fully embedded code.
- The embedded code represents a sequence of binary decisions that distinguish an image from the *null* image.
- The encoder can terminate the encoding at any point, allowing a target rate or distortion metric to be met exactly.
- The compression results are competitive with most of other compression algorithms.
- Features:
 - a discrete wavelet transform;
 - prediction of absence of significant information across scales;
 - entropy-coded successive-approximation quantization;
 - use of adaptive arithmetic coding.

Discrete Wavelet Transform

- trends and anomalies
- anomalies are well localized in time or space domain
- trends are well localized in frequency domain (periodical components), persist over a large range of samples in time domain.
- Wavelets provide a signal representation in which some of the coefficients represent long data lags (low frequency range) and others represent short data lags, corresponding to wide band, high frequency range.
- Hierarchical subband system, subbands are logarithmically spaced in frequency.
- First the image is divided into four subbands and critically subsampled: LL_1 , LH_1 , HL_1 , and HH_1 , by performing lowpass and high pass filtering separately on horizontal and vertical directions.
- Each coefficient represents a spatial area of 2×2 pixels in the original image. The lowfrequency represent a band $(0, \pi/2)$, while the high frequency represent the band from $(\pi/2, \pi)$.

Discrete Wavelet Transform (cont.)

- The process is repeated starting from the subband LL_1 , to obtain a coarser representation of the image. At a coarser scale the coefficients represent a larger spatial area of the image, but a narrower band of frequencies.
- The connection between the coefficients and the pixels of the original image is linear. Arranging both of them in two vectors, X and x

$$X = Wx \quad (23)$$

where W is the matrix of the linear transformation (on its rows are the coefficients of the filters).

- The filters used are not too long (e.g symmetrical 9-tap QMF) and are used to obtain a so called QMF pyramid. They have good localization properties, their symmetry allows for simple edge treatment.
- Most importantly, the transformation matrix obtained for a discrete wavelet transform using these filters is close to unitary. Therefore $\|X\|^2 = \|x\|^2$ or $\sum_i X_i^2 = \sum_i x_i^2$.

Zerotrees of Wavelet Coefficients

- First observation: we deal with low bit rate coding (under 1 bit/pixel, usually 0.5 or 0.25 or even 0.125, i.e. compressions of 8:1, 16:1, 32:1, 64:1).
- We finally transmit only (allocate bits for) the coefficients with large magnitudes. Therefore we need to transmit also the location of those coefficients. A quite large amount of the bit budget is spent with sending the significance map (a decision whether a coefficient is zero or nonzero). The Zerotrees represents an efficient way of transmitting this information.
- TotalCost=Cost of Significance Map + Cost of Nonzero Values
- Denote p the probability that a cost coefficient is quantized to zero.

$$H = -p \log_2 p - (1 - p) \log_2 (1 - p) + (1 - p)(1 + H_{NZ}) \quad (24)$$

H_{NZ} is the conditional entropy of the absolute values of the quantized coefficients (conditioned on them being nonzero).

- If we have the target 0.5 bits/pixel, $H=0.5$. The extreme case is to quantize a nonzero coefficient with 4 bits (16 possible levels) makes necessary to set the probability of zero to $p=0.954$ (i.e. 95.4% of all coefficients have to be set to zero, and transmit values only for 4.6% coefficients). The cost of the significance map is about 54% of the whole bit budget.

The data structure: Zerotree

- A wavelet coefficient X_i is insignificant with respect to a threshold T if $|X_i| < T$.
- HYPOTHESIS: if a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold T , then *all* wavelet coefficients of the same orientation in the same spatial location at finer scales are likely to be insignificant with respect to T .
- Every coefficient at a given scale can be related to a set of coefficients at the next finer scale of similar orientation. The coefficient at the coarse scale is called parent, and all coefficients corresponding to the same spatial location at the finer scales of similar orientation are called children.
- For a given parent, the set of coefficients at all finer scales of similar orientation corresponding to the same location are called descendants.
- The relations parent children are shown in Figure 4 (see also Figure 14.15 from the textbook).
- The scanning of the coefficients is performed in such way that no child node is scanned before its parent.
- For a N -scale transform, the scan begins at the lowest frequency subband, denoted LL_N , and scans subbands HL_N , LH_N , HH_N , at which point it moves to the scale $N - 1$. The scanning pattern for a three -scale QMF pyramid is shown in Figure 5.

167

- Given a threshold T (with respect to which we determine whether or not a coefficient is significant or not) we have the following definitions:
 - a coefficient is said to be an element of a zerotree if itself and all of its descendants are insignificant with respect to T .
 - a coefficient is said to be the root of a zerotree if it is an element in a zerotree and it is not a descendant of a previously found zerotree root for the threshold T .
 - a coefficient is an isolated zero, if it is not a zerotree root, but it is insignificant w.r.t T .
 - a coefficient is positive significant if it is larger than T .
 - a coefficient is negative significant if it is smaller than $-T$.
- Therefore we use 4 symbols: zerotree root (zr), isolated zero (iz), positive significant (ps) or negative significant (ns).
- the finer scale of coefficients is encoded only using two symbols (they have no descendants).
- The flow chart for the decisions made at each coefficient is shown in Figure 6.

168

Successive–approximation

- Zerotrees are very efficient to encode the significance map of wavelet coefficients. Using them produces a shorter code than the first order entropy or a run-length coding of the significance map.
- By using successive approximation (or successive-refinement) we will encode many significance maps, by using always zerotrees.
- The second reason for using successive–approximation is to develop an embedded code analogous with the binary representation of a real number.
- Successive-approximation quantization (SAQ) applies a sequence of thresholds $T_0, T_1, T_2, \dots, T_{N-1}$ to determine significance, where the thresholds are chosen such that $T_{n+1} = T_n/2$. The initial threshold is chosen such that $T_0 < \max X_j < 2T_0$.
- During encoding two separate list of wavelet coefficients are maintained.
- The dominant list contains the coordinates of those coefficients that have not been found yet to be significant (in the same relative order as the original scan).
- The subordinate list contains the magnitudes of the coefficients that have been found to be significant. For each threshold the list is scanned once.

- During a dominant pass the coefficients with coordinates in the dominant list are compared with the threshold T_i , and if they are significant, we also encode their sign. The significance map is encoded using the zerotree method. When a coefficient is found significant, it is removed from the dominant list and its magnitude is appended to the subordinate list, meanwhile its magnitude in the wavelet transform array is set to zero so the significant coefficient does not prevent the occurrence of a zerotree on future dominant passes.
- A dominant pass is followed by a subordinate pass in which all coefficients in the subordinate list are scanned and one more bit of precision refines the available magnitudes (as explained in the example).
- The process continues to alternate between dominant passes and subordinate passes.
- In the decoding process each decoded symbol, both during the dominant and the subordinate pass, refines and reduces the uncertainty interval in which the true value of the coefficient may occur.

Example

- The seven level decomposition is obtained with the six diadic filterbanks shown on the next page. The tree structure is formed by the root, which has 3 sons, each of the sons having in turn 4 sons.

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

- The initial threshold value is $T_0 = 2^{\lceil \log_2 26 \rceil} = 16$.
- We scan the coefficients, starting with 26, which is larger than the threshold, therefore we encode a *sp*.
- The next coefficient in the scan is 6, which is less than the threshold, therefore we check its descendants if they are all insignificant. The descendants are 13, 10, 6, 4, all being below the threshold. Therefore 6 is a zerotree root, and we encode a *zr*.
- The next coefficient in the scan is -7, which, together with its descendants 4, -4, 2, -2, is less than the threshold. We encode another zerotree root *zr*.
- The next coefficient in the scan is 7, which, together with its descendants 4, -3, -2, 0, is less than the threshold. We encode another zerotree root *zr*.

171

- We don't need to encode the rest of coefficients separately because they have been already encoded as parts of various zerotrees. The sequence of labels to be send at this point is

sp zr zr zr

Since each label requires 2 bits we have used 8 bits from the overall budget.

- The only significant coefficient in this pass, the one with value 26, is included in our list to be refined in the subordinate pass. Since the reconstruction value is $1.5T_0 = 24$ the reconstruction we can perform with the bits received up to now is

24	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

- the next step is the subordinate pass, in which we obtain a correction term for the reconstruction value of the significant coefficients. We have only 26 in the significant list; The difference between 26 and the reconstruction value is $26 - 24 = 2$, which quantized with reconstruction levels $\pm T_0/4$, i.e. $\{-4, 4\}$ we obtain the correction value 4, the reconstruction becoming $24 + 4 = 28$. The correction is transmitted using a single bit, therefore up to now we use 9

172

bits to get the reconstruction

28	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

- We reduce now the threshold value by a factor of two and repeat the process. The value of the new threshold is $T_1 = 8$. We re-scan the coefficient that have not yet been deemed significant (we use an asterisk to emphasize we skip the significant coefficients):

*	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

- The first coefficient scanned is 6. It is less than the threshold, but its descendants are not (i.e. $13 > 8$). This means 6 is an isolated zero *iz*.
- The next coefficient to scan is -7 , which we observe it is a zerotree root. The next coefficient to scan is 7, which we observe it is a zerotree root. The next coefficients are 13 and 10 which are both significant positive *sp*. The final two elements we scan are 6 and 4 which are both isolated zeros (they do not have descendants).

- The second dominant pass is coded as

iz zr zr sp sp iz iz

We have used up to know 23 bits. The significant coefficients found in this pass are reconstructed with values $1.5T_1 = 12$. The reconstruction at this point is

28	0	12	12
0	0	0	0
0	0	0	0
0	0	0	0

- The subordinate list is $\{26, 13, 10\}$. The subordinate pass we take the difference between the coefficients and their reconstruction and quantize it to $\pm T_1/4$, i.e. $\{-2, 2\}$. We obtain:
 $26 - 12 = 14$, with correction term = 2
 $13 - 12 = 1$, with correction term = 2
 $10 - 12 = -2$, with correction term = -2
 Each correction requires a single bit, therefore using 26 bits we have the reconstruction

26	0	14	10
0	0	0	0
0	0	0	0
0	0	0	0

- Now we reduce the threshold to $T_2 = 4$. The coefficients to be scanned are

*	6	*	*
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

- The dominant pass results in the sequence

sp sn sp sp sp sn iz iz sp iz iz iz

This pass costs 26 bits, so up to now we used 52 bits. The reconstruction values are $1.5T_2 = 6$. The subordinate list is $\{26, 13, 10, 6, -7, 7, 6, 4, 4, -4, 4\}$. The reconstruction up to now is

26	6	14	10
-6	6	6	6
6	-6	6	0
0	0	0	0

- We may continue until the bit budget is exhausted or other criterion is met.
- Sorting the subordinate list at the end using the information available to both encoder and decoder would increase the likelihood a larger coefficient being encoded first.
- We assumed a fixed number of bits for fixed length encoding. Using arithmetic coding can further reduce the rate.

Illustration of performance

Experimental conditions:

- Arithmetic coding with an adaptive model (the variant CACM) was used. The model is initialized every time the threshold is divided by two.
- There is no training of any kind, no ensemble statistics of the images are used in any way.
- The header (12 bytes) has to specify: the number of wavelet scales, the dimension of the image, the maximum histogram counts for the models in the arithmetic coder, the image mean and the initial threshold.
- The images are standard grayscale 512×512 images with 8 bit/pixel.
- PSNR (peak signal to noise ratio) is defined as $PSNR = 20 \log_{10} \frac{255}{MSE}$,
 $MSE = \frac{1}{512^2} \sum_{i=1}^{512} \sum_{j=1}^{512} (D_{ij} - \hat{D}_{ij})^2$.
- Coding results for Lena and Barbara are presented in Tables III and IV.

Comparison with other coders:

- JPEG does not allow to specify a target bit rate, but instead allows the user to select a "Quality factor".

- Barbara at 0.38 bpp. Jpeg reaches 26.99dB while EZW reaches 29.39 dB.
- Barbara at 26.99dB. JPEG needs 0.38 bpp while EZW needs 0.27 bpp.
- Visually the 0.38 bpp EZW looks much better than the 0.38 bpp JPEG (the latter presents disturbing blocking artifacts).
- in other progressive schemes using wavelets: the number of coefficients retained was 2019, using 0.26 bpp, the resulting SNR was 24.42 dB. Compare with EZW, which encodes at the same 0.26bpp a number of 9774 coefficients, to get 26.99 dB.
- When encoding and decoding are terminated in the middle of a pass, or in the middle of the scanning of a subband, there are no artifacts produced that would indicate where the coding stopped.
- The performance scales, even at very high ratios (the quality may be poor, but the image is recognizable). With block coding schemes at the same high ratio there will not be enough bits to even encode the DC coefficients of each block.